universität wien

DISSERTATION / DOCTORAL THESIS

Titel der Dissertation /Title of the Doctoral Thesis

Design of context-sensitive, multi-stable RNA molecules

verfasst von / submitted by

Mag.rer.nat. Stefan Hammer

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

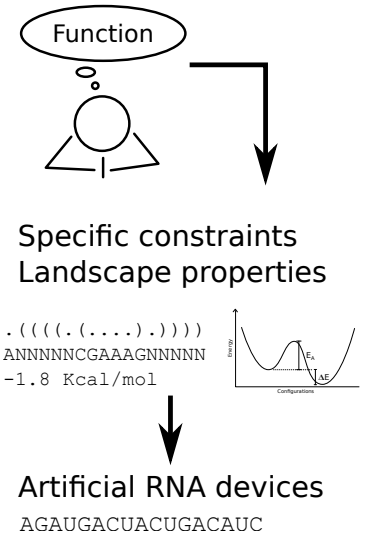
Doctor of Philosophy (PhD)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 794 685 490
Dissertationsgebiet lt. Studienblatt / field of study as it appears on the student record sheet:	Molekulare Biologie
Betreut von / Supervisor:	Univ.-Prof. Dipl.-Phys. Dr. Ivo Hofacker
Mitbetreut von / Co-Supervisor:	

Design of context-sensitive, multi-stable RNA molecules

Artificial gene regulation by *de novo* generated RNA devices



Stefan Hammer
Institute for Theoretical Chemistry
University of Vienna

ACKNOWLEDGMENTS

First of all I want to thank Sven Findeiß as without him this thesis would not have been possible. Thanks for helping and supporting me the last years, as a co-supervisor, coworker and as a friend! Furthermore, I want to thank my supervisor Ivo Hofacker for his extensive knowledge and his patience when transfering it, our associated professor Christoph Flamm for his ground-breaking ideas, passion and ambition, and our administrative staff Judith Ivansits and Richard Neuböck for their support in any matters. Another big thanks goes to Sebastian Will and Yann Ponty for catching the `RNAblueprint` ideas and developing them further, always patiently communicating new findings and improvements.

I am also grateful for all my co-workers at the TBI and for the friendly and supportive working environment they provided. Special thanks for their scientific input goes to Peter Kerpedjiev, Ronny Lorenz, Sebastian Will, Stefan Badelt, Fabian Amman, and my dear room colleges Birgit Tschiatschek, Christina Wagner, Roman Ochsenreiter, Bernhard Thiel, Dominik Steininger and Lukas Bartonek. I also want to thank our guest professor Craig Zirbel for introducing a broader viewpoint on RNA science.

Moreover, I want to say thanks to all members and organizers of our Doktoratskolleg RNA Biology, to Reneé Schröder, Andrea Barta, Gerlinde Aschauer, Nicola Wiskocil and in particular to Zahra Ayatollahi. I am also thankful for our collaborations within the EU project, Ribonets. In particular I want to thank Ilka Axmann and Andre Estevez Torres, and all the wonderful people associated with them.

I do not want to forget to thank my new chef Peter Stadler, my new coworkers at Bioinf Leipzig and Mario Mörl and his lab for the ongoing collaborations. A special thanks also to the brave reviewers of this thesis, Thomas Rattei and Daniel Merkle.

Last but not least, I want to thank Manuela Geiß, my friends and my family for their continuous support through the highs and lows of the past years.

FUNDING

This work was part of the RiboNets project, which acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323987. This work was furthermore supported by the FWF projects SFB F43 "RNA regulation of the transcriptome" and generously supported by the Austrian DK programme "Doktoratskolleg RNA Biology W1207-B09".

ABSTRACT

Introducing artificial logic into biological control systems enables to reprogram the behavior of living cells with applications in synthetic biology, biotechnology and medicine. This requires to design artificial molecules with prescribed functionality. RiboNucleic Acid (RNA) is perfectly suited for this purpose due to its close structure to function relationship, its ability to sense environmental factors or to perform enzymatic reactions.

The current approach to design multi-state RNA molecules, e.g. riboswitches, thermoswitches or small RNA (sRNA) triggered devices, is a multi-step pipeline. It ideally comprises extensive analysis of the biological system, computational sampling and optimization of sequences with prescribed constraints, an *in silico* analysis and filtering step, and experimental validation. The vision is to enable computational *de novo* design of RNA devices with high success rate and thus to reduce laboratory efforts significantly.

Thus, I developed methods and tools to improve especially the computational steps of this pipeline. A dynamic programming algorithm based on graph coloring allows to uniformly sample RNA sequences given various constraints. The resulting library implementation, `RNAblueprint`, enables to flexibly combine it with arbitrary evaluation and optimization methods. New objective function terms describe the mechanism of specific RNA devices in an elaborate way. We use probabilities and concentrations of structural elements, techniques to model ligand binding, target specific kinetic behavior, or apply extensive transcription/translation models. New visualization tools and the investigation of filtering and clustering techniques improve the analysis step. Finally, laboratory methods are proposed to verify the desired functionality of the designed systems in their target environment.

ZUSAMMENFASSUNG

Das Einbringen von künstlicher Logik in biologische Kontrollsysteme ermöglicht es, das Verhalten von lebenden Zellen umzuprogrammieren. Dies hat vielerlei Anwendungen in der synthetischen Biologie, Biotechnologie und der Medizin. Dafür ist es nötig, künstlich Moleküle mit definierter Funktionalität zu generieren. Ribonukleinsäure (RNA) ist durch ihre enge Struktur-Funktions Beziehung, der Fähigkeit Umgebungsfaktoren zu detektieren und dadurch, dass sie enzymatische Reaktionen vollziehen kann, perfekt dafür geeignet.

Der momentane Ansatz, RNA Moleküle mit mehreren Zuständen, wie zum Beispiel Riboswitches, Thermoswitches oder sRNA kontrollierte Elemente, zu bauen, ist ein Arbeitsablauf mit mehreren Schritten. Idealerweise beinhaltet dieser eine ausführliche Analyse des biologischen Systems, das computerunterstützte Generieren und Optimieren von Sequenzen mit vordefinierten Eigenschaften, einen *in silico* Analyse- und Filterschritt, und experimentelle Bestätigung. Die Vision ist es, computerunterstütztes Entwerfen von RNA Bauteilen mit einer hohen Erfolgsquote möglich zu machen, wodurch der Laboraufwand signifikant reduziert wird.

Ich entwickelte Methoden und Werkzeuge, um viele Schritte dieses Arbeitsablaufes zu verbessern. Ein Dynamischer Programmierungsalgorithmus basierend auf einer Graphfärbemethode erlaubt es, RNA Sequenzen mit vordefinierten Struktur- und Sequenzeigenschaften gleichverteilt zu ziehen. Die Implementierung, `RNAblueprint`, ermöglicht eine flexible Kombination mit beliebigen Evaluierungs- und Optimierungsmethoden. Neue Zielfunktionsterme beschreiben den Mechanismus von spezifischen RNA Bauteilen in umfangreicher Art und Weise. Wir verwenden Wahrscheinlichkeiten und Konzentrationen von Strukturelementen, Techniken um Ligandenbindung zu modellieren, streben spezifisches kinetisches Verhalten an, oder verwenden ausführliche Transkriptions-/Translationsmodelle. Neuartige Visualisierungsinstrumente und das Anwenden von Filter- und Clustertechniken verbessern den Analyseschritt. Abschließend werden Labormethoden vorgeschlagen, um die angestrebte Funktionalität der konstruierten Systeme in ihrer Zielumgebung zu bestätigen.

CONTENTS

0 PREFACE

Writing down a project spanning the time-frame of several years can be hard. Where to start? Which topics to focus on? What are the essential results and most interesting thoughts that should be preserved? Is a chronological story-line appropriate and helpful? Is it better to focus on a conceptual outline? These and many other questions are hard to answer and require a lot of thoughts on the beginning of a thesis like this one. I decided to focus only on some very central aspects of my work and explain these fundamentals in detail by introducing them in a partly chronological but mostly content-related manner.

Years ago, the work on my thesis started with some research questions, followed by literature search and the construction of a thesis proposal. Back then, the term "riboswitch" was very popular in the literature, especially in conjunction with the buzzword "design". The idea of creating logic circuits or even whole computers based on molecules like bio-polymers, such as RNA, was very popular and it was not uncommon to read about an upcoming century of effortless, rational design of such devices [11, 14, 16, 56, 98, 186, 204]. However, deduced from the concurrent slow progress in this field, it seems to be quite complicated to artificially draft a functional bio-component solely by using computers and algorithms. There were and still are just too many unknowns and crucial elements for the design pipelines missing.

I therefore formulated the question whether it is possible at all to create software that is able to accurately design artificial devices made from molecules that instantly work in cellular, biological systems? How should such a software or algorithm look like? Which choices need to be made by the designer, which aspects can be estimated by the program itself? How can the features be specified in an easily understandable but still very detailed way such that nothing is missed and the device really works as intended? Which ideas from the literature are brilliant and need to be developed further and which parts for rationally designing such bio-devices completely *in silico* are still missing?

The following chapters of the introduction, Part i, will introduce the relevant topics, starting by motivating the choice of the building material, some background in logic control systems (Chapter 1), biological gene expression regulation (Chapter 2), and RNA bioinformatics (Chapter 3). After a short historical outline of previous attempts to artificially design switching RNA molecules with a pre-

scribed functionality (Chapter 4), I quite extensively studied the literature and examined available tools and methods. This revealed that a design pipeline ideally comprises an extensive analysis of the biological system, computational sampling and optimization of sequences with prescribed constraints, an *in silico* analysis and filtering pipeline to determine best performing candidates and finally the experimental validation (Chapter 5).

My scientific contribution to this pipeline is finally outlined in Part ii, published work, and in Part iii, unpublished results and discussion, at the appropriate steps. The focus mainly lied on the developed methods to sample sequences given various constraints, an important prerequisite to enable efficient automated *de novo* design of RNA molecules. We introduced a dynamic programming algorithm based on a graph coloring strategy to uniformly sample RNA sequences given multiple structural constraints, see Chapter 6. The approach is implemented as a program library called `RNAblueprint` to allow for flexible combination of the sequence sampler with arbitrary evaluation and optimization methods, see Chapter 7.

Furthermore, I suggest new objective function terms which help to describe the functional goals of specific RNA devices much better than previously used terms. An objective function cannot comprise all design goals. Thus, I introduced novel building blocks to complement the *in silico* analysis pipeline in order to effectively filter, cluster and compare obtained candidates, see Chapter 8 and Chapter 10. As *in silico* designed systems need to be tested and analyzed for proper functionality in their target environment, either in a cell-free system *in vitro* or *in vivo* in a bacterial cell, I furthermore summarized various available laboratory methods in order to verify the proposed aspects of newly designed RNA molecules. In Chapter 10 I also highlight open questions and missing bits and pieces leading to a solid solution for the goal of computationally designing devices such as logic gates made completely from RNA, of engineering genetic networks or of building nano-structures that self-assemble within living cells.

Part I

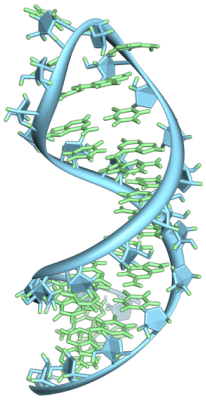
INTRODUCTION

1

ARTIFICIAL RNA MOLECULES TO REGULATE BIOLOGICAL PROCESSES

One of the first decisions any manufacturer is probably facing is the choice of material to use for realizing all the great ideas and concepts. This decision must be made very carefully as it may influence the later crafting process in almost any detail. However, the basic conditions are simple. The material must be easy to handle, its properties should be well understood and explored, and it must fulfill basic characteristics such as great electric conduction for power supply lines or water resistance for water tubes. Universality and flexibility can be another additional factor. Water pipes that can be used for any other liquid, or power supply lines with a common connector plug show clear advantages. Finally, the chosen material must be cheap in production and should be easily available.

For building artificial biological components a huge variety of materials are worth considering. There are small molecules such as lipids, fatty acids, sterols, vitamins, hormones or neurotransmitters, and there are biopolymers like saccharides made from monosaccharides, DesoxyriboNucleic Acid (DNA) and RiboNucleic Acid (RNA) build from nucleotides and proteins assembled from amino acids. While small molecules could be useful for some specialized tasks, they cannot be easily modified and adapted to fulfill a certain function. In contrast, biopolymers can be easily assembled from their monomers to build a completely new product with little effort. Saccharides have quite passive roles as scaffolding fibers or as special tags on cells and molecules. DNA is used for storing the genetic instructions in form of a double-helix like structure very well protected from its environment. Both are not very reactive compounds and they are not known to have catalytic activity in nature. RNA and proteins however seem to be a perfect material for crafting artificial components in biological systems. They can be handled easily in the lab as they can be created from their "blueprint" by using the natural occurring systems. Therefore, they are also easily available and reasonably affordable and it is known that they can fulfill many different functions, thus being universal and very flexible. Moreover, several more arguments show that RNA is most perfectly suited for artificial design, even compared to proteins.



A representation of an RNA hairpin loop, its bases colored green and the backbone blue. (CC-SA 3.0 by Vossman)

RNA is probably one of the oldest building blocks of life and according to the RNA world hypothesis [24, 152], certain forms of life could exist by almost exclusively using RNA as their building material. It is impressing that RNA is able to generate molecules with

an enormous variety in shape and function from only four simple building blocks. Moreover, it is possible for different RNA sequences to fold into the same structures, meaning that these macro-molecules can change their genotype (sequence) while still preserving the phenotype (structure), e.g. as it is the case for neutral mutations. In contrast, even small edits on the sequence level can fundamentally change the shape of the molecule and thereby induce a completely different functionality [166]. These features make it an ultimate evolvable molecule, as RNA molecules under strong selection pressure can only be slowly improved while others may lead to novel inventions in nature very rapidly.

It is known that RNA molecules have crucial and diverse regulatory functions in the cell due to their ability to interact with other nucleic acids, proteins and small molecules [28, 135, 179]. RNA can even serve as enzyme by very distinctively catalyzing chemical reactions. It also acts as transmitter of information from the genetic storage, the DNA, to the protein production machinery. In this context it is called messenger RNA (mRNA). In prokaryotes such a mRNA consists of the coding region, the part that gets translated into an amino acid sequence, and of untranslated regions (UTRs) flanking the coding region. UTRs can influence expression levels, attributes of the protein production and even the molecule's own stability.

There are many known natural examples of RNA molecules regulating biological processes [178]. We can learn from those and try to artificially integrate similar molecules into existing systems [47, 68, 106]. From the methodical view, such an integration can be easily done *in vivo* by inserting the templating DNA in form of plasmids in bacteria and eucaryotes and various similar methods [91, 100, 102]. After a successful integration, the DNA will be transcribed into the desired functional RNA. Even *in vitro*, RNA can be handled quite easily, as protocols exist to either synthesize or produce RNA by *in vitro* transcription and thus gain the molecule of interest in an extracellular context.

Furthermore, in order to prove its presence, RNA can be detected with various techniques such as Northern blot or quantitative real-time polymerase chain reaction (qPCR). Its sequence can be quickly and cheaply revealed using various sequencing techniques and we have protocols in place to enlighten the structure of a molecule, e.g. selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE), foot printing, and chemical probing. Most important for *in silico* analyses is the existence of solid mathematical models and energy parameters describing the RNA folding process (see Section 3.1). Those enable the usage of fast algorithms for structure prediction and thermodynamic as well as kinetic calculations (see Section 3.2 and Section 3.4). For proteins such models are rare. One reason could be that for the folding process of many proteins external factors like chap-

erons are necessary, which makes it much harder to mathematically describe the process. Additionally, environmental conditions such as ion concentrations or temperature have a big impact on the final shape of the molecules. For RNA however, there exist methods to take care of many of these variations. Nevertheless, some influences are still not well understood which makes *in silico* studies difficult.

1.1 INTRODUCTION TO RNA BIOCHEMISTRY

What exactly is an RNA molecule and what does it actually look like? It is a quite simple polymer made from only a small set of four components. These four nucleotides are two purines, called Adenine (A) and Guanine (G), and two pyrimidines, called Cytosine (C) and Uracil (U). Another pyrimidine, Thymine (T), only appears in DNA. Each nucleotide consists of a five-carbon sugar ribose, a nitrogenous base and additional phosphate groups (Figure 1). Polymerization happens through a phosphodiester bond connecting the 3′ and the 5′ carbon of the riboses of two nucleotides, thereby forming a sequence of the four different nucleotides. The composition of this sequence determines what the three-dimensional shape of the polymer will look like and even more exciting, how the molecule behaves and which properties it has. This is possible due to the nucleotides' capacity of forming hydrogen bonds, called base-paring.

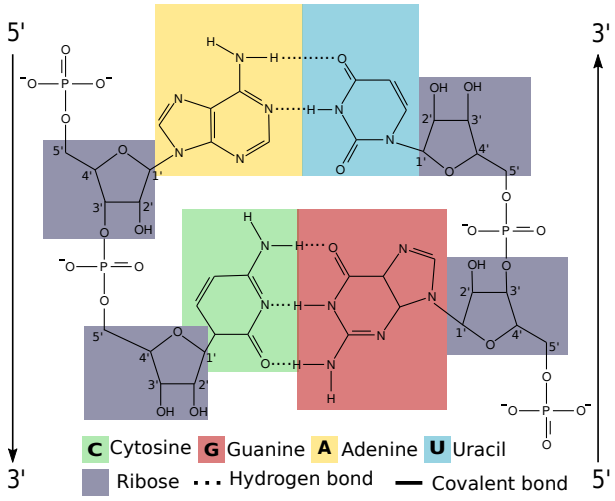


Figure 1: RNA consists of consecutive nucleotides connected through a phosphodiester bond between the riboses of adjacent molecules. Two strands in opposite direction can form base pairs leading to the formation of helices. Here, Watson-Crick pairs between A-U and C-G are shown. (CC-SA 4.0 by Florian Eggenhofer and Dominik Steininger)

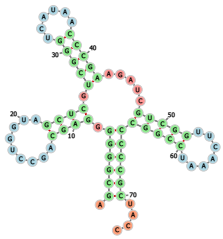
Pairing usually occurs between two nucleotides and leads to a certain spatial orientation relative to each other. Therefore, a so-called double helix is created whenever consecutive nucleotides form base

pairs. There is the basic distinction of two different so-called Watson-Crick pairs (A-U, C-G) and one wobble pair (G-U). While the C-G pair forms a strong interaction due to three hydrogen bonds, the A-U and G-U pairs are weaker as they are only capable of forming two hydrogen bonds. The noncanonical G-U pair exhibits an even weaker interaction due to a different geometry of interaction which also affects the shape of the double helical structure. Such slight differences and many other aspects of the pairing geometry are captured in the more detailed annotation scheme by Leontis and Westhof [104].

The general shape of an RNA molecule can be described in three abstraction levels. While the sheer polymer sequence is called primary structure, the secondary structure comprises information on the pattern of hydrogen bonds. These consist not only of the afore mentioned helices (or stems) but also of more advanced patterns which include unpaired bases such as hairpin-loops, internal-loops, multi-loops, or bulges. However, this level does not contain information on the specific atomic positions and possible additional atomic interactions, which the tertiary structure accounts for. The latter contains additional interactions like long range interactions, kissing loops and other weak tertiary interactions. Although it seems as if such a molecule architecture would generate a very restricted system, it is capable of creating a huge amount of different shapes [166].

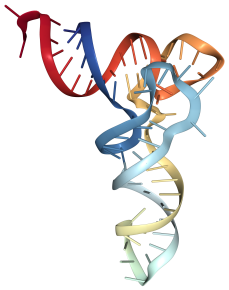
```
5'-AGCGGGGGGAGCA
GCCUGGUAGCUCGUCG
GGUCAUAACCCGAAGA
UCGUCGGUUCAAAUCC
GGCCCCCGCUACC-3'
```

Primary structure of the E. coli tRNAfMet A1-U72 variant [133]



Secondary structure indicates base-pairs and structural motifs such as helices and loops

Probably most interesting for the functionality of RNA molecules are unbound bases, and reactive or functional residues that are accessible on the surface of the molecule as these accomplish for the actual functions, such as enzymatic activity or binding abilities. While unbound nucleotides are sufficiently specified by the secondary structures, functional residues need a high resolution tertiary structure capturing the active site to be detected. However, these active sites are often specified in so-called sequence and structure motifs, which can be again detected by the primary and secondary structure alone. [85, 92, 183]



Tertiary structure shows detailed information about the spacial position of the nucleotides

A useful feature of RNA is the fact that for many purposes, most of the major aspects of the molecule can be sufficiently described by the secondary structure alone. It provides a strong initial scaffold which will not be fundamentally changed by the three-dimensional structure. Therefore, we use the secondary structure for many *in silico* calculations since it is easy to handle due to the existence of many algorithms mentioned in Section 3.2.

Moreover, the secondary structure has a major impact on the molecule's function, such as ribozymatic activity or the ability of binding ligands such as proteins, metabolites or other RNA molecules. This aspect was already summarized by Schuster et al. [166] in 1994. They stated that sequences correlate with genotypes while spatial structures relate to phenotypes. Therefore, RNA is one of the few biological examples where the genotype to phenotype mapping can be rel-

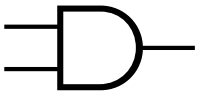
atively easily predicted computationally. Nevertheless, there is still a missing gap between this tight structure to function relationship. This concerns enzymatic activity as well as the ability to bind ligands. Both characteristics are hard to be revealed without experimental evidence. Only if measurements are describing the specific interaction or the ribozyme in detail, it is often possible to include these aspects in the prediction models [109].

1.2 INTRODUCTION TO LOGIC CONTROL AND REGULATION SYSTEMS

After the choice of material is made, we need to understand how regulation systems work. Probably a good start here is to look at such regulation systems in various domains and then try to map the knowledge to our desired field and combine it with existing know-how in this area.

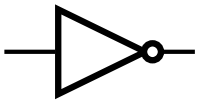
Control systems can be very diverse, ranging from a simple light switch that regulates the input command directly to huge systems such as rail networks or power plants, where many inputs regulate the behavior of complex devices or systems. They all have in common to use some predefined logic how to treat one or many inputs to react in a certain way on the output level. We generally differentiate between several fundamental types of control systems, which again can be combined to form bigger, more complex circuits. The easiest system is the sequential control system. It triggers a series of actuators in the correct order to perform an overall task. The next steps are often induced in a timely manner, thus without any external trigger.

The feedback control system is probably the most common form. It regulates a process in a way that it always follows a specific control signal, which is either a fixed or altering value. Such a system constantly checks and compares the output to the control signal and performs actions that bring their difference close to zero. Such feedback controller can not only be used to retain a certain state, but also to store or amplify input signals. The positive feedback controller triggers an additional amplifier as soon as an input signal is present leading to exponential growth and thus to the desired storage effect. In contrast, the negative feedback controller can be used to further reduce the output when the process variable reaches a predefined value, therefore promoting the stability of the system. Such feedback systems often lead to oscillations due to the temporal delay between measurement and effect of the response action. Sometimes this property is desired to construct so-called duty cycles which are also described in the activity of neurons, muscle fibers or even in more complex biological networks [4, 130, 170, 214]. However, many different techniques exist to avoid such oscillations and thereby lead to improved stability of the control system.



A	B	$A \cdot B$
0	0	0
1	0	0
0	1	0
1	1	1

A	B	$A + B$
0	0	0
1	0	1
0	1	1
1	1	1



A	$\overline{A}$
0	1
1	0

In order to calculate the output of a control system, boolean logic is often applied, which can handle very complex operations in a binary fashion. When implemented as electronic circuit, so-called logic gates implement the fundamental boolean functions to combine boolean inputs. The minimal set of logic gates needed to accomplish any boolean operation consists of AND ($A \cdot B$), OR ($A + B$) and NOT ($\overline{A}$) gates. However, additional elements exist that are able to store signals temporarily or permanently, such as transistors or switches. These building blocks and additional modules, such as sensors which act as inputs, can be wired to build a complex regulatory circuit.

However, as many control systems need to handle continuous inputs, boolean logic often fails in achieving certain tasks. In such cases, the design rules of fuzzy logic are preferred in order to build logic controllers of complex systems. In this case, inputs can not only be true or false, but also partly true i.e. true to a certain amount. In order to combine or use such inputs, the rules have to be formulated as a natural language with conditions, thresholds and boundaries. However, two signals can still be combined using boolean operators by implementing them as arithmetical operations. As a last operation of such control systems the output is defuzzified, leading to a distinct boolean output. If the rules and the logic are correctly defined, the system will produce a robust and true signal. Such systems are often used in digital electronics and mathematical models describing biological processes.

In the next section, I will describe and analyze the unique characteristics of biological control systems. Comparing these systems to well known logic implementations as electric circuits will help to realize the major differences and peculiarities of biological control mechanisms.

1.3 CHARACTERISTICS OF BIOLOGICAL REGULATION SYSTEMS

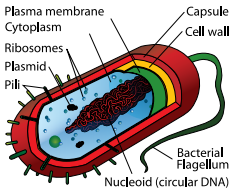
A biological cell could be envisioned as a huge control system with a variety of sensors for different inputs, such as inner or outer stimuli (e.g. environmental conditions) and with a well-defined logic how to react and modify its behavior in order to retain a stable system and thus to survive. However, it is unclear whether such a big biological system still acts in a deterministic manner. The latter means that it always reacts the same way given an identical set of inputs and inner state, i.e., distinct cells with the same inner state show the same reactions towards a set of identical stimuli. This is due to the high number of variables that have an effect on many parts of the system. Temperature, for example, will change the behavior of almost any component of the cell. Furthermore, it is almost impossible to attain an identical state of such a complex biological system, as almost any component involved bears its own complex state. This might be

a specific structural arrangement, a molecule's lifetime which influences e.g. the enzyme activity, or even simply interactions with other molecules. There are just too many variables and uncertainties to e.g. restore a complete biological cell and reproduce an identical output.

Thus, looking at biological systems we need to be aware that we cannot control or measure the full set of inputs and states of the control system. Therefore, effects such as fluctuations and varying outputs will always be present. However, if we look at a huge amount of more or less identical cells, the average of these systems shows deterministic properties. This can be considered as many single stochastic processes which lead to output signals with a certain distribution. This distribution will most probably be a stable one, given a set of well-defined inputs.

Another important characteristic of biological systems is the dose and time dependency of almost any reaction. In contrast to boolean logic, where any signal can either be on or off and there exist practically no time-delays between consecutive operations, a biological system uses continuous signals which might rise and fall with certain speeds after the states changed. Therefore such systems can be modeled much better using fuzzy logic where signals are continuous [3, 41, 208]. The final defuzzification of a measured signal is nevertheless a challenging task, as often only snapshots of the signal can be obtained, missing the full range of possible signal values. Additionally, there is the problem of fluctuations and stochastic outputs mentioned above. Thus, an answer to the apparently easy question whether a molecular toggle switch is on or off is already challenging to obtain. In many biological experiments this problem is solved by introducing positive and negative controls to be able to measure the boundaries of the extreme scenarios. It also helps to continuously measure the desired signal in order to determine the amount of background noise, leakage or the time period of action. Statistical analyses then help to defuzzify the signal to a boolean answer.

In biological systems, there exist no traditional wires as it is known from electronics. All components of a molecular circuit are present in a liquid environment and are thereby theoretically able to interact with any other element. To obtain a regulated system with cascades and organized pathways, wiring is achieved by allowing consecutive components to interact at so-called binding sites. These sites are highly specific in order to only allow interactions to their desired binding partner and are usually part of the component itself. Rewiring therefore means to alter the molecule itself, which is often a challenging task.



A procaryotic cell with its organelles (adopted from M.R. Villarreal)

Having binding sites still does not prohibit unspecific interactions. These so-called cross reactions need to be actively minimized when designing molecular components. One very efficient way to do so is to organize the circuits into organelles and thereby spatially segre-

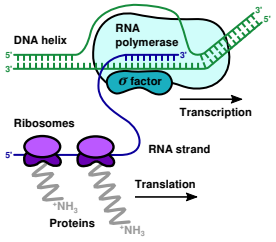
gate unrelated components from each other. Cells evolved that property by building structures such as a nucleus, mitochondria, endoplasmatic reticulum, golgi apparatus or many other forms of vesicles. The borderline of such structures is almost always formed by phospholipid bi-layers that contain various molecular gates and channels which only allow specific molecules to enter or leave the compartment. Another efficient way to obtain spatial segregation of various control circuits is cell differentiation. There, cells develop to fulfill a certain function by only expressing those genes relevant for a desired purpose. As a consequence, various control systems are separated from each other in specific cells such as neuronal or gut cells, communicating with each other by sending and receiving molecules on their surface. Famous examples for such communication interfaces are neurotransmitter and hormones in higher animals, or quorum sensing in cellular consortia of bacteria. In synthetic biology this concept was also tailored by engineering various cell types which implement certain logic gates and modules. By sensing the input on their surface using receptors, computing the internal logic and segregating a molecule as output, it becomes possible to easily combine basic components such as logic gates and build much bigger system by mixing various such cells. This might be a way to gain modularity and avoid interference and cross reactions [115].

2

INTRODUCTION TO GENE REGULATION IN BACTERIA

How did nature solve the problem of controlling cellular processes? What are common patterns used to regulate main tasks in cells? What exactly needs to be controlled in such a system? To tackle these questions I will focus on bacteria, as these unicellular organisms are generally much simpler than eukaryotes and therefore better understood. They lack cellular components such as the nucleus or mitochondria which eliminates a lot of complexity and makes their gene expression and regulation machinery much simpler.

In order to understand how bacteria are performing regulatory tasks, it is necessary to picture the central dogma how gene expression works. DNA serves as a storage molecule, encrypting the building plans for any component of the cell. DNA consist of regions which store the information for a molecule that has a function, so-called genes, and regulatory and organizational regions. The stored information can be read by the DNA polymerase for copying the information, or by the RNA polymerase (RNApol), which transfers it to newly generated RNA molecules. The latter process is called transcription. The resulting RNA is then capable of transmitting the information further, or can be functional, i.e. performs various tasks on its own. In the first case, the transmitted information gets processed by ribosomes that decrypt the genetic code and produce proteins from amino acids. This process is called translation. The produced proteins are then again able to perform many tasks in the cell.



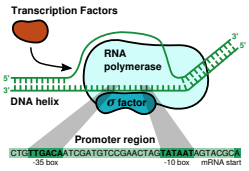
Transcription and translation in bacterial cells

Microorganisms struggle with a wide variety of continuously changing conditions in their environment, e.g. fluctuating nutrition conditions, changing pH value or availability of oxygen. They must be able to sense these changes and adapt their internal logic and controls on various levels. This can range from a simple fine-tuning of some expressed genes, switching to alternative metabolic pathways to major changes by undergoing developmental processes like sporulation or cell division. Any of these control systems and pathways finally leads to mechanisms regulating gene expression. They include transcriptional regulation, translational regulation and regulation of protein and RNA degradation. There also exist some regulatory processes in between such as splicing or protein and RNA modifications which are then called post-transcriptional and post-translational regulation. Keep in mind that in bacteria all of these mechanisms are happening simultaneously in the same spatial compartment, thereby influencing each other. [10]

To understand how transcription is regulated, it is important to first know how transcription, the process of producing RNA transcripts by reading the DNA template strand, works. It is generally split into three major steps called transcription initiation, elongation and termination. Initiation is the process of assembling the transcriptional machinery upstream of the transcription start site. It forms a stable complex able to start the catalytic reaction. This includes the formation of a so-called transcription bubble, a stretch of 12-14nt long melted and unwound base pairs within the double stranded DNA (dsDNA). While the initiation process is tightly regulated, elongation seems to be a simple catalytic approach which leads to monomeric ribonucleotides being polymerized at the 3′ end of the growing RNA chain. This polymerization is done within the active site of the DNA dependent RNApol, a huge enzyme consisting of two big β-subunits, two smaller α- and a ω-subunit.

In the end, termination stops the transcription and triggers the release of the nascent RNA chain into the cytoplasm. It also induces the recycling of the RNApol and post-transcriptional processes such as polyadenylation - the addition of a poly(A) tail to mRNA. Other RNA modifications might already happen cotranscriptionally by factors recruited to the transcript.

So how is it now possible to regulate transcription at these levels? Browning and Busby [17] mention four different mechanisms that control transcription initiation: promoter DNA sequences, sigma factors, small ligands and transcription factors (TFs).



Transcription initiation is mainly regulated by the promoter sequence, sigma factors, small ligands and transcription factors

The promoter is a sequence element responsible of attracting the RNApol to the DNA. It consists of a -10 hexamer and a -35 hexamer located 10 respectively 35 nucleotides upstream of the transcription start site and which are recognized by two different domains of the RNApol. Two more sequence motifs, the extended -10 element and the UP element, might help to promote this initial interaction. However, these motifs are not always present and show many variations [163, 169]. Nevertheless, the regulation due to various promoter sequences is a static one, meaning it cannot be easily modulated dynamically over time. It rather sets the ground level of expression for the individual genes as different promoter sequences lead to varying abilities to compete for the RNApols which are available in short supply.

Other important regulators are sigma factors which are part of the RNApol holoenzyme. These proteins are important for promoter sequence recognition by specifically binding to the DNA and help to stabilize the transcription initiation complex. There exists one major sigma factor in E. coli cells named sigma-70 which recognizes most promoters. Other factors accumulate only in response to specific

stresses and are thereby controlling many genes needed to respond to such situations. As these factors are also available in limited amounts, genes have to compete for their presence. The activity of sigma factors can sometimes be down-regulated by so-called anti-sigma factors which sequester the activity of the DNA binding ability.

Various small ligands provide good flexible alternatives to the before mentioned mechanism since they enable the cell to respond quickly and efficiently to environmental changes. Small ligands are capable of stabilizing or destabilizing the open complex at promoters, block interaction sites and much more.

TFs are one of the most important actors of transcription regulation. They modulate the initiation which efficiently leads to either activating or repressing RNA production. There are seven less specific factors which control a majority of all regulated genes whereas many others control only a single promoter and are therefore highly specific. The purpose of many TFs still needs to be revealed [123]. Life developed various different mechanisms of regulation involving TFs. Frequently, the production of a certain TF is directly regulated, meaning that its activity is directly related to its concentration. There also exist mechanisms to regulated the activity of an already present factor by small ligands or environmental stimuli. Similarly, TFs can be regulated by covalent modifications. For example, so-called response regulators can only bind the target DNA when phosphorylated by their respective sensor kinase. [17]

Transcription termination is a highly regulated process, too. In bacteria, termination regulation is particularly important as genes are often assembled in operons, meaning more than one coding region is associated to the same promoter and are therefore transcribed right after each other forming one long RNA transcript. Therefore, a termination signal after the first gene of such a polycistronic operon induces the down-regulation or complete silencing of the subsequent genes.

There basically exist two distinct mechanisms for transcription termination, the intrinsic termination which does not require any protein factors and the Rho-dependent termination. Intrinsic termination solely works by a hairpin structure formation upstream of a polyuridine RNA stretch. The weak interaction between RNA and DNA triggers the dissociation of the transcript from the template strand and the RNApol. Additional pause sites often help to support this process by allowing some time to complete the folding of the terminator hairpin and the required conformational changes in the RNApol to release the nascent RNA. Such pause sites and also protein factors influencing the transcription speed thus provide a great way for regulating this process. The Rho-dependent transcription termination requires a small protein called Rho. In E. coli this is a hexameric RNA-DNA helicase that binds to the RNA transcript at a specific

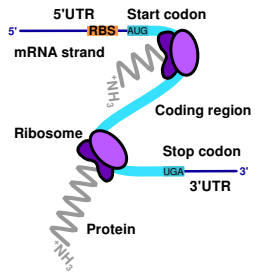
loading site. It is capable of interacting with certain TFs and translocates along the RNA chain until it meets the progressing RNApol. There it uses its helicase activity to unwind the RNA-DNA hybrid in the transcription bubble and thereby terminates the transcription, a highly dynamic process which is still not fully understood [168].

Both termination mechanisms are used to control gene expression by immature termination, where a termination site exists within the coding frame which only gets active under certain conditions. If so, the early termination will lead to an unfinished mRNA piece which will be an immediate target of degradation. Again, we have to distinguish the two termination machanisms. For intrinsic termination, the early terminator hairpin is down-regulated by the formation of a so-called anti-terminator structure. In contrast, for the Rho-dependent termination it is known that e.g. the boxA sequence on the RNA acts as a nucleation signal for assembling a ribosomal anti-termination complex. [15, 67]

The RNA strand produced using the described transcription process can now fulfill various purposes. If it is functional on its own by triggering enzymatic reactions due to a certain structural fold, it is called a ribozyme. The ribosomal RNA (rRNA), transfer RNA (tRNA) or small nuclear RNAs (snRNAs) belong to this group. Other produced strands help in various regulatory processes due to binding of complementary regions or proteins. Examples are regulatory small RNAs (sRNAs), transfer-messenger RNAs (tmRNAs) or 6S RNA [198]. However, many transcripts called mRNAs carry the genetic code to produce proteins and are passed on to the translation machinery.

2.2 TRANSLATIONAL CONTROL

Translation is perfectly suited as entry point for regulatory control, as the initiation of this process is a rate-limiting step in producing proteins. Nature developed many post-transcriptional regulation mechanisms that fine-tune protein expression at this level. To be able to understand these systems, I will shortly explain the structure of a mRNA and the general translation process first.

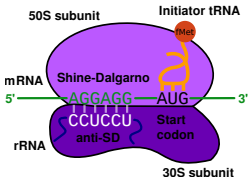


Translation of mRNA strands leads to the formation of proteins

A mRNA transcript consists of a region which contains the information for generating the according protein, and two flanking parts at the 3′ as well as the 5′ end, called UTRs that are not translated. The part encoding for the amino acid sequence is called coding region and always begins with the start codon and ends with an in-frame stop codon. The 5′ untranslated region (5′UTR) is the main player for translational regulation. Its most important sequence elements are the ribosome binding site (RBS), translational enhancers and the start codon. Together they are sometimes referred to as translation initiation region (TIR).

Translation initiation begins with the formation of a 30S ribosomal complex by interaction of the mRNA start codon with the anti-codon of fMet-tRNA which has joined the 30S ribosomal subunit. The 70S initiation complex is formed by the interaction with the large 50S subunit. Then, the first regular aminoacyl-tRNA binds the second codon in the reading frame which starts the elongation process. The elongation factor EF-Tu helps in this process. Initiation factors IF1-3 are bound to the 30S subunit and dissociate during the 70S formation. They determine the accuracy, fidelity and kinetics of the initiation process. [71]

In contrast to promoter sequences at the transcription level, where DNA sequence motifs represent a rather static regulation, the sequences in the UTRs are made from RNA. As a consequence, the mRNA sequence motifs can be functional by themselves without the need of protein factors and thereby highly contribute to flexible regulation, either in response to ligands or other stimuli like temperature. These sequences are therefore very diverse in bacteria and heavily influence the translation initiation as they represent the major variable component.



Translation initiation happens at the Ribosomal Binding Site (RBS)

The start codon is a triplet of nucleotides with the central one being a universally conserved U. Although the other two nucleotides vary, we find AUG to be most common. The bacterial RBS consists of a specific sequence motif called Shine-Dalgarno sequence (SD) which is complementary to the anti-Shine-Dalgarno sequence (aSD) contained in the 30S ribosomal subunit sequence. The distance of 4-9 nucleotides between the RBS and the start codon seems to be crucial for proper functionality. Efficient SD-aSD duplex formation promotes translation initiation by bringing the start codon near the 30S subunit. In this context it seems obvious that the RBS is responsible for recruiting the ribosome to the mRNA. [71]

However, not all bacterial mRNAs contain or need SDs. Some studies even suggest that only a minor fraction of bacterial genes contain a SD. The currently dominating opinion that most genes in bacteria contain a SD, is probably biased by the over-representation of SD dependent genes in E. coli. Many mRNAs do not even contain a 5'UTR. These so-called leaderless mRNAs use an ancestral mechanism of translation initiation conserved in bacteria, archaea and eukaryotic cells. [132]

Interestingly, there exists a whole regulation mechanism based on the leaderless expression system. The endoribonuclease MazF can remove 3'terminal nucleotides of ribosomes and even the complete 5'UTR of specific transcripts. Under certain stress conditions cells use this system to selectively regulate translation of various stress response genes due to the complementary effect of MazF on ribosomes and mRNAs. [131, 190]

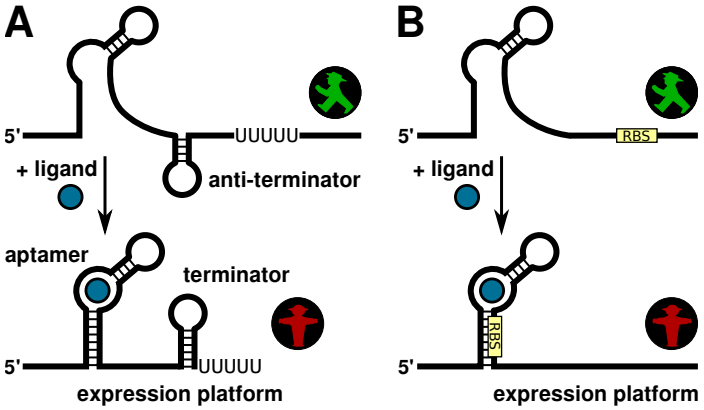
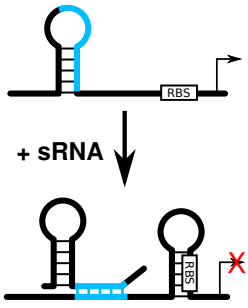


Figure 2: A riboswitch is capable of changing its state in response to an environmental trigger. A) Ligand binding stabilizes the aptamer which leads to terminator formation and thus transcription termination. B) Aptamer stabilization due to ligand binding sequesters the RBS and thus inhibits translation initiation.

Although it is hard to determine general rules valid for all genes, it was found that the accessibility and unstructuredness of a mRNA around the start codon is the most important feature for a successful initiation [71, Fig 1]. This is especially true in the absence of an SD. This feature is included in most of the common translation regulation patterns where translational repression is achieved through TIR sequestration, either directly or through structural rearrangements.

Antisense RNAs make use of direct binding and silence gene expression by base pairing to compatible anti-sense strands around the RBS which then blocks and represses translation [63, 88]. Moreover, so-called sRNAs, a class of small trans-encoded regulatory RNAs, often operate by directly binding the TIR and thereby blocking translation initiation [178]. These transcripts are typically only 40 to 200 nucleotides long [172] and diffuse through the cell cytoplasm until they find a proper target to interact with.



An sRNA triggered riboswitch regulates translation initiation

Binding of sRNA, metabolites or other ligands can also trigger structural rearrangements in the 5'UTR and thus change the accessibility of the TIR. This cannot only lead to gene repression but also to activation of gene translation [64, 116]. Such a switching region on the mRNA is called riboswitch. The term riboswitch comes from the material, namely RNA, and the ability to switch between at least two conformations triggered by an external stimulus. Riboswitches contain a so-called aptamer domain which is able to sense the ligands due to a highly specific binding pocket, and an expression domain responsible for controlling translation.

This enables a specific and tightly controlled expression of the target gene depending on the ligand concentration. The external trigger leading to the conformational switch can also be temperature. Such a temperature sensitive structural element is called a thermosensor. Although riboswitches are common mRNAs elements for translational

control, they are not limited to this purpose. There are examples of riboswitches that control Rho dependent and independent transcription, splicing and virus replication in a ligand-dependent manner [83, 180, 193]. I will introduce several natural examples in Section 2.5. [10, 48, 88]

2.3 REGULATION OF TRANSCRIPT STABILITY

Not only the production of RNA is highly regulated in a cell, but also the degradation. Nature put many antagonistic systems in place in order to ensure RNA stability as well as proper degradation of obsolete and unwanted transcripts. The main enzymes involved in cutting down the polymers are endonucleases and exonucleases. While the latter are only able to digest from the ends of an RNA strand, endonucleases are also able to cut in the middle of the polymer. As exonucleases are always active in a cell, eukaryotes developed structures on the mRNA to protect it against its activities exonucleases' activity. These include a modification at the 5′ end of the transcript called 5′CAP and the poly(A)-tail at the 3′ end. The latter is gradually shortened by the deadenylation dependent mRNA decay pathway which acts as a kind of destruction timer. Bacterial mRNAs lack such protective structures and therefore get degraded much faster. Some protection is obtained for mRNAs that are heavily translated and therefore constantly covered by polymerases and other factors involved in translation. Furthermore, stable structures at the transcript ends often help to increase the lifetime of an RNA molecule as they protect it from various RNases [36].

As soon as a transcript is defective, the degradation machinery will rapidly degrade it from both directions. In this process, the nucleases get help from other enzymes and proteins such as helicases which unwind complex RNA structures, chaperons which facilitate RNA-protein interactions and even polymerases that are capable of adding tails for attracting specific nucleases [142].

Endonucleases on the other side are often very specific for certain targets and are heavily controlled. As soon as they break a transcript apart in the middle, exonucleases will again take care of the dysfunctional RNA trash. A similar functionality is provided by self-cleaving ribozymes. These are RNA transcripts capable of cutting themselves or other pieces specifically. This only happens under certain conditions and when folded into the right catalytic active conformation. As a result, the target is again completely degraded. Such a targeted mRNA degradation is an effective way of post-transcriptional repression of the despised target gene [88].

We just learned that there are various options to regulate the production of proteins and RNA molecules and thereby the activity of functional components such as enzymes. As summarized in Figure 3 this generates a quite complicated construct of interplaying regulation mechanisms as most of them happen simultaneously and influence each other. This section will showcase some interesting characteristics of the different regulation approaches and highlight their advantages and limitations in terms of responsiveness, accuracy or efficiency. In a study by Shimoni et al. [171], mechanisms like post-transcriptional regulation by sRNA-mRNA interaction, transcriptional regulation by protein-DNA interaction and post-translational regulation by protein-protein interactions were compared which led to stunning results. Interesting aspects of riboswitch regulation were enlightened by Beisel and Smolke [10] due to mathematical modeling. However, many interrelations and facets of the gene expression regulation system are still unexplored and need experimental investigation as well as modeling to uncover universal patterns.

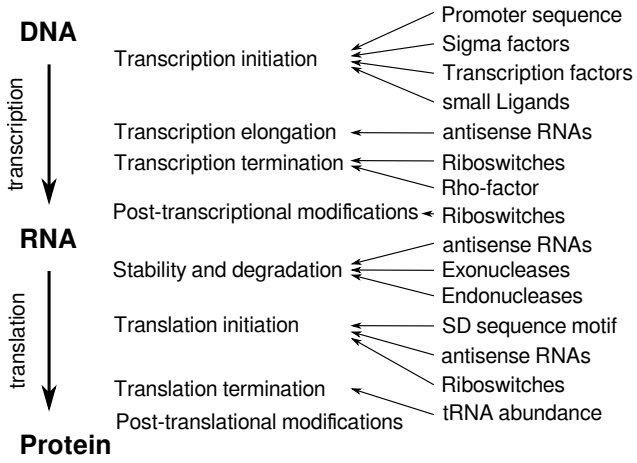


Figure 3: Transcription and translation are highly regulated at various phases of the process. This figure is a summary of the described regulation mechanisms in Chapter 2. The central column indicates the phases whereas the right column highlights possible control mechanisms. Adapted and simplified from Storz et al. [178, Fig 1]

TRANSCRIPTION regulation is an exclusively irreversible process for the individual molecules, meaning that an inhibited initiation or successful premature termination results in the destruction of the specific RNA molecule. Already produced RNA transcripts however will continue to perform their tasks until they eventually degrade [171]. Therefore, regulation on the level of transcription represents a rather static control system with slow response times but efficient inhibition

without much leakage. The latter is especially true for transcription initiation regulation where only a single trigger molecule can switch off the entire RNA production of this gene, as only one copy of DNA is present. As a consequence, only little concentrations of a trigger molecule are needed for a big impact on gene expression, therefore such triggers are often less specific, i.e. they can serve as regulator for multiple targets. [171] Such transcription initiation regulators are thus perfectly suited for global, long lasting regulations with little need for rapid responses. Transcription initiation regulation is also very efficient in terms of material usage as the signal interferes right at the beginning of the production chain. Thus, no or very little unneeded material is produced.

In contrast, regulation of transcription elongation or termination will produce possibly futile RNA strands and also high concentrations of trigger molecules are needed as every transcript has to be regulated individually. This however has the advantage of fine-tuning the amount of expression by varying trigger concentrations leading to different rates of termination or read-through. Contrarily, for transcription initiation control only binding affinities between the trigger and the promoter or RNApol can be used to adopt the sensitivity. For any transcription control system various rates also heavily influence properties of the mechanism, e.g. the rate of structural formations or the switching speed between states in comparison to ligand binding play important roles. Moreover, the dynamic range is bigger if the conformational changes between ligand competent and alternative structures happen faster than the formation of the terminator stem [10, Fig 4].

Another interesting aspect which heavily influences the response times is the nature of the trigger. If the trigger is an external stimulus which can be directly sensed by the target molecule, regulation will of course be quite quick. If the regulator needs to be produced first, sRNA regulation is even faster than protein regulation, as mRNAs are much longer than regulatory sRNAs and the necessary protein translation takes much additional time [171]. This is also true for translation regulation.

TRANSLATION is mainly regulated during the initiation, the process of TIR recognition and ribosome binding. As a major advantage, regulation on translational (or post-transcriptional) level is much faster and more responsive as RNA transcripts are already present in case of activation, and are only disabled but not destructed in case of repression [171]. This means that transcripts stay intact and will still be produced upon repression and hence, a subsequent reactivation can happen very fast. Such a quick response was experimentally measured by Isaacs et al. [88] to be less than five minutes from trigger to

gene expression change. Therefore, translational control is perfectly suited for quick and specific regulation of individual genes.

Translational regulation is often restricted to only regulate single, very specific targets. This specificity comes from the fact that the trigger molecules need to control many copies of a certain mRNA which requires sufficient concentrations of these triggers. This is especially true for cis regulating riboswitches which respond to external triggers such as metabolites or peptides. An exception are sRNA regulated mechanisms. Due to length differences, sRNA production is much faster than mRNA transcription, leading to much higher concentrations of sRNA compared to mRNA. Therefore, sRNA induced regulation might be also suitable for multiple targets.

Regulation triggered by sRNA has the additional advantage of being very responsive to external stimuli, which was shown by Shimoni et al. [171]. In such systems, the production rate ratio of sRNA and mRNA and the half live of their resulting sRNA/mRNA complex determine the steady-state level of the gene expression. On the other hand, if a protein is regulating the mRNA translation, this situation changes. As the resulting mRNA/protein complex cannot be degraded while being bound, a much smaller copy number of regulatory proteins is needed to trigger strong negative regulation in this case [171].

Nevertheless, the requirement of huge trigger concentrations also brings a huge advantage. Translational control mechanism can be easily fine-tuned by changing various properties of the system, resulting in different concentrations and thus response intensities. For example, a simple mathematical model of a 5'UTR riboswitch with a ligand binding competent state B leading to RBS sequestering and an alternative, non-functional state A shows that if the equilibrium is more on state A compared to B, we need higher ligand concentrations to see an effect on protein expression. However, the switching effect is then also much bigger and the dynamic range of activation grows [10]. Additionally, a higher ligand affinity to the aptamer increases the speed of the response in the initial switching, but also makes the reversible process slower [10, Fig 2].

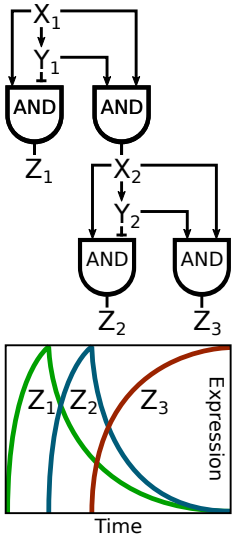
STABILITY regulation of RNA molecules and proteins is also great for fine-tuning response intensities by providing negative feedback systems. In such a system, higher concentrations of the target molecule lead to a faster and more efficient destruction which tightly regulates the availability of the corresponding RNA or protein. Such a feedback system is particularly important for the specification of an upper or lower bound of response. However, also the final concentration of RNAs or proteins can be regulated by increasing or reducing degradation. The ratio of the production and degradation speeds define the amount of change in a system, while a higher degradation or

lower production will decrease the overall concentration of the target molecule.

Regulation of stability might also be important for global regulation with the need of quick response times. In such a system, a released trigger that degrades all transcripts or proteins of a certain class leads to a fast but not necessarily long-lasting response while transcriptional control makes sure to retain the achieved state on the long run. Thus, the introduction of this additional degradation mechanism significantly accelerates the regulation response. In addition, mRNA stability can also be influenced by post-transcriptional processes and modifications such as 5′ capping or polyadenylation. Both heavily influence the half-time of the transcript and the absence of such a modification strongly promotes RNA degradation.

All the described regulation mechanisms on the level of transcription, translation and stability are tightly connected, which makes it difficult to analyze such complete regulation systems. Furthermore, there are just too many variables in already one single regulation pathway to be able to easily understand it in a timely and context sensitive manner. Therefore we need mathematical models with measured and predicted parameters to see the effects of input variations and modulation. This is especially important if we combine more pathways to complex systems, as it is the case in a living cell. Some developed methods use regulation networks to model such interactions. With that it is possible to analyze the complete network in order to detect motifs like feedback loops, signal sinks or regulatory pinholes. [4, 130, 159, 170, 214]

In summary, it is notable to say that gene regulation systems are very complex. The actual properties of the mechanism often depend on small details in the implementation and many diverse factors influence the output. However, main variables of such systems are the concentration of the interaction partners, the interaction affinities, production speed of products as well as the lifespan of the involved molecules. These main characteristics influence response times, sensitivities to the trigger signal or the ability to hypersensitivity or background noise reduction.



Transcription network of B. subtilis spore development. Four feedforward loops generate pulses of Z_1 and Z_2, and a delayed Z_3 expression. Adopted from Alon [4, Fig 7]

2.5 RIBOSWITCHES IN GENE REGULATION SYSTEMS

The previous sections emphasized that RNA has a central role in regulating gene expression at any level. Of course there exist many protein-based control systems that regulate gene expression, but RNA is in most cases involved and works together with other compounds. In this section I want to focus on one particular module already mentioned several times, namely the riboswitch, as this module was and still is perfectly suited as a genetic switch in ancient and modern settings. Early life forms probably controlled molecular functions by

using riboswitches even before proteins emerged and these elements are still crucial parts of regulation. [64, 119, 203]

In Section 2.2 I already mentioned that riboswitches are located in 5'UTRs of mRNAs where they regulate translation initiation by sequestering or releasing the RBS. Similarly, riboswitches can also regulate transcription termination by triggering either the formation of a terminator hairpin or of an anti-terminator structure in response to ligand binding. Newer studies even discovered riboswitches in Rho-dependent transcription termination systems [83]. The authors speculate that this kind of control mechanism is widespread as many riboswitches appear to lack Rho-independent terminators or RBS-sequestering hairpins. Wachter et al. [193] found riboswitches to be involved even in the control of splicing. [185]

All riboswitches share the ability to sense external triggers which leads to the change of their conformation and as a result the transduction of the signal to the underlying control system [167]. These triggers are very diverse and range from metabolites, amino acids, cofactors, and metals to peptides or sRNAs. The binding is often highly specific as the ligand is recognized with high selectivity and affinity. Examples for riboswitch triggers in gene control are thiamin pyrophosphate (TPP), flavin mononucleotide (FMN), S-adenosylmethionine (SAM), guanine, lysine or glucosamine-6-phosphate (glmS gene). It was shown for many of them that no protein factors are involved in these systems [167]. Other triggers can be temperature or ion concentrations such as such as Mg^{2+}. Temperature-dependent RNA structures are called RNA thermometer. Kortmann and Narberhaus [98] categorized several such devices found in bacteria into molecular zippers and switches. While zippers gradually shift the equilibrium between the conformations with the temperature and therefore step by step change the gene expression, switches rapidly change their structure once a certain threshold is reached. The latter are often found in front of cold shock genes [138, 160].

The reversible structural switching between several states, at least one active and one inactive state, is crucial for the function of a riboswitch [167]. This multi-state property is one reason that makes riboswitches so fascinating since their functionality does not depend on additional protein factors. Moreover, it is not determined by just a simple sequence motif as it is the case for sRNAs or antisense RNAs. Only the right structural conformations combined with the aptamers ligand binding functionality will form a working device.

Quarta et al. [144] therefore analyzed the structural changes and the dynamic energy landscape of some riboswitches using *in silico* methods. They described that many riboswitches exhibit several phases during their transcription and therefore called the various RNA lengths or states *sensing window*, *downhill folding window* and *functional window*.

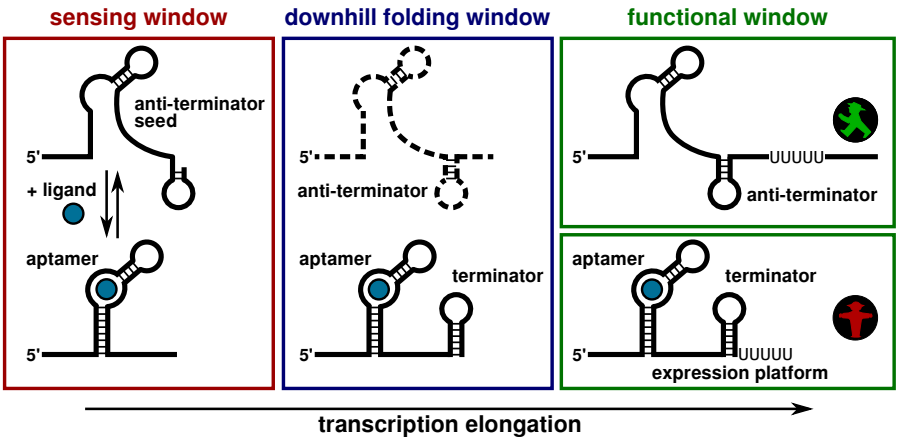


Figure 4: A riboswitch regulating transcription termination (Figure 2 A) while being transcribed. During the *sensing window*, the aptamer and the structure containing the anti-terminator seed are able to switch easily upon ligand binding. As transcription elongation continues, the current state is stabilized, either through terminator or anti-terminator formation (*downhill folding window*). Thus, switching is not possible any more. Finally, as soon as the poly-U stretch is transcribed, termination happens and we reached the *functional window*. Adopted from [144, Fig 3]

These windows can be in any order depending on the functionality of the riboswitch and the nature of the aptamer and ligand binding.

During the *sensing window* the ligand binding competent structure must be already available as well as some part of the alternative structure. During this phase the riboswitch is able to switch its conformation quickly and allows binding or dissociation of the ligand. This binding leads to the decision whether to favor one or the other structure. The *downhill folding window* stabilizes the current conformation in order to obtain a compact and stable structure. This can either be one of the structures after the decision through the trigger, or the ligand competent structure at the beginning of the transcription. If this window occurs after the sensing window where the decision takes place, the amount of stabilization for both conformations influences the prioritized activity of the riboswitch and the ability to revert the decision unidirectionally. In extreme cases this phase can lead to a complete insensibility of the full transcript to ligand-binding. In the so-called *functional window*, the alternative state and the ligand binding state are distinct and separated from each other on the energy landscape. At this state it is not easily possible to switch between the functional conformations any more due to a high energy barrier in between. [144]

If the sensing window comes first or second during the transcription of the riboswitch, it exhibits a kinetic or cotranscriptional mechanism. However, if the sensing window occurs last it underlies a thermodynamic mechanism. Riboswitches from the latter class are able to change their state and sense ligands even if fully transcribed.

Quarta et al. [144] highlight that transcription pause sites can ultimately change the kinetic behavior of riboswitches.

Screening the literature for natural examples revealed that validating natural riboswitches is as challenging as designing some since the relationship between the RNA structure and the nature of their cognate metabolites has to be understood. This is often only possible through binding studies and crystal structure studies, important details often missing in previous studies. Nevertheless the understanding of naturally occurring riboswitches is crucial for the ability of synthetic design. We need to know important features such as binding affinities, detailed structural conformations in natural conditions or information about the switching kinetics in order to design new variations. The Breaker lab [119, 203] analyzed and summarized many natural riboswitch classes and explained details of their functionality.

The previous chapters revealed why it is a good idea to use RNA as material to artificially regulate gene expression and other cellular processes in bacteria. They explained how general processes are controlled and showed general pattern of control systems in biological environments. We also learned which mechanisms nature implemented to regulate gene expression and finally, which role RNA and especially riboswitches play in regulatory tasks. Now it is time to focus on the design of RNA devices to artificially regulate biological processes in gene regulation.

3

INTRODUCTION TO RNA BIOINFORMATICS

In this chapter I will shortly introduce formal definitions of RNA bioinformatics that are required for designing these polymers by using computational power. Many highly sophisticated *in silico* methods are available to predict the behavior of RiboNucleic Acid (RNA), including algorithms for structure prediction and a quite detailed measured energy model. Using these computational methods it is therefore possible to obtain information about the structure and therefore also some idea about functionality directly from the sequence.

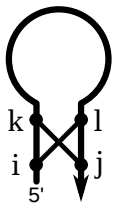
3.1 FROM SEQUENCE TO STRUCTURE

To begin, I will briefly outline the formal definition of an RNA sequence and structure as well as their relation to each other in this section.

Let us define an RNA sequence with length n as $x = \{x_1, x_2, x_3, ...x_n\}$ where each $x_i \in \mathcal{A} = \{A, C, G, U\}$ represents a nucleotide and $\mathcal{A}$ is the alphabet of nucleotides. For a given RNA sequence (x), I will denote by $x_{i,j}$ the subsequence from position i to j. The nucleotides of x are able to form base pairs according to the set of base pairs $\mathcal{B} = \{AU, UA, GC, CG, GU, UG\}$. This set contains the *Watson-Crick* base pairs as well as the so-called *Wobble pair* between G and U and can also be defined as a pairing matrix ($\mathcal{P}$):

	A	**C**	**G**	**U**
A	0	0	0	1
C	0	0	1	0
G	0	1	0	1
U	1	0	1	0

A set of base pairs (i, j) with $1 \leqslant i < j \leqslant n$ on a specific RNA sequence (x) is called secondary structure (Θ). Each base pair must be in the set of base pairs $(x_i x_j) \in \mathcal{B}$ and each nucleotide is allowed to pair with at most one partner. That means for any two unequal distinct base pairs $(i, j), (k, l) \in \Theta, i < k$ that belong to the same RNA structure, the condition $\{i, j\} \cap \{l, k\} = \emptyset$ must be true. If base pairs cross ($l > j$), this secondary structure (Θ) contains so-called pseudoknots. In the following, I will only use structures without pseudoknots, if not stated otherwise.



The definition of a pseudoknot

For a better understanding and illustration of these often quite complex structures, various visual representations have been developed

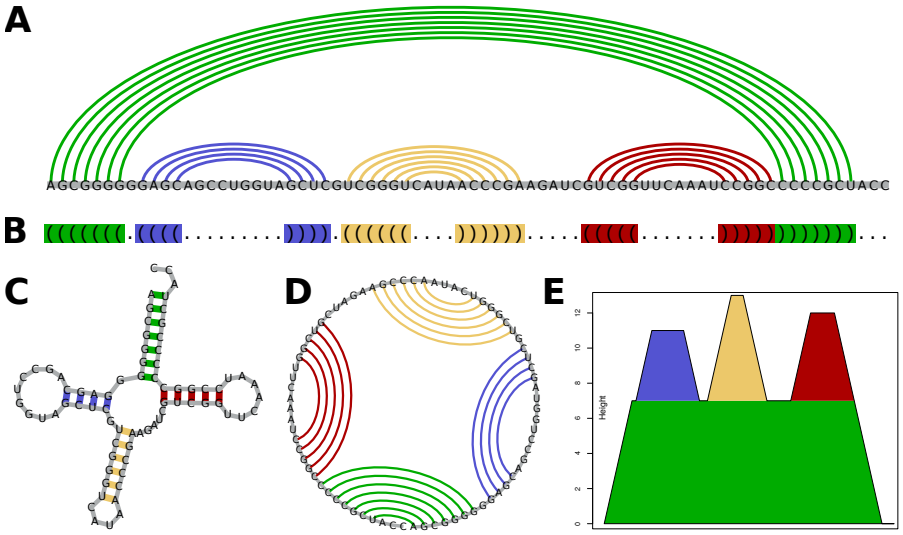


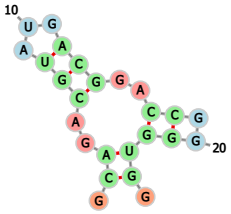
Figure 5: Representations of RNA secondary structures on the E. coli tRNAfMet A1-U72 example. A) The arc plot connects bases with an arc to depict pairing. B) The dot-bracket notation is widely used as its a machine readable form only consisting of dot and bracket characters. C) The radial drawing brings pairing bases in close proximity and thus is a rather native representation. D) The circular drawing arises when outlining the arc plot on a circle. This representation is well suited to picture long range tertiary interactions. E) To obtain a mountain plot, a peak with height one that spawns between the connected positions is drawn for each base-pair. This plot us useful when comparing large structures.

over time. For example, a pair of a sequence and structure can be presented as a *Circular Drawing* first used by Nussinov et al. [141]. For a more native representation, so-called *Radial Drawings* are useful, or *Mountain Plots* and *Arc Plots* are perfectly suited to compare huge RNA structures (Figure 5). Structures are very often represented in the dot-bracket notation, a simple string of dots for unpaired bases and matching brackets for base pairs. As an RNA sequence (x) can fold into various different secondary structures, a common representation to display multiple conformations is to intersect several circular drawings in various colors. This is easily possible, as such a circular drawing is from a more mathematical point of view nothing else than a graph representation. This is possible since the secondary RNA structure can be represented by a graph, where the vertices $V = \{1, \ldots, n\}$ represent nucleotides and the edges $E = \{(i, j)\}$ indicate base pairing. Such an undirected graph is also-called dependency graph $G = \{V, E\}$. In Part ii it is shown how this notation is used to sample sequences that are compatible to multiple structures.

Moreover, there exists the term of an RNA shape which is a mathematical abstraction of not only one single secondary structure (Θ), but a set of secondary structures. Such a shape describes these secondary structures in a way we would also naturally communicate their similarities and structural features. The arrangement and the number of

helices for example characterizes a particular set of secondary structures quite comprehensively while omitting details such as loop sizes or detailed lengths. Various abstraction levels help to regulate the level of details in this representation. [60, 90]

How are sequences and structures related to each other? The following mathematical formulation helps to get an understanding of this relationship and how they depend on each other. The total number of possible RNA sequences (x) with length n is $|\mathcal{A}|^n$. Let us define the set of all those sequences as sequence space (Ω). This space can be traversed by simple point mutations, the distance between two arbitrary sequences can be measured by the Hamming distance which is simply the number of differences between the two RNA sequences. Subsequently, for a given RNA sequence (x), the structure space ($\overline{\Omega}$) is defined as the set of all possible secondary structures Θ_x.



_[_[]_[]_
[[][]]

Example of an RNAshape representation, abstraction level 1 (top) and level 2 (bottom).

Several distance measures have been proposed as a metric for the structure space [80, 166]. Frequently used are the tree edit distance which is based on a tree comparison and the base pair distance which counts the number of distinct base pairs between two secondary structures from $\overline{\Omega}$. In contrast to the previous measurements, it is not a trivial task to find an explicit formula for the number of possible secondary structures (Θ) for a given x. However, it can be calculated in a recursive manner [69]:

$$|x_{i,j}| = |x_{i+1,j}| + \sum_{i<k\leqslant j \mid (x_i x_k)\in\mathcal{B}} (|x_{i+1,k-1}| \cdot |x_{k+1,j}|) \tag{1}$$

The total number of secondary structures (Θ) that a RNA sequence (x) is able to form, is given by the sum of the first nucleotide being unpaired and all possibilities for this nucleotide to form upstream base-pairs. Omitting the pairing condition $(x_i x_k) \in \mathcal{B}$ in the sum of Equation 1 yields the maximal amount of possible secondary structures (Θ) for any RNA sequence (x) of length n. Note however, that Equation 1 ignores the commonly used definition of a hairpin loop being at least three nucleotides long. If we include this constraint, the number of structures is significantly lower [69].

Another interesting measure in this context is the number of different RNA sequences (x) that can fold into the a given secondary structure (Θ). This however requires some knowledge about the inverse folding problem which will be discussed in Chapter 6. It is worth noting that due to the combinatoric explosion of n independent choices, the number of RNA sequences of length n by far exceeds the number of secondary structures (Θ) possible. Recently, Clote [31] developed an algorithm to analyze the expected degree of secondary structure networks, thus further characterizing the structure space ($\overline{\Omega}$).

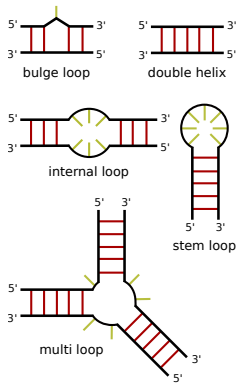
The previous definitions only introduce the relationship between sequence space and structure space without taking any energies into account. In order to be able to predict a structure that is thermodynamically likely in nature, we need to include a measured energy

parameter set [124, 187] which enables us to put weights in form of energies on every structure in $\overline{\Omega}$.

This process of predicting a thermodynamic likely structure is called RNA folding and is the actual mapping between the two metric spaces Ω and $\overline{\Omega}$. Schuster et al. [166] and follow-up publications intensively studied this relationship and highlighted several interesting properties. First of all, they realized that RNA sequences fold into relatively few common shapes and many rare ones. Interestingly, mutations of sequences with small Hamming distances often lead to very similar structures. However, even a single point mutation can in some cases substantially alter the shape, leading to many structures in the ensemble with big tree edit distances to the original. Sequences with Hamming distances above three, i.e. with more than three distinct base positions, will only very rarely produce similar structures and above Hamming distance fifteen they effectively produce uncorrelated structures. Nevertheless, it seems as if neutral paths in the sequence space Ω – neighboring RNA sequences (x) that produce identical secondary structures (Θ) – can be seen in one fifth of all random RNA sequences (x) throughout Ω. [69, 166]

The next section will introduce how such insights were possible by incorporating the energy parameter set to predict minimum free energy (MFE) structures and perform other thermodynamic calculations with our model of an RNA molecule.

3.2 Structure Prediction and Thermodynamic Calculations



Next to the double helix, there are various types of loop types possible (CC-SA 4.0 by Florian Eggenhofer and Dominik Steininger).

RNA structure prediction and other *in silico* calculations of RNA behavior are only possible due to the availability of measured energy parameter sets. This parameters can be obtained through melting experiments of small oligo-nucleotides and tabulated as free energies [187]. Initially, only energy contributions of Watson-Crick interactions were known, but as a significant contribution to the overall free energy of an RNA polymer comes from other interactions such as stacking of adjacent bases, contributions from loop regions, terminal mismatches or dangling ends, the energy model was soon updated to a nearest neighbor energy model including these contributions [187]. Using this energy model, the total free energy of an RNA molecule can be determined by summing over all individual contributions. The additivity of these energies is an important aspect for the availability of advanced RNA bioinformatic methods and tools.

However, for a given sequence predicting the structure with the minimal free energy, the MFE, would still require to enumerate all possible structures Θ_x and then calculate the total free energy for each structure $E(x, \Theta)$, which would be a very demanding task. Waterman et al. [197] formulated a recursive decomposition scheme which is ad-

missible to RNA secondary structures including only secondary base pair interactions without pseudoknot crossing pairs. This recursion paved the way for numerous dynamic programming (DP) algorithms that enable to solve the afore mentioned task in feasible time. Its first application was the calculation of the number of compatible structures Θ_x (Equation 1) [69].

The first DP approach was developed in 1978 by Nussinov et al. [141] and tackled the problem of finding the MFE structure by solving the so-called maximum matching problem. The DP table was filled only with contributions of Watson-Crick base pairs and maximized the number of those pairs. The MFE structure was later obtained as a list of base pairs by backtracking through the DP energy arrays.

Zuker and Sankoff [219] improved this algorithm by including the individual energy contributions of hairpin loops, interior loops and multi-branch loops. This was possible by filling several additional DP tables by applying additional generation rules. This algorithm uses the full nearest neighbor energy model and is therefore still used by implementations such as *RNAstructure*, *UNAfold* or the *ViennaRNA Package* [108, 122, 212].

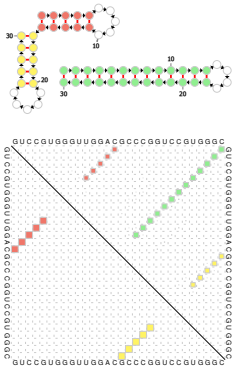
To predict not only the structure with the lowest free energy but also suboptimal structures, Zuker [218] modified the backtracking approach of the afore mentioned algorithm. This allowed to generate for any chosen base pair the optimal secondary structure (Θ) that includes this pair. However, it soon turned out that although this approach generated many suboptimal structures, in some cases it missed important ones. Wuchty et al. [209] therefore came up with an algorithm that is able to enumerate all secondary structures in a certain energy range above the MFE structure. This is possible due to the construction of a special list of partial solutions during the backtracking procedure.

By modifying the Zucker algorithm's generation rules, McCaskill [126] made it possible to calculate the canonical partition function of the system and therefore introduced the field of statistical mechanics in RNA bioinformatics. The partition function (Q) describes a system with statistical properties in the thermodynamic equilibrium and is a function of the temperature (T) and the micro-state energies, in our case the energies of all the possible Θ in the structure space denoted as $E(\Theta)$.

$$Q = \sum_{\Theta \in \overline{\Omega}} e^{-\beta E(\Theta)} \tag{2}$$

The temperature is included in the term $\beta = \frac{1}{kT}$, with $k \approx 1.987 \frac{cal}{molK}$ being the Boltzmann constant. Using the partition function, we can calculate many important characteristics of the system, such as the total free energy of all states, given by

$$G = -\frac{1}{\beta} \ln Q \tag{3}$$



Dot plot showing the base-pair probabilities of the ensemble and the MFE structure (lower triangle)

As the states are Boltzmann distributed in the equilibrium, the probability of a certain Θ in the ensemble is given by

$$p(\Theta) = \frac{e^{-\beta E(\Theta)}}{Q} \tag{4}$$

The probability to observe a certain structural feature such as a hairpin stem, an unpaired region or a multi-loop within all possible structural states needs the extension of *hard constraints* included in the McCaskill algorithm [109]. This probability can be obtained by dividing the constraint partition function $Q_{\mathcal{F}}$ by the full partition function (Q).

$$p(\mathcal{F}) = \frac{Q_{\mathcal{F}}}{Q} \tag{5}$$

The hard constraint partition function only takes structures compatible with the specified structural features into account and can be calculated analogously.

$$Q_{\mathcal{F}} = \sum_{\Theta_{\mathcal{F}}} e^{-\beta E(\Theta_{\mathcal{F}})} \tag{6}$$

On the other hand, *soft constraints* allow to take all states into account while preferring some structures with a certain structural feature. This enables to add a certain bonus energy to structural features which can be used to model RNA-ligand interactions or to perform experimental guided folding predictions [109].

Using these algorithms, it is also possible to predict the folding of interacting RNA molecules using various tricks. This can either be done by concatenating two interacting sequences and treat the junction as a simple loop region without any energy contribution, or by using a two-step approach such as *RNAup* [136]. The latter first predicts a potential interaction site using opening energies of local structures, followed by calculating the interaction energy and structure of the RNA hybrid.

There exist many more extensions to the afore mentioned algorithms for the calculation of various properties of an RNA sequence such as the structural diversity, positional entropy, structures with maximal expected accuracy, or the density of states. An extensive summary of all the possible calculations and history of algorithms was reviewed by Lorenz [107][111]. Most commonly used programs implementing all or a subset of these algorithms are `mfold/ UNAfold` [122], `RNAstructure` [212], the `NUPACK` package [215] and the `ViennaRNA` package [108].

3.3 ENERGY LANDSCAPES OF RNA MOLECULES

The previous section introduced how structures and their properties in the ensemble can be predicted in the thermodynamic equilibrium.

However, these methods tell us nothing about how the ensemble changes over time and how all the structures are related to each other. In the following I will discuss the kinetic properties during structure formation, the refolding from one structure into another and the process of RNA transcription in conjunction with folding, called cotranscriptional folding.

To get a better understanding of the matter, I will first introduce the concept of an energy landscape. This landscape is a high-dimensional space of all possible secondary structures (Θ) in the ensemble $\overline{\Omega}$ weighted by their free energy $E(\Theta)$. The property of being high-dimensional comes from the introduction of a neighborhood relation which can be an one to many relationship. Two structures are neighbors if they can be converted into each other by applying only one move of the chosen move set. The move set can be quite diverse but it normally contains at least the formation and dissociation of a single base pair, two basic modifications reversible to each other [50]. More sophisticated move sets may contain moves such as shift-moves or helix slipping. The choice of the move set highly influences how coarse grained the calculations are and thus the runtime of the algorithms. Within a move set, it must always be possible to access any of the structures from any other structure through a series of moves, i.e. the move set connects all structures in $\overline{\Omega}$, otherwise ergodicity does not hold.

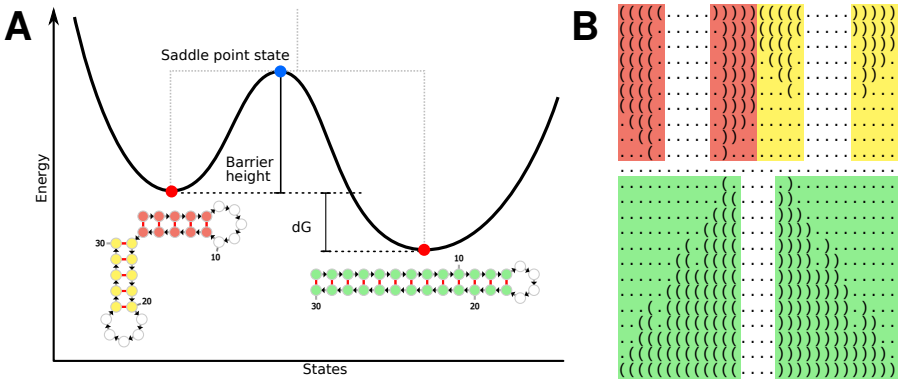


Figure 6: A) An energy landscape representation of a bistable RNA molecule. Two local minima are separated by an energy barrier, the state with the highest energy on this path is called *Saddle point*. Grey dashed lines indicate how the *Barrier tree* for this landscape looks like. B) Folding path obtained with `Findpath` between the structures with a move set that allows only the formation and dissociation of single base pairs.

With the consecutive application of moves, we can generate multiple paths from one to another structure. The shortest distance possible is thereby given by the definition of the move set, while the longest acyclic possible path is restricted by the size of the structure space ($\overline{\Omega}$). Algorithms such as `Morgan-Higgs` [134] or `Findpath` [51] are able to detect the shortest path between two structures, also called the *direct path*. However, such a path must not necessarily be the

energetically favored way in the energy landscape to reach the targeted structure, as the path might contain intermediate states with quite high free energies. The intermediate structure with the highest energy is called *saddle point* and the energy difference to the start structure is denoted by activation energy or energy barrier (Figure 6). Algorithms that are able to find paths with minimal energy barriers rather than short paths, also-called *indirect paths*, are for example the `Morgan-Higgs-revisited` [134] or the `RNAtabupath` algorithm [39].

The Barriers flooding algorithm [52] is able to calculate *indirect paths* by an exact approach that requires the exhaustive enumeration of a subset of structure space ($\overline{\Omega}$) [120]. The trick to be able to handle this demanding task is to coarse grain the energy landscape by merging structures (micro-states) into gradient basins (macro-states). A gradient basin consists of a local minimum structure and any structures that ends up in this local minimum through a gradient walk. The latter is defined as a path where the next neighbor is always chosen by the steepest energy difference. Using this algorithm it is possible to generate a conclusive representation of the energy landscape. The so-called *barrier tree* shows local minima, saddle points and energy barriers in a tree representation. Moreover, the algorithm is able to calculate the transition rates between all macro-states, an information that is necessary for solving the master equation of a Markov process described below. Unfortunately, the algorithm is despite the coarse graining still very demanding and is therefore only feasible for RNA molecules of up to about 120 nucleotides.

3.4 KINETIC FOLDING PREDICTIONS AND CALCULATIONS

There basically exist two major methods to include time into the system and therefore allow for kinetic folding predictions. The system can be described as a continuous-time Markov chain of elementary moves from the chosen move set. In this system the probability for an RNA molecule to adapt a certain state j with $\Theta_j \in \overline{\Omega}$ at time t is denoted as $P_j(t)$. The dynamic change of conformations is then governed by the master equation

$$\frac{dP_j}{dt} = \sum_{i \mid \Theta_i \in \overline{\Omega}} k_{i \to j} P_i(t) \tag{7}$$

where Θ_i and Θ_j are any two states on the energy landscape and $k_{i \to j}$ describes their transition rate, which is zero if the two states are not directly connected by the move-set. After obtaining the transition rates from the `barriers` program, the master equation (Equation 7) can be solved which results in a global picture of the folding kinetics over time. Although this equation could also be applied to micro-states, it is advantageous to only use it for predicting the kinetics

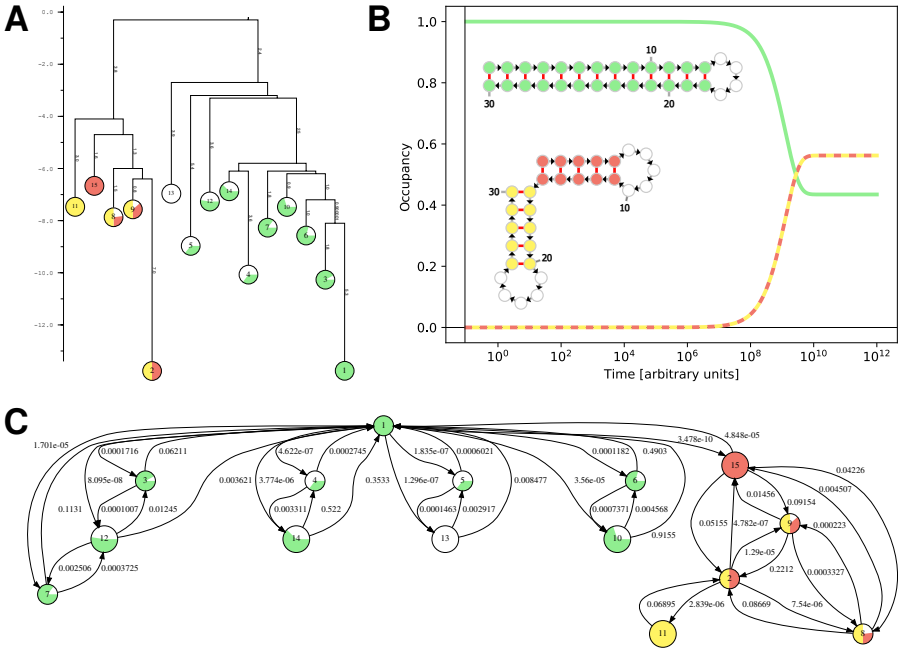


Figure 7: Energy landscape, kinetic trajectory and transition rate graph of a bistable example riboswitch (see also Figure 6). A) Barrier tree representation of the energy landscape. The macro-states are visualized by showing their stem colors in addition to the similarity. B) Kinetic simulation starting in the green conformation. The yellow/red depicted state exhibits a major occupancy after 10^9 arbitrary time units. C) Transition rate graph produced by `barriers`. It shows the rates between the considered macro-states.

of macro-states in order to save computational resources. This approach is implemented in the `treekin` program [207]. A downside of this approach is the necessity of exhaustively processing numerous secondary structures (Θ) in the structure space ($\overline{\Omega}$) in order to obtain the gradient basins and their transition rates, which is a demanding task. An example[1] of a `barriers` and `treekin` output is shown in Figure 7.

A faster way is a Markov Chain Monte Carlo approach that stochastically samples possible kinetic trajectories through the structure space ($\overline{\Omega}$). This will however only lead to a local picture of a single molecule over time. In this approach, each move is randomly chosen from the set of move sets and applied to the current structure $\Theta_i \in \overline{\Omega}$ in order to obtain the temporal subsequent structure $\Theta_j \in \overline{\Omega}$. The transition rates $k_{i\to j}$ and $k_{j\to i}$ can be calculated on the fly from the energy difference between the two states (Equation 8).

$$K = \frac{k_{j\to i}}{k_{i\to j}} = e^{-\frac{\triangle G_{j\to i}}{RT}} \tag{8}$$

Thus, no enumeration of $\overline{\Omega}$ is necessary. Furthermore, it is computationally feasible to sample folding trajectories of micro-states that

1 GUCCGUGGGUUGGACGCCCGGUCCGUGGGC

way. Such strategies are implemented in the programs *Kinfold* [50] and *Kinefold* [210]. To get a global statement concerning the whole ensemble, it is necessary to sample many folding trajectories that way and use statistical approaches to analyze them.

Transcription, the process of producing an RNA molecule by elongating it, is not decoupled from folding events in a cellular environment. Therefore, it is important to take structure formation and the kinetics of it into account. The program *BarMap* [81] does this by subsequently elongating the RNA sequence and simulating the kinetics for the current transcription state. The latter is achieved by using the `barriers` and `treekin` approach starting from the final structural ensemble of the previous transcription state. This requires a mapping of structures from the consecutive transcription states, which can be a challenging task as new states might appear or previous ones vanish.

This chapter gave a brief overview of the methods available to predict structures, calculate energy landscapes and to run kinetic simulations given an RNA sequence as input. With this introduction it is now time to focus on the reverse problem of obtaining a sequence given structures, energy landscapes or even kinetic folding curves.

4

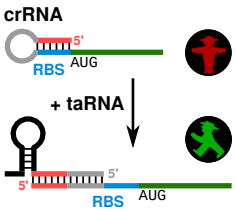
HISTORY OF DESIGNING RNA SWITCHES

The previous chapters demonstrated why non-coding RNA provides an interesting and perfectly suited framework for designing devices to implement artificial logic in a cellular context. So how is it possible to obtain an RNA molecule that is able to accomplish the desired task? It could be used to detect a certain metabolic state, balance metabolic pathway expressions, tightly regulate toxin genes or even to detect environmentally harmful chemicals [113]. Moreover, such RNA devices can be helpful for the visualization of certain cellular molecules and for analytic purposes.

4.1 BEGINNING OF RNA DESIGN

In the early beginning of RNA engineering studies, researchers retrieved naturally occurring systems, analyzed them in detail and eventually modified the RNA sequences in order to improve and optimize the system. Such mutational studies were used to gain higher specificity for a target molecule, to adapt the system to a new environment such as exchanging the trigger molecule or the target gene, or to circumvent undesired crosstalk to other devices. [113, 118, 143, 196]

Designs were generated by combining known sequence elements with well-studied functionality to generate a completely new logic device. For example, Isaacs et al. [87] used a ribosome binding site (RBS) (see Section 2.2) and generated a complementary region upstream to sequester the site in a stable hairpin. A second molecule that is complementary to the repressor region acts as small RNA (sRNA) and activates the translation by opening this hairpin. Neupert et al. [140] built a similar device triggered by temperature changes. A region complementary to the Shine-Dalgarno sequence (SD) (see Section 2.2) was added upstream and again forms a hairpin structure that melts at higher temperatures. A system that was also built by combining known sequence motifs but rather works on the transcriptional level, was designed by Dawid et al. [34]. A terminator was extended by an upstream anti-terminator region complementary to part of the terminator sequence and therefore hampering termination. An additional molecule, called antisense RNA, is then able to bind parts of the anti-terminator which leads to terminator formation and therefore to early termination. This system was further extended by an anti-anti-terminator to rather act as an OFF-switch. Ceres et al. [25] built two transcriptional ON-switches by combining the naturally oc-



Isaacs et al. [87] translational control of a cis-repressed device (crRNA) by an sRNA trigger (taRNA)

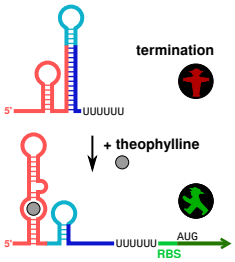
curring expression platform of a riboswitch (Figure 2) with various aptamers that bind e.g. to theophylline or tetracycline.

However, such designed sequences only occasionally showed the desired functionality and often led to more or less dysfunctional devices as these designs required exchanging, adapting and extending parts of the original sequence. All of these are highly risky operations, as already very little modifications can completely change the energy landscape of an RNA molecule [166].

4.2 AUTOMATED IN SILICO TESTING

To minimize the risk of loosing the desired functionality, various *in silico* predictions were used to test the newly obtained sequences before they were synthesized and analyzed in the lab. Researchers came up with a variety of measures which should either describe the desired function or allow for filtering dysfunctional sequences. The probably most obvious *in silico* test is to check whether the sequence still folds into the desired structure. While Man et al. [118], for example, used `mfold` to predict the MFE structure of their sequences and exclude miss-folded devices, Lucks et al. [113] decided to predict the most probable structure using another implementation of the algorithm, `RNAstructure`. In the followup study of the latter, a kinetic model was developed which could explain the activation observed for STARs, small transcription activating RNAs [26].

Subsequently, more and more such analyzing and filtering steps were added after a sequence was manually edited and designed. For example, for testing two theophylline triggered transcriptional and translational riboswitches *in silico* several free energies – of the full length and parts of the sequence – were computed, including local RNA structures and hybridization energies [143]. Wachsmuth et al. [191] even created a complete computational testing and filtering pipeline to get rid of potential dysfunctional sequences. To design a transcriptional riboswitch able to activate gene expression in the presence of theophylline, they generated random spacer sequences between the aptamer and the partially complementary terminator and rejected candidates which exhibited undesired behavior in the prediction. This includes the formation of base-pairs between spacer and aptamer, the absence of the terminator in the MFE structure, an overstated energy difference between aptamer and terminator as well as a z-score for MFE structure stability. Moreover, even cotranscriptional folding paths were simulated to further investigate the candidates [191].



Wachsmuth et al. [191] theophylline controlled transcriptional riboswitch.

4.3 RATIONAL DE NOVO DESIGN

It turns out that manual sequence generation and selection is a demanding and time consuming task, as many sequences do not have the desired features and need to be discarded anyway. Thus, this leaves much room for improvement by automating this tasks. A process that generates most of the RNA sequence from scratch concerning some design goals, such as structural constraints, is called rational *de novo* design. The functionality of the desired molecule is not encoded as a sequence but rather as the mechanistic understanding of the regulatory principles. Therefore, the generated RNA devices are only based on a functional idea describing the design, encoded as structural constraints, sequence constraints or properties of the molecule's energy landscape. The latter includes thermodynamic features, kinetic processes, binding affinities or specificity to ligands. This way, the design idea (sequence and structure constraints, energy landscape) is clearly separated from the specific implementation (sequence) and can thus be reused in other contexts and environments or be adopted to fulfill altering needs which is not easily possible for an RNA sequence. Rational *de novo* design is a much more general, flexible and adaptable way of constructing synthetic RNA molecules which fulfill specific functions.

Over the years, many algorithms and computational tools have been developed that are able to *de novo* design sequences with respect to a variety of properties [30, 84, 97, 153, 216]. They all have in common that they attempt to solve the *inverse folding problem*, the problem of finding an RNA sequence that folds into the desired structure as its MFE structure. Furthermore, a successful algorithm should enable to specify additional properties and inputs necessary to describe the functional idea properly. Solving this problem is a challenging task, and methods producing an exact solution are still missing. Many tools try to solve the inverse folding problem by implementing an optimization approach. Such an approach bypasses the actual inverse problem by generating many possible solutions to rather solve the forward problem, the efficient RNA structure prediction, in a recurring fashion. It typically consists of a method to generate valid solutions, a scoring or objective function to evaluate the current sequence and a method to decide whether to accept or reject the newly generated solution.

4.4 EMERGED DESIGN PROGRAMS

`RNAinverse` was the first design tool published [80]. It performs a stochastic local search which is able to find a sequence that exhibits as its MFE structure the prescribed structure, or alternatively that has this structure with the highest frequency in the ensemble. To speed

up the calculation of the objective function, the structure is chopped into local substructures. Then, the problem is solved for these substructures and optimal solutions are concatenated subsequently. A technique that was also frequently used by other algorithms. However, concatenation occasionally fails as joined sequences might fold differently than their isolated pieces which requires re-optimization of the obtained parts.

Subsequently, more tools following the same idea with slight variations were published. `RNA-SSD` sets the initial seed sequence according to a probabilistic model to have it as close as possible to the target structure [6]. `INFO-RNA` uses a dynamic programming approach to compute the sequence where the prescribed secondary structure is as stable as possible [19]. This sequence is then set as initial seed. Lastly, `Inv` extends the approach to be the first inverse folding algorithm that handles a certain type of pseudoknotted structural inputs [57].

The first method able to design RNA sequences that fold into two alternative conformations, and therefore create switching RNA molecules, was a Perl script called `switch.pl` by Flamm et al. [51]. It makes use of probability theory to allow fair sampling and implements an adaptive walk as optimization approach. By that it is possible to design switches within desired temperature ranges, and states separated by specific energy barrier heights and with prescribed energy differences. `ARDesigner` extends these objectives by many features suggested by Flamm et al. [51] to fold into prescribed multiple stable conformations with additional properties such as mutational robustness [173]. Another program, `multiSrch`, again revamped these objectives to allow for interacting molecules with multiple stable conformations, two per single molecule [146].

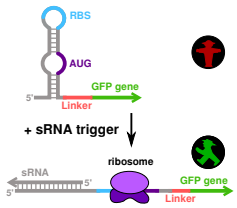
Others changed the optimization method from single objective local search to evolutionary algorithms. `ERD` (Evolutionary Algorithm for Designing) recursively applies an evolutionary algorithm for all sub-structures of a decomposition tree, where sequences on the leaves are sampled from a natural pool of sequence-structure pairs [42, 43]. `MODENA` introduces a genetic algorithm which is a multi-objective optimization approach. It optimizes towards multiple goals such as thermodynamic stability and structural similarity. In a subsequent version of `MODENA` the algorithm got extended to allow for multiple structural constraints with pseudo-knots and is able to include a list of various programs in the objective function terms [181, 182]. Another method, `Frnakenstein`, also uses a genetic algorithm and a heuristic sampling approach to create sequences with multiple structural targets. Its fitness evaluation is done positional along the sequence, hence it is possible to gain a fitness value of the full sequence or only of parts of the sequence, which accelerates the evaluation process [114].

The methods presented above can be classified according to three major factors. The optimization method, the possible objective function terms and various techniques to accelerate the evaluation or optimization process. The way how the RNA sequences are sampled, i.e. how the solution space is traversed during optimization, was often neglected. Among the methods presented here, only Flamm et al. [51] guaranteed to obtain every sequence with the same probability, for the others this distribution is non-uniform and often unknown. The following methods impose more importance on this aspect and improved the sampling step by obtaining sequences from well-specified distributions. Levin et al. [105] for example developed `RNA-ensign` which implements a global sampling algorithm to explore large regions of the mutational landscape under structural and thermodynamic constraints. This is achieved by new sequences being sampled from a set of Boltzmann distributed k-mutants during each optimization round. `IncaRNAtion` expands this idea of Boltzmann distributed sampling by not only sampling from neighboring mutants, but from the whole solution space using dynamic programming and a reduced energy model [149]. `NUPACK:Design` rather uses ensemble-defect weighted mutation sampling to complement their method in order to generate stable and robust sequence of one or more interacting nucleic acid strands [216]. An enhanced objective function which accounts for multiple strands in multiple solution tubes was published as `NUPACK:TestTubeDesign` [205].

`RNAifold` does not use any optimization approach at all, but tries to find a valid solution solely by sampling. It uses Constraint Programming to solve the RNA inverse folding approach by exhaustively finding all sequences which fold into the given structure as their MFE structure. For bigger instances, a stochastic large neighborhood search only searches parts of the solution space. Version 2.0 of `RNAifold` allows to not only specify a MFE structure, but also one compatible structure [58, 59]. Although it seems a great idea to squeeze any necessary features into the sampling approach, this is not always possible. There are just too many aspects to take care of in the sampling procedure. The authors therefore use analysis and filtering to cut down their huge list of possible solutions.

4.5 TOOLS SUCCESSFULLY UTILIZED

Many computational tools were already developed to *de novo* generate RNA sequences regarding a set of inputs. However, only very little of these programs were used for biological applications. Waldminghaus et al. [196] for example published an RNA thermometer made by `switch.pl` using the temperature bistable objective function. However, they could only obtain working riboswitches after an *in vivo* selection screening with error prone PCR mutagenesis. Another



Green et al. [65] toehold switches can be translationally switched on by a trigger sRNA

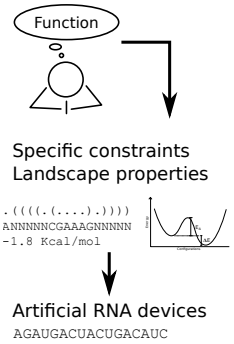
successful system, a cis-repressed RNA sequence able to be trans-activated by an sRNA, was obtained through optimization towards various energy terms describing the duplex formation [154, 155, 157]. The developed tool was published as `RiboMaker` [153].

`RNAifold` was successfully applied to design a cis-cleaving hammerhead ribozyme. However, many of the design goals such as the positional entropy, ensemble defect or structural flexibility/rigidity were included as filtering steps after the sampling step [40]. Green et al. [65] used `NUPACK:TestTubeDesign` and an exhaustive *in silico* analysis pipeline to generate and evaluate so-called toehold switches. These riboswitches sequester the region around the RBS and the start codon. A designed sRNA or another messenger RNA (mRNA) then triggers the opening of this region and therefore activates gene expression. The latter publication is a great example for how an RNA design process on the *in silico* side as well as on the *in vivo* side can and should be done to obtain a really high percentage of functional devices.

5

DESIGN AS A COMMON PIPELINE

Looking at the rich history of various attempts to craft artificial riboswitches with prescribed functionality, it seems that the interplay between computational tools and laboratory experiments is often missing. This leads to two distinct kinds of publications, one kind created by wetlab researchers focusing on a nice experimental testing setup as well as solid functional and analytical biological tests, but missing the possible advances of *in silico* tools such as their great prediction powers. In contrast, publications written by theory based researchers often comprise great computational details, sophisticated biophysical methods, and a huge variety of mathematical and algorithmic tricks [30], but frequently neglecting the biological applicability. This leads to design software that misses the applicability and ignores the bigger context, such as *in vivo* testing, initial analysis of the system or finding the right objective which describes the optimization goal to gain the desired functionality. This fact would not be hampering if the published tools were flexible and customizable to be able to adopt them to the desired scenarios, but this if often not easily possible.



How can we describe and build a novel RNA device de novo?

Successful design implementations show that the problem of finding an RNA sequence which fulfills all the desired objectives to form a functional device artificially has to be formulated as a multi-step approach including computation and laboratory, predictive and analytical methods. Early design publications already followed the same scheme but included manual steps instead of computational methods, as there were just no computational tools available or known which implemented the features actually needed by the experimentalists. However, this changed over the time so that the common trend today is to perform as many steps as possible with the support of advanced *in silico* methods. Such a pipeline usually consists of the following steps.

ANALYSIS OF THE BIOLOGICAL SYSTEM At first, it is important to get an idea about the current situation, the biological system, the cellular environment and most importantly all the properties of the building blocks to use. No designer starts to work with unknown material and without knowing the context and environmental factors.

SAMPLING SEQUENCES TO GIVEN CONSTRAINTS Obtaining a sequence which is compatible to target structures and other important constraints might seem easy, but is a quite tricky task. The method has to be fast as we often run optimization problems where this step

is repeated numerous times, and they have to be accurate to avoid missing, duplicated or wrong solutions.

OPTIMIZATION PROBLEM The currently most common way to efficiently handle the huge set of possible solutions is to solve an optimization problem subjected to a formulated objective function. A variety of well-established optimization methods helps to perform this task. However, finding the right objective to evaluate the solutions is quite challenging.

IN SILICO ANALYSIS AND FILTERING A subsequent step to analyze and filter the obtained solutions is generally inevitable. Even if only a small set of solutions was generated, it is important to create a meaningful visualization to be able to explore the differences and advantages of various solutions. Moreover, some computational methods are too demanding to be applied during an optimization approach and can only be handled in such a filtering step.

BIOLOGICAL TESTING As many biological relevant aspects cannot be easily included in the computational models and algorithms, it is crucial to always biologically test for the desired functionality of the designed molecule. This includes extensive studies *in vitro* or *in vivo*, or preferable both. To really gain knowledge about the device's mechanism and about potential mistakes or pitfalls in case of dysfunctionality, a pure qualitative answer will not be enough. Therefore, a complete testing pipeline includes the determination of structures, binding affinities or elucidating kinetic properties.

The following sections will elucidate each of these points of the pipeline in more detail in order to collect and organize already published knowledge and to highlight the great advances of the various studies.

5.1 ANALYSIS OF THE BIOLOGICAL SYSTEM

Whenever we desire to include an artificial element in a cellular context, we rely on existing biological parts to some extend. Even for *in vitro* systems, knowledge about many factors and variables is required to be able to design the interfaces and prohibit undesired interactions.

A successful design pipeline therefore begins with enlightening the essential properties of the current system and analyzing available building blocks in order to be able to implement an artificial extension. Hence, a great start is to analyze and understand the folding structures and kinetic folding processes of the components. As experimental structure elucidation is often hard, the probably minimal required effort is to use *in silico* structure prediction tools. Man et al.

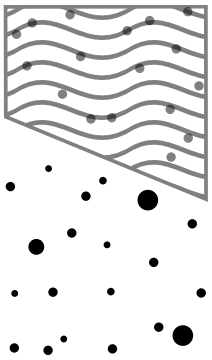
[118] for example used the `mfold` structure prediction tool to analyze the natural RNAs desired to be included in their system. Dawid et al. [34] in addition to folding experiments performed kinetic simulations using `Kinefold`, and Neupert et al. [140] studied natural occurring RNA thermometers, including their structures and energies.

It is also advantageous to gain insights into the relevant biological pathways and how natural components similar to the designed ones are embedded in the cellular context. Waldminghaus et al. [196] extensively studied natural RNA thermometers such as ROSE (repression of heat shock gene expression)-like thermometers in proteobacteria and their mechanisms before designing such artificial devices.

Without prior analytical efforts, designing ligand sensing systems is not possible at all as we cannot predict ligand binding properties, such as binding pockets, affinities or kinetic rates of the interaction. Thus, to create such sensing RNA molecules we rely on time consuming methods such as systematic evolution of ligands by exponential enrichment (SELEX), or we have to use naturally occurring aptamers instead. The properties of the ligand interaction have to be enlightened in tedious studies, such as affinity studies, structural probing or stop-flow kinetics. Ceres et al. [25] for example performed extensive isothermal titration calorimetry (ITC) experiments to measure the ligand-affinity of the chosen aptamers. In order to be able to model the interaction *in silico*, a minimal list of necessary features includes structural and sequence constraints of the binding pocket, binding energies and kinetic rates. Such a model then allows to design the context of the sensing aptamer such as an expression platform which fulfills the communication to the downstream reporter expression. Similar to ligand binding, also enzymatic cleaving activity of an RNA molecule cannot be easily predicted. Dotu et al. [40] therefore used information gained by previous studies about the efficiency of various hammerhead cleavage sites.

Many published artificial RNA devices were built upon previous, quite extensive studies of natural systems. Lucks et al. [113] started by modifying the natural antisense-RNA-mediated transcriptional attenuator plasmid pT181 which regulates the copy number control mechanism. Similarly, [143] also adopted the pT181 anti-sense RNA for gaining an artificial transcriptional and extended IS10 for a translational system. An experimental screening of 500 IS10 mutants helped to support the objective function used by Rodrigo et al. [154] for their designs. In a follow-up publication they then used information from their previous publication [155].

Garcia-Martin et al. [59] even created a tool to automate such an initial analysis step. The RFAM-based design pipeline analyses naturally occurring families and finds relevant design goals and constraints automatically. It is then able to build novel sequences similar to the existing ones.



While constraints restricting the solution space (top) lead to specificity, raising the probability of obtaining desired solutions (big dots) leads to affinity.

How can we generate a sequence that fulfills our requirements of a functional RNA device? Dirks et al. [38] stated that an RNA design problem needs to solve two distinct problems, the *specificity* and the *affinity* problem. Specificity in this context means the compatibility between base pairs e.g. to form a certain structural element or to be able to bind to a sRNA molecule. However, this does not yet include how likely this structural element is in the ensemble of possible structures and whether or not the sRNA will be attracted to the desired target. Therefore, it is important to take the energy properties of the involved molecules into account which specify the affinity to the desired structure compared to other possible states.

While the specificity can be easily achieved even manually, the affinity problem is much harder to solve as it requires the inclusion of the RNA energy model (see Section 3.2). A design approach that generates sequences regarding some inputs describing the desired functionality of the RNA is therefore often formalized as an optimization problem. This approach is able to take care of both, the specificity and the affinity problem. The specificity problem is solved in the sequence sampling step while the affinity problem is taken care of by the iterative optimization of the initial sequence. However, in many tools algorithmic tricks were used to include some kind of affinity already into the sampling step in order to accelerate the optimization process. A great example which illustrates such an optimization process is `Eterna` [103], an online game[1] to manually create single-stable RNA molecules with a prescribed structure. A MFE folding view shows the most likely structure of the current sequence. Users are encouraged to modify the nucleotide composition with respect to base-pairing (specificity) in a try and error fashion (optimization) until they obtain the desired MFE structure (affinity).

In this section I will explain how the method of creating a sequence evolved, which features were developed and why this step is hard to solve.

MANUAL SEQUENCE GENERATION Early designs were generated by manually generating sequences due to rational thinking. With this technique it is possible to generate complementary regions such as antisense strands, introduce bulges and loops, and take care of sequence constraints like SD motifs, restriction sites or aptamer sequences [34, 87, 139, 140, 191]. While with manual sequence sampling the specificity problem can successfully be solved, the affinity problem cannot. It becomes obvious when one tries to generate a sequence with many unpaired positions. This sequence will probably never fold into the desired structure. However, for rather small

1 http://www.eternagame.org

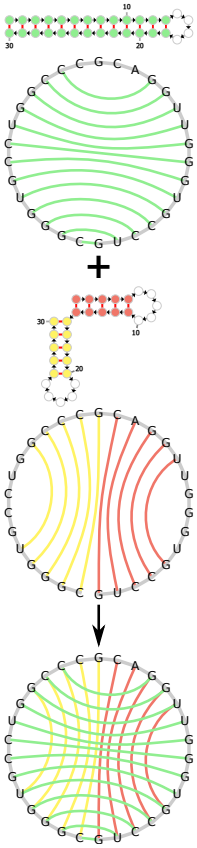
changes in a well-structured region, e.g. exchanging base-pairs in a stable helix, this method works quite well.

AUTOMATIC SEQUENCE SAMPLING As manual sequence generation is a quite tedious and error prone task, many different algorithms were developed to automatically generate sequences with various properties and constraints. The problem of automatic sequence sampling can be described as randomly picking a sequence or solution from a huge solution space. Thus, two variables can be adjusted here: The size of the feasible solution space, meaning the set of solutions that fulfill the prescribed constraints, and the probability of a particular solution to be chosen. Most straight forward would be to choose any possible solution with the same probability. In such a case, there exist no constraints and the solutions are uniformly distributed. This can be achieved by picking the four possible nucleotides randomly for every position with an equal probability. However, as shown in Section 3.1, the complete solution space is enormously big and probably includes many undesired solutions.

Better would be to add some meaningful constraints which will exclude the majority of unwanted sequences and therefore raise the probability of any desired sequence to be picked randomly. In order to restrict the solution space we could use sequence constraints which only allow a subset of all possible nucleotides at certain positions. As individual nucleotides are picked independently of each other, such constraints will enormously decrease the search space even for a small number of restricted positions. For example, Man et al. [118] decided to include sequence constraints of a functional SD site, an HFQ/HF-I protein (HFQ) binding site and a Rho-independent terminator from natural trans-encoded sRNAs in their design. Ceres et al. [25] optimized an aptamer sequence by shortening and mutating some loop regions and included this sequence as a constraint.

It is also possible to use constraints based on a desired target structure to solve the specificity problem directly in the sampling step. This way it is possible to enormously restrict the solution space. A structural constraint excludes all sequences which are not able to form the desired base pairs, meaning that base-pair compatibility should be given for certain positions at any time. This property can be achieved by always exchanging both nucleotides of a base-pair constraint, and single nucleotides or unconstrained regions. `RNAinverse` [80] and many of its successors generate sequences compatible to a single prescribed structural constraint this way.

BIASES TOWARDS UNDESIRED SOLUTIONS As a negative consequence of their increasing complexity, methods that sample solutions respecting some prescribed constraints often introduce undesired biases towards certain nucleotide compositions, energies and structural



Flamm et al. [51] introduced the dependency graph, a super-imposition of the circle representation of all structures.

patterns. This means, instead of only making the solution space smaller, we will unintentionally also change the probability of certain solutions to be randomly picked and therefore add a systematic affinity towards a probably undesired property. Thereby it can happen that good solutions become very unlikely to be found so that we might miss them. The above mentioned very simple method for single structural constraints is not affected by this problem. However, to be able to build riboswitches, it must be possible to constrain the search space to two or more possible structures, which is much more complicated [2].

Flamm et al. [51] therefore developed a method that allows to uniformly sample sequences compatible to two alternative secondary structures. This methods constructs a so-called dependency graph and decomposes it into its connected components which are unpaired bases, paths and cycles. Moreover, the authors found a recursive method for these paths and cycles that uses Fibonacci numbers to uniformly assign nucleotides [51]. Abfalter et al. [1] extended this method to multiple structural constraints by further decomposing more complex connected components of this dependency graph using the ear decomposition algorithm [121]. A dynamic programming graph coloring approach is then used to count all possible nucleotide assignments for these components and their subgraphs, which paves the way for uniform sequence sampling. Part ii describes how my work contributed to this field by further extending this approach.

Version two of `MODENA` [182] is also able to design multi-stable RNA molecules. But instead of implementing the previously mentioned graph coloring approach, the authors use a sampling technique which introduces undesired biases towards certain solutions. Chapter 7 describes this in detail and shows how this affects the efficiency and quality of the applied optimization approach. Similarly, also Lyngsø et al. [114] use a simple recursive method on the dependency graph, which introduces an uncharacterized bias.

DESIRED BIASES TO GAIN AFFINITY As previously explained, including meaningful constraints will narrow down the solution space. However, sampling sequences compatible to multiple structures in a fair way is a challenging task as it is quite hard to avoid unwanted biases. In addition to such constraints, it can be advantageous to sample according to a specified distribution, meaning with a desired bias. Increasing the likelihood of obtaining desired solutions will significantly accelerate the process of finding better sequences. This makes the applied optimization problem faster and easier to solve since the affinity problem is already tackled in the sampling step to some extent. The probably easiest example for a distorted distribution consists in sampling according to a prescribed nucleotide composition. The G/C content for instance is important for better transcription

rates and structural plasticity. In order to bias the sampling probabilities towards G/C richer sequences, it is sufficient to increase the probability for choosing G and C for every nucleotide.

Similarly, `RNA-SSD` [6] sets the initial seed sequence according to a probabilistic model defined by a structural constraint. In this model, base-pairs must be complementary, only non-complementary nucleotides are assigned to unpaired positions after helices, and G-C pairs are preferred in helices as they are energetically more stable. This increases the chance of obtaining a sequence with an MFE structure closer to the target structure. A related, but more advanced method is to include the RNA energy model to some extent into the sampling process.

`INFO-RNA` [19] generates its initial sequence using a dynamic programming approach with sophisticated energy evaluations to more likely find the sequences that adopt the target structure with the lowest energy possible. However, there is no guarantee that other structures have lower energy and are more stable, therefore an additional optimization step is still required. Both methods exhibit the side effect to prefer sequences with high G/C content during the sampling process. Moreover, they gain sequences with very low free energies and thus highly stable structures, both features that might not always be desired. `IncaRNAtion` [149] enhances this idea and performs Boltzmann sampling from the full sequence space (Ω) given a specific structure. It uses a dynamic programming approach and a reduced energy model which only includes base-pair and stacking contributions. Moreover, the Boltzmann factor is weighted towards a G/C dependent factor to allow sampling of certain G/C nucleotide contributions and avoid biases. Its predecessor, `RNA-ensign` [105] uses `RNAmutants` to generate the initial random seed sequence. This algorithm also performs Boltzmann distributed sampling, but in a less sophisticated way. All sequences with k mutations (k-mutant) of a given target structure are generated and their MFE and partition function (Q) values exhaustively computed to achieve Boltzmann sampling within this local neighborhood. This approach enables to measure the resilience of an RNA towards point-wise mutations [194, 195].

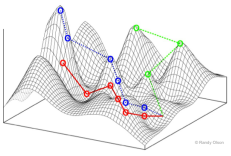
`NUPACK:Design` uses a similar Boltzmann sampling approach, but with a different goal. Their method performs ensemble defect weighted sampling during the optimization process, which increases the likelihood of solutions which fold into secondary structures close to the desired target structure [216].

CONSTRAINT PROGRAMMING After learning that not only specificity in form of constraints, but also affinity towards desired features can be achieved in the sequence generation step, one could argue that it is probably optimal to include all objectives into the sampling step.

Garcia-Martin et al. [58] attempted to do so by using a constraint programming (CP) framework for `RNAifold`. In such a framework, the general model such as base-pairing and the types of nucleotides and constraints like structural constraints, sequence constraints and channeling constraints need to be specified. A CP search then exhaustively explores all solutions or it is proven that none exists. As the list of possible solutions can be enormously big, the authors also implemented a large neighborhood search (LNS) that allows to enumerate only parts of the complete solution space. [40, 59]

However, it is not yet possible to include any possible constraint and goal into such a CP framework. A rather small amount of meaningful constraints leads to a huge number of possible solutions enumerated by the framework. These solutions then need to be analyzed and filtered for additional features. Alternatively, an optimization problem could be formulated to explore the gained solutions. Similar to any other biased and constraint sampling, CP could also potentially miss good solutions if restrictions were falsely or too strongly specified or if undesired preferences were introduced. Especially when exploring only parts of the solution space, biases towards a certain region of the solution space cannot be avoided.

5.3 Optimization Approach for Finding Desired Solutions



A representation of an optimization progress on a fitness landscape. (CC BY-SA 3.0 by Randy Olson)

To be able to use a stochastic sampling method for an optimization problem, several prerequisites must be fulfilled. Concerning the sampling procedure, there must be a way to fairly choose a good starting point or initial solution for the optimization procedure. Moreover, a so-called move set must be specified which defines the way how to mutate this initial sequence to reach neighboring sequences. The latter can be achieved by formulating a neighborhood relation between the individual solutions, which is equivalent to specifying generation rules for the solution space. However, no undesired bias should occur towards any solution, or towards moves that then unwittingly favors some mutations over the others. These generation rules make sure that only valid solutions can be obtained and that it is possible to obtain all solutions when sampling long enough.

Downstream of this sampling engine lies the optimization approach which is basically an iterative process of obtaining a new solution, evaluation of its fitness and the decision whether to keep it or not. There exist many different algorithms with varying amount of complexity which I will shortly introduce in Section 5.3.1. The evaluation of any solution is achieved by calculating a score using the so-called *objective function*. In case of multi-objective optimizations, it is possible to define more than one, sometimes even competing goals. Finding the right objective is crucial as it specifies the ideal solution and the direction of the optimization. In Section 5.3.2 I will summarize

which values were previously used to define an objective function describing the fitness of a *de novo* designed RNA device.

5.3.1 *Optimization algorithms used for RNA design*

Most publications on RNA design use optimization algorithms based on the Metropolis–Hastings algorithm [78, 127] which itself is a Markov chain Monte Carlo (MCMC) method for stochastically drawing samples from a certain probability distribution. It is mostly used for huge, multi-dimensional distributions for which direct sampling is not feasible. The Metropolis–Hastings idea is based on generating sample states of a thermodynamic system. An arbitrary initial candidate is chosen, followed by recursively invoking the following steps. Given the current solution, a neighboring candidate is chosen from a symmetric probability density distribution specified by the neighborhood relation. This relation exhibits symmetric probabilities, so that the principle of detailed balance is given. The new solution is evaluated against the current one by calculating the so-called acceptance ratio. This ratio includes the scores of the current and the new candidate by evaluating the specified objective function. It decides whether to accept or reject the newly obtained candidate. It is always accepted when greater than one, and accepted with a certain probability otherwise. In case of a rejection, the algorithm forgets about the new candidate and continues with the previous one instead.

The derived algorithms used for RNA design are all based on the Metropolis–Hastings idea described above and mainly vary in the way the acceptance ratio is calculated. Hofacker et al. [80] implemented an adaptive walk approach, where new solutions are only accepted if the ratio is better than one and rejected otherwise. This means that the suggested mutation of the RNA sequence is only accepted if and only if it improves the score. Also Flamm et al. [51], Reinharz et al. [149] or Zadeh et al. [216] use an adaptive walk to optimize either the whole or just parts of the newly obtained solutions.

Andronescu et al. [6], Busch and Backofen [19] and Gao et al. [57] rather implemented a stochastic local search which is similar to the adaptive walk, but softens the restriction of not accepting any worse solutions. Here, worse solutions are accepted with a constant probability. The acceptance of worse mutations allows to overcome local optima, i.e. sequences that perform better than their direct neighbors, but not necessarily better than any other solution in the sequence space (Ω).

A more sophisticated way to decide whether to keep worse solutions is used by the Monte Carlo simulated annealing (MCSA) approach. While better solutions are always accepted, worse solutions are accepted dependent on the acceptance probability function which encompasses the score difference and a the global parameter temper-

ature. This temperature follows a specified cooling schedule during the course of the optimization, i.e. it continuously decreases and eventually tends to zero. The probability of accepting a worse solution is positively linked to the temperature, which yields a declining acceptance ratio. This means that mutations with the same score difference are more likely to be accepted early in the optimization run than close to the end. Many RNA design software packages are based on this optimization method [7, 125, 154, 155, 173]. In Chapter 7 we compared the performance of MCSA with a simple adaptive walk approach coupled with uniform sampling for designing RNA.

A different approach is to follow nature's advice and utilize one of a wide variety of evolutionary algorithms (EAs). Instead of only one single solution, a whole population of solutions is simultaneously improved. Biological mechanisms such as mutation, reproduction, recombination, and selection change the genetic composition of a population which tends to higher fitness over time in an unchanged environment. Major drawbacks of such an approach are the frequent lack of a clear genotype-phenotype distinction or a missing method of encoding the solutions into a gene-like and therefore mutable structure (genetic representation). However, none of these apply to the RNA design problem as an RNA sequence already naturally represents a genotype and the resulting secondary structure expresses the phenotype.

An EA is always initiated by generating a random population, the so-called first generation. Within each step of an iterative process, the individuals of this population are then evaluated according to the given objective function, where best performing candidates are selected for reproduction. The latter involves mutation operations or breeding between individuals such as crossover. Best performing new individuals will then replace bad candidates of the initial population. Esmaili-Taheri et al. [42] implemented such an EA in `ERD` [43]. As an objective they use a distance measure between the current MFE structure and the target structure. Similarly, Lyngsø et al. [114] use a genetic algorithm (GA) to optimize their population. GAs usually use binary strings as genetic structure and are the most popular form of an EA for solving optimization problems. Another detail makes the `Frnakenstein` [114] approach quite special. Instead of evaluating substructures or the full sequence, the fitness is always calculated positional, so it is possible to gain a score for any part of the RNA sequence. However, this makes it necessary to distinguish between the fitness in each round and the objective for the determination if the found solution is adequate.

Another GA called multi-objective genetic algorithm (MOGA) was utilized in the `MODENA` design package [181, 182]. Although the general evolutionary approach is still present in this method, it uses quite different evaluation procedure. Instead of calculating a single score

for each individual, multiple objective functions are evaluated leading to a set of scores. The goal is then to find for all the given objectives a solution which is Pareto-optimal. Such solutions are not dominated by any other solution which means that they have equal and better scores for all objectives. Instead of obtaining solutions which are optimized towards one single goal, many varying solutions are gained. These solutions differ in how well they are optimized towards the various objectives. The advantage of this approach is that the user can decide posteriori which goal was most important. Ramlan and Zauner [146] also implemented a multi-objective optimization approach in addition to single aggregate objective and their non-deterministic, stochastic variant `StochSrchMulti`. Instead of an EA, they decided to use a sorting-bins approach.

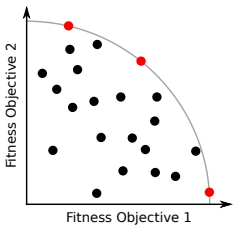


Illustration of a Pareto-front with two objective functions. It is possible to posteriori decide which Pareto-optimal solution (red) to choose.

The introduction of the multi-objective approaches directly leads to the probably most important aspect of an optimization approach, namely the formulation of the objective function.

5.3.2 *Objective functions previously used*

In order to be able to optimize an RNA sequence by one of the before mentioned optimization methods, it is necessary to specify how the fitness of a solution is calculated. This is usually done by the formulation of a so-called *objective function*, a mathematical model of the relation between an RNA sequence and its important properties and functions. In biological terms this relation between fitness and sequence can be seen as an assignment of a phenotype (score) to a genotype (sequence). Selection pressure due to the optimization method ensures that better phenotypes are advantageous and thus preferred, which optimizes the sequence to fall into fitness optima.

It is crucial that this model not only specifies the overall goal of the optimization, meaning the best possible solution, but that at any time during the optimization a better solution exhibits a larger score. Only this guarantees that local optima of the solution landscape exhibit better properties than their neighbors. The mathematical formulation is therefore often quite tricky, especially if several competing goals need to be joined into one value. It is often better to simplify and write an objective function well and precise to the point, skipping optional goals. Moreover, as the evaluation of this function is often the most time consuming step, a simpler function will save time.

In this section I will mainly focus on how objective functions for RNA design problems historically emerged and evolved, evincing some of their interesting properties.

SIMPLE DISTANCE MEASURES The first objective function formulated for solving the inverse RNA folding problem was quite simple. Only the structure distance between the target structure and the cur-

rent MFE structure was used [80]. This required only one computationally demanding calculation, therefore being computationally fast. For more information on distance measures and folding predictions see Section 3.1 and Section 3.2. This objective was used many times since then, however with slightly varying distance measures such as base-pair distance, incorrectly bound bases or the number of altering paired or unpaired bases [6, 19, 57, 181].

PROBABILITY OF TARGET STRUCTURE Hofacker et al. [80] introduced an alternative approach in `RNAinverse`, namely to optimize the probability of the target structure in the ensemble of structures. This required the calculation of the partition function, therefore being computationally more demanding (also see Section 3.2). Others also used this probability measure either as overall goal or for sorting and ranking competing solutions during their approach [105, 149, 181]. This objective function is quite useful as it introduces a strong affinity towards the target structure. However, the ultimate goal of obtaining a sequence that folds just into a single specified structure is hard be reached with this approach. RNA will always adopt alternative conformations, which can be suppressed by specifying negative design goals.

NEGATIVE DESIGN GOALS In addition to the mentioned positive goal it is preferable to also specify negative design goals which specify unwanted goals. In the case of RNA inverse folding we would like to include undesired secondary structures in the objective function to actively get rid of these competing states. Thus, Zadeh et al. [216] introduced the *ensemble defect* which can be described as the average number of incorrectly paired nucleotides over the ensemble of structures. It is calculated such that the probability of every structure is weighted by a factor correlating to the distance to the target structure. If many frequent structures are distant to the target structure, the ensemble defect will be big. In contrast, if all frequent structures are very close or equal to the target state, the ensemble defect will be minimal.

MULTI-STRAND OBJECTIVES The previous mentioned ensemble defect was extended further to include concentrations and values for multiple interacting RNA molecules [205]. However, the idea of including the properties of multiple molecules into the objective function was used before. Rodrigo et al. [154] already built sRNA triggered riboswitches by minimizing the free energy of the complex formation. This can be achieved by calculating the free energy difference of interacting and free species. Also the activation energy of complex formation was included since it influences the speed of the reaction. This paved the way for developing a kinetic model using a Michaelis

Menten system in subsequent work [155]. Moreover, `multiSrch` [146] is also able to design multi-strand switches by combining the principles of `RNAcofold` and `RNAup` [12, 137].

BISTABLE AND MULTI-STABLE RIBOSWITCHES To support the design of riboswitches, it must be possible to specify an unlimited number of desired structures including their energy differences, barrier heights or kinetic properties for a prescribed timely succession. Flamm et al. [51] introduced the first objective function that can optimize an RNA molecule able to adopt two distinct states. This bistable objective function contains two terms. The first term increases the probability of both structures in the ensemble and the second one specifies the desired energy difference between both states. It is also possible to vary the temperature of the states in order to gain a bistable thermoswitch. Shu et al. [173] and Ramlan and Zauner [146] expanded many of these ideas to more than two structures, including extensions for multi-state energy barrier calculations or objective functions to design multi-stable thermoswitches. Chapter 6 and Chapter 7 contain further developments on this topic.

MUTATIONAL ROBUSTNESS AND NEUTRALITY In addition to the previous mentioned terms required to gain affinity towards the desired target structure, Shu et al. [173] included a term which measures the mutual neutrality of the sequence concerning the target structure. This term optimizes the sequence in a way such that the fraction of one-mutant neighbors to the original structure is as big as possible. This way, the structure is perfectly preserved even when mutations are introduced. Avihoo et al. [7] followed this idea and also developed objectives to optimize for mutational robustness [7].

KINETIC AND MATHEMATICAL MODELS It is often not sufficient to only optimize towards energy landscape properties of the RNA molecules. The inclusion of biochemical and biophysical models describing cellular processes brings additional information which can lead to improved design success. Thus, Carothers et al. [23] used mechanistic modeling with ordinary differential equations (ODEs) to engineer ligand controlled ribozymes. Similarly, Salis et al. [162] developed a biophysical model able to predict the translation initiation rate of a given gene. Using this model they designed translation regulating riboswitches with great success. More recently, they enhanced the model to account for ligand concentrations, RNA-ligand complex stability and macro-molecular crowding [44, 45, 46].

MULTI-OBJECTIVE OPTIMIZATION In case of more than one design goals, it is necessary to concatenate them to fit into one mathematical representation of an overall goal. This is often quite tricky

as the individual terms are influencing each other and need to be weighted differently. An alternative is to use multi-objective optimization methods. There, several objective functions, even competing ones, can be specified without the need of assigning arbitrary weighting factors. The solutions are all optimal with respect to at least one objective and thus arranged on the so-called Pareto optimal front. Several RNA design tools already use this approach [146, 181, 182]. However, especially these methods would need an extensive visualization of the obtained solutions and their multiple scores to be able to quickly spot the differences and similarities, the advantages and disadvantages of specific solutions.

EFFICIENCY OF OBJECTIVE EVALUATION Objective functions are used to evaluate the fitness of possible solutions iteratively during the process. That means that the function is called many times, therefore consuming a lot of computation time. Thus, it would be advantageous to minimize the load of every step, or alternatively the amount of iterations must be decreased which might lead to less optimal solutions. As the complexity of RNA prediction algorithms scales polynomial with the sequence length, it is best to split the sequence apart. Many divide and conquer strategies have previously been developed which are able to speed up demanding objectives [6, 19, 42, 80, 216].

However, structural folding might change as soon as optimized parts of the solution are concatenated again. As explained in Section 3.1, the addition of nucleotides to a sequence can completely change the energy landscape of the molecule. In such a case, these parts of the solution must be re-optimized to avoid such problems. However, a high degree of structural neutrality allows us to still take advantage of such strategies.

Another interesting method of saving computation time is used by Lyngsø et al. [114]. Instead of dividing the sequence at fixed positions, they established a way to evaluate the fitness positional. However, as this leads to some inaccuracies, this method is just used to evaluate the fitness of the population but not to calculate the global score.

If the desired objective is still too demanding to be repeatedly evaluated for any newly sampled sequence during the optimization, the only way to include it efficiently in the design pipeline is to add a subsequent filtering step, which is outlined in the following section.

5.4 FILTERING AND IN SILICO ANALYSIS

Although it was not explicitly mentioned before as such, but almost any published RNA design approach included an *in silico* analysis step to find potentially dysfunctional sequences or to narrow down a huge list of potential candidates.

In early design approaches, such a filtering step was frequently the only computational part in the design pipeline. A small variety of RNA sequences was generated manually without utilizing computational sampling or optimization approaches. Some computational methods then helped to analyze the obtained candidates before they were tested in the laboratories. Often, the only performed analysis was a simple MFE structure prediction. For this task, Man et al. [118] for example used `Mfold`, Lucks et al. [113] `RNAstructure` or Waldminghaus et al. [196] `RNAfold` at various temperatures. The latter also included melting curve analyses generated by `RNAheat` to discover possible malfunctioning candidates.

Subsequently, sequences were generated automatically or semi-automatically, thereby generating more potential solutions which needed to be handled and organized. Thus, sophisticated analysis and filtering pipelines were developed including various measures to gain additional information about the quality and differences of the potential solutions. This information helped to make educated choices of which solution to test experimentally. Wachsmuth et al. [191] for example developed a pipeline of step-wise analysis and filtering to narrow down the huge list of semi-automatically generated sequences. They reject candidates forming undesired base-pairs between several domains, check for the existence of certain structural features or filter according to certain predicted energy values. As final steps, even a statistical model and kinetic folding simulations are used to analyze the remaining solutions. Similarly, Dotu et al. [40] also use concatenated filtering steps to narrow down their list of sequences generated with constraint programming (CP). They calculate the average structural positional entropy, the ensemble defect, expected base-pair distance, and the ViennaRNA and Morgan-Higgs structural diversity for any of their solutions to be able to evaluate and rank them. Both methods have in common that they utilize meaningful constraints when generating their pool of solutions, but lack an optimization method towards any of the features.

Using optimization techniques highly increases the chance of obtaining solutions with the desired properties, as the iterative objective evaluation already discards undesired candidates. This increases the performance of the pipeline, especially if the solution space is big as it is usually the case in RNA design. Nevertheless, even in the latter case, a subsequent *in silico* filtering step is often inevitable due to several reasons.

One reason might be that the calculation of the desired property is just too demanding to be included in the objective function. As the objective function is evaluated iteratively in each sampling step, even small performance losses will quickly add up. Rodrigo et al. [155] thus chose to exclude the prediction of pseudoknots from the objective function and rather moved this design goal to the analy-

sis step. However, more recent publications already include pseudoknot evaluations in their objectives using the NUPACK package [182]. Similarly, also mutational analyses or genomic wide searches are just too exhaustive to be incorporated as an objective term. Shu et al. [173] thus applied an analysis step for mutual robustness by using the RNA Structural Robustness Evaluator (RSRE) while a Basic Local Alignment search tool was used by Rodrigo et al. [154] to check for significant similarities to known noncoding RNA sequences. Moreover, demanding kinetic simulations were often used to verify the obtained solutions. For example, Dawid et al. [34] applied Kinefold to obtain folding paths and Ramlan and Zauner [146] Kinfold to evaluate the kinetic foldings of the generated sequences. Chappell et al. [26] even developed a kinetic model that explains the observed activation and used this to analyze subsequent solutions. In Chapter 8 we also applied demanding kinetic calculations to verify our optimization approach based on a fast thermodynamic objective function.

Another reason for the necessity of a filtering step might be the pure amount of essential goals. It is often hard to mathematically join many competing terms in one single objective value. Therefore, it is often better to optimize towards the most important ones and filter for the presence or absence of other, less important factors. Green et al. [65] chose to solely optimize the ensemble defect of reaction wells using the NUPACK package and append an exhaustive *in silico* screening to predict crosstalk and orthogonality. To further optimize the design process, they calculated many thermodynamic properties of the candidates and tried to correlate them with the laboratory results. These included duplex formation energies or accessibility values for the RBS and the toehold region. Sanity checks for in-frame stop codons or an integrity score that reflects perturbations to the ideal stem region structure helped to filter unwanted solutions.

Even if it was possible to incorporated all desired goals into a meaningful objective function, it is often hard to tell which solution to prefer in the end, especially if the scores are close together. This problem arises from the fact that multiple terms are often concatenated by using so-called weighting factors. The weight of a factor positively correlates with the importance of the corresponding term in the objective function, hence it influences the overall score. However, weighting factors are frequently assigned in an arbitrary way, as the data for a precise parameter fitting is often missing. Therefore an analysis step where all terms are listed in tables and diagrams often helps to choose specific solutions from a broader range of scores. This procedure is known from multi-objective optimization strategies where the resulting Pareto front of optimal solutions is often nicely represented in order to choose the preferred combination. In Section 10.3 I will discuss how it might be possible to enhance such an analysis step by using sophisticated visualization tools and clustering techniques.

After performing all necessary and possible predictions on the *in silico* side, the potential candidates of RNA devices need to be tested and analyzed in the laboratory to verify their functionality and properties.

5.5 BIOLOGICAL TESTING OF THE GENERATED CANDIDATES

The last step of a successful pipeline consists of testing the functionality of the *de novo* designed artificial RNA molecule and verifying the proposed mechanistic model.

Therefore, we need to somehow generate and deploy the system into a biological environment. There are several possibilities how to generate an RNA in the laboratory. The cheapest way is to synthesize a template DesoxyriboNucleic Acid (DNA) molecule, clone it into a vector and transfect this plasmid into bacteria. These will then transcribe the DNA template into the desired RNA molecule. The molecule can then be either extracted from a lysate or used for functional studies *in vivo*. Sometimes it may be advantageous to do this transcription step *in vitro*. This prohibits contamination of the RNA with unwanted components from the cell lysate and assures a well-defined composition of the obtained solution. Alternatively, RNA can also be synthesized chemically by Phosphodiester, Phosphotriester or Phosphite triester synthesis. This way a pure solution of the desired RNA molecule is obtained which can be used for *in vitro* studies. However, the latter methods have the downside of being quite expensive and labor intensive compared to letting bacteria build the molecule in return for sugar. Moreover, chemical synthesis has a quite limited length restricion concerning the number of nucleotides. Nevertheless, if certain chemical modifications on the RNA are needed, *in vitro* transcription or chemical synthesis are inevitable.

Preceding publications mainly focused on deploying the design *in vivo* and performing functional tests. These tests normally reveal whether the artificial RNA molecule triggers the proposed output on the desired input signals. The outputs of such artificial devices which can be handled best are either RNA or protein levels as they can be measured with established methods. In the following I will showcase several methods capable to quantify levels of specific RNA molecules or proteins.

5.5.1 *Measuring RNA transcript levels*

Very basic but well-established ways to identify specific RNA molecules are the so-called Northern Blot analysis and the Ribonuclease Protection assays [62]. They have in common that the desired RNA is directly detected by antisense probes. In Northern Blots, the complete RNA extract is first separated by size using polyacrylamide gel

electrophoresis (PAGE) followed by labeling the desired RNA. In a Ribonuclease Protection assay, single stranded molecules of the extract are chopped into pieces by RNases. Only molecules which are protected by the antisense probe survive this treatment and are visible on the gel afterwards. Both approaches are low throughput and labor intensive methods.

Any other method always starts with a reverse transcription step where the RNA extract is transcribed into a library of complementary DNA (cDNA) strands. This way, the library is more stable against degradation and can be used for polymerase chain reaction (PCR) amplification. The most common tool to quantify specific RNA is the so-called quantitative real-time polymerase chain reaction (qPCR). It is a highly sensitive, mid-throughput method where multiple molecules can be quantified in parallel. A PCR run is observed real-time during each amplification round by measuring double-stranded DNA due to non-specific fluorescent dyes that intercalate with the DNA helix. Specificity is gained through individual primers in each reaction well that just bind the desired cDNA strand. Downsides are pipetting errors that often occur due to the individual treatment of the wells. Moreover, the desired RNA molecule can not be quantified in an absolute manner but only in relation to another RNA level, which is often a steady housekeeping gene transcript. The latter often underlie slight variations, thus influencing the quantitative result of the measurement.

In order to circumvent the latter, the competitive polymerase chain reaction (cPCR) method can be used. There, a standard gene transcript and a gene of interest are treated in the same reaction well. As the added concentration of the standard is known, the absolute amounts of the gene of interest can be calculated. Drawback of this method is that standard and probe need to have different sequence lengths for the probe to be separable by PAGE, which leads to varying PCR efficiencies and thus uncertainties. Furthermore, as the final measurement readout is done by quantifying a dye on a gel this method is low throughput, has little sensitivity and delivers only poor quantitative values.

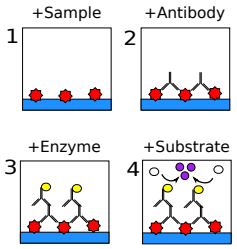
The real competitive polymerase chain reaction (rcPCR) [37] extends cPCR by analyzing the amplified strands with a matrix-assisted laser desorption/ionization-time-of-flight (MALDI-TOF) mass spectrometry system which can even distinguish single point mutations. The amplification standard can therefore have the same size as the probe, exhibiting only one single-nucleotide polymorphism (SNP). This method exhibits a high sensitivity and leads to real quantitative results. Isaacs et al. [87] used it for their design studies.

DNA micro-arrays can also be used to detect many different RNAs on a large scale. However, the sensitivity for lowly or highly expressed transcripts is often bad and besides, such a large scale ap-

proach is often not necessary for design applications. Similarly, RNA sequencing (RNA-seq) is a quantitative and extremely high-throughput method. Its big advantage is that there is no need to know the sequences a priori which makes this approach very well suited for *in vivo* or *in vitro* selection studies. The pipeline of this method starts with a reverse transcription step. The RNA extract is often filtered according to various criteria which enable the separation of sRNAs, mRNAs and others, depending on the research question. The cDNA library is then directly sent to high-throughput (next generation) sequencing facilities. Finally, the sequences are bioinformatically analyzed to deliver a quantitative result. Recent publications such as Chappell et al. [26] make use of RNA-seq in RNA design studies.

5.5.2 *Measuring protein levels*

Western blotting [5] is a method to directly detect the occurrence of a specific protein. The protein extract is separated by PAGE first followed by the detection of the protein of interest by specific antibodies and a binding dye such as fluorescent markers or radioactive labeling. This method is semi-quantitative as electrophoresis bands cannot be quantified in a very exact way and only in relation to a control standard. The enzyme-linked immunosorbent assay (ELISA) [213] is thus better suited for qualitative answers as the resulting dye can be measured and quantified nicely using a plate-reader.



ELISA is an antibody-based detection and quantification method for proteins. (adapted from CC-BY-SA 3.0 by Wektoryzacja Harkonnen2)

A similar quantitative readout as ELISA but with less experimental overhead is gained by using reporter gene assays. There, the artificial device is programmed to produce a certain enzyme as an output. This enzyme is not directly detected, but it triggers a reaction which can be quantified by measuring one of its products. For example, in the beta-galactosidase activity assay the reporter protein, glycoside hydrolase enzyme, hydrolyzes the added product X-gal, a galactose linked to a substituted indole and thereby produces an intense blue product. The latter can be easily quantified by using a plate-reader.

Similarly, fluorescent proteins are often used as reporter genes. These proteins exhibit bright fluorescence when exposed to light in a certain wavelength range. The green fluorescent protein (GFP) is probably the most common one, but there exist many variations which emit many kinds of different colors. For RNA design studies, this readout method is very popular as the experimental setup is easy and cheap. There are no antibodies or dyes necessary, and the protein can be detected and quantified by many fluorescence detecting devices.

Instead of measuring the mean fluorescence of a cell culture, it is also possible to quantify the fluorescence of single cells by using single-cell flow cytometry [54]. There, a liquid stream of cells passes the measuring system consisting of a detector and an amplifier which

increases the signal. With this high-throughput method, thousands of cells or particles can be analyzed per second, making this method very popular for RNA design applications. Using a fluorescence-activated cell sorting (FACS) machine, these cells can even be sorted into various containers, e.g. separating functional or dysfunctional devices, which opens the way for further analysis steps.

5.5.3 *Analytical methods*

Any of the afore mentioned detection methods is perfectly suited to perform functional tests in order to verify certain steps and the overall goal of the artificial device. However, these methods reveal no information about the exact mechanism of the actual functionality. Especially if the designed construct does not show the proposed behavior, more analytical tests must be performed. This includes to unveil the structure of the designed molecule, to analyze kinetic processes, and to utilize smart controls or mutational studies.

RNA STRUCTURE DETERMINATION Sequences of artificial RNA devices are designed in a way that they adopt certain structural conformations which hopefully lead to the desired functionality. Therefore, besides testing for proper functionality it is important to also verify the structural conformation of the designed molecule as an intermediate result between sequence and function. Some of the most commonly used methods are presented below.

In-line probing is a very basic and stable method to get an idea about changing RNA structural conformations by detecting single stranded regions of the molecule. It uses a natural property, which is that the backbone of single stranded RNA regions degrades much faster over time than the double stranded parts as they are more flexible and unstable. After an initial incubation step, the partially degraded RNA is separated by size through gel electrophoresis, where only pieces containing the attached 5′ label are visible. Thus, bands only occur at frequently degraded positions and their size correlates to the nucleotide position. Riboswitches or aptamers can therefore be incubated with and without trigger molecule. Changing band patterns will directly unveil structural changes or ligand binding, the positions of these changes and to some extent the binding affinity. [147]

Similar information is gained by an RNase footprinting assay. In addition to natural degradation, a ribonuclease (RNase) is added which digests the RNA gradually. Binding of a ligand to the RNA molecule protects the binding region from being cut by the RNase. PAGE is again used to separate the 5′-labeled fragments by size. At any cut position there will be bands with the corresponding size. Thus, the binding site leaves a band free area, the so-called footprint.

The toeprint assay [77], or primer extension inhibition assay, works despite its similar name a little different from the footprinting assay. Nevertheless, both methods have in common that they reveal positions bound by a protein or other interactions with the RNA molecule. In the toeprint assay, a reverse transcription process starting from a labeled DNA primer at the 3'end of the RNA will produce a library of corresponding cDNAs. At ligand bound positions, the transcriptase cannot proceed and falls of the RNA template. Again, sizes of the cDNA pieces can be analyzed by separating them on an electrophorese gel.

Chemical probing techniques also use reverse transcription and PAGE readout for RNA structure determination. Here, a chemically modified base or backbone will form a roadblock for the reverse transcriptase, leading to cDNA transcripts with the size corresponding to the distance of the modification to the 3' end. There are different chemicals in use which label various positions of single- or double-stranded RNA. Dimethyl sulfate (DMS) for example modifies cytosines and adenines, either single stranded, paired at an end of a stem structure, or next to a structured GU pair. By using several chemicals with similar constraints, it is possible to reveal the RNA structure from the PAGE gel bands. In case of enzymatic probing, site specific RNases cut the RNA strand at certain positions. Similar constraints also allow a detailed analysis of the electrophoresis gel patterns. Waldminghaus et al. [196], for example, used RNase T1 enzymatic probing which introduced cuts at single-stranded guanine positions in order to analyze the designed thermoswitches at varying temperatures.

The selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) method works similar but uses reagents that modify the backbone of the RNA dependent of the accessibility and flexibility of this region, largely unbiased by base identity. Unstructured regions show more adduct formation than structured positions, leading to electrophoresis bands of cDNA showing up at the corresponding sizes. As PAGE readout is only quantitative to a certain extent, more information can be gained in doing a deep sequencing run of the obtained (bar-coded) cDNA fragments. This so-called SHAPE-Seq is a high-throughput method with accurate quantitative outputs. Recently, this method was extended to even work *in vivo* [199]. Furthermore, the resulting SHAPE reactivities can used to guide *in silico* structure prediction and modeling [35, 72, 110].

X-ray crystallography, the most popular 3D structure determination method for revealing protein structures in a detailed fashion, is of limited use for RNA molecules. Reasons are that it is quite hard to obtain crystals from RNA macro-molecules and that RNA tertiary structures are very versatile [82, 151]. The "frozen" crystal state only represents one possible structure of the whole ensemble

present in a liquid solution and it is not guaranteed that it is the most frequent one. Various types of atomic force microscopy (AFM) experiments were developed to reveal the structure of RNA molecules [76] with similar downsides to crystallography like strongly distorted structures due to the fixation techniques, artificial rigidity and in case of AFM also low resolution images.

nuclear magnetic resonance spectroscopy (NMR) [189] is sometimes better suited for determining local tertiary structural features of RNA molecules [33]. The molecule can be analyzed in liquid solution as no crystal is needed for this technique. Although this methods provides useful and detailed information about the 3D structure, it is still tricky and demanding, especially for bigger molecules [55]. Nevertheless, this technique is promising and leads to stunning insights, even kinetic observations over time are possible.

KINETIC MEASUREMENTS Buck et al. [18] used NMR to obtain a time-resolved structure of a ligand induced folding process. In a recent study, it was even possible to determine cotranscriptional intermediate structures using NMR, therefore directly revealing the kinetic mechanism of riboswitch regulation [79]. Watters et al. [200] use SHAPE-seq to experimentally characterize cotranscriptional RNA folding at nucleotide resolution.

A so-called Förster resonance energy transfer (FRET) experiment can also be utilized to obtain folding information over a period of time. In this experiment, fluorophores are attached to either the C_5 or the C8 atom of the backbone sugar. Thus, if those nucleotides are close together a high energy transfer efficiency can be measured between the fluorophores. If repeated with different nucleotide positions, these experiments lead to time-resolved structural information of the RNA molecule [176].

Although such sophisticated techniques allow for detailed kinetic insights, some studies use less demanding methods to measure various kinetic properties of artificial RNA devices. Isaacs et al. [87] successfully determined the dissociation constant of RNA duplex formation using a simple reverse transcriptase assay. The *in vitro* synthesized RNAs are incubated for various time-spans while one molecule is reverse transcribed into cDNA. The transcription stalls at the stable duplex binding site, making a quantification of duplex formation during time possible. A more accurate method to measure the affinity of a ligand to the RNA molecule is ITC. Ceres et al. [25] utilized it to distinguish the affinity of a ligand to its transcriptional expression platform designs. Man et al. [118] showed that a simple electrophoretic mobility shift assay (EMSA) also returns a semi-qualitative answer to the ligand affinity question.

Dawid et al. [34] demonstrated that utilizing standard methods can lead to useful kinetic insights. They used northern blots to deter-

mine the effect of transcription speeds of different RNA polymerases (RNApols) or nucleotide concentrations on the designed devices.

BIOLOGICAL AND MECHANISTIC CONTROLS Complex methods are rarely necessary to confirm proposed mechanisms and functional details of artificial constructs. Often smart and well considered experimental setups are sufficient to reveal stunning results. Several smart controls designed in a way that immediately infer specific mechanistic details, lead to comprehensive insights and should be incorporated in every design pipeline. Such controls are, for example, simple but conclusive mutational studies where e.g. the functionality is disrupted and regained. These confirm ligand binding sites, complementary regions of RNA binding or even the ability to form certain structural motifs. Despite the current approaches, such controls should always be designed *in silico*, or at least sanity checked by using structure prediction algorithms to avoid undesired behavior.

For instance, to confirm the proposed model of an sRNA being the only reason for successful translation activation, Green et al. [65] performed extensive cross-activation tests. These experiments were important to prove that the specific binding of the proposed sRNA indeed triggered the designed toehold switches.

Moreover, the effect of RNA degradation should always be considered when interpreting the results of a functional experiment, especially when designing OFF switches. It can always happen that the trigger activates specific or unspecific degradation of the designed mRNA and thus the output vanishes. This can be easily confirmed or falsified by expressing the artificial system in RNase deficient strains. Rodrigo et al. [154] used RNAse III knockout strains and qPCR quantification to exclude double stranded RNA (dsRNA) degradation. Similar experiments with RNase deficient strains were performed by Man et al. [118] to confirm RNase E dependency of their artificial system. Besides, not only RNases can interfere with designed RNA mechanisms. The chaperon protein HFQ is known to often interfere with sRNA structure and promote duplex formation [117].

Nevertheless, its is probably not feasible to perform every experiment for sanity checking the proposed mechanism and to confirm every mechanistic detail in a well-designed control experiment. However, much can be learned from consistent and reliable experimental data. This information has the potential to speed up and streamline future RNA design studies.

5.6 LIMITATIONS OF CURRENT METHODS

In summary, the introduced universal design pipeline was present in any of the analyzed studies, however with parts missing in some cases. Especially in early papers, almost no bioinformatic work was

performed. Manual and rational designs were often the starting point of these artificial devices. In contrast, most of the evaluated bioinformatic publications of design tools completely miss any biological application. Thus it is often difficult to verify if the theoretical idea is actually working for designs and if it is at all necessary and helpful for generating a proper artificial RNA device. More recent publications such as the design by Green et al. [65] [66] follow the pipeline in every step, including biological analysis of the system, *de novo* design by sophisticated algorithms, *in silico* filtering and analysis and biological testing [27].

Nonetheless, many parts of the pipeline are not perfect yet with important aspects and tools missing. I want to use this last section of the introduction to evince these missing pieces and highlight where, in my opinion, future research is needed.

ANALYSIS OF BIOLOGICAL DATA This important first step is often missing or not specifically outlined in the studied publications. The existent systems were probably analyzed or information gained from published work to minimize the risk of running into unexpected difficulties, however it seems as if the information could be used more efficient in order to streamline and enhance the design process. Furthermore, it is evident that many aspects of fundamental cellular processes are still not fully understood which makes it even more complicated to introduce artificial compounds with desired functionality.

SEQUENCE SAMPLING Schuster et al. [166] and succeeding publications [69, 70, 148] developed many theoretical fundamentals on how sequences and structures of RNA and other biopolymers are linked and how to explore these spaces. Nevertheless, many aspects are still not properly known, including how the exploration strategy of the solution space impacts the outcomes of sequence optimization approaches. Moreover, it is tricky to include sophisticated constraints to exclude undesired solutions and still sample in a fair way from the remaining sequence space. For example, methods allowing more than two structural constraints [114, 182] perform biased sampling leading to warped solution spaces or completely missing solutions.

Software solutions are often limited to specific scenarios and cannot be easily adapted to new needs. This is problematic, as RNA design problems can be very diverse and require flexible adaptation to the specific design. So far, there is no sampling software available that is completely decoupled from a particular optimization method and specific objective goals which makes it hard to experiment with different optimization methods and novel objective functions.

OPTIMIZATION METHOD AND OBJECTIVE FUNCTION Although several optimization methods were successfully used to design RNA

sequences with prescribed properties, it is still unclear how much the choice of the actual method influences the quality of the solutions. Only little research has been done so far to theoretically describe the solution space given certain move-steps and objective functions. Thus, no rational choice can be made when trying to choose the perfect method to traverse the solution space. Also statistical approaches comparing various techniques are still missing.

Moreover, the choice of a proper objective function was often neglected. Many design studies used common, generic objectives to optimize their sequences instead of developing a function that precisely describes the mechanism and properties of the desired device. This probably results from the fact that theoretical publications often used the same objectives for benchmarking their novel tool. The study of Rodrigo and Jaramillo [153] serves as counterexample. They developed an individual objective function, describing the main features of a sRNA induced riboregulation in the 5′ untranslated region (5′UTR) of a target gene. Also Wolfe and Pierce [205] contributed a lot to finding meaningful and conclusive objective functions. Over the years the concept of the ensemble defect was expanded to describe test tubes and recently also to be used in multi-state optimization problems [206].

Nevertheless, only very little research was done concerning the development of protocols and methods to reveal the mechanistic model of an RNA device from biological data. Only this way it can be ensured that the objective function derived from this model targets the right optimization goal. Such a protocol would require extensive testing of existing RNA devices and statistical analyzes to confirm functional details of the process.

IN SILICO FILTERING This part of the pipeline, which aims to narrow down the list of potential candidates, was introduced very early in design studies. It evolved from the urgent need of being unable to handle the huge amount of designs in the laboratory. Thus, this step was often satisfied in a quick and dirty manner with only little attention payed on using proper and perfectly fitting methods and tools. Two examples of publications exhibit a quite exhaustive filtering procedure worth mentioning. Wachsmuth et al. [191] showed that filtering can lead to working results even without any optimization approach. They nicely pinpointed which aspects of their design could go wrong and excluded such solutions already *in silico*. Dotu et al. [40] also exhaustively searched for the right filters which describe the functionality of their hammerhead ribozymes designs best. Nevertheless, this part of the pipeline urgently needs improvements which would benefit the whole process. For example, sophisticated visualization tools would help a lot to cluster, filter, compare, and explore such predicted and measured biological data, especially if the

amount of designs becomes large. Unfortunately, this is still only a small research field missing specified tools and novel ideas on how to solve these problems.

BIOLOGICAL TESTING Testing the designed RNA devices in the lab is still a tedious and time consuming task. Thus, many studies limit the experiments to an absolute minimum necessary to verify the function of the novel riboswitch. Little attention has been payed to systematically prove the proposed mechanistic details of the obtained logic device as such a detailed analysis is labor and budget intensive. Hence, only little knowledge about the chemical and cellular processes involved in the obtained behavior could be gained in such studies. As a result it was often completely unclear why some of the tested candidates worked while others did not. Recent attempts to reveal such information focused on performing various *in silico* predictions and searching for correlations to the functional test results [65, 192]. Unfortunately, these correlations were weak and many of the results could not be explained. However, exactly such mechanistic details are vastly important for a successful iterative design approach where mechanistic testing results are immediately used to enhance the design, leading to iterative rounds of design and analysis cycles. This way, detailed analyses of previous iterations directly help to refine the current design goals. Green et al. [65] followed such a design approach in a well-structured and methodically clear way with great success.

I hope this introduction helped to get detailed insights into the current state of this fascinating research field. Although many milestones were already achieved that enable the design of artificial RNA devices with prescribed functionality, many hurdles are still blocking the way. Nevertheless, I am confident that it will be possible to design functional RNA molecules solely *in silico* in a high throughput manner, to better support the fascinating area of cellular engineering and *in vivo* computation. The following chapters highlight my contribution to this research field and summarize the work of the recent years.

Part II

PUBLISHED WORK

6

RNADESIGN GRAPH COLORING PROTOTYPE

Christian Höner zu Siederdissen, Stefan Hammer, Ingrid Abfalter, Ivo L. Hofacker, Christoph Flamm, and Peter F. Stadler.
"Computational Design of RNAs with Complex Energy Landscapes.", 2013 in Biopolymers 99 (12): 1124–1136.
doi:doi:10.1002/bip.22337

SUMMARY

This publication contributed to the sequence sampling step of the described design pipeline. The published graph coloring algorithm solves the problem of sampling sequences with multiple structural constraints in a guaranteed uniform manner. Thus, it provides a solution to the specificity problem of obtaining candidate sequences. `RNAdesign` is a prototype implementation written in Haskell, which includes the sampling algorithm and a Monte Carlo simulated annealing (MCSA) optimization approach where the user is able to define an objective from a set of given functions.

AUTHORS CONTRIBUTION

IA, PS, IH and CF developed the fundamental ideas of the algorithm, CS and SH implemented software and performed the statistical analyses. All Authors contributed to the manuscript mainly written by CS.

LICENSE

Computational Design of RNAs with Complex Energy Landscapes

Christian Höner zu Siederdissen,[1] Stefan Hammer,[1] Ingrid Abfalter,[2] Ivo L. Hofacker,[1,3,4] Christoph Flamm,[1] Peter F. Stadler[1,4,5,6,7,8]

[1] *Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Wien, Austria*

[2] *Research Support, Johannes Kepler University Linz, Altenberger Str. 69, 4040 Linz, Austria*

[3] *Bioinformatics and Computational Biology Research Group, University of Vienna, A-1090 Währingerstraße 17, Vienna, Austria*

[4] *Center for Non-coding RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, DK-1870 Frederiksberg, Denmark*

[5] *Department of Computer Science and Interdisciplinary Center for Bioinformatics, Bioinformatics Group, Härtelstraße 16-18, D-04107 Leipzig, Germany*

[6] *Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany*

[7] *RNomics Group, Fraunhofer Institut für Zelltherapie und Immunologie, Deutscher Platz 5e D-04103 Leipzig, Germany*

[8] *Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501*

Received 2 May 2013; accepted 17 June 2013
Published online 2 July 2013 in Wiley Online Library (wileyonlinelibrary.com). DOI 10.1002/bip.22337

ABSTRACT:

RNA has become an integral building material in synthetic biology. Dominated by their secondary structures, which can be computed efficiently, RNA molecules are amenable not only to in vitro and in vivo selection, but also to rational, computation-based design. While the inverse folding problem of constructing an RNA sequence with a prescribed ground-state structure has received considerable attention for nearly two decades, there have been few efforts to design RNAs that can switch between distinct prescribed conformations. We introduce a user-friendly tool for designing RNA sequences that fold into multiple target structures. The underlying algorithm makes use of a combination of graph coloring and heuristic local optimization to find sequences whose energy landscapes are dominated by the prescribed conformations. A flexible interface allows the specification of a wide range of design goals. We demonstrate that bi- and tri-stable "switches" can be designed easily with moderate computational effort for the vast majority of compatible combinations of desired target structures. RNAdesign is freely available under the GPL-v3 license. © 2013 Wiley Periodicals, Inc. Biopolymers 99: 1124–1136, 2013.

Keywords: RNA sequence design; inverse folding; multi-stable structures; graph coloring

This article was originally published online as an accepted preprint. The "Published Online" date corresponds to the preprint version. You can request a copy of the preprint by emailing the Biopolymers editorial office at biopolymers@wiley.com

INTRODUCTION

A wide variety of RNA elements requires transitions between two or more different spatial conformations. A prime example are riboswitches. These regulatory elements, which are abundant in prokaryotes, regulate mRNA transcription or translation in response to metabolite concentrations, reviewed by

Correspondence to: Peter Stadler; e-mail: studla@bioinf.uni-leipzig.de
Contract grant sponsor: Austrian Fonds zur Förderung der wissenschaftlichen Forschung (FWF) "RNA regulation of the transcriptome"
Contract grant number: SFB F43
Contract grant sponsor: Deutsche Forschungsgemeinschaft
Contract grant number: STA 850/15-10

Serganov and Nudler.[1] Substrate binding or unbinding at the aptamer component of the riboswitch triggers a conformational change of the molecule that is propagated to the effector location, where it causes the formation or destruction of a terminator hairpin, or the exposure or sequestration of the Shine-Dalgarno sequence. RNA thermometers are a variation of this theme.[2] Similar mechanisms have been reported in eukaryotic genome regulation for elements in untranslated parts of mRNAs that respond to protein binding.[3] Major conformational changes also play a crucial role in viroid processing,[4] in the replication cycle of self-replicating RNA synthesized by Q_β-replicase,[5] the folding of rRNA after excision of self-splicing introns,[6] and the functioning of the hok/sok host-killing system.[7]

Regulatory RNA elements that respond to external triggers are attractive components in synthetic biology,[8–10] making the design of novel RNA components an interesting task of practical importance.[11] Recent success in designing a synthetic riboswitch acting on transcription emphasizes the feasibility and usefulness of rational design approaches for RNAs with distinct prescribed conformations.[12] Similar principles have been used to construct a riboswitch based on IRES structures.[13]

The task of designing an RNA sequence with a prescribed secondary structure as its ground state is known as the "inverse folding problem". Although this combinatorial problem is hard in general,[14] most instances of practical interest can be solved by simple hill-climbing heuristics: An initial random seed is progressively "mutated" to approach the desired folding properties. This simple idea is the basis of RNAinverse[15] and later, more efficient approaches such as RNA-SSD,[16,17] as well as the very efficient optimization algorithm[18] implemented in NUPACK.[19] INFO-RNA[20] uses a dynamic programming approach to compute the most stable sequence for the prescribed secondary structure as a starting point for a local search heuristic. A multi-objective optimization approach considering the trade-off between thermodynamic stability and structural similarity is used in MODENA.[21] Inverse folding problems can also be solved by an exact branch and bound algorithm.[22] An alternative, essentially enumerative approach that covers certain classes of pseudoknots is described in Ref. 23. RNAexinv[24] includes some additional attributes and also the mutational robustness and the minimum free energy. As an alternative to iterative improvement, a global sampling approach was proposed in Ref. 25.

Much less is known about the design problem for multistable RNAs. In this case, the design goals involve more complex properties of the energy landscape such as prescribed local optima and energy barriers. A web tool for this type of design problem is ARDesigner,[26] which implements many of the ideas discussed in Flamm et al.[27] The most salient difference between the inverse folding problem for single and multiple structural constraints is that a solution need not exist in the latter case.[27] Thus, computing feasible solutions as starting points for subsequent optimization steps, and—in particular—sampling these starting points so that biases can be avoided, becomes a nontrivial problem. Flamm et al.[27] describe a uniform sampling procedure for two prescribed secondary structures. In this case, a nonempty set of feasible solutions always exists.[28]

The general case has been discussed by Abfalter et al.,[29] but no corresponding software has become available. Lyngsø et al.[30] implemented a much simpler, approximate sampling of initial conditions together with a genetic algorithm in their Frnakenstein tool. Similar ideas have been used by Ramlan and Zauner.[31] In the present contribution, we consolidate and expand on our earlier computational approaches to designing RNA sequences with multiple prescribed conformations that satisfy additional, complex constraints and provide with RNAdesign an implementation for a wide variety of RNA design tasks.

THEORY

Notation

Let $\mathcal{A}$ denote the alphabet of monomers and let $\mathcal{B} \subset \mathcal{A} \times \mathcal{A}$ be the set of allowed base pairs. We assume that B is symmetric. For RNA, we have $\mathcal{A} = \{A,U,G,C\}$ and $\mathcal{B} = \{AU, UA, GC, CGGU, UG\}$. We denote a sequence consisting of n monomers $x_i \in \mathcal{A}$ by $x_1x_2\ldots x_n$.

A secondary structure Θ on x is a set of pairs (i,j), (i,j), $1 \le i < j \le n$ such that for all $(i,j), (k,l) \in \Theta$ holds

1. $(x_ix_j) \in \mathcal{B}$
2. $(i,j) = (k,l)$ or $\{(i,j)\} \cap \{(k,l)\} = \varnothing$, i.e., Θ is a matching on $\{1,2,\ldots,n\}$
3. If $i < k < j$ or $i < l < j$, then $i < k < l < j$, i.e., base pairs do not cross.

Given a secondary structure Θ, we write $C[\Theta] = \{x \in A^n | (x_i, x_j) \in B$ for all $(i,j) \in \Theta\}$ for the set of sequences that can form the structure Θ. We say that a sequence $x \in C[\Theta]$ is compatible with Θ.

To every pair (x, Θ) of a sequence x and a secondary structure Θ compatible with X, an energy $f(x, \Theta)$ can be assigned. In practice, $f(x, \Theta)$ is computed as the sum of energy contributions of stacked base pairs and loops, which in turn are derived from a large body of accurate thermodynamic measurements.[32]

The energy landscape for a fixed sequence x is defined by the function $f_x : \Theta \to f(x, \Theta)$ together with an adjacency relation $\sim$ defined between secondary structures. As usual, we

regard two secondary structures as adjacent if they differ by a single base pair. Later, we will use properties of f_x in the specification of the design goals. This energy landscape is a high-dimensional combinatorial object that cannot be visualized in its entirety.

Coarse-grained representations muct thus be employed. Ding et al.,[33] proposed clustering of a Boltzmann sample. Quarta et al.[34] favoured a scatter plot of folding energy versus base pair distance from the ground state. RNA2Dfold[35] considers an abstracted energy surface with two anchor points. Throughout this presentation we will make use of the barrier tree of the landscapes as a comprehensive presentation.[36,68] The leaves of the barrier tree are the local minima (metastable states) of the landscapes, which are connected by the saddle points separating them. The height of a node corresponds to the energy of the corresponding secondary structure, so that both energy differences between (meta)stable states and their separating barriers can be read off the tree immediately.

For the examples in this contribution we use exhaustive enumeration with the programs RNAsubopt[37] and barriers to construct exact barrier trees. Approximations could be obtained by folding algorithms that directly address metastable structures[38,39] and heuristics to estimate saddle points.[40] Using the barrier trees enables a wide variety of design goals to be expressed in a concise manner.

Now consider a collection $\{\Theta_1,\Theta_2\ldots,\Theta_M\}$ of M distinct secondary structures of the same length n. Is there a sequence x that is simultaneously consistent with all the Θ_i ? If so, our task is to determine x such that all the prescribed Θ_i features as prominently as possible among the structures formed by x. We first address the existence question.

The Search Space C

Given $\{\Theta_1,\Theta_2\ldots,\Theta_M\}$, the set of sequences simultaneously consistent with all these secondary structures is

$$\mathcal{C} = \mathrm{C}[\Theta_1] \cap \mathrm{C}[\Theta_2] \cap \ldots \cap \mathrm{C}[\Theta_M]. \quad (1)$$

Hence, the design problem is solvable if $\mathcal{C} \neq \varnothing$. This question is addressed in Flamm et al.[27]

The dependency graph $G = G(\Theta_1, \Theta_2, \ldots, \Theta_M)$ has n vertices corresponding to the sequence positions of x. There is an edge connecting $k \in V(G)$ with $l \in V(G)$ if and only if (k, l) is a base pair in at least one of the secondary structures Θ_i, i.e.,

$$E(G) = \bigcup_{i=1}^{M} \Theta_i. \quad (2)$$

see Figure 1.

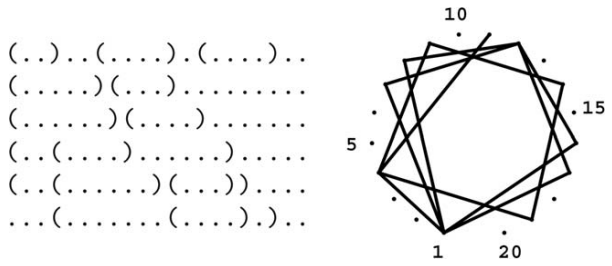


FIGURE 1 The M=6 secondary structures on the l.h.s. give rise to the dependency graph G in which each edge corresponds to a base pair in at least one of the input structures. Each edge thus constrains the set of possible sequences: the endpoints of each edge must be different nucleotides that can pair with each other. To make the example smaller, the minimum number of unpaired positions in a hairpin is reduced to 2 here. Adapted from Ref. 29 with permission from belleville Verlag Michael Farin.

Generalized Intersection Theorem. Suppose $\mathcal{B} \subseteq \mathcal{A} \times \mathcal{A}$ contains at least one symmetric base pair, and G is the dependency graph of a set of secondary structures.[27] Then:

1. $\mathcal{C} \neq \varnothing$ if G is bipartite. For the RNA alphabet, bipartiteness of G is also a necessary condition.
2. There are $|\mathcal{C}| = \prod_{\text{components } \psi \text{ of G}} F(\psi)$ sequences in $\mathcal{C}$, where $F(\psi)$ is the number of sequences that are compatible with a connected component ψ of G.

The proof of the intersection theorem makes use of two ingredients. (1) Base-pairing in the natural alphabet divides the letters into two subsets $V_1 = \{G, A\}$ and $V_2 = \{C, U\}$ with base-pairs allowed only between the subsets but not within them. (2) Since edges in the dependency graph are base pairs and must have a nucleotide from v_1 at one end and a nucleotide from v_2 on the other end, it must be possible to color the vertices of the dependency graph with v_1 and v_2. A simple breadth-first-search coloring algorithm can be used to test whether G is bipartite.

Even for $M = 3$ different secondary structures, it is simple to construct triples of structures with conflicting base pairs that lead to a triangle in $G(\Theta_1, \Theta_2, \Theta_3)$. In order to estimate the probability of a nonempty C, we sampled secondary structures with uniform probability as described by Tacker et al.[44] and checked whether the dependency graph of M-tuples of structures is bipartite. For $M \geq 3$, we find an exponential decrease with sequence length, see Figure 2. However, the exponent is very small for $M = 3$, indicating that tri-stable switches in particular should not be uncommon.

The exponential decrease with length n can be explained as follows: the obstructions to bipartiteness can be small, i.e., triangles corresponding to just three incompatible base pairs in three sequences. It appears with some finite probability in a

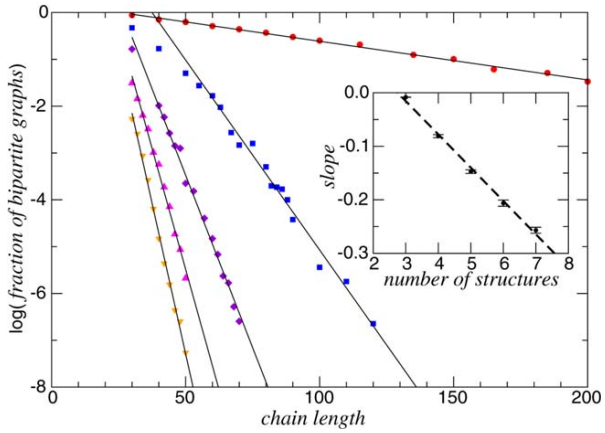


FIGURE 2 Statistics of the fraction of bipartite graphs versus sequence length with different numbers M of prescribed structures generated with uniform distribution for the set of all secondary structures of fixed chain length. •M=3, ■M=4, ◆M=5, ▲M=6, and ▼M=7.

triple of positions. Hence the chance to avoid such configurations in long sequences decreases exponentially in the case of random, unrelated input structures. If the mutual structure distances are bounded, however, so is the chance to find inconsistent configurations.

Sequence Design as Graph Coloring

In this section, we outline a dynamic programming approach that can be used to enumerate and uniformly sample from $\mathcal{C}$. To this end, we consider sequences as $\mathcal{A}$-colorings of the dependency graph G, that is, as maps $c : V(G) \to \mathcal{A}$ which obey the pairing rules, i.e., $(c(k), c(l)) \in \mathcal{B}$ for all $(k, l) \in E(G)$.

The important observation for our purposes is that colorings can be obtained by combining partial colorings: Let H be a subgraph of G, and consider two vertex sets $U, W \subseteq V(H)$. A partial coloring of U in H is a map $c_U : U \to A$ such that $(c(u), c(v)) \in \mathcal{B}$ for all $u, v \in U$ with $(u, v) \in E(H)$. Partial colorings c_u and c_w on U and W, respectively, are compatible if (i) $c_u(y) = c_w(y)$ for all $y \in U \cap W$ and (ii) $(c_U(u), c_w(v)) \in \mathcal{B}$ for $u \in U$ and $v \in W$ with $(u, v) \in E(H)$. Denote by $\partial(U, W)$ the set of vertices in which U and W overlap or are adjacent. Denote by $c(U, a)$ and $c(W, b)$ the sets of all those colorings on U and W that are fixed to some assignments a and b on $\partial(U, W)$. Then the set of colorings in $U \cup W$ consists exactly of the combinations of colorings on U and W for which a and b are consistent, i.e., identical on $U \cap W$ and satisfying the color constraints on adjacent vertices. For simplicity, write $c(U \cup W) = \cup a, b, c(U) \circ c(U \cup W)$.

The idea is to use this type of composition of the set of all conflict-free colorings for the step wise construction of $\mathbf{c}(G)$. Graph coloring is a well-known NP-complete problem.[41] Of course, our approach cannot overcome this in general. We can, however, search for a decomposition of $\mathbf{c}(G)$ that allows us to concatenate partial colorings with as little resource consumption as possible.

It is particularly easy to compose colorings at cut vertices. In the first step, we therefore decompose (each connected component of) G into its blocks, i.e., the two-connected components and those edges that are not contained in a cycle, Figure 3. The blocks and cut-vertices, i.e., the vertices common to two or more blocks, can be determined in linear time. For each block B, we then determine the sets of colorings $\mathbf{c}(B,q)$ with fixed colors q assigned to cut vertices of G in B. The two-connected components are arranged in a tree. Choosing an arbitrary root, we can compose the colorings recursively by traversing from the leafs up.

Since two-connected components can be large, we need to decompose them further. While it might seem natural to use successive higher-order connectivities for this purpose, we explore here an alternative approach.

Let G be a two-connected graph. An ear decomposition of G is a sequence $\varepsilon = (P_0, P_1, \ldots)$ of paths where $G_0 = P_0$ is a single vertex,

$$G_k = \bigcup_{i=0}^{k} P_i \tag{3}$$

and $G_\mu = G$, with $\mu = |E| - |V| + 1$ being the dimension of the cycle space of G. An ear decomposition is "open" if, P_i, $i \geq 1$, has two distinct end points in G, see Figure 4 for an example. An open ear decomposition exists exactly for two-connected graphs.[42] We use the EDS algorithm by Maon et al.[43] to produce the decomposition. For details, see Implementation section.

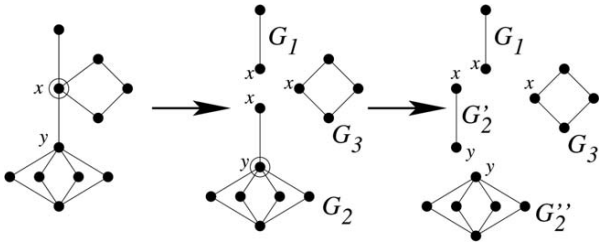


FIGURE 3 Decomposition of the dependency graph of Figure 1 into its blocks. Colors need to be constrained only at the cut vertices x and y. The number of colorings in this example is $|c(G)| = \sum_{c_y} |c(G''_2, c_y)| \cdot \sum_{c_x} |c(G'_2, c_x c_y)| \cdot |c(G_1, c_x)| \cdot |c(G_3, c_x)|$. Adapted from Ref. 29 with permission from belleville Verlag Michael Farin.

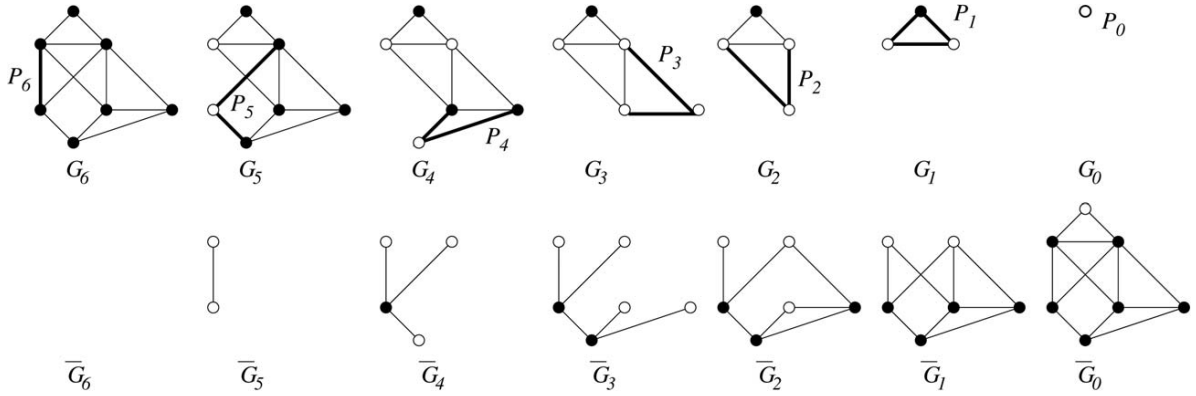


FIGURE 4 Graphs associated with an ear-decomposition. Top, Ear-decomposition of a block: In each step from G_6 to G_0, a path (ear) is removed until a central cycle is left. Bottom, The corresponding $\bar{G}_k$ of each step is shown. The attachment points of the ears are depicted by unfilled vertices. For more compact illustration, a non-bipartite graph is shown. Adapted from Ref. 29 with permission from belleville Verlag Michael Farin.

With the ear decomposition ε of G we associate a sequence of subgraphs of G for which we construct the colorings:

$$\bar{G}_k = \bigcup_{i=k+1}^{\mu} P_i. \tag{4}$$

By definition $\bar{G}_0 = G$ and $\bar{G}_\mu = \varnothing$, the empty graph. Further, we have

$$\bar{G}_k = P_{k+1} \cup \bar{G}_{k+1}. \tag{5}$$

The intersection $A_k := G_k \cap \bar{G}_k$ is completely disconnected for each k and by construction forms a cut in G. We call these vertex sets the *attachment points* of $\bar{G}_k$ on G_k.

Our task is now to construct and evaluate the sets $c(\bar{G}_k, a_k)$ of colorings of the graph $\bar{G}_k$ with colors a_k fixed on the set A_k of its attachment points. To this end, we start from the outer-most path $\bar{G}_{\mu-1}$, for which the colorings are easily constructed and counted, and proceed inwards until we reach $\bar{G}_0 = G$.

These sets $\mathrm{c}(\bar{G}_k, a_k)$ can be computed by combining, in the above sense, colorings of the path P_{k+1} with colorings of the subgraph $\bar{G}_{k+1}$, again with prescribed assignments A_{k+1} at its attachment points A_{k+1}.

$$\mathrm{c}(\bar{G}_k, a_k) = \mathrm{c}(P_{k+1}, b) \circ \mathrm{c}(G_{k+1}, a_{k+1}). \tag{6}$$

Since A_k, A_{k+1}, and P_{k+1} are not disjoint in general, the colorings a_k, b, and a_{k+1} at the sets of attachment points must, of course, coincide at their intersections. However, as A_{k+1} and $\bar{G}_{k+1}$ are not connected by any other edges in G, the concatenation $\circ$ of the coloring sets is constrained only by the common vertices. In particular, the end points of P_k are attachment points in A_k, and the attachment points of $\bar{G}_{k+1}$ are either contained in the interior of P_{k+1} (b and a_{k+1} coincide on $A_{k+1} \setminus A_k$), or they are attachment points of $\bar{G}_{k+1}$, and thus $a_k = a_{k+1}$ on these vertices.

The path P_{k+1} is subdivided by the interior attachment points into $|A_{k+1} \setminus A_k| + 1$ subpaths. For any coloring condition B, it is straightforward to compute the set of colorings on a path of length l with fixed colors at its endpoints, see e.g. Flamm et al.[27] From these, colorings of longer paths with fixed colors at the attachment points are easily obtained.

This decomposition of the sets of colorings forms the basis for the recursive enumeration of colorings by dynamic programming. In each decomposition step k, we need to store the number of colorings $|\mathrm{C}(\bar{G}_k, a_k)|$ of $\bar{G}_k$ given a fixed coloring of the attachment vertices, i.e., $|\mathcal{A}|^{|A_k|}$ values. The maximum $\alpha = \max_k |A_k|$ over the steps of the ear decomposition thus determines the memory requirements.

The CPU time required to compute one entry in this matrix is determined by the set $A_{k+1} \setminus A_k$ of attachment points of $\bar{G}_{k+1}$ that are attached to the interior of P_{k+1}. The total effort to count all colorings is therefore $|\mathcal{A}|^{\beta}$ with $\beta = \max_k(|A_k| + |A_{k+1} \setminus A_k|)$. The exponents α and β depend explicitly on the spanning tree of G used in the construction of the ear decomposition. Figure 5 shows that α and β can vary dramatically, depending on the choice of the spanning tree. Note that α and β are strongly correlated. The data suggest that $|A_{k+1} \setminus A_k| \in \{0, 1, 2\}$, i.e., in each step at most two earlier attachment points are consumed.

Generating Colorings

For small connected components of $\mathcal{C}$ it is possible (and efficient) to explicitly enumerate all colorings. For large

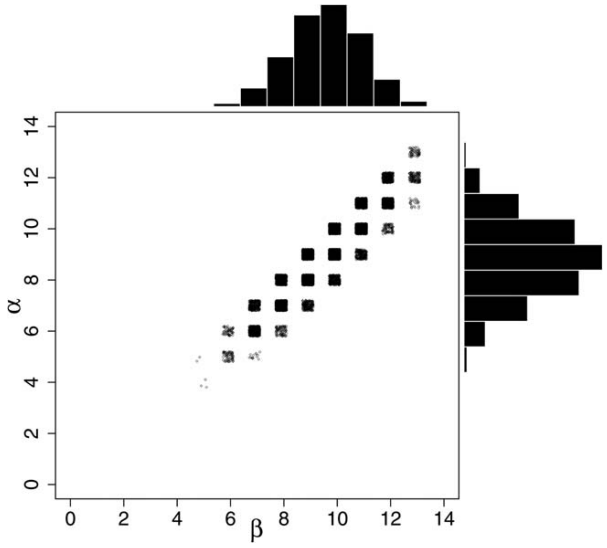


FIGURE 5 Distribution of the exponents α (memory requirements) and β (CPU runtime) for a fixed graph with $\mu=13$. Thousand spanning trees were generated by replacing randomly selected tree edges with non-tree-edges. Each vertex was used as root for the Maon/Schieber algorithm to compute the ear decomposition. The (rare) best spanning trees yield $\alpha=4$ and $\beta=5$ in this example.

components, however, this becomes inefficient, and we must resort to a sampling technique. To this end, we use the generic idea of stochastic backtracing in dynamic programming, which is used in a similar context, for instance, to generate samples of secondary structures.[38,44,45] Here, we set for each k from 1 to μ the colors a_k for the attachment points with probabilities proportional to $|\mathrm{C}(\bar{G}_k, a_k)|$. In the second step, we sample the colorings for the connecting paths, whose end points now have fixed colors, as described in Flamm et al.[27] This simple procedure ensures uniform sampling from C and hence unbiased generation of feasible solutions. Biased samples could, of course, be generated with less effort, for example, by depth first search with a random vertex order. In many cases, however, it is desirable to minimize *a priori* sequence biases.

Local Moves and Optimization

All heuristics for RNA design use "local moves" to navigate $\mathcal{C}$ in an attempt to further improve the sequence. The most obvious move, i.e., changing the color of a single vertex of G, however, will typically not be feasible as it destroys compatibility. Instead we need to always replace all vertices belonging to one connected component of the dependency graph. For designs with a single target this reduces to mutating a single unpaired base or replacing a base pair with another one. This type of structure-dependent move is also used, for instance, to explore neutral networks of sequences folding into the same secondary structures.[46] As the number of target structures grows, the dependency graph will have larger but fewer connected components. This means that the fraction of the sequence changed in a single "local move" becomes larger and larger.

In the context of sequence design with design goals specified in terms of the energy landscape, locality in terms of sequence is highly desirable. The RNA energy model has the interesting property that the difference in minimum free energy between two sequences that differ by a single nucleotide is bounded by a constant C.[47] This is a consequence of the additivity of the energy model, which limits the effect of a mutation to the maximum energy difference between two adjacent loops upon removing their separating base pair, in practice twice the maximum stacking energy. As an immediate consequence

$$|f(x,\Theta) - f(x',\Theta)| \leq c d_H(x,x'), \tag{7}$$

where $d_H(.,.)$ denotes the Hamming distance. Small changes in the sequence therefore cause only moderate changes in the Boltzmann distribution of structures and are thus less prone to destroying achievements of past optimization steps.

The design goals are represented by an objective function $\chi_i : \mathcal{C} \to \mathbb{R}$ that assigns a "fitness" to each sequence x, i.e., a feasible coloring of G. We use a simple, Simulated Annealing-like strategy to optimize χ_i. In each step, a candidate x' is generated by a local move in one of the components of G. We accept x' if

$$\chi_i(x') \leq \chi_i(x) + t, \;\; t \sim \exp(\lambda) \tag{8}$$

The new candidate sequence x' is always accepted if it is better according to the optimization criterion Θ than its parent x. To avoid locally optimal traps, a candidate sequence is also accepted if the energy difference is less than an exponentially distributed random variate (drawn new each time). The parameter λ controls the speed with which local optima are left again.

Design Goals

This fitness function χ_i can combine many features of the energy landscape of x that can be expressed in terms of the secondary structure model. Examples of such building blocks are properties of the Boltzmann ensemble of secondary structures of x such as its partition function $Z(x)$, the ensemble free energy $g(x) = -RT\ln Z(c)$, the minimum free energy $f(x) = \min_\Theta f(x,\Theta)$, the base-pairing probability matrix $\boldsymbol{P}(x)$, and the energy of a given structure $f(x,\Theta)$. All these properties are readily computed by RNA folding algorithms as implemented, for instance, in the Vienna RNA Package.[15,48]

A basic design task, on which we focus here, is to construct RNA sequences for which the prescribed structures Θ_i have

nearly the same folding energy and which together dominate the Boltzmann ensemble. The Θ_i will thus correspond to the ground state and the most important metastable states in the fitness landscape. The simplest fitness function for this task aims at simultaneously minimizing the energy of the Θ_i, for instance

$$\chi_i(x) = \max_{i=1\ldots M} f(x, \Theta_i). \quad (9)$$

Since optimization of Eq. (9) forces an increase in the fraction of the most under-represented target structure, it leads to comparable abundances of all prescribed structures. The advantage of this ansatz is that it can be evaluated very efficiently, requiring only the determination of the energy of M individual secondary structures and avoids the use of the computationally demanding RNA folding algorithm. The effort to evaluate $\chi_i(x)$ is only $\mathrm{O}(Mn)$, compared with the cubic in n runtime of RNA folding. A disadvantage, however, is the lack of direct control over the ground state and hence over the ensemble in which the Θ_i are embedded.

Zadeh et al.[18] argued that design fitness functions should not only contain the positive design goals but also encapsulate negative design goals, i.e., they should explicitly penalize unwanted structures in the Boltzmann ensemble. A good example is the ensemble defect $d(x, \Theta)$, defined as the expected base pair distance of a random structure picked from the Boltzmann ensemble of the target structure Θ. It can be computed in quadratic time from $\mathrm{P}(x)$.[18] The sum of the ensemble defects is one of several conceivable generalizations of the multi-target design problem.

Flamm et al.[27] used a different form of the objective for bistable structures, aiming directly at minimizing the difference between the energies of the individual structures and the ensemble free energy. For M structures, this approach yields

$$\chi_i(x) = \left(\sum_{k=1}^{K} f(x, \Theta_k) - g(x)\right) + \gamma \sum_{k<1} \left(f(x, \Theta_k) - f(x, \Theta_1)\right)^2. \quad (10)$$

The first part of Eq. (10) minimizes the difference between the energies of the target structures and the Gibbs free energy of the ensemble, while the second part yields targets that have approximately the same energy. The weight γ allows us to favour one goal over the other. Fitness functions based on RNA folding are expensive to evaluate but promise better designs. An appealing approach is thus to first find a sequence using Eq. (9), which is then used as the initial seed for further optimization using Eq. (10) or another scheme.

Additional design goals can easily be included in χ_i. For instance, a prescribed sequence composition can be approached by suitable penalty terms for sequence bias. In particular, a log-multinomial function is available that allows penalizing mono-nucleotide distributions that deviate from a user-selected probability vector. More elaborate features of the fitness landscape, such as minimum heights of energy barriers, could also be included as discussed in Flamm et al.,[27] albeit at high computational cost.

Summary of the Design Algorithm

The complete design algorithm consists of the following steps:

1. INPUT: a set of secondary structures $\{\Theta_i | 1 \leq i \leq M\}$ and the objective function Θ.
2. Construct the dependency graph $G(\Theta_{1,\ldots,}\Theta_M)$.
3. If G is not bipartite, stop since the design problem is unsolvable.
4. Decompose the graph first into its connected components, then further into the biconnected components, and finally construct an open ear decomposition for each block.
5. Compute the numbers $|\mathrm{C}(H, a)|$ of colorings for the various subgraphs in the decomposition with fixed color assignments at their attachment and cut points.
6. Using these tables, generate sequences with uniform distribution on the set of compatible sequences.
7. Optimize these start sequences by local search with respect to the desired cost function χ_i for the design problem at hand.
8. OUTPUT: Optimized nucleic acid sequence compatible with all predefined structures.

RESULTS

Implementation

We opted to implement the algorithm described above in the functional programming language Haskell.[49,50] Haskell promotes a high-level style of programming and makes it easy to separate the logically distinct facets of an algorithm. In terms of implementation, the functional style of programming sometimes requires expressing an algorithm differently than known from the imperative world.[51,52] Here, this concerns in particular the graph decomposition algorithm and the evaluation of candidate sequences.

The ear decomposition algorithm of Maon et al.,[43] which we use to handle complex components of the dependency graph, is implemented using the functional graph library.[53] The decomposition algorithm by Maon et al.[43] adapts well to a functional description as it is not described in terms of an explicit graph coloring, but rather as a decomposition of the original graph into a spanning tree, tree edges, and non-tree edges. The resulting ears are then colored by legal assignments

of base pairs. The laziness properties of algorithms implemented in Haskell make it possible to handle assignments with a large number of legal assignments without having to explicitly store them.

The evaluation of candidate sequences is a potential performance bottleneck, as it requires evaluation of the energy of sequence candidates given the structure constraints. We make use of fusion,[54] a compiler optimization technique aimed at removing intermediate data structures in functional programs, which often yields executables with a runtime performance comparable to that of C implementations. In particular, we use stream fusion[55,56] during sequence sampling. Energy evaluations are performed in a functional version of the Vienna RNA folding algorithms,[48] which are also fused.[57]

In order to facilitate the exploration of different objective functions, the user can supply χ_i on the command line as a function of the primitive features outlined in the Design Goals section. It is easy to extend both, the design algorithm and the command line parser to include additional terms if necessary. The current implementation of RNAdesign uses Eq. (10) as the default objective function.

In many cases it is important to enforce sequence constraints. For example, the Shine-Dalgarno sequence, the start codon, and the sequence of the ligand-binding aptamer are typically fixed in design problems for riboswitches. We therefore provide an option to restrict the set of nucleotides that may be varied during the design process. The current implementation allows the user to specify, for each sequence position, the set of allowed nucleotides. It is important to note that sequence constraints further shrink $\mathcal{C}$ and may render a design problem infeasible even if the prescribed target structures are consistent. RNAdesign of course detects such cases.

In addition to the functional implementation, we are developing a memory-efficient implementation in C++ to extend the range of applications to large complex problems, i.e., very long sequences and $M \gg 3$ independent target structures.

Artificial SV11-Like Bistable Riboswitches

SV11, a 115 nt long RNA, is a recombinant of the plus and minus strands of the phage-derived MNV-11 RNA. Both molecules are efficient substrates for Q_β replicase and arise consistently in artificial selection experiments.[5] SV11 is frequently used as an extreme example of an RNA whose properties are determined by folding intermediates rather than its thermodynamic ground state alone. Co-transcriptional folding results in a metastable conformation consisting of a Y-shaped multibranched structure and two additional exterior hairpin loops, which is replicated by Qβ replicase. The ground state, in contrast, is a single long helix structure with a hairpin which no longer serves as a template for the Q_β replicase. The metastable structure can spontaneously rearrange to the ground state. This transition is effectively irreversible because of an energy difference exceeding 30 kcal/mol, as computed by RNAeval.[48] For the same reason, the base pair probabilities in the equilibrium ensemble give no indication of important structural alternatives.

Because of its extreme properties, the SV11 structure pair has been used repeatedly as an example, including for design tasks[30] whose goal is to find a sequence that realizes the two conformations with nearly equal energy. In Figure 6, we show that our software readily solves this computational problem. The sequence proposed by our algorithm using the default optimization criterion of Eq. (10) is almost optimal. Both structures differ by only *0.9* kcal/mol. The minimum-free energy structure differs from the complex (red) structure by a single AU base pair.

Using only the criterion in Eq. (10), the algorithm required 120 s on an Intel Core i5-3570K. In total, 50 candidates were created, with a thinning of 100 (i.e., only every 200th candidate is retained) with an initial burn-in period of 100 candidates. Of the 50 candidates that are returned, only the top-most was selected. Other, suboptimal solutions are returned to provide alternatives that can be evaluated before running the algorithm again.

A Tri-Stable Riboswitch

In Figure 7 we present a small, artificial example of a tri-stable system with three prescribed target structures (red, green, and blue). The computational design problem is solved by our tool using the default fitness function Eq. (10) within 10,000 optimization steps, amounting to 45 s on standard PC hardware. The designed sequence readily folds into exactly these three structures. The red structure is the minimum-free energy structure, the green and the blue ones are the first two suboptimal local minima in the energy landscape. Alternative structures with non-negligible probability have a small base pair distance from one of the targets. As indicated by the partition function dotplot, the targets are very well represented in the structural ensemble. Base pairs that are not part of one of the three design goals are very rare. There are, however, some "mixed" structures that facilitate the transition between the three local optima. We use a barrier tree to visualize the landscape of the designed sequence. Simulated folding kinetics, starting from the open chain, shows that the three target structures are, again, the three most prominent structures.

A Large-Scale Set of Multi-Stable Targets

The example above shows the effectiveness of our algorithm in designing sequences for tri-stable targets. To demonstrate that

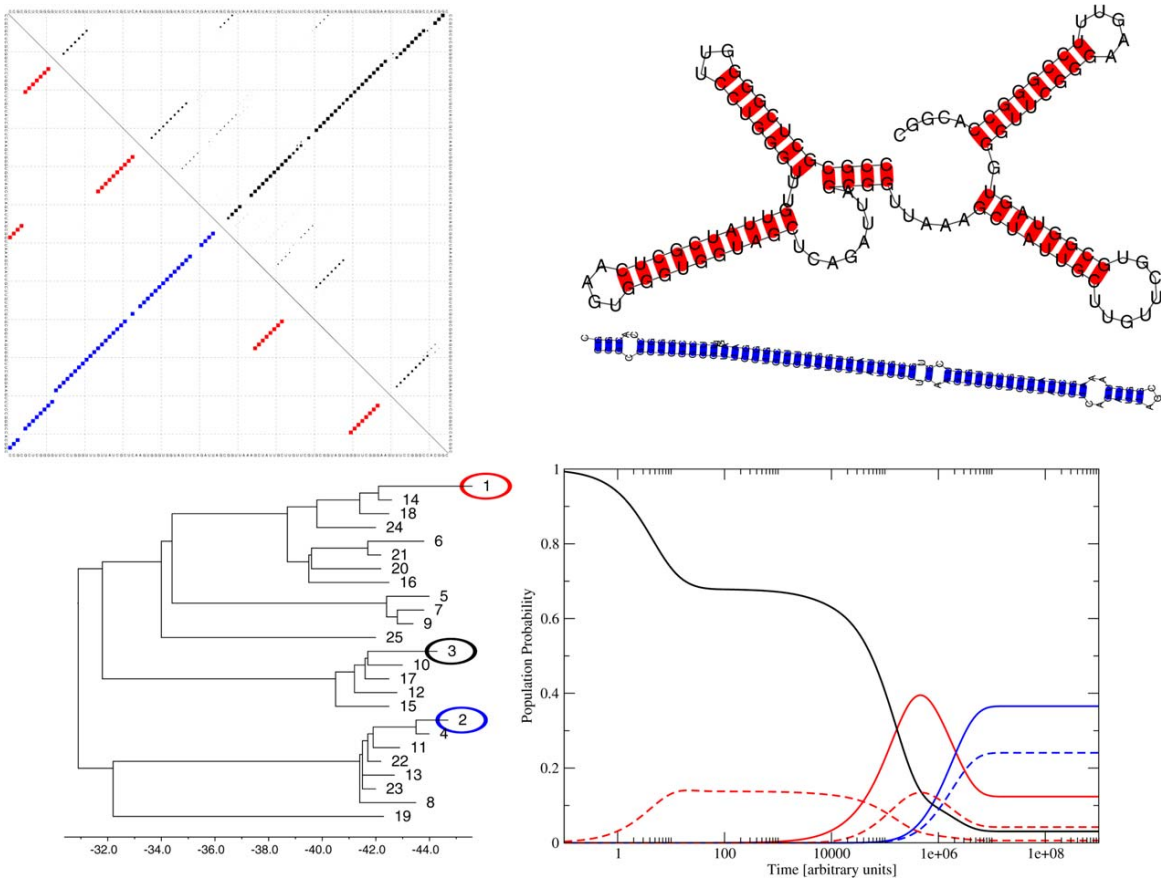


FIGURE 6 A designed sequence for the SV11 riboswitch. Dotplot, The upper triangular matrix plots the probability for each individual nucleotide to be paired. The lower triangular plot shows the two structural constraints. Structures, In red, the lower-energy structure that forms a Y-shaped multi-branched loop and two additional exterior loops, and in blue the single long helix structure. The red structure is almost equivalent to the minimum-free energy structure, which forms a single additional AU base pair, resulting in a three-nucleotide hairpin instead of a five-nucleotide hairpin with an additional gain of 200 kcal/mol. Accordingly, the second structure in the suboptimal ensemble is the red structure, followed by the blue structure at the third position in the ensemble with a difference of 0.6 kcal/mol. Barrier tree (left bottom) and folding kinetics (right bottom), The red and blue curves correspond to the target structures and are dominant in the kinetics. The dashed lines are structures that are very similar (base pair distance of five or less) to the target structures. As the energy distance to the open chain is too large to be included in the barrier and kinetics calculations, we started from a structure (colored black) that is somewhat related to the red target.

our method scales well to larger design problems, we generated an ensemble of 100 design problems, each produced from a random sequence of length 100 as follows: We first used RNAshapes[58] to extract the three most stable coarse-grained structures and their most stable fine-grained representatives ("shreps"). The SV11 sequence, for example, has (at least) two shapes: the low-energy rod-like structures with shape [ ] and the high-energy complex structure [[ ][ ]][ ][ ] with a Y-shaped multi-branched loop and two additional external stems. This way we ensure that the design problem is feasible. We solve the design problems using the fitness function Eq. (10), perform a single optimization run for each problem, and retain a single top-scoring sequence, as output.

In order to evaluate the quality of the designed sequences we investigated their energy landscapes in more detail. Using RNAsubopt[48] we produced all suboptimal structures within 5 kcal/mol of the ground state, and determined the suboptimal structure Θ_i' within this energy band that is closest to the design goal Θ_i. Ideally, the base-pairing distance $d_{bp}(\Theta_i\Theta_i')$ should vanish and $f(\Theta_i)$ should be very close to the ground

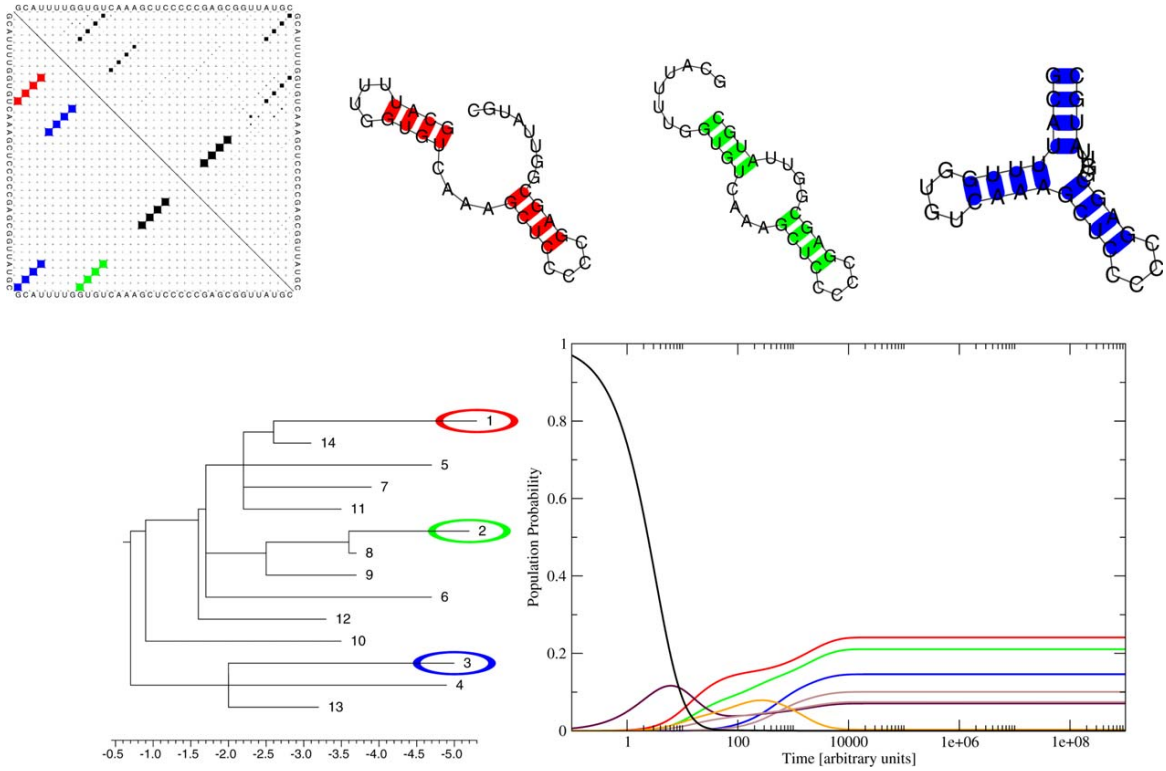


FIGURE 7 Example of a tri-stable switch molecule. (Dotplot) All structures have great statistical weight within the thermodynamic equilibrium. Base pairs belonging to one structure are colored red, green, or blue, respectively. All structures share four base pairs, which are colored black. (Structures) The three structures correspond to the three lowest energy structures of the RNA. The red (top-left) structure is the mfe structure, the green and blue structures have free energies 0.1 and 0.3 kcal/mol above the ground state, respectively. (Barrier tree (left, bottom) and folding kinetics (right, bottom)), The desired structures of the tri-stable switch form the most prominent local minima in the folding landscape. Kinetic curves in brownish colors correspond to mixed conformations where compatible structural features of different local minima structures have been blended into a single structure.

state for all three design targets. In 96% of the design problems, all three target structures were indeed contained within the 5 kcal/mol energy band, and even in the remaining few cases, a very similar structure was observed within this range. Figure 8 shows the energy differences for the 100 designs as a scatter plot. In most of the designs, one of the three targets is the ground state. The mean and median energy differences between the ground state and the worst of the three target structures are only 1.0 and 1.5 kcal/mol, respectively. Overall, these data show that our approach produces close-to-optimal tri-stable designs reliably and efficiently.

To assess the increase in difficulty of designing sequences for more than three target structures simultaneously, we repeated the large-scale experiment, but this time using four target structures instead of three. The results shown in the lower panel of Figure 8 lead us to believe that our approach does indeed scale to very complex multi-stable targets. As the quality of the generated sequences also degenerated slightly with median differences from 1.6 to 2.3 kcal/mol, further investigation into complex multi-structure targets will be required. Nevertheless, these differences only amount to roughly 1 to 3 stacked base pairs. Again, we automatically selected the top-scoring sequence for each target instead of trying the, say, best five sequences for each structure.

DISCUSSION AND CONCLUSIONS

We have shown that multi-stable RNA sequences with prescribed alternative secondary structures can be constructed efficiently by means of a generic computational approach. With RNAdesign we provide an efficient implementation that combines an exact solution of a graph coloring problem with the

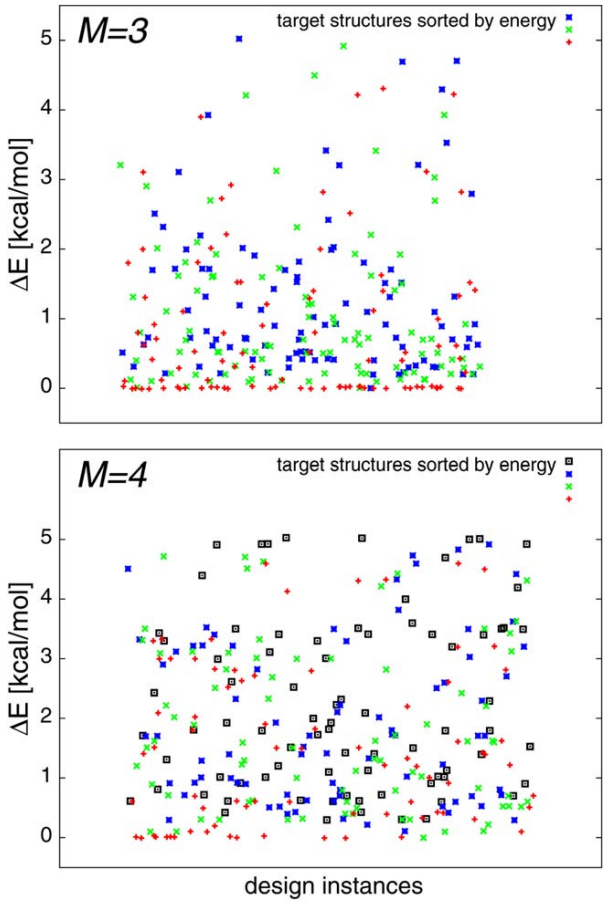


FIGURE 8 Performance of the sequence design algorithm on 100 randomly generated instances. Top, Tri-stable targets. Of the 300 structures, the 288 that appear within the 5 kcal/mol window are shown. The remaining 12 structures have $1 \leq d_{bp} \leq 14$. The lowest-energy structure (red) is typically very close (mean 0.93, median 0.4 kcal/mol difference) to the mfe structure, the second (mean 1.17, median 0.7 kcal/mol; green) and the third (mean 1.41, median 0.9 kcal/mol; blue) structure are more distant. (A small amount of jitter was added to better separate data points). Below, Four target structures. Only 24 of the 400 target structures lie outside the 5 kcal/mol window. The energy differences to the ground state for the energy-sorted targets are red (mean 1.85, median 1.6 kcal/mol), green (mean 2.17, median 1.9 kcal/mol), blue (mean 2.40, median 2.05 kcal/mol), and black (mean 2.62, median 2.3 kcal/mol).

heuristic optimization of feasible solutions by local search. When more than two target structures are prescribed, a combinatorial consistency condition must be satisfied. For triples of targets and moderate sequence length, the design problem frequently has feasible solutions, although the probability decreases exponentially with chain length. Randomly generated sets of four or more target structures, however, typically cannot be realized by the same sequence. Since very few multistable RNAs have been described in the literature, we resorted to artificial test cases to verify that our approach solves the computational problem at hand.

RNAdesign can accommodate a wide range of design goals. Although our test cases focus on nearly equal enrichment of target structures in the Boltzmann ensemble, more complex features of the fitness landscapes can easily be incorporated. As discussed by Flamm et al.[27] for the case of bi-stable structures, it is feasible (albeit computationally demanding) to estimate for a candidate sequence x the energy barrier $f^{\neq}(x, \Theta_i, \Theta_j)$ between target structures Θ_i and Θ_j. For moderate sequence lengths, this can be computed exactly by using RNAsubopt and barriers, and for longer sequences a path-based heuristic provides at least an upper bound.[27,40] On this basis, it is even possible to estimate kinetic parameters such as first passage times to target structures.[36] It will be easy to extend the RNAdesign so that kinetic parameters of this type can be included into the design fitness function χ_i.

The current version of RNAdesign already supports inclusion of prescribed energy differences between the target conformations. This is desirable, for instance, for the rational design of riboswitches that are triggered by ligand binding. In this case, the fitness landscape is distorted by the binding energy of the ligand in certain structures. This causes a re-folding of the molecule in which conformational changes in the ligand binding domain are used to change adjacent structural domains. Our recent construction of a transcriptional riboswitch based on the theophylline aptamer domain[12] shows that the RNA energy model is sufficiently accurate to capture such effects.

We can, therefore, argue that the relative ease with which multistable structures can be designed reflects the evolutionary accessibility of such molecules. Our data suggest, in particular, that RNA sequences with three or four disparate local optima with energies close to the ground state are abundant and can readily be optimized by a local search in sequence space. A similar observation has been made by Ramlan and Zauner.[31] If such structures provide a selective advantage, evolution should therefore be able to evolve them de novo in different contexts. This immediately raises the question of whether multi-stable RNAs have arisen in the history of life and how abundant they are in nature.

For the case of two alternative structures the answer is, of course, affirmative, as demonstrated by a diverse set of riboswitches for a wide variety of ligands[1] and several classes of RNA thermometers.[2] Self-induced conformational switches[59] act as a kind of timing device. Here, the molecule is trapped in a metastable structure that either allows or blocks the RNA's function. Decay to the ground state then flips the switch. Molecules undergoing such conformational changes have also been

observed as the outcomes of artificial selection experiments, for instance, selecting for suitability as a template for Q_β replicase.[5,60]

For more than two structural alternatives, the situation is less obvious. No self-induced or small metabolite-triggered RNA switch with three or more structural alternatives has so far been characterized. Complex conformational changes, however, play a role in splicing and the action of ribozymes, including self-splicing introns and other allosteric nucleic acid catalysts.[61,62] A well-understood system that comes at least close to a self-induced tri-stable RNA switch is the *Hok/Sok* system of plasmid R1 in *Escherichia coli*.[7,63] Here, the binding of one effector causes a change in the structure of the ribozyme molecule, which in turn allows the binding of a second effector necessary for the final activation of the enzymatic function. An artificial catalytic system consisting of two RNAs that catalyze their ligation with the help of a transient hammerhead ribozyme structure relies on several coordinated structural rearrangements.[64]

The possibility that multi-stable conformational switches are a common element beyond simple ON/OFF switches in RNA-based regulation leads to the question of whether RNA-based circuits provide a compact—and hard to disentangle—implementation of complex regulatory programs. Beyond such intriguing perspectives on RNA biology, we encountered also several nontrivial computational problems that provide interesting avenues for future research on rational RNA design.

Instead of a "fair" starting sequence sampled uniformly from $\mathcal{C}$, one might want to "stack the deck" as much as possible in favour of a successful design. This invites the question of whether there are any efficient dynamic programming algorithms that compute the sequence that minimizes the sum of free energies on a prescribed set of structures. A promising way to address this question is a generalization of the intaRNA approach of Busch and Backofen[20] to multiple structures. Another obvious challenge is to improve the coloring step using an explicit construction for ear decompositions that guarantee small values of α and β.

Since the design goals for more than two sequences are not feasible in general, one may be interested in a slight relaxation of the structure in $\{\Theta_i\}$, i.e., in a set $\{\Theta'_i\}$ that is as close as possible to the original and for which the design is feasible. A natural objective function for this task is, for instance, $\sum_i d_G(\Theta'_i, \Theta_i)$ for some graph edit distance $d_G(.,.)$. A simpler, but maybe less natural, approach is to directly edit the dependency graph G, i.e., by removing a minimal number of edges.

An alternative approach to relaxing the structural constraints is to allow a small number of noncanonical base pairs. The CONTRAfold algorithm by Do et al.[65] considers all 16 possible base pairings instead of just the canonical six. Another solution is to use the space of extended secondary structures,[66] which also considers all 16 possible base pairs and, in addition, explicitly annotates nucleotide pairings with the nucleotide edge engaged in pairing. As both of these models have basically no constraints, the space of candidate sequences is unrestricted. However, since canonical base pairs are more likely than non-canonical base pairs, it makes sense to always constrain the search space to those sequences for which canonical pairings predominate. Formally, this equates to allowing some—but not too many—color conflicts.

Finally, it is desirable to impose more sophisticated conditions on sequence composition. We currently allow penalizing candidate sequences according to a mononucleotide model. It seems feasible to explore di-nucleotide distributions instead of the current mononucleotide model. Such models have already been used in a gene prediction context,[67] and their impact on sequence design will be interesting to explore.

The RNAdesign tool opens the door to the largely unexplored realm of tri-stable and even higher-level multistable structures, which is of utmost interest for synthetic biology. With small modifications of the energy model our approach can easily be extended to interacting multistable RNA molecules, a topic that is of particular interest for the design of small trans-acting and multistable self-assembling RNAs.

Availability

The RNAdesign software can downloaded from http://www.bioinf.uni-leipzig.de/Software/RNAdesign/.

REFERENCES

1. Serganov, A.; Nudler, E. Cell 2013, 152, 17–24.
2. Kortmann, J.; Narberhaus, F. Nat Rev Microbiol 2012, 10, 255–265.
3. Ray, P. S.; Jia, J.; Yao, P.; Majumder, M.; Hatzoglou, M.; Fox, P. L. Nature 2009, 457, 915–919.
4. Baumstark, T.; Schroder, A. R.; Riesner, D. EMBO J 1997, 16, 599–610.
5. Biebricher, C. K.; Diekmann, S.; Luce, R. EMBO J 1992, 11, 51129–5135.
6. Cao, Y.; Woodson, S. RNA 2000, 6, 1248–1256.
7. Gultyaev, A.; Franch, T.; Gerdes, K. J Mol Biol 1997, 273, 26–37.
8. Wieland, M.; Benz, A.; Klauser, B.; Hartig, J. S. Angew Chem Int Ed Engl 2009, 48, 2715–2718.
9. Win, M. N.; Liang, J. C.; Smolke, C. D. Chem Biol 2009, 16, 298–310.
10. Topp, S.; Gallivan, J. P. ACS Chem Biol 2010, 5, 139–148.
11. Isaacs, F. J.; Dwyer, D.; Collins, J. J. Nat Biotechnol 2006, 24, 545–554.
12. Wachsmuth, M.; Findeiß, S.; Weissheimer, N.; Stadler, P. F.; Mörl, M. Nucleic Acids Res 2013, 41, 2541–2551.
13. Ogawa, A. RNA 2011, 17, 478–488.

14. Schnall-Levin, M.; Chindelevitch, L.; Berger, B. In Proceedings of the 25th International Conference on Machine learning (ICML '08); ACM: New York, NY. 2008; pp 904–911.
15. Hofacker, I. L.; Fontana, W.; Stadler, P. F.; Bonhoeffer, L. S.; Tacker, M.; Schuster, P. Monatsh Chem 1994, 125, 167–188.
16. Andronescu, M.; Fejes, A. P.; Hutter, F.; Hoos, H. H.; Condon, A. J Mol Biol 2004, 336, 607–624.
17. Aguirre-Hernández, R.; Hoos, H. H.; Condon, A. BMC Bioinformatics 2007, 8, 34.
18. Zadeh, J. N.; Wolfe, B. R.; Pierce, N. A. J Comput Chem 2011, 32, 439–452.
19. Zadeh, J. N.; Steenberg, C. D.; Bois, J. S.; Wolfe, B. R.; Pierce, M. B.; Khan, A. R.; Dirks, R. M.; Pierce, N. A. J Comput Chem 2011, 32, 170–173.
20. Busch, A.; Backofen, R. Bioinformatics 2006, 22, 1823–1831.
21. Taneda, A. Adv Appl Bioinform Chem 2011, 4, 1–12.
22. Burghardt, B.; Hartmann, A. K. Phys Rev E Stat Nonlin Soft Matter Phys 2007, 75, 021920.
23. Gao, J. Z.; Li, L. Y.; Reidys, C. M. Alg Mol Biol 2010, 5, 27.
24. Avihoo, A.; Churkin, A.; Barash, D. BMC Bioinformatics 2011, 12, 319.
25. Levin, A.; Lis, M.; Ponty, Y.; O'Donnell, C. W.; Devadas, S.; Berger, B.; Waldispühl, J. Nucleic Acids Res 2012, 40, 10041–10052.
26. Shu, W.; Liu, M.; Chen, H.; Bo, X.; Wang, S. J Biotechnol 2010, 150, 466–473.
27. Flamm, C.; Hofacker, I. L.; Maurer-Stroh, S.; Stadler, P. F.; Zehl, M. RNA 2001 7, 254–265.
28. Reidys, C.; Stadler, P. F.; Schuster, P. Bull Math Biol 1997, 59, 339–397.
29. Abfalter, I.; Flamm, C.; Stadler, P. F. In Proceedings of the German Conference on Bioinformatics (GCB), Vol. 1; Mewes, H. W.; Heun, V.; Frishman, D.; Kramer, S., Eds.; belleville Verlag Michael Farin: München, 2003; pp 1–7.
30. Lyngsø, R. B.; Anderson, J. W.; Sizikova, E.; Badugu, A.; Hyland, T.; Hein, J. BMC Bioinformatics 2012, 13, 260.
31. Ramlan, E. I.; Zauner, K. P. Biosystems 2011, 105, 14–24.
32. Mathews, D. H.; Disney, M. D.; Childs, J. L.; Schroeder, S. J.; Zuker, M.; Turner, D. H. Proc Natl Acad Sci USA 2004, 101, 7287–7292.
33. Ding, Y.; Chan, C. Y.; Lawrence, C. E. RNA 2005, 11, 1157–1166.
34. Quarta, G.; Kim, N.; Izzo, J. A.; Schlick, T. J Mol Biol 2009, 393, 993–1003.
35. Lorenz, R.; Flamm, C.; Hofacker, I. L. In German Conference on Bioinformatics. Vol. 157: Lecture Notes in Informatics; GI: Koellen Verlag, Bonn, 2009; pp 11–20.
36. Wolfinger, M. T.; Svrcek-Seiler, W. A.; Flamm, C.; Hofacker, I. L.; Stadler, P. F. J Phys A Math Gen 2004, 37, 4731–4741.
37. Wuchty, S.; Fontana, W.; Hofacker, I. L.; Schuster, P. Biopolymers 1999, 49, 145–165.
38. Waldispühl, J.; Clote, P. J Comput Biol 2007, 14, 190–215.
39. Li, Y.; Zhang, S. Bioinformatics 2011, 27, 2994–3001.
40. Morgan, S. R.; Higgs, P. G. J Phys A 1998, 31, 3153–3170.
41. Jensen, T. R.; Toft, B. Graph Coloring Problems; John Wiley & Sons: New York, 1994.
42. Whitney, H. Trans Am Math Soc 1932, 34, 339–362.
43. Maon, Y.; Schieber, B.; Vishkin, U. Theor Comp Sci 1986, 47, 277–298.
44. Tacker, M.; Stadler, P. F.; Bornberg-Bauer, E. G.; Hofacker, I. L.; Schuster, P. Eur Biophy J 1996, 25, 115–130.
45. Ding, Y.; Lawrence, C. E. Nucleic Acids Res 2001, 29, 1034–1046.
46. Schuster, P.; Fontana, W.; Stadler, P. F.; Hofacker, I. L. Proc Roy Soc Lond B 1994, 255, 279–284.
47. Fontana, W.; Stadler, P. F.; Bornberg-Bauer, E. G.; Griesmacher, T.; Hofacker, I. L.; Tacker, M.; Tarazona, P.; Weinberger, E. D.; Schuster, P. Phys Rev E 1993, 47, 2083–2099.
48. Lorenz, R.; Bernhart, S. H.; Höner zu Siederdissen, C.; Tafer, H.; Flamm, C.; Stadler, P. F.; Hofacker, I. L. ViennaRNA Package 2.0. Algorithms for Molecular Biology 6; 2011.
49. The GHC Team. The Glasgow Haskell Compiler (GHC); 1989. Available at: http://www.haskell.org/ghc/. Last accessed on April 21, 2013.
50. Hudak, P.; Hughes, J.; Peyton Jones, S.; Wadler, P. In Proceedings of the Third ACM SIGPLAN Conference on History of Programming Languages, HOPL III; ACM: New York NY, 2007; pp 1–55.
51. Okasaki, C. 1999. Cambridge University Press: Cambridge, UK.
52. Bird, R. Pearls of Functional Algorithm Design; Cambridge University Press: Cambridge, 2010.
53. Erwig, M. ACM SIGPLAN Notices 1997, 32, 52–65.
54. Hinze, R.; Harper, T.; James, D. W. Theory and Practice of Fusion. Implementation and Application of Functional Languages; Springer: Berlin, Heidelberg, 2011; pp 19–37.
55. Coutts, D.; Leshchinskiy, R.; Stewart, D. In Proceedings of the 12th ACM SIGPLAN International Conference on Functional Programming (ICFP'07); ACM: New York, NY, 2007; pp 315–326.
56. Leshchinskiy, R. Recycle Your Arrays! Practical Aspects of Declarative Languages; Springer: Berlin, Heidelberg, 2009; pp 209–223.
57. Höner zu Siederdissen, C. In Proceedings of the 17th ACM SIGPLAN International Conference on Functional programming (ICFP '12); ACM: New York, NY, 2012; pp 215–226. Available at: http://doi.acm.org/10.1145/2364527.2364559.
58. Reeder, J.; Giegerich, R. Bioinformatics 2005, 21, 3516–3523.
59. Nagel, J. H.; Pleij, C. W. Biochimie 2002, 84, 913–923.
60. Biebricher, C. K.; Luce, R. J Mol Biol 1982, 154, 629–648.
61. Jose, A.; Soukup, G.; Breaker, R. Nucleic Acids Res 2001, 29, 1631–1637.
62. Soukup, G.; Breaker, R. Curr Opin Struct Biol 2000, 10, 318–325.
63. Møller-Jensen, J.; Franch, T.; Gerdes, K. J Biol Chem 2001, 276, 35707–35713.
64. Gwiazda, S.; Salomon, K.; Appel, B.; Müller, S. Biochimie 2012, 94, 1457–1463.
65. Do, C. B.; Woods, D. A.; Batzoglou, S. Bioinformatics 2006, 22, e90.
66. Höner zu Siederdissen, C.; Bernhart, S. H.; Stadler, P. F.; Hofacker, I. L. Bioinformatics 2011, 27, 129–136.
67. Gesell, T.; Washietl, S. BMC Bioinformatics 2008, 9, 248.
68. Flamm, C.; Fontana, W.; Hofacker, I.; Schuster, P. RNA 2000, 6, 325–338.

Reviewing Editor: Sarah A. Woodson

7

RNABLUEPRINT SEQUENCE SAMPLING LIBRARY

Stefan Hammer, Birgit Tschiatschek, Christoph Flamm, Ivo L. Hofacker, and Sven Findeiß.
"RNAblueprint: Flexible Multiple Target Nucleic Acid Sequence Design.", 2017 in Bioinformatics 33 (18): 2850–58.
doi:10.1093/bioinformatics/btx263

SUMMARY

In this publication the `RNAdesign` algorithm was further developed and its dynamic programming approach reformulated in a more generic manner. The software is implemented as a sequence sampling library to be able to formulate any objective function and optimization method in a scripting language. Thus, this contribution is a first step towards making computational design tools more flexible and adaptable to the varying needs in RiboNucleic Acid (RNA) design.

Moreover, a new objective function corrects the weighting of previous published versions and the library was used to perform statistical analyses of the uniform sampling method, the suggested neighborhood relations in the solution space, and the runtime of the optimization method. It also includes extensive benchmarks against other multi-stable design software.

AUTHORS CONTRIBUTION

SH, SF, CF reformulated the `RNAdesign` algorithm. SH implemented the software. SH and SF wrote the majority of the article. SH, BT, SF developed python design scripts and performed statistical analyses. CF and IL contributed with essential scientific advise.

LICENSE

Bioinformatics, 2017, 1–9
doi: 10.1093/bioinformatics/btx263
Advance Access Publication Date: 25 April 2017
Original Paper

OXFORD

Structural bioinformatics

RNAblueprint: flexible multiple target nucleic acid sequence design

Stefan Hammer[1,2,*], Birgit Tschiatschek[2], Christoph Flamm[1,3], Ivo L. Hofacker[1,2,4,*] and Sven Findeiß[1,2,*]

[1]Faculty of Chemistry, Department of Theoretical Chemistry, [2]Faculty of Computer Science, Research Group Bioinformatics and Computational Biology, [3]Research Network Chemistry Meets Microbiology, University of Vienna, 1090 Vienna, Austria and [4]Center for Non-Coding RNA in Technology and Health, University of Copenhagen, Copenhagen DK-1870, Denmark

*To whom correspondence should be addressed.

Associate Editor: Cenk Sahinalp

Received on September 21, 2016; revised on February 24, 2017; editorial decision on April 7, 2017; accepted on April 21, 2017

Abstract

Motivation: Realizing the value of synthetic biology in biotechnology and medicine requires the design of molecules with specialized functions. Due to its close structure to function relationship, and the availability of good structure prediction methods and energy models, RNA is perfectly suited to be synthetically engineered with predefined properties. However, currently available RNA design tools cannot be easily adapted to accommodate new design specifications. Furthermore, complicated sampling and optimization methods are often developed to suit a specific RNA design goal, adding to their inflexibility.

Results: We developed a `C++` library implementing a graph coloring approach to stochastically sample sequences compatible with structural and sequence constraints from the typically very large solution space. The approach allows to specify and explore the solution space in a well defined way. Our library also guarantees uniform sampling, which makes optimization runs performant by not only avoiding re-evaluation of already found solutions, but also by raising the probability of finding better solutions for long optimization runs. We show that our software can be combined with any other software package to allow diverse RNA design applications. Scripting interfaces allow the easy adaption of existing code to accommodate new scenarios, making the whole design process very flexible. We implemented example design approaches written in `Python` to demonstrate these advantages.

Availability and implementation: `RNAblueprint`, `Python` implementations and benchmark datasets are available at github: https://github.com/ViennaRNA.

Contact: s.hammer@univie.ac.at, ivo@tbi.univie.ac.at or sven@tbi.univie.ac.at

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

RNA molecules are omnipresent in all domains of life. They execute diverse functions including small molecule sensing, signal transduction and gene regulation. RNA is a molecule well-suited for designing with predefined functionality. This is mainly due to its close structure to function relationship and the physio-chemically grounded energy models for straightforward *in silico* calculations at the level of secondary structure. In recent years, due to the advent of synthetic biology, more researchers are focusing on the design of synthetic RNAs. There has been increasing success in modifying existing systems and incorporating novel functionality in RNAs within a cellular context (Chappell *et al.*, 2015; Espah-Borujeni *et al.*, 2015; Green *et al.*, 2014; Rodrigo *et al.*, 2012)

To produce an RNA molecule with a prescribed function, the close structure to function relationship must be incorporated into the design process, along with a rationally defined specification of the structure performing that function. In the simplest form one could generate all 4^n possible nucleic acid sequences of length n and test for each sequence if it fulfills the design criteria, e.g. its most stable fold is the structure of interest. Only a small subset of all possible sequences will be actually able to fold at all into the target structure and it is therefore favorable to generate those candidate sequences at least comply with the structural constraints, i.e. are able to fold into the defined structure. Generating only those sequences able and likely to fold into the target structure is known as the 'inverse folding problem' (Hofacker *et al.*, 1994) where the applied structural constraints reduce the size of the solution space to be investigated. Biologically active RNA molecules such as aptamers or ribozymes frequently require specific nucleotide patterns in binding or catalytic domains. Therefore, the designed RNA must also comply with certain sequence constraints. Several computational tools capable of solving this hard combinatorial optimization problem have been published. These tools differ mainly in how the initial sequence is selected and which search strategy, e.g. stochastic local or global search, is applied (see Supplementary Table S1). Both algorithmic characteristics have a big impact on the success of the optimization.

A variety of RNA molecules, natural as well as artificial, have been described that exploit structural change as their functional mechanism. Usually, the structural switching of these RNAs between an inactive and the active conformation is induced by an external trigger, which can be as diverse as temperature, small organic molecules, or other small RNAs (Berens and Suess, 2015). The design of such RNA devices requires finding a sequence compatible with two or more structural constraints. Designing a bi-stable RNA was first solved by Flamm *et al.* (2001) using a graph coloring approach. Recent tools can now also design multi-state (three or more) RNA molecules (Höner zu Siederdissen *et al.*, 2013; Lyngso *et al.*, 2012; Taneda, 2015; Wolfe and Pierce, 2015; Zadeh *et al.*, 2011a). The most recent version of the `RNAiFold` server seems to accept more than two target structures, this extension is however not yet described in the latest publication (Garcia-Martin *et al.*, 2015). Algorithms able to handle multi-state as well as multi-sequence folding and pseudoknotted structures are required if multiple RNA molecules are used as triggers. The latter are implemented in the `NUPACK` design and analysis framework (Zadeh *et al.*, 2011b).

Sampling sequences compatible with multiple structural constraints can be achieved using a complex graph coloring algorithm (Abfalter *et al.*, 2003; Höner zu Siederdissen *et al.*, 2013). It solves this problem in a defined way where each solution is drawn statistically fairly with equal probability. In contrast, other sampling approaches use ad hoc sampling heuristics that introduce biases and often exhibit undefined runtime complexities (Lyngso *et al.*, 2012; Taneda, 2015). Thus, good solutions may be missed because the sampled part of the solution space is not clearly specified and therefore cannot be fully explored. Furthermore, frequent re-evaluation of already discovered solutions due to biased sampling leads to inefficient optimization, especially if the calculation of the objective involves demanding computations such as pseudoknot structure prediction.

A review of the literature revealed that published RNA designs were either achieved by manual ad hoc approaches or very specific software implementations, which can handle only restricted design problems on a case-by-case basis (Isaacs *et al.*, 2004; Neupert *et al.*, 2008; Qi *et al.*, 2012; Rodrigo and Jaramillo, 2014; Wachsmuth *et al.*, 2013). Very recent publications focus on the flexibility of the design approach and provide methods and interfaces to allow the specification of broader objectives (Höner zu Siederdissen *et al.*, 2013; Taneda, 2015). However, the diversity of the objectives is still limited and introducing a new feature in the objective function requires changes in the program code (some of which are closed source). Furthermore, the mechanisms of optimization in existing tools are always predefined and very rigid.

To address these limitations, we developed `RNAblueprint` which solves the problem of sampling RNA sequences compatible with multiple structural and sequence constraints in a well defined way. The library is able to specify the runtime complexity and memory requirements of the problem for any given constraints, calculate the number of possible solutions, and to stochastically sample uniformly from all solutions. Furthermore, our technique can be easily integrated into existing tools, henceforth making it possible to focus on the formulation of the objective function as the most crucial part of the design process. Until now this aspect was largely neglected, even though the objective describes best how the design should function. The actual optimization process is performed using the scripting interface, where we offer predefined solutions but also give the user the opportunity to easily implement new ideas without having to change the source code of the core library. Next to the well defined way of describing and exploring the solution space, this flexibility is a major advantage of our approach.

With our framework, in addition to predicting RNA structure and RNA–RNA interactions, and allowing for pseudoknot incorporation (Janssen and Giegerich, 2015; Lorenz *et al.*, 2011; Zadeh *et al.*, 2011a,b) recent methods for the calculation of RNA-ligand interactions can also be incorporated (Lorenz *et al.*, 2016). Using `RNAblueprint` and its scripting interface we here implemented a multi-state design, which we used to analyze and benchmark our software. The developed software allows us to effectively solve problems including the design of translational and transcriptional on/off switches, triggered by diverse inputs such as small RNAs, ligands, temperature, salt concentration or proteins. `RNAblueprint` can also be used to specify the design of RNA or DNA scaffolds in synthetic biology, and to construct RNA/DNA origami.

2 Approach

An RNA sequence $x = \{x_1, x_2, x_3, \ldots, x_n\}$ is constructed from a set of monomers $x_i \in \mathcal{A} = \{A, U, G, C\}$ that can interact by forming base pairs (i, j), $1 \leq i < j \leq n$ where i and j are positions separated by at least three bases and $(x_i, x_j) \in \mathcal{B} = \{AU, UA, GC, CG, GU, UG\}$ the set of allowed base pairs with $\mathcal{B} \subset \mathcal{A} \times \mathcal{A}$. A set of base pairs of a sequence x is called secondary structure Θ.

`RNAblueprint` implements a method to sample RNA sequences compatible with all structures of a given set $\{\Theta_1, \Theta_2, \ldots, \Theta_M\}$ and sequence constraints $\{\Upsilon_1, \Upsilon_2, \Upsilon_3, \ldots, \Upsilon_n\}$ where $\Upsilon_i \subseteq \mathcal{A}$ is the set of allowed nucleotides at position i. To be able to uniformly sample from the entire solution space $\mathcal{C}$ (which is the set of all x compatible with all $\Theta_t, 1 \leq t \leq M$, given all sequence constraints $\Upsilon_i, 0 \leq i \leq n$), we implemented the graph-theoretical coloring approach developed by Abfalter *et al.* (2003), which is depicted in Figure 1 and described in the following. The goal is to generate sequences that are compatible with a sequence constraint and a set of target structures. Such a design problem is represented as a *dependency graph* $G = (V, E)$ constructed as the union of the circle plot representations of the structural constraints (Supplementary Fig. S1). Each vertex $v_i \in V$ of the graph corresponds to a position $1 \leq i \leq n$ in the sequence to be designed, and the edges E represent base pairs (i, j) that are formed between two vertices. Each base pair

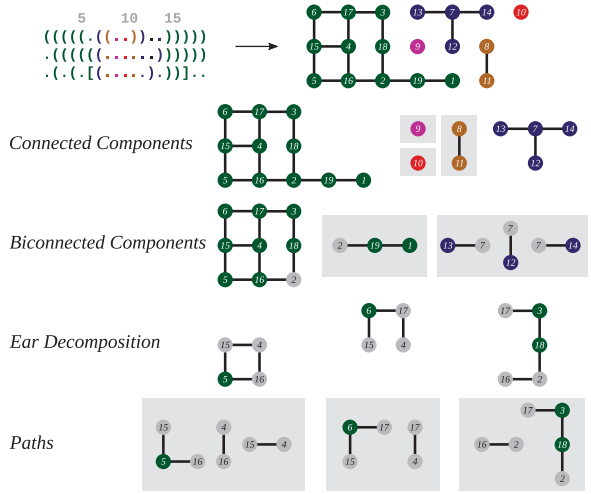


Fig. 1. A dependency graph is hierarchically decomposed starting from the top and moving down through four levels to generate a decomposition tree. The dot-bracket strings (top left) denote three structural input constraints which are converted into a dependency graph (top right) by intersecting their circle representations, see Supplementary Figure S1. For an easy visual mapping sequence positions are indicated above the dot-bracket string by an increment of five. Gray boxed subgraphs are not decomposed further as their number of possible colorings can be obtained with the path coloring approach. ● nodes represent articulation points

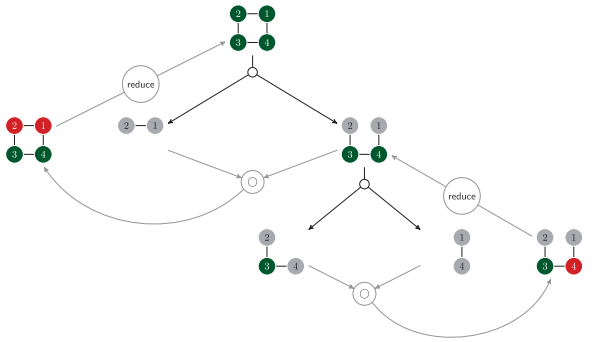


Fig. 2. Algorithmic implementation of the decomposition (black arrows) and the reassembly (gray arrows) of a biconnected component. ● nodes are ordinary nodes and ● nodes indicate articulation points. ● nodes are internalized articulation points which can be converted to ordinary nodes with the `reduce` function. During the dynamic programming forward recursion, the matrix concatenation operator calculates the number of possible colorings of the combined subgraphs given any assignments on $\mathcal{S}$

occurs at least in one of the input structures. According to the generalized intersection theorem, there exists a solution given the structural constraints, *iff* the resulting graph is bipartite (Flamm *et al.*, 2001). Finally, a coloring or base assignment on a vertex v_i is a single nucleotide $x_i \in \Upsilon_i$ assigned to the position i. Note, that poorly chosen sequence constraints might lead to an unsolvable design problem if they contradict the base pairing pattern enforced by the structural constraints. However, this can already be detected during the graph construction process.

For the design problem with two structural states, Flamm *et al.* (2001) showed that connected components and the underlying sequence positions of the corresponding dependency graph belong to one of the following classes: (i) isolated positions that are unpaired in both structures, (ii) positions that are paired with the same pairing partner in both structures and therefore form paths of length one and (iii) positions that are paired differently in both structures and therefore form paths or cycles. Connected components in (i) and (ii) can be assigned with any element of $\mathcal{A}$ and $\mathcal{B}$, respectively. Paths and cycles belonging to (iii) can be stochastically colored by a simple recursion. Furthermore, it was demonstrated how Fibonacci numbers can be used to determine the number of possible solutions for the latter.

Following these results, it is desirable to decompose a more complex dependency graph generated by more than two structural constraints into the aforementioned classes. This decomposition happens at vertices with degree greater than two, denoted as a set of *articulation points* $\mathcal{S}$. Depending on the decomposition algorithm, these nodes are also called cut points or attachment points. As the subsequently explained coloring approach can be very memory and CPU demanding, it is important to follow a specific order on how to decompose the dependency graph into paths. Connected components containing articulation points are decomposed into biconnected components and if they still contain articulation points, they are further decomposed using the ear decomposition algorithm, see Figure 1. An ear decomposition of a graph starting with a path P_0 is a decomposition of its edge set $E = P_0 \cup P_1 \cup \ldots \cup P_k$ where P_{i+1} is a simple path or ear whose endpoints belong to $P_0 \cup \ldots \cup P_i$, but its internal vertices do not (Maon *et al.*, 1986). Our step-wise decomposition approach ensures that the dependency graph falls apart into paths and cycles in a fixed order. As soon as the maximal degree of a subgraph H is two, either a path or a circle is reached and further decomposition is terminated. Using this decomposition approach, a binary tree of subgraphs is generated where the complete dependency graph sits at the root and each step of decomposition leads to a fixed order of subgraphs.

After the graph decomposition, the coloring problem therefore reduces to the determination of possible colorings of articulation points (● and ● vertices in Fig. 2) in the generated subgraphs H. This information can be efficiently calculated by a dynamic programming procedure (Abfalter *et al.*, 2003). Uniform sampling from $\mathcal{C}$ can then be achieved by stochastic backtracking. First articulation points are assigned, followed by the sampling of colors for adjacent paths. For ear decompositions this has been described in (Höner zu Siederdissen *et al.*, 2013). In this contribution we describe a generalized approach that covers the dynamic programming for all decomposed components of the dependency graph.

The dynamic programming forward recursion we implemented traverses the binary decomposition tree from the bottom up, ending at the complete assembled dependency graph G. For every subgraph H the possible colorings for the set of articulation points $\mathcal{S}_H$ and the according number of available solutions for H given these colors are stored in a memorization table during the dynamic programming procedure. The dimension of such a table is determined by $|\mathcal{S}_H|$. Since this number differs during the recursive traversal of the graphs in the decomposition tree (smaller graphs are connected at articulation points to larger units) the dimension of the memorization tables also varies. A table dimension itself is indexed by the elements of $\mathcal{A}$. For unbranched paths of length l the number of colorings can easily be looked up in the l-th power of the pairing matrix $\mathcal{P}$. The memorization table of any other subgraph H (parent node in the decomposition tree) is always calculated from the memorization tables of its two smaller constituting graphs (child nodes in the decomposition tree) in a type of concatenation procedure (Fig. 2). The corresponding entries of the articulation points (table dimensions) are first multiplied component-wise and then inserted into the new table. In our

implementation the memorization tables are sparse objects and the above construction procedure only increases dimensionality of the tables. The result would be a sparse memorization table with $|\mathcal{S}|$ dimensions at the root node of the decomposition tree. To avoid wasting of memory resources, we introduced a dimension reduction step during the successive construction of the memorization tables. This reduction step rests on the observation that whenever the vertex degree of an articulation point in a partially assembled graph is equal to the vertex degree of the corresponding node in the union graph (root node of the decomposition tree) no further subgraph will be 'attached' to this particular vertex in subsequent memorization table concatenation operations (see Fig. 2). Hence, the corresponding dimension of the memorization table is collapsed via summing up the values over that internalized articulation point, which shrinks the memorization table and removes the articulation point from the table. This implies that memorization tables for connected components have dimension zero since all articulation points have been internalized and removed via summation. In other words a memorization table with zero dimensions stores the total number of possible colorings for the respective subgraph. The memorization table for the root graph (i.e. the original union graph) therefore stores the size of the solution space, $|\mathcal{C}|$, which is equal to the total number of sequences compatible with the design constraints. With the help of the total number of sequences, the coloring count entries of the memorization tables can be re-interpreted as probabilities, paving the way for uniform sampling approaches.

The sampling procedure works exactly in the opposite order of the memorization table calculation. For each subgraph, articulation points are colored by stochastic backtracking from the probability matrix, which corresponds to the re-interpreted memorization table, followed by the sampling of the graph itself, if it is a path. Otherwise the next hierarchical level of subgraphs is processed. If an articulation point has a base assigned already, this information is used during the stochastic backtracking. Finally, when the last child has been processed, all bases are assigned and a solution was fairly drawn from the complete solution space.

Besides *global sampling*, i.e. generating a completely new sequence all the time, `RNAblueprint` offers two more procedures to mutate or resample parts of the sequence. *C-local sampling* resets the base assignments of all vertices of a random connected component and draws new colors, i.e. nucleotide assignments, for these vertices. *P-local sampling* randomly selects one path at the leaves of the decomposition tree and resamples only vertices which are not articulation points. This way we ensure the compatibility within a connected component. For both C-local and P-local sampling it can be useful to restrict the random selection of subgraphs by minimal and maximal size constraints or to directly select the connected component or path. The possibility to resample a specific position in the sequence also exists. This either involves a P-local sampling of the path containing the position or, in cases where the selected position corresponds to an articulation point, a C-local sampling of the corresponding connected component. In this way, the ranges of positions to be sampled can be specified. A history of previous sampled sequences is stored, making it convenient to revert to those previous sequences if necessary.

The complexity of our program strongly depends on the number of articulation points $|\mathcal{S}|$. The minimum time complexity $\mathcal{O}(n)$ is specified by running the graph decomposition algorithms or path colorings. For every subgraph H, the memory and CPU requirements of the dynamic programming coloring approach can be denoted as $\mathcal{O}(|\mathcal{A}|^{\alpha}), \alpha = |\mathcal{S}_H|$ and $\mathcal{O}(|\mathcal{A}|^{\beta}), \beta = |\cup_{b\in C(H)} \mathcal{S}_b|$, respectively. $C(H)$ represents the set of child subgraphs of H. The overall complexity is therefore defined as the sum over all H. The latter varies, as the ear decomposition is not done in a deterministic way. It derives from one of many possible spanning trees of the corresponding graph and it follows that α and β can vary dramatically as investigated in (Höner zu Siederdissen *et al.*, 2013). Therefore, we generate a set of random instances of spanning trees and select the one with lowest α and β values.

The implementation is written in `C++` using the boost graph library and other parts of the boost library available at http://www.boost.org/. Using the `SWIG` framework, we offer an easy to use `Perl` and `Python` scripting interface to the library. Additionally, we developed a `Python` module so that code can be reused for many central components.

3 Materials and methods

3.1 Objective function

The original objective function $f(x)$ proposed by Flamm *et al.* (2001) for two target designs was extended to the multi-target case (Höner zu Siederdissen *et al.*, 2013) and is

$$f(x) = \underbrace{\sum_{i}^{M}(E(x,\Theta_i) - G(x))}_{\text{dominate ensemble}} + \xi \underbrace{\sum_{i<j}^{M}(E(x,\Theta_i) - E(x,\Theta_j))^2}_{\text{minimize energy difference}} \quad (1)$$

where $G(x)$ is the ensemble free energy, $E(x,\Theta_i)$ is the free energy of the sequence x folded into structure Θ_i and ξ is a weighting factor typically set to 1. The first term is to maximize the energy contribution of each target structure to the ensemble to achieve dominance whereas the second term is to minimize the energy difference of target structures to get them to the same energy level. In Taneda (2015) the latter was changed to $\sum_{i<j}^{M}|E(x,\Theta_i) - E(x,\Theta_j)|$ which brings most of the target structure energies close to the minimum free energy (MFE) and outliers are possible. In contrast, the original version attempts to minimize the number of outliers and therefore the distance to the MFE of all states might be higher. Either way, weighting of the two terms is essential in single objective approaches. Although objective function (1) showed good performance on two-target designs, the straight-forward extension to three or more structures neglects the varying number of target structures. We therefore modified the objective function to

$$\begin{aligned} f(x) = &\frac{1}{M}\sum_{i}^{M}(E(x,\Theta_i) - G(x)) \\ &+ \xi\frac{2}{M(M-1)}\sum_{i<j}^{M}|E(x,\Theta_i) - E(x,\Theta_j)| \end{aligned} \quad (2)$$

as we sum up M elements in the first term and build $\binom{M}{2}$ differences in the latter. With this new objective function, the ratio between the two terms is independent of the number of structures M. To preserve the good performance for the two-target structure case and keep the 1:1 ratio between the two terms in the objective we set ξ to 0.5.

3.2 Benchmark datasets

The number of target structures is only a rough estimate of the complexity of a given design problem. If the given structural constraints have no conflicting base pairs, the complexity of the connected components are just single vertices or paths of length one. If more overlap between the structural constraints exists, paths get longer, and

complex subgraphs such as cycles and blocks occur. Based on a published tri-stable switch (Höner zu Siederdissen *et al.*, 2013), which contains only two cycles and eight paths of length one, we generated more complex examples by adding a fourth and fifth structural constraint, see Supplementary Figure S2A–C. These three example inputs of increasing complexity were used to evaluate the implemented sampling procedures of `RNAblueprint`. The effect of uniform sampling is tested on an extreme example that contains one large and complex connected component and a base pair as well as an unpaired position. To further reduce the solution space size, two sequence constraints were introduced, see Supplementary Figure S2D.

Comparison with existing approaches was performed on the published datasets containing two-, three- and four-target designs as well as pseudoknotted two-target structure examples (Taneda, 2015). The applied optimization is described in Section 3.3.

3.3 Multi-state design

To be able to benchmark against existing design software, we implemented an optimization approach consisting of `RNAblueprint` for uniform sequence sampling, the value of the objective function (2) to determine the cost of a solution, and an adaptive walk. The latter works as follows: Consecutive sequence candidates are generated by randomly applying one of the three sampling methods, i.e. P-local, C-local or global and calculating the cost. The new sequence is only kept if the cost is lower than the current best solution. Depending on the chosen method, one randomly selected subgraph (P-local and C-local sampling) or all subgraphs (global sampling) are redrawn. The stop condition was set to 1000, being the maximum number of optimization trails with no cost improvement. An optimization run would therefore be stopped earliest after 1000 sampling steps, which gave reasonable results for design problems with increasing complexity, see Figure 4. To compare this approach to existing multi-target design tools we created 100 solutions for each member of the two-, three- and four-target design datasets described in Taneda (2015). Energy calculations for these datasets were made using the scripting bindings of the `ViennaRNA package` v2.2.5 (Lorenz *et al.*, 2011). As we are not restricted to nested base pairs in the structural input, the pseudoknotted two-target datasets described in Taneda (2015) were also used with stop condition 100. This is set to be much smaller because the runtime dramatically increases when using the `Nupack package` v3.0.4 (Zadeh *et al.*, 2011b) for pseudoknotted structure prediction. Furthermore, only 30 solutions were generated for each of the latter benchmark tasks.

4 Results and discussion

4.1 Effect of uniform sampling

Implementing the complete graph coloring algorithm (Abfalter *et al.*, 2003; Höner zu Siederdissen *et al.*, 2013) and assigning all possible base pairs, `RNAblueprint` guarantees to uniformly sample the complete solution space. We show that this leads to an extreme value distributed frequency of uniquely found solutions (Fig. 3A). It follows that the solution space, by means of unique solutions generated, can be efficiently explored (Fig. 3B). The expected number of samplings required to explore $\mathcal{C}$ is $|\mathcal{C}|\log(|\mathcal{C}|)$ a fact that is known for the related Coupon Collector's problem (Michael Mitzenmacher, 2005). As the redundancy increases with the sample size d (known for the Birthday problem) and $|C|$ grows not more than exponentially with n, the average number of times sequences are generated when uniform sampling is bounded by

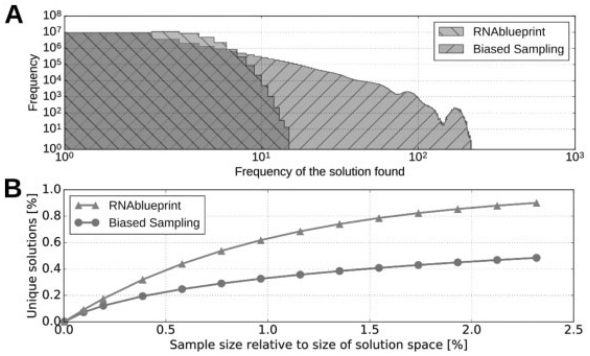


Fig. 3. Differences in uniform and biased stochastic sampling shown on a small example with a rather complex dependency graph, see Supplementary Figure S2D. (**A**) The histogram shows how frequent unique sequences were found when sampling completely new candidates using `RNAblueprint` and the biased sampling method. In total 9.6×10^9 sequences (sample size d) were sampled from 4.1×10^7 possible unique sequences (size of solution space $|C|$). While uniform sampling led to an extreme value distribution with the mean (2.57) count being slightly above the relative sample size and the maximum number of times a sequence is rediscovered being 15, biased sampling led to a input specific distorted distribution where a sequence is found 4.78 times on average and 227 times maximal. (**B**) When the sample size was chosen to be much bigger than the solution space ($\sim$ 230%), only about 50% of all possible sequences with biased sampling were obtained for this example, while the uniform sampling method generated about 90%. The performance of `RNAblueprint` is independent of the underlying problem whereas the curve of the other approaches heavily depends on the properties of the dependency graph

$\log(|C|) \in \Theta(n)$. Thus sequences will typically be sampled at most a linear number of times. The advantage of uniform sampling is most apparent when the amount of generated sequences d is large, $d \gtrsim |C|$. In this case, uniform sampling generates a much larger number of unique solutions. To highlight these properties, we implemented a biased sampling method by using the same algorithm as `RNAblueprint`, but making every backtracking decision uniformly. Thus, we sample all articulation point combinations with the same probability independent of the number of possible solutions of the attached subgraphs. We show that the biased sampling approach produces sequences with varying probabilities heavily dependent on the structure of the dependency graph. Therefore, while generating the same amount of sequences, only a fraction of unique sequences were found compared to `RNAblueprint` (Fig. 3B). Note, that for very small d the curves are almost identical, as expected. However, utilizing an approach that produces many different solutions, avoids heavy re-evaluation of already found sequences.

A simplified version of the graph coloring algorithm was implemented in `MODENA` (Taneda, 2015). Therein a naïve nucleotide assignment algorithm is used that is able to generate solutions of a design problem but not uniform sampling of the solution space. Furthermore, during the assignment of paired nucleotides without a sequence constraint, the G-U base pair is neglected unless a sequence constraint forces such an assignment. This generates a biased initial population of sequences that are subsequently optimized by applying a genetic algorithm. Unfortunately, `MODENA` is available as binary only, of which the maximum population size is restricted to 1000 and at least one iteration of the genetic algorithm optimization is enforced. Therefore, we could not compare the effect of the implemented nucleotide assignment algorithm alone. However, their

sequence sampling essentially corresponds to our biased sampling method described in the previous paragraph.

The Haskell prototype implementation in Höner zu Siederdissen *et al.* (2013), `RNAdesign`, used lazy enumeration of all solutions and therefore allows uniform sampling. It opts for $\mathcal{O}(1)$ sampling, with low overhead in components. However, this applies only for sufficiently small problems. The way the prototype samples does not scale well for designing sequences with many and complex constraints as each component may get prohibitively large. These limitations get obvious when comparing the memory and CPU requirements of both implementations. While `RNAblueprint` needs 7 MB and about 0.35 s to generate one compatible sequence (without any optimization) for the complex design example shown in Supplementary Figure S2D the prototype implementation needs 15 GB and about 45.8 s on an Intel i7-6700, 3.4 GHz machine. The memory requirement is not independent of the sample size and further increases during the sampling process. We are aiming on a flexible approach where the sequence sampling step should not be the bottleneck as it might be necessary to generate multiple dependency graphs for exploring sequence and solutions spaces and application of computationally demanding objective functions, e.g. including pseudoknot prediction, will anyway slow down the design process.

In summary, our method is capable of generating sequences with a well defined distribution independent of the input constraints or the sample and solution space size. Note, that `RNAblueprint` can be easily incorporated in any multi-state design software such as `MODENA` in order to explore the complete solution space of complex multi-state design problems in an unbiased way.

4.2 Sequence sampling

In a typical RNA design scenario, sequences compatible with the structural constraints are scored using an objective function, which gets either minimized or maximized. Thereby, the sequence space is transformed into a landscape of complex and typically unknown structure that needs to be explored. Sampling completely new sequences generates solutions distributed over the complete landscape. This way, for an infinite sampling time the global optimum is always found. However, the optimization is rather slow because in each sampling step the reachable neighborhood is the complete solution space. The solution space of small examples is already of size 4.1×10^7 to 1.4×10^{14} (Supplementary Fig. S2) and therefore only a small fraction of all solutions is evaluated during a typical optimization run. The other sampling methods, i.e. P-local and C-local, described in Section 2 dramatically reduced the size of the reachable neighborhood. An adaptive walk using these move steps led to the solution ending up in local minima. The quality of these minima and how fast they were reached depends on the number of nucleotides changed in each step, Supplementary Figure S3.

In Figure 4, the published three state design example (Höner zu Siederdissen *et al.*, 2013) was extended to four and five input structures. The extension was made in a way that the complexity of the dependency graph from short paths and cycles in the three state example was increased to larger connected components, Supplementary Figure S2. We compared the performance of different sampling methods that differ in the size of their largest move step, see Figure 4. One method, called *global*, always generates a completely new sequence. When sample *C-local* is applied, the assignments of a randomly selected connected component are redrawn. The random selection is weighted by the number of possible solutions associated to the connected components. In contrast, *P-local* resamples only vertices which are not articulation points of a randomly selected path.

If the dependency graph contained only short paths and cycles (three state example), the C-local sampling approach was similar to the P-local sampling, i.e. reached a local minimum relatively fast and the cost converged. The relative mean cost difference between P-local and C-local sampling minima results from the fact that articulation points were redrawn by the latter only. This allowed a maximum step size of up to six nucleotides (complete circle) compared to three nucleotides (longest path), Supplementary Figure S3. The more complex the dependency graph, i.e. the more articulation points and larger connected components exist, the more pronounced this difference between P-local and C-local sampling, Supplementary Figure S3. If one large connected component contained most of the bases (five state example), performing a C-local sampling where all assignments of the large component are most likely reassigned (Supplementary Fig. S3), was similar to a completely new sampled sequence, i.e. the slope of the corresponding curves in Figure 4 are similar. However, the hamming distance to

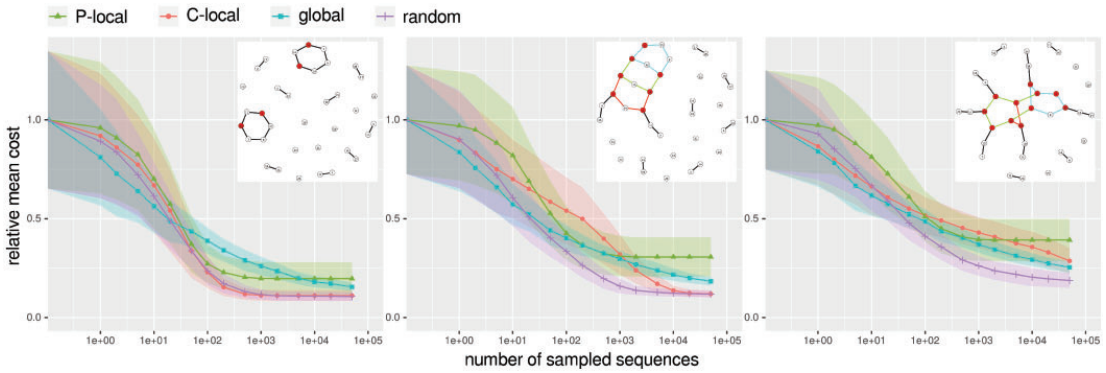


Fig. 4. Cost change during the optimization procedure using different move steps and dependency graphs. We minimized function (2) with $\xi = 0.5$ to calculate the cost. The x-axis shows the number of sampled sequences while the y-axis resembles the mean cost from 100 optimization runs, normalized to the mean cost of the initial randomly chosen sequences. For each trend curve the confidence interval ($\pm\,\sigma$) is indicated. Three different move steps (P-local, C-local and global) and an additional run, where one of these moves was randomly picked at every step (random), are compared. At the leftmost plot a very simple dependency graph was generated, only consisting of paths and two cycles, in the middle plot the graph already contains a block and on the right hand side many vertices are captured in one big connected component. The slope of the cost change mainly results from two aspects, the rejection rate and the quality of the newly found solutions. Both are heavily dependent on the size of the move step, therefore we see a change from the left to the right plot, as the move steps of C-local, global and random become bigger, Supplementary Figure S3

reachable neighbors was different for C-local and global sampling, Supplementary Figure S3. Reaching a local minimum indicates that most likely no further cost improvement can be made using the same sampling method. For the analyzed examples applying a simulated annealing approach, i.e. using an adaptive walk that allows for the acceptance of worse solutions on the way, did only slightly improve the results, see Supplementary Figure S4. Changing the method and thereby changing the move step allows other local minima with better solutions to be reached. Interestingly, our analyses showed that randomly changing the sampling method in each step, *random* in Figure 4, gave significantly better results faster in most cases. We investigated the reachable neighborhood of selected time points during optimization of the four state design example in more detail, Supplementary Figure S5. After 1000 sampling steps, the mean cost of sequences optimized with the random approach was significantly lower than the cost reached with global sampling (student's *t*-test *P*-value: 10^{-55}). Furthermore, the number of neighbors with a cost below the current best solution was similar, Supplementary Figure S5. At the end point of the trend curves (after 500 000 sampling trails), C-local and random sampling reached the same mean costs and within their analyzed neighborhood of size 350 600 no better solution was found, Supplementary Figure S5. Interestingly, the sequences optimized with global sampling did not reach the same mean cost and the likelihood of generating a better solution was very low, Supplementary Figure S5. We stress again that these observations are highly dependent on the design problem, e.g. the complexity of the dependency graph and the length of the sequence to be designed. However, we show in the following that applying the random sampling method to a diverse benchmark dataset of nested and pseudoknotted structural input gives reasonably good results.

4.3 Impact of normalization and weighting

To analyze the effect of the corrected objective function (2) and the applied optimization procedure we used the recently published benchmark dataset (Taneda, 2015), which consists of two-, three- and four-target design problems as well as three pseudoknotted two-target sets. These examples were either taken from naturally occurring RNAs that are able to switch between structural states or were generated in a way that reachable, sub-optimal structures are taken as input constraints for the design process. `RNAblueprint` itself does no optimization but rather implements a move set on uniformly sampled sequences. We implemented an adaptive walk that, given a start sequence, randomly selects one of the three sampling methods and applies it to generate the next sequence candidate. The generated sequence is retained if its cost is lower than the best prior solution. On the small examples evaluated in Figure 4, this approach adapted best to the varying complexity of the underlying dependency graphs. To score sequences, we applied an objective function that ensures on one hand that the target structures of a good solution dominate the ensemble while on the other hand the energy difference between the target structures is minimized. In its original version (1), proposed for the two state design case in (Flamm *et al.*, 2001), the corresponding two terms were summed up without any weighting. Designs for two states gave reasonable results compared to other approaches, see Table 1. However, a systematic extension to three or even more states needs individual normalization of the individual terms. Therefore, we proposed a corrected objective function (2), which is adjusted to the good performing original two state objective. Especially for the four structure designs this yielded a significant improvement over the original one, see Table 1. Note, when using a multi-objective approach it is assumed that the weighting is implicitly found during optimization (Taneda, 2015).

Comparing the results of our naïve optimization procedure with multi-objective approaches that implement complex genetic algorithms to optimize sequences we performed similar or even better on the benchmark dataset as measured by δe_1, i.e. the difference of the lowest energy target structure to the ground state and δe_2, i.e. the difference between the ground state and the highest energy target structure, on the benchmark dataset. Furthermore, we also compared how often the desired target structures are energetically equal to the predicted MFE structure, see Supplementary Tables S2–S7. These values are termed n_i, i being the number of target structures with equal energy to the MFE. Given this benchmark measure, `MODENA` and `RNAblueprint` performed similarly. A notable difference between our approach and `MODENA` is that the latter uses a genetic algorithm to optimize a population of 500 individuals of which the best 100 are taken, while we performed 100 independent optimizations. We expect to get similar sequences from a population-based approach while the solutions generated with our approach are extremely diverse.

Although δe_1, δe_2 and n_i together are a good measure of the solution quality of this specific design problem, they do not describe the

Table 1. Comparison of currently available approaches to solve multi-target designs

	Nested Structure Input												Pseudoknotted Structure Input					
	`RNAblueprint`						`MODENA`[a]			`Frnakenstein`[a]			`RNAblueprint`			`MODENA`[a]		
	original			*corrected*														
	2str	3str	4str	2str	3str	4str	2str	3str	4str	2str	3str	4str	LE80	PK60	PK80	LE80	PK60	PK80
$\mu(\delta e_1)$	**0.28**	0.22	1.46	0.31	**0.10**	**0.48**	0.38	0.27	0.84	0.35	0.39	0.92	**0.82**	**0.03**	**0.15**	0.89	0.12	0.29
$\tilde{x}(\delta e_1)$	**0.00**	**0.00**	0.70	**0.00**	**0.00**	**0.05**	0.10	**0.00**	0.39	0.10	0.10	0.55	0.30	**0.00**	**0.00**	**0.20**	**0.00**	**0.00**
$\mu(\delta e_2)$	**0.34**	0.43	1.96	0.36	**0.26**	**1.21**	0.76	0.54	1.78	1.09	0.96	1.89	**1.09**	**0.08**	**0.17**	1.22	0.32	0.56
$\tilde{x}(\delta e_2)$	**0.00**	0.20	1.30	**0.00**	**0.10**	**0.80**	0.50	0.30	1.40	0.60	0.80	1.60	**0.55**	**0.00**	**0.00**	**0.55**	**0.00**	0.05

Results of two-, three- and four-target designs are shown. For `RNAblueprint` and `MODENA` two-target designs of pseudoknotted structures are also presented. For each resulting sequence, we evaluated the difference between the most stable target structure to the ground state (δe_1) and the highest energy target structure to the ground state (δe_2). The mean (μ) and median ($\tilde{x}$) energy difference for 100 and 30 generated sequences is presented for the nested and pseudoknotted structure input, respectively. Performance of the individual sequences is listed in Supplementary Tables S2–S7. Boldface values highlight the best performing approach on a specific dataset. For `RNAblueprint` the values for the *original* (1) and *corrected* (2) objective functions are listed.

[a]Values taken from the original publication (Taneda, 2015).

actual functionality of the switch *in vitro* or *in vivo*. An objective function describing every aspect necessary to create a functional switch might contain many more features, some of which cannot easily be calculated. Furthermore, it is questionable whether the creation of 100 solutions is even useful. It might be better to run the optimization longer and retrieve 10-20 heterogeneous solutions, as this is a more realistic number for experimental validation.

4.4 Flexibility matters

Three example objective functions were proposed by Flamm and coworkers to design two-state temperature and structural switches (Flamm *et al.*, 2001). Those objectives have been adapted to multi-state design and are still used to benchmark new software (Höner zu Siederdissen *et al.*, 2013; Taneda, 2015). `MODENA` enables the user for the first time to choose from a catalog of different structure prediction methods to calculate features of a given sequence and derive new objectives. However, this catalog is fixed and therefore the complete functionality of the applied software might not be available. This is especially true for recent developments, such as the soft constraint framework implemented in the `ViennaRNA package` (Lorenz *et al.*, 2016) and the test tube ensemble defect available in `NUPACK` (Wolfe and Pierce, 2015). Furthermore, the methods to optimize sequences, in the case of `MODENA` by applying a genetic algorithm, cannot be changed. Therefore, we implemented `RNAblueprint` as a library and equipped this sequence generator with a flexible scripting interface where the user can easily implement its own optimization procedures and come up with new objective functions. Thus, features such as molecule concentrations, specific nucleotide compositions, or various probabilities can be captured in the design process.

5 Conclusion

We have developed a software solution that makes it possible to uniformly sample RNA sequences compatible with structural and sequence constraints. Sampling in an uniform way from a well defined solution space ensures to efficiently investigate the entire solution space and avoids heavy re-evaluation of repeatedly generated sequences. Therefore, it is possible to review many more solutions, which potentially leads to better results. We are currently investigating how to adapt the graph coloring algorithm to implement other desired sampling distributions, such as Boltzmann sampling according to a state energy model in a manner similar to what is done for single target design in `IncaRNAtion` (Reinharz *et al.*, 2013). This way promising sequences that are able and likely to fold into the target structures would be already favored during the sampling procedure.

Scripting interfaces make it easy to freely combine different optimization algorithms and to incorporate evaluations of different software packages into the objective function. We used the `NUPACK` and the `ViennaRNA` package to design multi-stable RNA structures with and without pseudoknots, respectively. With the scripting interface, any software such as the recently published `RNA shapes studio` (Janssen and Giegerich, 2015) and the approach by Wolfe and Pierce to reduce the amount of unwanted complexes when designing interacting molecules (Wolfe and Pierce, 2015), can be easily integrated. As the correct sequence generation problem for a set of fixed structural constraints is now efficiently solved, further research can focus on the challenging task of finding objective functions that better describe the goals and functions of RNA molecules. Using `RNAblueprint` it is now feasible to explore a much broader range of objectives and it is easy to adapt and recombine existing software and optimization techniques to generate an RNA molecule that perfectly suits the specific needs and goals of the task.

We illustrated the usefulness of our approach with typical but small sample applications. A general solution for solving all the diverse RNA design problems does not exist and there is also no universal way how to benchmark existing tools or novel approaches against each other. Applied measurements heavily depend on the goal and the objective of the design and therefore user knowledge is always necessary to choose an appropriate optimization method, move set and objective function.

Acknowledgements

Thanks to Christian Höner zu Siederdissen for assistance with the prototype Haskell implementation, Peter F. Stadler and Daniel Merkle for fruitful discussion and our private `boost` help desk Jakob L. Andersen. We thank Life Science Editors for proofreading and editing. Computational results have been achieved in part using the Vienna Scientific Cluster (VSC). We further like to thank the reviewers helping us to improve the manuscript significantly.

Funding

The project RiboNets acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323987. This work was furthermore supported by the COST-Action CM1304 'Systems Chemistry', the FWF projects SFB F43 'RNA regulation of the transcriptome' and 'Doktoratskolleg RNA Biology W1207-B09'.

Conflict of Interest: none declared.

References

Abfalter,I.G. *et al.* (2003) Design of multi-stable nucleic acid sequences. In: Mewes,H.W. *et al.* (eds.) In: *Proceedings of the German Conference on Bioinformatics (GCB)*. Belleville Verlag, Michael Farin, München, vol. **1**, pp. 1–7.

Berens,C. and Suess,B. (2015) Riboswitch engineering — making the all-important second and third steps. *Curr. Opin. Biotechnol.*, **31**, 10–15.

Chappell,J. *et al.* (2015) Creating small transcription activating RNAs. *Nat. Chem. Biol.*, **11**, 214–220.

Espah-Borujeni,A. *et al.* (2015) Automated physics-based design of synthetic riboswitches from diverse RNA aptamers. *Nucleic Acids Res.*, gkv1289.

Flamm,C. *et al.* (2001) Design of multistable RNA molecules. *RNA*, **7**, 254–265.

Garcia-Martin,J.A. *et al.* (2015) RNAiFold 2.0: a web server and software to design custom and rfam-based RNA molecules. *Nucleic Acids Res.*, **43**, W513–W521.

Green,A.A. *et al.* (2014) Toehold switches: de-novo-designed regulators of gene expression. *Cell*, **159**, 925–939.

Hofacker,I.L. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Monatshefte Für Chemie/Chem. Mon.*, **125**, 167–188.

Höner zu Siederdissen,C. *et al.* (2013) Computational design of RNAs with complex energy landscapes. *Biopolymers*, **99**, 1124–1136.

Isaacs,F.J. *et al.* (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotechnol.*, **22**, 841–847.

Janssen,S. and Giegerich,R. (2015) The RNA shapes studio. *Bioinformatics*, **31**, 423–425.

Lorenz,R. *et al.* (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol.*, **6**, 26.

Lorenz,R. *et al.* (2016) RNA folding with hard and soft constraints. *Algorithms Mol. Biol.*, **11**, 8.

Lyngso,R.B. *et al.* (2012) Frnakenstein: multiple target inverse RNA folding. *BMC Bioinformatics*, **13**, 260.

Maon,Y. *et al.* (1986) Parallel ear decomposition search (EDS) and ST-numbering in graphs. *Theor. Comp. Sci.*, **47**, 277–298.

Michael Mitzenmacher,E.U. (2005) *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University, Puerto Rico.

Neupert,J. *et al*. (2008) Design of simple synthetic RNA thermometers for temperature-controlled gene expression in *Escherichia coli*. *Nucleic Acids Res*., **36**, e124.

Qi,L. *et al*. (2012) Engineering naturally occurring trans-acting non-coding RNAs to sense molecular signals. *Nucleic Acids Res*, **40**, 5775–5786.

Reinharz,V. *et al*. (2013) A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, **29**, i308–i315.

Rodrigo,G. and Jaramillo,A. (2014) RiboMaker: computational design of conformation-based riboregulation. *Bioinformatics*, **30**, 2508–2510.

Rodrigo,G. *et al*. (2012) De novo automated design of small RNA circuits for engineering synthetic riboregulation in living cells. *Proc. Natl. Acad. Sci. U. S. A*., **109**, 15271–15276.

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

Wachsmuth,M. *et al*. (2013) De novo design of a synthetic riboswitch that regulates transcription termination. *Nucleic Acids Res*., **41**, 2541–2551.

Wolfe,B.R. and Pierce,N.A. (2015) Sequence design for a test tube of interacting nucleic acid strands. *ACS Synth. Biol*., **4**, 1086–1100.

Zadeh,J.N. *et al*. (2011a) Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem*., **32**, 439–452.

Zadeh,J.N. *et al*. (2011b) NUPACK: analysis and design of nucleic acid systems. *J. Comput. Chem*., **32**, 170–173.

Supplementary Material
RNAblueprint: Flexible multiple target nucleic acid sequence design

Stefan Hammer, Birgit Tschiatschek, Christoph Flamm, Ivo L. Hofacker and Sven Findeiß

Contents

1 Supplementary Text

1.1 Local neighborhood of various move steps

To get a better understanding of the solution landscape based on the introduced move steps, we analyzed the local neighborhood of three small examples with dependency graphs of varying complexity shown in Supplementary Figure S2. Using one of the introduced sampling methods (*global*, *C-local* and *P-local*, see section 2 in the main text), the local neighborhood was explored by stochastic sampling. The analysis includes the actual hamming distance to the start sequence (Supplementary Figure S3), and the cost change (Supplementary Figure S5) for the two parts of the multi-state objective function (formula (2) in the main text). Additionally the *random* move, where one of the three sampling methods is chosen randomly at each step was investigated. For C-local, 85% of the reachable neighborhood, i.e. 3506 neighbors, was generated for each of the 100 sequences. The same absolute number was used for the global and random approach. With the P-local move, only very few neighbors can be reached, therefore we sampled as many solutions as possible using a stop condition.

The hamming distance describes the size of the move step in terms of actually changed nucleotides, see Supplementary Figure S3. The distribution of these distances for any move step was very much dependent on the structure of the dependency graph. For the four and five structure example the C-local move always showed a flat distribution at smaller hamming distances and one defined peak at a certain distance. This results from the fact that the corresponding dependency graphs consisted of one bigger connected component in addition to several small ones (see Supplementary Figure S2). The connected components could be divided into two disjoint sets due to the bipartite property of the base assignments. If the coloring pattern of the disjoint sets of the bigger component was changed in a way that the coloring switches, all nucleotides of this big connected component are changed, resulting in the defined peak with a hamming distance of exactly the size of this component. If the sets maintained the coloring pattern, we obtained a flat distribution of several smaller distances. Global sampling of the full sequence resulted in similar peaks, however with a shift towards higher distances, as all the smaller connected components are also resampled at every move. The peaks at higher distances show a more even distribution for the same reason. In the analyzed examples, no decomposed path was longer than three nucleotides, excluding special vertices. Therefore, we only obtained hamming distances between 0 and 3 with the P-local approach. Sampling with a randomly picked move step resulted in a very nice superimposition of all the hamming distance distributions, see Supplementary Figure S3.

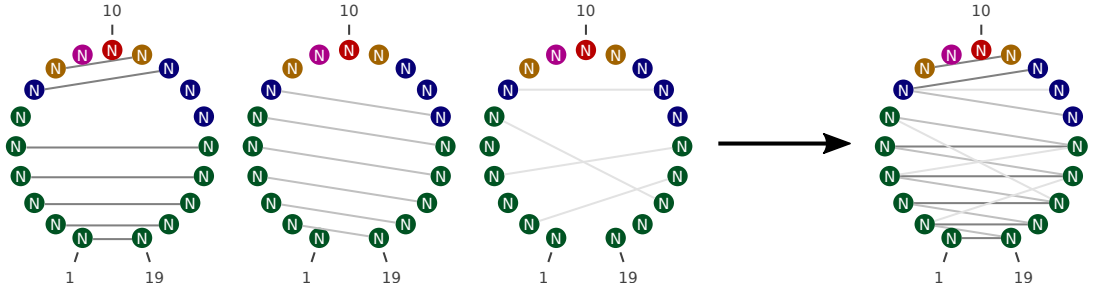
We further investigated the cost change from a start sequence to its local neighborhood reachable by applying the described sampling methods, Supplementary Figure S5. This is depicted in two-dimensional density plots as cost changes for the two parts of the multi state objective function (objective 1 and objective 2) at the x- and y-axis. The weighted overall cost change can be obtained by following the inclined lines. The purple line indicates neighbors with a constant overall cost, the scale of the actual improvement or decline can be read from the x-axis. From left to right, plots with further optimized sequences obtained from different time steps of Figure 4 were used as start sequences for the analysis. The degree of optimization is therefore measured as "number of sampled sequences".

In the most left plot (number of sampled sequences = 0), the local neighborhood showed a quite similar distribution in terms of cost improvements on objective 1 and objective 2 for any sampling method. After 100 iterations of optimization, C-local showed the highest number of neighbors with better costs (number in purple box), furthermore the cost improvement possible for individual neighbors was highest for the C-local approach. Both might result from the low quality of the optimized sequences compared to the other approaches. When analyzing even more optimized sequences after 1000 steps, the C-local move still showed highest number of better neighbors. However, the cost improvement possible was quite similar between the various methods. Only with the P-local approach, the cost could not be substantially improved as the local minimum had almost been reached. After $5 \cdot 10^5$ iterations no better solutions

could be obtained for the C-local and P-local approach as the optimization appeared to have reached a local minimum. Overall, P-local sampling behaved similar to C-local sampling, but reached the local minimum of the optimization much earlier due to the smaller size of the reachable neighborhood. In a local minimum no better solution could be obtained with the same move step. Only the global approach could not reach a optimization minimum, as we sampled from the whole solution space with this move. However, better solutions could only rarely be found (see Supplementary Figure S5).

100

2 Supplementary Figures

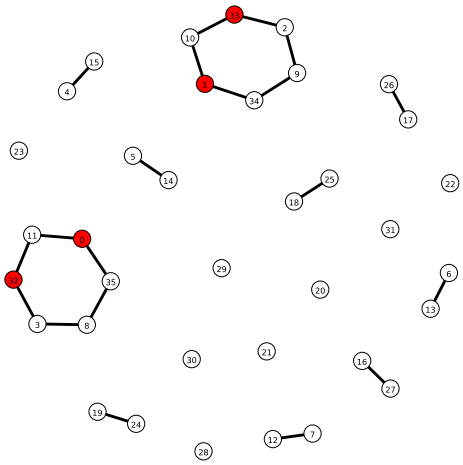


Supplementary Figure S1: The dependency graph G can be generated by the union of the circle plot representations of the given structural constraints. The three input structures in dot-bracket notation are first converted to the circle representation depicted on the left hand side and then the union is formed as shown on the right hand side. Vertices represent bases in a given order along the backbone of the molecule. Edges in different shades of gray represent the base pairs of the three input structures. Colors on the vertices show the different connected components into which the graph can be decomposed. The further decomposition and graph coloring approach of this example is shown in Figure 1 of the main text. Layout by `VARNA`[4]

A)

```
((((....))))....((((....))))........
........((((....((((....))))....))))
(((((((((....))))((((....))))....))))
```

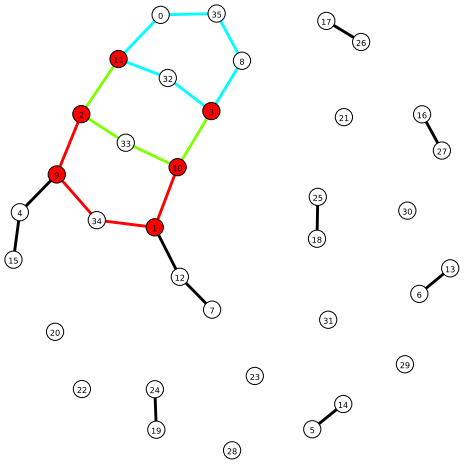
```
Size of solution space: 1.42658e+14
```



B)

```
((((....))))....((((....))))........
........((((....((((....))))....))))
(((((((((....))))((((....))))....))))
.((((....))))...((((....))))........
```

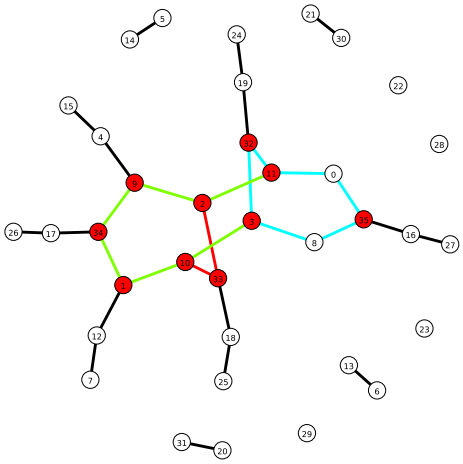
```
Size of solution space: 1.24018e+13
```



C)

```
((((....))))....((((....))))........
........((((....((((....))))....))))
(((((((((....))))((((....))))....))))
.((((....))))...((((....))))........
.((((....))))...((((((........))))))
```

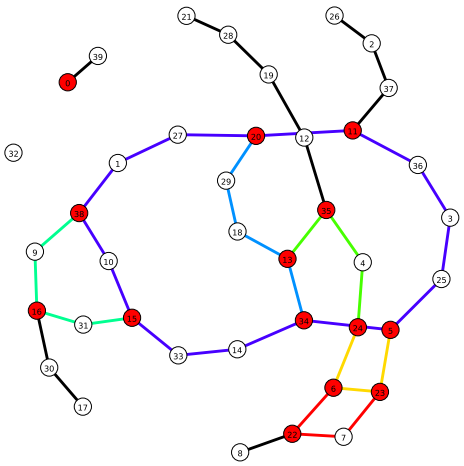
```
Size of solution space: 7.08853e+10
```



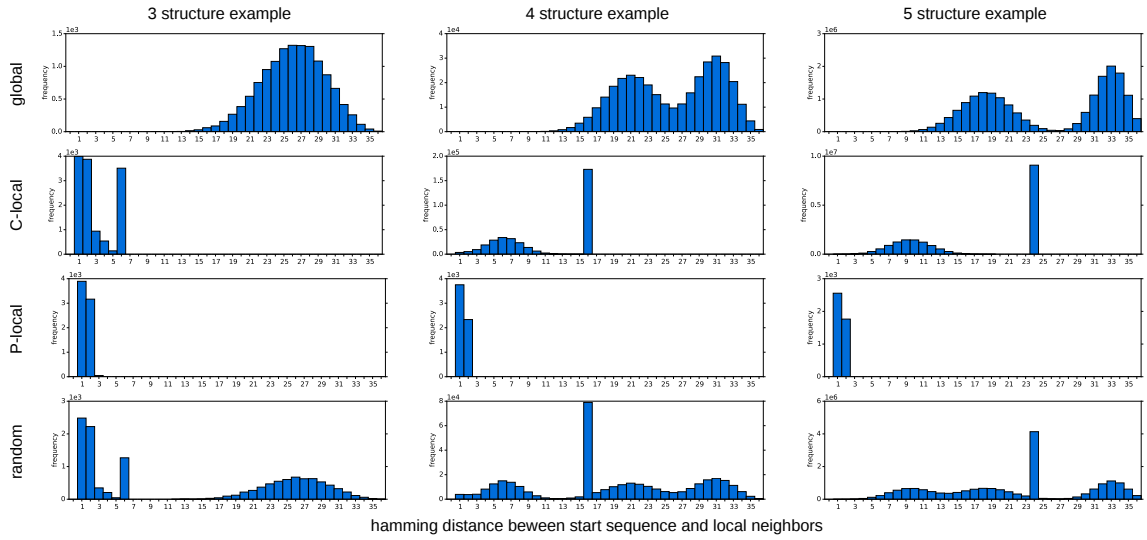
D)

```
.((((((..((....)).....))))))............
..........((.(((....((......))...))).)).
((((((..........((((.......))))))..))))))
.........(.((((((.............)).)))).).
.....((((..(((....))).))))..............
.....(((..............)))................
ANNNNNNNNNNNNNNNNNNNNNNNCNNNNNNNNNNNNNNN
```

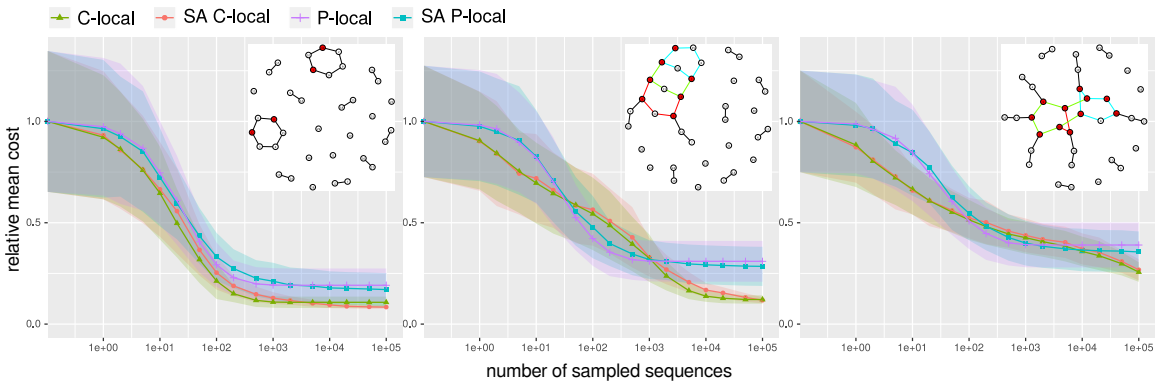
```
Size of solution space: 4.14142e+07
```



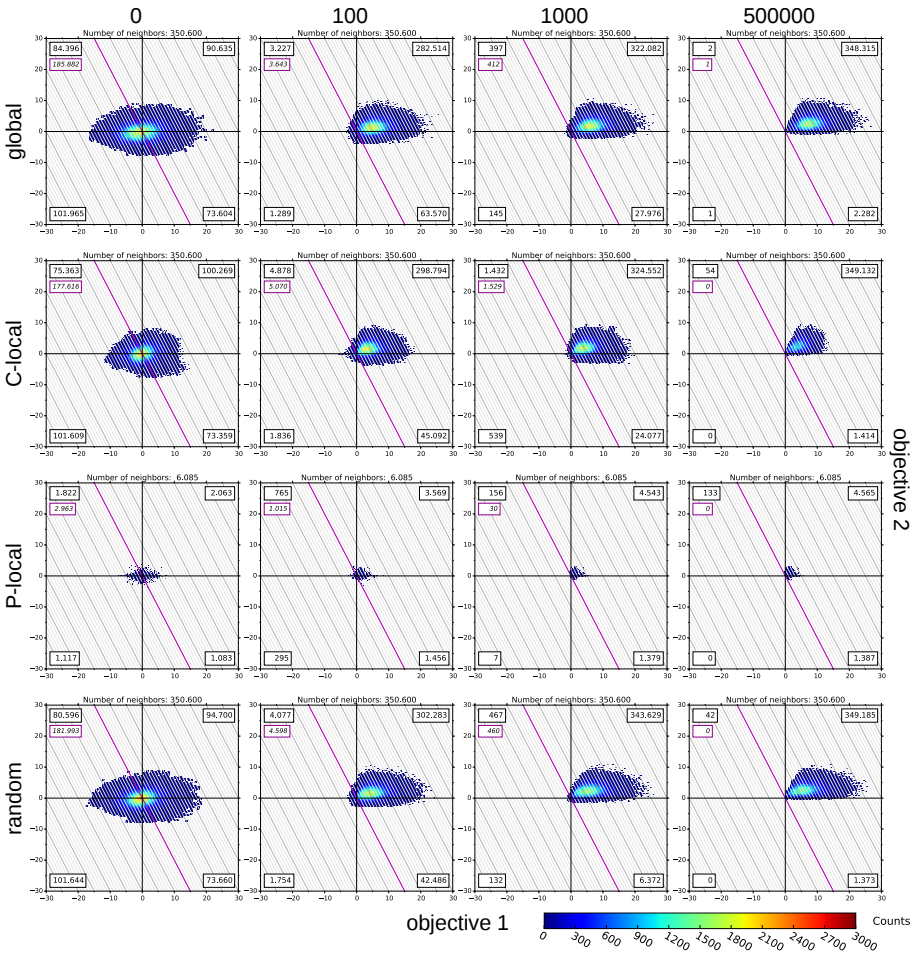
Supplementary Figure S2: Example design inputs and the corresponding graph representations. The depicted examples were used to evaluate different sampling methods provided in `RNAblueprint` (A-C) as well as uniform and biased stochastic sampling (D). The structure and sequence constraint input is shown at the top and the corresponding dependency graph at the bottom in each example. Articulation points are highlighted in red. Edges of circles and paths resulting from the ear decomposition of complex sub-graphs are colored individually.



Supplementary Figure S3: Size of different move steps measured as hamming distance between an initial start sequence and most of the reachable neighbors. Rows depict the used sampling method/move steps while columns show the three different small design examples with various complexity (see Supplementary Figure S2A-C). The global method always generates a completely new sequence. When C-local sampling is applied, the assignments of a randomly selected connected component are redrawn while P-local resamples only vertices which are not articulation points of a randomly selected path. One of these three methods is randomly chosen in each sampling step when the random approach is used. For C-local, 85% of the reachable neighborhood was sampled, while for global and random the same absolute number was used (3str: $1.3 \cdot 10^4$, 4str: $\approx 3.5 \cdot 10^5$, 5str: $\approx 1.6 \cdot 10^7$ sequences). The P-local approach had a much smaller neighborhood, which was sampled with a stop condition to reach most of the neighbors (3str: 7108, 4str: 6076, 5str: $1.3 \cdot 10^4$ sequences).



Supplementary Figure S4: Cost change during the optimization procedure applying simulated annealing (SA) or an adaptive walk optimization approach. We minimized function (2) from the main text with $\xi = 0.5$ to calculate the cost. Furthermore, P-local as well as C-local were used as move steps for comparison. The shadowed area depicts the confidence interval ($\pm\sigma$). The x-axis shows the number of sampled sequences while the y-axis resembles the mean cost from 100 optimization runs, normalized to the mean cost of the initial randomly chosen sequences. We used structural inputs with varying complexity listed in Supplementary Figure S2A-C. The simulated annealing approach followed a linear and continuous cooling schedule with $\Delta T = \frac{1}{d}, T_0 = 1$ where d is the number of sampled sequences (x-axis).



Supplementary Figure S5: Relative cost change to local neighborhood with various move steps on the 4 structure example Supplementary Figure S2B. Each row corresponds to a different sampling method, columns represent the neighborhood of differently optimized sequences obtained from Figure 4 in the main text. The degree of optimization is measured in "number of sampled sequences" during the optimization procedure. The density plots depict the cost change to local neighbors reachable with the used move step. The cost difference is split into changes of the two parts of function (2) in the main text. These are shown as objective 1 and objective 2 on the x-axis and y-axis, respectively. The weighted overall cost change to the start sequence can be obtained by the inclined lines, the purple line indicating unchanged cost. The size of the neighborhood varies, for C-local we sampled 85% of unique neighbors and used the same absolute number for global and random. For the P-local move we sampled as many unique sequences as possible in a reasonable time using an stop condition as this neighborhood is very small. The numbers in the boxes display the count of solutions in this quadrant, the purple box the absolute number of neighbors with a cost change smaller than zero, meaning better solutions than the initial sequence.

3 Supplementary Tables

Supplementary Table S1: Published software to solve the inverse folding problem with single-, two- and multi-target structural input. For each tool how the initial sequence is generated and what kind of search strategy is applied. For further details we like to refer to the original referenced publications.

Name	Initial Sequence Selection	Search Strategy	Reference
single-target input			
`RNAinverse`	random	stochastic local search	[12]
`RNA-SSD`	random	stochastic local search	[1]
`INFO-RNA`	energy optimized	stochastic local search	[3]
`RNAexinv`	from `RNAinverse`	stochastic local search	[2]
`RNA-ensign`	random	global sampling	[17]
`IncaRNAtion`	seedless	local/global sampling	[20]
`DSS-Opt`	seedless	Newtonian dynamics simulation and simulated annealing	[19]
`EteRNABot`	random	stochastic local search	[16]
`NUPACK:Design`	random	stochastic local search	[5, 26, 25]
`ERD`	RNA sub-sequences of different structural elements are sampled from natural occurring RNA sequences	evolutionary algorithm	[6]
`antaRNA`	a graph that represents all possible paths to generate compatible sequences is used	ant colony based optimization	[14, 15]
two-target input			
`switch.pl`	random	stochastic local search	[8]
`RiboMaker`	random	stochastic local search	[21]
`RiboSwitch Calculator`	random	genetic algorithm	[7]
`RNAiFold2.0`	seedless	local or global sampling	[9, 10, 11]
multi-target input			
`ARDesigner`	random	stochastic local search	[22]
`Frnakenstein`	random or from `RNAinverse`	genetic algorithm	[18]
`MODENA`	random	multi objective genetic algorithm	[23, 24]
`RNAdesign`	random	stochastic local search	[13]

The following tables show the benchmark results summarized in Table 1 in the manuscript. The benchmarks were adapted from Taneda [24] and calculated using the multi-stable design optimization approach and the weighted objective function. For the two-, three- and four-structure inputs (see Methods section in main text) we generated 100 independent solutions using the `ViennaRNA` package and the stop condition set to 1000. For the pseudoknotted structure data sets only 30 solutions were generated utilizing the `NUPACK` package for evaluation and the stop condition of 100.
We used the same measures as in [24] and expanded the table by a probability value. δe_1 and δe_2 are the minimal and maximal energy difference between the evaluated energies of the target structures and the minimum free energy. The values shown in each row of the table are for the solution with the lowest δe_2 and, if multiple solutions existed, also with the the lowest δe_1. Furthermore, $n1, n2, n3, ...nM$ are the number of solutions such that $1, 2, 3, ...M$ of the target structures have the lowest free energy. We also introduced a new measure called "$\sum$ prob", which is the sum of the probabilities of all target structures in the Boltzmann ensemble for the solution picked using the δe_1 and δe_2 values. We furthermore list the time it took to construct the dynamic programming tables and to sample and optimize the sequences in seconds (calculated on VSC3: Intel Xeon E5-2650v2, 2.6 GHz, Ivy Bridge-EP family). Note, the latter includes the energy calculations by `ViennaRNA` or `NUPACK`. The column named "max dim" shows the maximal number of dimensions of all dynamic programming tables. This number is the main measure for the complexity and memory requirements of the specific problem (see main text for calculation of the

complexity).

Supplementary Table S2: Detailed results of [24] two-target design inputs (SV11 & RNAtabupath dataset)

RNA	l	n1	n2	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim
alpha operon	130	74	5	0.00	0.00	0.10	0.00	0.28	0.20	7521.18	7071.50	0.46	0.00	185.23	0
amv	145	3	0	0.30	0.40	0.96	0.85	1.16	1.10	9776.94	9353.50	0.22	0.00	308.39	0
attenuator	73	82	71	0.00	0.00	0.08	0.00	0.09	0.00	5149.67	4884.00	0.30	0.00	39.78	0
dsrA	85	2	0	0.10	0.10	1.64	1.50	1.67	1.50	7312.21	6543.00	0.21	0.00	78.00	0
hdv	153	62	44	0.00	0.00	0.19	0.00	0.24	0.10	7583.76	7228.00	0.05	0.00	276.75	0
hiv	280	12	7	0.00	0.00	1.28	1.10	1.36	1.20	15442.83	14653.50	0.02	0.00	2176.04	0
ms2	73	45	35	0.00	0.00	0.33	0.10	0.37	0.20	6092.20	5601.50	0.14	0.00	52.11	0
rb1	148	79	64	0.00	0.00	0.12	0.00	0.16	0.00	7579.62	7279.00	0.10	0.01	230.27	0
rb2	113	42	30	0.00	0.00	0.45	0.20	0.49	0.30	6818.82	6481.50	0.08	0.00	119.27	0
rb3	141	69	42	0.00	0.00	0.14	0.00	0.18	0.10	8740.21	8104.50	0.08	0.01	258.44	0
rb4	146	0	0	2.00	2.00	4.90	4.95	4.99	5.00	9327.93	8506.00	0.00	0.00	319.34	0
rb5	201	73	56	0.00	0.00	0.16	0.00	0.21	0.00	9001.18	8677.00	0.12	0.00	455.17	0
ribD	304	0	0	1.10	1.10	2.99	2.85	3.06	2.95	13307.82	13106.50	0.01	0.00	2078.01	0
s15	74	54	40	0.00	0.00	0.33	0.00	0.37	0.10	5043.62	4666.00	0.25	0.00	40.58	0
sbox	247	0	0	0.70	1.00	1.65	1.20	1.69	1.25	12247.86	11819.50	0.25	0.00	1141.45	0
spliced	56	3	0	0.00	0.20	0.98	1.00	1.05	1.05	4914.67	4528.50	0.19	0.00	31.28	0
sv11	115	9	5	0.00	0.00	1.12	0.90	1.16	1.05	5644.82	5254.50	0.02	0.00	129.32	2
μ		36	23	0.25	0.28							0.15			
$\tilde{x}$		42	7	0.00	0.00							0.12			

Supplementary Table S3: Detailed results of the three-target design inputs (RNAdesign dataset [3str]).

RNA	l	n1	n2	n3	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim
sq1	100	98	7	0	0.00	0.10	0.00	0.00	0.47	0.30	5975.53	5582.00	0.38	0.00	89.11	0
sq2	100	0	0	0	1.10	1.30	3.51	3.25	4.19	4.20	6957.17	6847.50	0.01	0.00	121.06	1
sq3	100	17	2	0	0.00	0.10	0.60	0.60	1.04	1.00	6980.33	6462.50	0.11	0.00	91.63	2
sq4	100	83	22	5	0.00	0.00	0.08	0.00	0.42	0.30	6566.18	5761.00	0.31	0.00	97.05	2
sq5	100	98	40	13	0.00	0.00	0.00	0.00	0.30	0.20	6353.41	5893.50	0.69	0.00	101.30	0
sq6	100	38	7	0	0.00	0.10	0.40	0.20	1.57	1.30	6569.40	6253.50	0.16	0.00	92.84	0
sq7	100	24	6	1	0.00	0.00	0.64	0.55	1.02	0.90	7195.74	6381.50	0.26	0.00	100.42	3
sq8	100	71	13	3	0.00	0.00	0.19	0.00	0.63	0.50	7156.62	6581.00	0.22	0.00	109.25	0
sq10	100	21	3	0	0.00	0.10	0.68	0.60	1.09	1.10	6510.08	5987.50	0.29	0.00	101.15	1
sq11	100	61	5	2	0.00	0.00	0.21	0.00	0.69	0.60	6276.30	5800.50	0.33	0.00	95.72	0
sq12	100	96	10	0	0.00	0.10	0.01	0.00	0.45	0.30	6201.07	5806.00	0.23	0.00	92.06	0
sq13	100	3	0	0	0.00	1.00	2.78	2.65	3.79	3.60	8529.50	7977.00	0.06	0.01	151.32	4
sq14	100	30	0	0	0.00	0.20	0.69	0.45	1.62	1.50	5814.88	5631.50	0.08	0.00	87.77	0
sq15	100	1	0	0	1.10	1.50	3.04	2.85	4.14	4.00	7987.25	7928.00	0.01	0.00	124.29	3
sq16	100	60	20	7	0.00	0.00	0.18	0.00	0.43	0.30	6868.02	6379.50	0.32	0.00	92.37	0

sq17	100	19	0	0	0.20	0.70	0.74	0.70	1.67	1.50	6226.68	5645.00	0.05	0.00	97.91	0
sq18	100	49	3	0	0.00	0.10	0.58	0.10	1.49	1.40	6430.86	6154.50	0.12	0.00	90.79	0
sq19	100	5	0	0	0.00	0.10	0.80	0.70	1.18	1.10	5899.59	5739.50	0.21	0.00	91.30	2
sq20	100	61	21	4	0.00	0.00	0.24	0.00	0.55	0.40	6217.49	6122.50	0.15	0.01	96.93	0
sq21	100	30	2	0	0.00	0.10	0.69	0.40	1.65	1.50	6281.56	5960.50	0.07	0.00	100.94	0
sq22	100	13	3	0	0.00	0.20	0.96	1.00	1.35	1.30	10181.89	9857.00	0.13	0.00	161.28	2
sq23	100	0	0	0	2.10	2.20	4.42	4.40	5.68	5.60	9113.57	8313.00	0.00	0.00	154.49	3
sq24	100	43	11	3	0.00	0.00	0.40	0.25	0.75	0.70	6759.85	6487.00	0.18	0.00	99.73	1
sq25	100	100	48	19	0.00	0.00	0.00	0.00	0.19	0.15	6919.91	6745.50	0.69	0.00	109.11	0
sq26	100	6	1	0	0.00	0.30	0.87	0.90	1.52	1.50	5995.44	5181.00	0.10	0.00	91.57	0
sq27	100	1	0	0	0.00	0.50	1.24	1.20	1.68	1.60	8398.12	8038.00	0.09	0.00	129.17	2
sq28	100	100	2	0	0.00	0.30	0.00	0.00	0.87	0.80	8230.88	7525.50	0.42	0.00	118.18	0
sq29	100	23	6	2	0.00	0.00	0.62	0.55	1.00	0.90	5672.03	5407.50	0.18	0.00	90.37	1
sq30	100	34	7	2	0.00	0.00	0.54	0.30	0.99	0.85	6888.60	6825.00	0.08	0.00	108.52	2
sq31	100	17	5	1	0.00	0.00	0.59	0.40	1.11	1.00	5486.78	5328.50	0.05	0.00	81.05	0
sq32	100	57	13	3	0.00	0.00	0.38	0.00	0.83	0.70	5432.50	5263.50	0.05	0.00	79.82	0
sq33	100	93	35	18	0.00	0.00	0.02	0.00	0.21	0.20	7759.08	7266.00	0.42	0.00	122.29	0
sq34	100	0	0	0	2.20	2.60	4.99	5.10	5.99	5.90	7943.20	7670.00	0.00	0.00	135.51	2
sq35	100	31	7	2	0.00	0.00	0.59	0.50	1.03	0.85	6722.60	6324.00	0.07	0.00	97.98	0
sq36	100	91	2	0	0.00	0.20	0.03	0.00	0.57	0.40	7397.27	6793.00	0.31	0.00	99.75	0
sq37	100	62	10	1	0.00	0.00	0.18	0.00	0.58	0.50	6480.41	6218.00	0.53	0.01	99.90	0
sq38	100	96	10	0	0.00	0.10	0.01	0.00	0.72	0.70	6178.97	5870.50	0.39	0.00	80.82	2
sq39	100	15	3	0	0.10	0.10	1.33	1.25	2.17	2.05	8087.73	7853.50	0.07	0.00	130.01	2
sq40	100	85	1	0	0.00	1.60	0.05	0.00	1.74	1.70	5812.31	5644.50	0.09	0.00	79.74	0
sq41	100	33	9	1	0.00	0.00	0.47	0.30	0.86	0.70	6434.00	5555.00	0.16	0.00	100.89	0
sq42	100	15	0	0	0.60	1.00	0.96	0.90	2.02	1.70	7347.52	6932.50	0.09	0.00	96.47	0
sq43	100	78	5	0	0.00	0.20	0.10	0.00	1.15	1.10	6068.68	5794.00	0.35	0.00	87.94	0
sq44	100	5	0	0	0.00	0.10	1.19	1.05	1.75	1.70	6358.53	5973.50	0.08	0.00	92.26	1
sq45	100	66	17	5	0.00	0.00	0.14	0.00	0.46	0.40	6132.27	5821.50	0.34	0.00	84.31	0
sq46	100	5	0	0	0.10	0.30	0.94	0.90	1.23	1.30	6289.39	5981.00	0.09	0.00	93.21	1
sq47	100	97	4	0	0.00	0.70	0.01	0.00	1.60	1.10	7376.39	6475.50	0.49	0.00	101.43	0
sq48	100	41	3	0	0.00	0.20	0.37	0.20	1.43	1.30	7333.19	6815.50	0.14	0.01	105.53	2
sq49	100	37	0	0	0.00	0.20	0.32	0.20	1.02	0.90	6034.74	5457.00	0.27	0.00	86.76	0
sq50	100	48	2	0	0.00	0.20	0.22	0.10	0.94	0.80	7113.65	6233.50	0.42	0.00	98.32	0
sq51	100	0	0	0	0.60	1.10	2.43	2.25	3.41	3.25	9160.54	8825.00	0.00	0.01	150.62	3
sq52	100	2	1	0	0.10	0.60	2.43	2.15	3.26	3.25	8423.05	7965.00	0.01	0.00	152.05	2
sq53	100	98	23	13	0.00	0.00	0.00	0.00	0.20	0.20	7125.22	6411.50	0.29	0.00	104.47	0
sq54	100	6	2	1	0.00	0.00	1.44	1.45	1.99	2.00	9938.95	9303.50	0.05	0.00	147.97	2
sq55	100	1	0	0	0.30	0.60	0.96	0.90	1.48	1.40	6368.29	5597.00	0.02	0.01	98.01	0
sq56	100	7	1	0	0.00	0.20	1.24	1.20	1.54	1.50	7552.24	6787.00	0.17	0.00	113.88	0
sq57	100	99	2	0	0.00	0.20	0.00	0.00	0.49	0.40	7489.52	7270.00	0.77	0.00	109.02	0
sq58	100	60	6	0	0.00	0.10	0.23	0.00	1.01	0.90	7231.36	6586.50	0.37	0.00	109.54	0
sq59	100	14	4	3	0.00	0.00	0.89	0.85	1.25	1.30	7664.53	7550.50	0.23	0.01	109.84	3
sq60	100	16	0	0	0.10	0.20	1.10	0.90	1.57	1.40	6759.46	6397.00	0.09	0.00	94.99	0
sq61	100	5	0	0	0.10	0.20	0.47	0.30	0.95	0.70	6632.91	6339.50	0.28	0.00	99.29	0
sq62	100	0	0	0	1.10	1.60	3.21	3.20	4.22	4.15	9448.37	8549.50	0.01	0.01	156.53	4
sq63	100	98	19	14	0.00	0.00	0.00	0.00	0.28	0.30	6861.21	5966.00	0.51	0.00	89.88	0

sq64	100	30	6	0	0.00	0.10	0.73	0.50	1.23	1.10	6767.57	6568.00	0.09	0.01	99.83	1
sq65	100	0	0	0	1.50	1.70	3.10	3.00	4.31	4.10	7224.15	6862.50	0.00	0.00	133.55	3
sq66	100	29	1	0	0.00	0.10	0.70	0.40	1.56	1.50	6621.04	6369.50	0.13	0.01	112.09	1
sq67	100	0	0	0	1.40	3.10	7.66	7.75	9.11	9.15	6991.76	6707.50	0.00	0.01	131.77	5
sq68	100	8	0	0	0.10	0.20	1.29	1.20	1.87	1.80	5628.45	5342.00	0.07	0.01	94.53	2
sq69	100	0	0	0	0.90	1.30	3.47	3.25	4.98	4.90	7484.97	7254.50	0.02	0.00	128.51	2
sq70	100	95	7	0	0.00	0.40	0.03	0.00	0.94	0.60	6528.09	5973.50	0.59	0.00	90.05	0
sq71	100	31	0	0	0.90	1.10	0.53	0.40	2.71	2.75	7094.99	6478.00	0.02	0.00	113.87	0
sq72	100	99	38	24	0.00	0.00	0.00	0.00	0.20	0.20	6428.78	5759.00	0.22	0.00	90.74	0
sq73	100	95	18	1	0.00	0.00	0.01	0.00	0.44	0.40	6848.43	6540.50	0.49	0.01	106.04	2
sq74	100	8	0	0	0.00	0.50	1.33	1.30	2.05	2.00	6568.45	6118.50	0.09	0.00	102.99	1
sq75	100	15	2	0	0.00	0.10	1.03	0.95	1.45	1.30	6588.74	6486.00	0.04	0.00	108.73	2
sq76	100	9	2	0	0.20	0.40	1.73	1.30	2.54	2.30	7323.90	6833.00	0.07	0.00	122.64	1
sq77	100	12	1	0	0.00	0.10	1.25	1.20	1.85	1.70	5951.63	5626.50	0.05	0.00	89.68	0
sq78	100	41	2	0	0.00	0.90	0.47	0.20	2.28	2.10	6518.50	6233.50	0.21	0.01	106.74	1
sq79	100	2	0	0	0.10	0.50	1.88	1.90	2.63	2.70	8657.86	7776.00	0.04	0.01	158.22	3
sq80	100	100	7	2	0.00	0.00	0.00	0.00	0.39	0.40	6933.06	6410.00	0.58	0.00	102.45	0
sq81	100	85	16	7	0.00	0.00	0.03	0.00	0.42	0.30	6684.96	6007.50	0.21	0.00	98.78	0
sq82	100	99	29	18	0.00	0.00	0.00	0.00	0.27	0.20	6867.12	5824.50	0.44	0.00	96.96	0
sq83	100	56	14	2	0.00	0.00	0.25	0.00	0.74	0.60	7350.49	6866.00	0.27	0.00	96.74	0
sq84	100	18	2	1	0.00	0.00	0.41	0.30	0.71	0.70	6109.86	5690.00	0.20	0.00	97.84	0
sq85	100	85	22	13	0.00	0.00	0.04	0.00	0.21	0.20	7482.15	6858.00	0.24	0.00	111.06	0
sq86	100	28	2	0	0.00	0.10	0.60	0.45	1.03	1.00	6010.22	5885.00	0.12	0.00	86.18	0
sq87	100	22	4	0	0.00	0.40	0.47	0.45	1.78	1.60	7865.87	7464.50	0.30	0.00	118.78	0
sq88	100	88	11	1	0.00	0.00	0.03	0.00	0.46	0.40	7470.32	6667.00	0.73	0.00	118.53	0
sq89	100	97	37	21	0.00	0.00	0.01	0.00	0.25	0.20	6052.79	5557.00	0.36	0.00	91.79	0
sq90	100	97	29	6	0.00	0.00	0.02	0.00	0.38	0.30	6660.64	6167.50	0.48	0.00	87.25	0
sq91	100	78	25	9	0.00	0.00	0.09	0.00	0.38	0.30	6520.25	6181.00	0.15	0.00	109.23	1
sq92	100	9	1	0	0.10	0.20	0.68	0.70	1.21	1.30	6748.22	6620.00	0.12	0.00	98.68	0
sq93	100	64	15	7	0.00	0.00	0.15	0.00	0.46	0.30	6222.48	5679.00	0.14	0.00	97.16	0
sq94	100	6	0	0	0.00	0.20	1.07	0.80	1.42	1.30	6180.87	5925.50	0.25	0.00	91.79	0
sq95	100	0	0	0	2.00	2.20	4.69	4.40	5.83	5.70	6537.24	6214.00	0.00	0.04	109.39	6
sq96	100	21	5	0	0.00	0.10	0.49	0.50	0.95	0.90	7243.22	6611.00	0.29	0.00	93.34	0
sq97	100	41	12	1	0.00	0.00	0.52	0.20	1.03	0.95	5482.89	5306.50	0.47	0.01	79.39	0
sq98	100	95	2	1	0.00	0.00	0.01	0.00	0.65	0.50	7096.59	6784.00	0.37	0.00	96.66	2
sq99	100	28	4	2	0.00	0.00	0.45	0.45	0.70	0.70	6321.52	5836.00	0.20	0.00	84.55	0
sq100	100	12	0	0	0.00	0.30	1.02	1.10	1.63	1.50	8625.88	8190.50	0.32	0.00	142.78	2
μ		42	7	2	0.17	0.36							0.22			
$\tilde{x}$		30	3	0	0.00	0.10							0.17			

Supplementary Table S4: Detailed results of the four-target design inputs (RNAdesign dataset [4str]).

RNA	l	n1	n2	n3	n4	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim

sq1	100	98	11	0	0	0.00	0.30	0.00	0.00	1.15	1.05	7065.64	6290.50	0.27		0.00	107.09		0
sq3	100	0	0	0	0	1.20	3.70	5.11	5.00	7.80	7.80	10313.01	10232.50	0.00		0.01	256.82		4
sq4	100	61	6	1	1	0.00	0.00	0.21	0.00	1.07	0.90	6232.92	5801.00	0.30		0.00	148.02		2
sq5	100	87	12	0	0	0.00	0.20	0.04	0.00	0.75	0.70	6139.30	5751.00	0.49		0.00	158.41		0
sq6	100	27	1	0	0	0.10	0.30	0.61	0.50	1.56	1.45	7384.96	7053.00	0.02		0.00	177.77		0
sq7	100	46	10	2	0	0.00	0.10	0.49	0.10	1.13	0.85	7489.11	6977.50	0.21		0.01	166.58		3
sq8	100	4	0	0	0	0.20	0.70	2.05	1.85	3.49	3.30	7959.74	7439.00	0.02		0.00	206.02		2
sq10	100	10	0	0	0	0.00	0.50	1.05	0.90	2.49	2.30	9071.55	8659.00	0.16		0.00	147.87		2
sq11	100	16	3	1	0	0.00	0.40	0.96	0.90	1.83	1.80	6080.15	5837.50	0.10		0.00	101.82		1
sq12	100	90	0	0	0	0.20	0.60	0.03	0.00	1.62	1.80	6572.47	5914.00	0.09		0.00	107.29		0
sq13	100	1	0	0	0	3.30	4.60	5.87	5.85	8.78	8.65	7085.27	6552.50	0.00		3.17	140.17		10
sq14	100	0	0	0	0	1.20	4.30	3.84	3.70	8.60	8.50	8844.75	8589.50	0.00		0.00	156.09		3
sq15	100	0	0	0	0	4.80	5.80	6.59	6.30	9.05	9.00	9336.95	8877.00	0.00		0.01	169.58		6
sq16	100	91	15	1	1	0.00	0.00	0.04	0.00	0.66	0.70	7928.64	7092.50	0.23		0.00	110.73		0
sq17	100	4	0	0	0	0.40	1.70	2.36	2.10	4.77	4.75	10917.99	10392.50	0.00		0.00	182.14		3
sq18	100	16	1	1	0	0.90	1.10	1.38	1.30	3.40	3.35	6310.94	6095.00	0.01		0.00	104.57		1
sq19	100	1	0	0	0	2.30	3.10	3.64	3.50	7.12	7.00	10726.86	10257.00	0.00		0.01	198.35		5
sq20	100	9	0	0	0	0.20	0.80	1.61	1.35	3.03	2.80	9639.32	9307.50	0.08		0.00	164.04		3
sq21	100	0	0	0	0	4.50	6.30	7.45	7.40	10.40	10.50	6735.71	6625.00	0.00		0.01	124.69		3
sq22	100	5	0	0	0	0.20	1.30	1.51	1.40	3.22	3.00	8779.18	8195.50	0.01		0.00	145.07		2
sq23	100	0	0	0	0	2.20	3.10	3.84	3.85	5.63	5.45	9278.87	8664.50	0.00		0.00	159.84		3
sq24	100	34	5	2	0	0.00	0.10	0.47	0.30	1.28	1.25	8271.60	8009.00	0.26		0.00	132.48		2
sq25	100	84	18	5	1	0.00	0.00	0.06	0.00	0.86	0.80	6895.81	6451.50	0.42		0.01	117.67		0
sq26	100	29	1	0	0	0.10	0.50	0.35	0.30	1.92	1.80	5791.33	5159.00	0.04		0.00	93.58		0
sq27	100	11	0	0	0	0.00	0.10	1.05	1.00	1.90	1.75	9275.04	8329.50	0.12		0.00	168.69		2
sq28	100	74	0	0	0	0.00	0.90	0.14	0.00	3.08	3.30	6816.69	5996.00	0.11		0.00	110.12		0
sq29	100	58	0	0	0	0.00	0.30	0.24	0.00	2.41	2.50	6148.43	5642.00	0.38		0.00	96.53		1
sq30	100	31	0	0	0	0.00	0.90	0.71	0.50	2.96	2.90	8032.09	7731.00	0.06		0.00	164.35		3
sq31	100	2	0	0	0	0.10	1.20	1.53	1.35	3.23	3.15	7308.63	6979.00	0.01		0.00	140.47		2
sq32	100	48	7	3	0	0.00	0.10	0.37	0.15	1.50	1.30	5720.23	5626.50	0.11		0.00	92.73		0
sq33	100	51	2	0	0	0.00	0.40	0.30	0.00	1.45	1.30	7065.44	6734.00	0.09		0.00	143.89		3
sq34	100	0	0	0	0	3.00	3.70	5.83	5.65	7.70	7.80	8253.23	7950.00	0.00		0.00	198.48		3
sq35	100	9	0	0	0	0.00	0.60	1.30	1.10	3.67	3.55	8194.17	7667.50	0.05		0.00	169.42		1
sq36	100	95	2	0	0	0.00	0.30	0.01	0.00	1.41	1.35	7365.93	6961.00	0.20		0.00	137.03		0
sq37	100	0	0	0	0	2.40	3.60	6.38	6.55	9.58	9.70	6678.52	6130.00	0.00		0.01	168.64		5
sq38	100	58	2	0	0	0.00	1.90	0.16	0.00	2.84	2.60	6538.08	6166.50	0.10		0.00	133.12		2
sq39	100	6	0	0	0	0.20	0.70	1.54	1.20	3.11	3.00	8281.63	7869.50	0.06		0.00	212.28		2
sq40	100	63	15	0	0	0.00	1.60	0.19	0.00	2.11	2.00	5824.55	5322.50	0.21		0.00	127.14		0
sq41	100	37	0	0	0	0.00	0.90	0.54	0.40	3.39	3.25	7220.99	6520.00	0.06		0.00	182.06		2
sq42	100	0	0	0	0	2.30	2.80	3.89	3.75	6.67	6.65	12878.88	12269.50	0.00		0.00	289.70		5
sq43	100	23	0	0	0	0.60	1.20	0.74	0.70	3.02	2.90	6901.26	6801.00	0.03		0.00	166.81		1
sq44	100	6	0	0	0	0.10	0.90	1.30	1.20	2.64	2.55	6531.29	6417.50	0.09		0.00	156.84		1
sq45	100	17	0	0	0	0.00	0.20	0.62	0.50	1.44	1.45	6637.74	5920.00	0.30		0.00	147.20		0
sq46	100	2	0	0	0	1.00	1.20	1.97	1.95	3.13	3.05	7283.80	6953.00	0.02		0.01	181.38		1
sq47	100	63	0	0	0	0.00	0.70	0.12	0.00	2.08	2.30	7732.64	6780.00	0.22		0.00	178.27		0
sq48	100	1	0	0	0	2.00	4.30	5.07	5.20	8.65	8.70	9143.70	8535.50	0.00		0.00	249.03		4
sq49	100	53	9	0	0	0.00	0.20	0.23	0.00	0.98	0.75	6428.33	5834.00	0.48		0.01	151.69		0

sq50	100	26	0	0	0	0.00	0.20	0.51	0.50	2.23	2.40	6363.72	5870.50	0.12	0.00	144.42	0
sq51	100	0	0	0	0	4.50	8.30	10.03	9.75	14.70	14.75	7018.94	6673.00	0.00	87.27	233.70	12
sq52	100	1	0	0	0	1.70	3.10	2.65	2.50	7.41	7.50	8101.18	7775.50	0.00	0.00	241.83	4
sq53	100	79	4	0	0	0.00	0.10	0.10	0.00	1.09	1.00	7633.86	7340.50	0.55	0.00	187.27	1
sq54	100	0	0	0	0	2.30	3.60	4.73	4.80	7.10	7.10	9548.92	9327.00	0.00	0.05	246.69	6
sq55	100	5	0	0	0	0.10	1.50	1.29	1.10	3.00	2.80	10526.75	10247.00	0.03	0.00	256.75	1
sq56	100	0	0	0	0	2.10	2.60	3.80	3.65	5.83	5.85	9434.16	9225.00	0.00	0.00	247.34	3
sq57	100	48	2	0	0	0.20	0.80	0.29	0.15	2.11	2.20	6805.85	6176.50	0.19	0.00	165.91	1
sq58	100	0	0	0	0	0.10	0.80	2.51	2.20	4.04	3.90	9235.33	9054.50	0.02	0.00	236.55	1
sq59	100	2	0	0	0	0.00	1.80	3.16	3.20	5.17	5.00	10804.93	10002.00	0.06	0.02	262.15	5
sq60	100	0	0	0	0	3.40	4.10	6.42	6.20	8.36	8.05	9291.52	8451.00	0.00	0.00	237.74	5
sq61	100	3	0	0	0	0.40	0.60	1.37	1.30	2.47	2.20	6120.63	5988.00	0.04	0.00	162.91	1
sq62	100	3	0	0	0	0.90	1.50	2.41	2.50	5.50	5.35	9477.27	8898.00	0.02	0.00	240.81	4
sq63	100	85	3	2	0	0.00	0.10	0.06	0.00	1.05	1.00	6974.64	6704.00	0.27	0.00	160.11	1
sq64	100	0	0	0	0	0.30	0.80	4.82	4.90	6.72	6.60	7698.47	7467.00	0.02	0.00	214.51	4
sq65	100	2	0	0	0	1.10	2.00	2.56	2.55	5.09	5.10	8350.45	7688.00	0.01	0.00	231.70	3
sq66	100	11	0	0	0	0.00	1.20	1.43	1.20	3.40	3.10	7893.92	7655.50	0.09	0.00	213.50	2
sq67	100	0	0	0	0	1.70	4.30	4.91	4.80	9.17	9.10	7524.53	7470.00	0.00	0.01	225.66	5
sq68	100	3	0	0	0	0.50	2.40	2.93	2.70	6.89	6.85	8046.23	7704.50	0.00	0.01	234.49	4
sq69	100	0	0	0	0	0.60	6.90	8.46	8.50	12.67	12.25	6120.45	5843.50	0.00	6.00	197.66	10
sq70	100	86	26	1	0	0.00	0.40	0.04	0.00	1.27	1.00	5584.74	5475.00	0.29	0.00	134.45	0
sq71	100	78	11	0	0	0.00	1.10	0.08	0.00	3.07	2.90	6819.99	6293.00	0.07	0.00	168.24	0
sq72	100	25	2	0	0	0.10	0.10	0.74	0.60	1.68	1.55	6050.92	5871.00	0.05	0.00	152.68	1
sq73	100	7	0	0	0	0.00	1.70	1.62	1.50	3.75	3.55	11254.62	10838.00	0.16	0.00	281.52	3
sq74	100	4	0	0	0	0.50	1.00	1.73	1.65	4.21	4.20	11157.47	10327.50	0.04	0.00	276.41	3
sq75	100	20	1	0	0	0.10	0.30	0.86	0.80	2.43	2.30	8990.84	8706.00	0.08	0.00	228.23	2
sq76	100	0	0	0	0	0.80	3.60	4.46	4.60	7.12	7.05	7761.07	7004.50	0.00	0.00	219.19	3
sq77	100	1	0	0	0	0.00	0.90	2.25	2.20	3.74	3.80	7804.50	7213.00	0.04	0.00	200.62	1
sq78	100	1	0	0	0	1.40	4.50	4.13	4.00	9.38	9.10	9087.38	8932.50	0.00	5.59	252.72	10
sq79	100	5	0	0	0	2.10	2.60	2.34	2.20	6.37	6.30	8310.17	7803.50	0.00	0.01	248.67	5
sq80	100	99	1	0	0	0.00	0.40	0.00	0.00	0.86	0.80	6328.88	5673.50	0.29	0.00	151.96	0
sq81	100	0	0	0	0	3.70	5.90	4.85	4.50	10.24	10.50	8385.73	7724.00	0.00	0.01	239.02	4
sq82	100	86	12	5	2	0.00	0.00	0.06	0.00	0.84	0.70	6291.30	6136.00	0.17	0.00	156.94	1
sq83	100	26	2	0	0	0.00	0.30	0.62	0.45	1.56	1.50	7059.81	6874.50	0.09	0.00	158.64	1
sq84	100	11	3	2	0	0.00	0.20	0.81	0.70	1.83	1.75	5831.79	5639.00	0.19	0.00	153.44	2
sq85	100	0	0	0	0	0.10	2.50	3.09	3.00	5.44	5.15	6862.03	6648.50	0.03	0.00	193.01	2
sq86	100	6	0	0	0	0.20	0.50	1.62	1.50	2.78	2.80	8441.08	8123.50	0.08	0.00	211.64	2
sq87	100	33	1	0	0	0.10	1.00	0.37	0.20	2.98	2.90	8004.00	7104.00	0.12	0.00	190.71	1
sq88	100	19	2	0	0	1.00	1.60	0.83	0.70	3.04	2.90	6663.26	6216.50	0.07	0.00	154.29	1
sq89	100	63	5	1	0	0.00	0.10	0.19	0.00	1.09	1.00	5485.78	5196.00	0.17	0.01	138.94	2
sq90	100	73	6	1	0	0.20	0.60	0.09	0.00	2.26	2.20	6627.58	6184.50	0.12	0.00	148.26	1
sq91	100	49	6	1	0	0.00	0.20	0.38	0.10	1.58	1.40	9550.33	9248.50	0.06	0.00	245.08	4
sq92	100	0	0	0	0	3.30	7.70	7.12	7.05	12.43	12.45	7151.99	6737.50	0.00	0.03	133.44	6
sq93	100	2	0	0	0	1.10	1.40	1.99	1.90	4.91	4.80	9986.64	9377.50	0.02	0.00	163.46	2
sq94	100	35	4	0	0	0.00	0.20	0.41	0.20	1.32	1.30	6049.44	5813.00	0.19	0.01	91.66	0
sq95	100	0	0	0	0	5.80	7.60	9.06	9.05	13.01	13.20	6065.87	5951.00	0.00	1.44	112.62	9
sq96	100	16	0	0	0	0.00	0.40	0.68	0.60	2.12	2.05	7191.67	6812.00	0.05	0.00	101.05	2

sq97	100	67	1	0	0	0.00	0.20	0.22	0.00	2.30	2.20	5925.38	5806.00	0.16	0.00	83.26	2
sq98	100	84	0	0	0	0.00	0.80	0.04	0.00	2.11	2.10	6817.99	6247.00	0.18	0.00	95.28	2
sq99	100	25	13	3	3	0.00	0.00	0.42	0.40	0.70	0.70	6207.71	6054.00	0.12	0.00	84.98	0
sq100	100	1	0	0	0	0.00	2.60	2.79	2.80	5.84	5.70	11922.50	11469.50	0.01	0.00	206.90	3
μ		27	2	0	0	0.75	1.69							0.10			
$\tilde{x}$		10	0	0	0	0.10	0.90							0.06			

Supplementary Table S5: Detailed results of two-target pseudknot design inputs (LE80 dataset).

RNA	l	n1	n2	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim
PKB00002 PKB00004 0	50	11	5	0.00	0.00	0.57	0.15	0.71	0.35	635.77	638.50	0.24	0.00	3.41	0
PKB00005 PKB00015 0	41	0	0	0.60	0.60	1.42	1.40	1.63	1.70	568.57	534.50	0.10	0.00	3.03	0
PKB00008 PKB00031 0	40	0	0	0.20	0.40	1.34	1.10	1.67	1.40	656.43	579.50	0.19	0.01	3.46	0
PKB00010 PKB00066 0	40	12	5	0.00	0.00	0.52	0.20	0.70	0.45	593.07	573.50	0.38	0.00	3.11	0
PKB00012 PKB00268 0	40	8	4	0.00	0.00	0.85	0.65	0.95	0.70	453.80	399.00	0.05	0.00	2.39	0
PKB00030 PKB00045 0	41	0	0	0.70	0.90	1.64	1.40	1.85	1.60	588.83	529.00	0.19	0.01	3.08	0
PKB00047 PKB00069 0	61	0	0	3.00	3.30	5.61	5.45	5.80	5.70	487.47	449.00	0.00	0.00	2.61	0
PKB00048 PKB00265 0	61	0	0	1.10	1.20	3.40	3.10	6670.27	3.65	517.87	452.00	0.01	0.00	2.75	0
PKB00050 PKB00128 0	59	10	4	0.00	0.00	0.65	0.40	0.93	0.75	518.03	481.50	0.15	0.00	2.71	0
PKB00052 PKB00107 0	52	4	1	0.00	0.00	6667.73	0.85	6667.99	1.15	437.93	385.50	0.25	0.00	2.33	0
PKB00057 PKB00072 0	67	0	0	1.30	2.10	5.41	5.30	5.69	5.50	495.33	436.00	0.00	0.00	2.59	0
PKB00068 PKB00129 0	68	0	0	2.60	3.30	5.34	5.10	5.65	5.45	681.17	640.50	0.00	0.00	3.41	0
PKB00070 PKB00244 0	55	2	0	0.00	0.10	1.91	1.75	2.32	1.90	481.70	394.50	0.17	0.00	2.52	0
PKB00078 PKB00106 0	62	4	0	0.00	0.10	1.17	1.00	2.25	1.90	525.20	466.50	0.24	0.00	2.77	0
PKB00080 PKB00132 0	49	10	4	0.00	0.00	0.50	0.40	0.70	0.60	407.60	407.50	0.07	0.00	2.11	0
PKB00088 PKB00127 0	62	10	1	0.00	0.00	0.74	0.55	1.54	1.15	676.53	658.00	0.25	0.00	3.58	0
PKB00098 PKB00232 0	62	1	0	0.00	0.40	2.53	2.65	2.80	2.80	609.90	580.00	0.04	0.00	3.64	0
PKB00131 PKB00205 0	48	0	0	1.70	2.20	3.02	3.00	4.16	3.90	570.73	567.50	0.01	0.00	3.35	0
PKB00139 PKB00141 0	70	0	0	1.50	1.60	3.17	3.00	3.27	3.20	745.73	662.00	0.01	0.00	3.89	0
PKB00142 PKB00231 0	71	1	0	0.00	0.80	2.86	2.55	3.33	2.90	498.97	468.50	0.05	0.00	2.75	0
PKB00143 PKB00233 0	71	0	0	1.40	1.50	13337.10	3.70	13337.27	3.75	603.03	577.00	0.01	0.00	3.22	2
PKB00148 PKB00218 0	72	0	0	3.30	3.60	5.52	4.95	5.77	5.10	642.50	571.00	0.00	0.00	3.35	0
PKB00175 PKB00259 0	57	0	0	0.30	0.50	1.69	1.65	1.98	1.90	648.77	643.00	0.08	0.00	3.95	0
PKB00179 PKB00280 0	68	0	0	0.60	0.60	2.61	2.70	2.90	2.90	565.13	587.50	0.01	0.00	3.05	0
PKB00180 PKB00212 0	64	0	0	0.30	0.40	6670.49	3.05	20003.69	3.65	448.53	402.50	0.13	0.00	2.62	0
PKB00190 PKB00266 0	47	21	7	0.00	0.00	0.18	0.00	0.34	0.20	534.43	530.00	0.29	0.00	3.19	0
PKB00207 PKB00213 0	45	7	1	0.00	0.00	13334.10	0.60	13334.26	0.85	364.37	339.50	0.26	0.00	2.17	0
PKB00211 PKB00239 0	80	0	0	0.80	1.10	4.15	3.75	4.62	4.50	486.80	464.00	0.02	0.00	2.27	0
PKB00222 PKB00305 0	80	0	0	0.50	0.90	6670.15	3.35	6670.54	3.85	595.60	592.00	0.02	0.00	2.61	0
PKB00224 PKB00281 0	43	9	3	0.00	0.00	0.70	0.55	1.01	0.75	513.63	485.00	0.19	0.00	3.05	0
PKB00230 PKB00273 0	48	0	0	2.00	2.50	4.11	4.10	6671.29	4.65	374.43	353.00	0.00	0.00	2.19	0
PKB00248 PKB00257 0	66	0	0	4.40	6.70	6675.06	8.55	33343.25	10.90	214.17	221.00	0.00	0.00	1.20	0
PKB00263 PKB00270 0	62	6	1	0.00	0.00	13334.31	0.95	13334.51	1.15	620.23	629.50	0.16	0.00	3.11	0

PKB00269	PKB00272	0	66	0	0	1.50	2.30	6670.63	3.80	20004.32	4.40	444.57	411.00	0.00	0.00	1.25	0
μ				3	1	0.82	1.09							0.11			
$\tilde{x}$				0	0	0.30	0.55							0.06			

Supplementary Table S6: Detailed results of two-target pseudknot design inputs (PK60 dataset).

RNA	l	n1	n2	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim
no1	60	0	0	0.20	0.20	1.64	1.35	1.82	1.60	620.13	659.00	0.11	0.00	3.26	0
no2	60	2	0	0.20	0.30	1.48	1.55	1.87	1.70	1015.93	1008.50	0.06	0.00	6.12	0
no3	60	17	5	0.00	0.00	0.39	0.00	0.65	0.20	806.70	707.00	0.12	0.00	4.85	2
no4	60	11	5	0.00	0.00	0.58	0.20	0.70	0.30	810.63	769.00	0.14	0.00	4.44	0
no5	60	3	1	0.00	0.00	1.20	1.30	1.41	1.40	696.63	686.50	0.08	0.00	3.99	0
no6	60	2	0	0.00	0.30	1.16	0.95	1.40	1.30	787.03	736.00	0.15	0.00	4.29	0
no7	60	29	18	0.00	0.00	0.04	0.00	0.11	0.00	929.33	897.50	0.45	0.00	4.70	0
no8	60	0	0	0.30	0.30	2.50	2.30	2.72	2.60	647.67	651.00	0.06	0.00	3.32	0
no9	60	27	11	0.00	0.00	0.03	0.00	0.16	0.10	908.17	887.00	0.27	0.00	2.53	0
no10	60	21	9	0.00	0.00	0.15	0.00	0.31	0.10	944.97	917.00	0.39	0.01	5.07	0
no11	60	27	15	0.00	0.00	0.05	0.00	0.17	0.05	862.27	806.00	0.41	0.00	4.65	0
no12	60	17	3	0.00	0.00	0.38	0.00	0.61	0.55	779.47	779.00	0.33	0.00	4.24	0
no13	60	8	1	0.00	0.00	0.45	0.25	0.56	0.40	784.07	791.00	0.17	0.00	4.24	0
no14	60	28	14	0.00	0.00	0.03	0.00	0.11	0.10	852.70	798.50	0.28	0.00	4.59	0
no15	60	3	1	0.00	0.00	1.31	1.25	1.57	1.50	615.13	595.00	0.18	0.00	3.31	0
no16	60	7	2	0.00	0.00	0.60	0.45	0.77	0.55	729.23	736.00	0.17	0.00	3.95	0
no17	60	0	0	0.20	0.50	1.98	1.90	2.17	2.15	657.07	619.00	0.06	0.00	3.50	0
no18	60	25	7	0.00	0.00	0.15	0.00	0.32	0.20	719.77	717.50	0.46	0.00	3.85	0
no19	60	14	3	0.00	0.00	0.62	0.10	0.89	0.55	784.93	683.50	0.27	0.00	4.20	0
no20	60	4	0	0.00	0.10	1.14	0.85	1.44	1.30	652.77	640.50	0.19	0.00	3.49	0
no21	60	3	1	0.00	0.00	1.71	1.70	1.94	1.85	592.00	576.50	0.09	0.00	3.11	0
no22	60	23	7	0.00	0.00	0.10	0.00	0.27	0.20	880.83	885.50	0.31	0.00	4.65	0
no23	60	30	15	0.00	0.00	0.00	0.00	0.11	0.05	881.77	860.50	0.75	0.00	4.65	0
no24	60	3	0	0.00	0.20	1.21	1.20	1.43	1.35	532.30	486.50	0.37	0.00	2.81	0
no25	60	20	15	0.00	0.00	0.13	0.00	0.18	0.05	841.13	822.00	0.27	0.00	5.12	0
no26	60	28	11	0.00	0.00	0.04	0.00	0.14	0.10	845.93	900.50	0.44	0.00	5.18	0
no27	60	7	1	0.00	0.00	1.44	1.30	1.92	1.45	633.67	625.00	0.16	0.00	3.90	0
no28	60	27	4	0.00	0.00	0.02	0.00	0.26	0.15	902.00	877.50	0.45	0.00	5.45	0
no29	60	17	5	0.00	0.00	0.36	0.00	0.50	0.30	639.13	604.50	0.06	0.00	3.95	0
no30	60	22	6	0.00	0.00	0.15	0.00	0.30	0.20	672.63	652.00	0.14	0.00	4.07	0
no31	60	1	0	0.00	0.20	1.84	1.25	2.06	1.55	696.03	704.00	0.07	0.00	4.25	0
no32	60	23	3	0.00	0.00	0.14	0.00	0.30	0.20	905.87	876.00	0.32	0.00	5.45	0
no33	60	12	4	0.00	0.00	0.47	0.40	0.63	0.55	652.57	589.00	0.41	0.01	4.01	0
no34	60	9	4	0.00	0.00	0.75	0.50	0.86	0.55	528.40	475.50	0.14	0.00	3.27	0
no35	60	21	8	0.00	0.00	0.21	0.00	0.39	0.40	844.40	802.00	0.46	0.00	5.18	0
no36	60	9	1	0.00	0.00	0.59	0.40	0.83	0.55	805.07	812.00	0.33	0.00	4.69	0

no37	60	0	0	0.60	1.00	1.88	1.80	2.16	2.00	937.03	1000.50	0.03	0.00	5.69	0
no38	60	1	1	0.00	0.00	1.55	1.25	1.79	1.55	647.67	640.00	0.24	0.00	3.96	0
no39	60	30	18	0.00	0.00	0.00	0.00	0.06	0.00	865.63	862.00	0.48	0.00	5.33	0
no40	60	24	15	0.00	0.00	0.13	0.00	0.27	0.05	1026.00	1072.00	0.27	0.00	6.07	0
no41	60	22	9	0.00	0.00	0.13	0.00	0.34	0.10	810.83	743.00	0.22	0.00	4.89	0
no42	60	26	7	0.00	0.00	0.09	0.00	0.22	0.10	885.23	835.00	0.45	0.00	5.27	0
no43	60	2	0	0.00	0.80	2.10	1.90	2.52	2.40	539.30	485.00	0.05	0.00	3.24	0
no44	60	29	9	0.00	0.00	0.01	0.00	0.15	0.10	1038.80	1084.00	0.46	0.00	6.08	0
no45	60	13	4	0.00	0.00	0.76	0.35	1.01	0.60	828.50	831.00	0.12	0.00	4.91	0
no46	60	4	1	0.00	0.00	1.51	1.10	1.78	1.60	437.30	422.50	0.05	0.00	2.67	0
no47	60	2	0	0.00	0.10	1.90	1.80	2.14	2.05	645.33	591.00	0.03	0.00	3.87	0
no48	60	0	0	0.10	0.10	1.52	1.30	1.72	1.60	695.47	681.00	0.12	0.00	4.11	2
no49	60	18	7	0.00	0.00	0.17	0.00	0.29	0.15	878.80	829.00	0.32	0.00	4.90	0
no50	60	18	7	0.00	0.00	0.35	0.00	0.53	0.25	666.67	657.50	0.10	0.00	3.78	0
μ		14	5	0.03	0.08							0.24			
$\tilde{x}$		14	4	0.00	0.00							0.21			

Supplementary Table S7: Detailed results of two-target pseudknot design inputs (PK80 dataset).

RNA	l	n1	n2	d1	d2	μ d1	$\tilde{x}$ d1	μ d2	$\tilde{x}$ d2	μ nom	$\tilde{x}$ nom	$\sum$ prob	μ constructing	μ sampling	max dim
no1	80	3	2	0.00	0.00	1.13	1.00	1.31	1.30	871.63	915.00	0.09	0.00	4.80	0
no2	80	23	15	0.00	0.00	0.15	0.00	0.20	0.05	846.37	799.00	0.17	0.00	4.74	0
no3	80	28	4	0.00	0.00	0.05	0.00	0.19	0.10	1178.70	1119.50	0.22	0.00	6.33	0
no4	80	22	8	0.00	0.00	0.18	0.00	0.34	0.10	1007.97	940.00	0.17	0.00	5.56	0
no5	80	12	7	0.00	0.00	0.64	0.25	0.74	0.40	1079.90	1108.50	0.27	0.00	5.99	0
no6	80	25	18	0.00	0.00	0.20	0.00	0.25	0.00	971.53	933.50	0.10	0.01	5.34	0
no7	80	5	1	0.00	0.00	0.96	0.80	1.12	0.95	833.23	816.00	0.12	0.00	4.70	0
no8	80	1	0	0.00	0.20	1.80	1.45	2.00	1.70	847.50	803.00	0.14	0.00	4.80	0
no9	80	20	14	0.00	0.00	0.29	0.00	0.34	0.10	931.57	895.50	0.19	0.00	2.43	0
no10	80	20	8	0.00	0.00	0.27	0.00	0.38	0.15	925.50	929.00	0.26	0.00	5.17	0
no11	80	29	10	0.00	0.00	0.01	0.00	0.15	0.10	1083.83	1006.00	0.21	0.00	6.01	0
no12	80	30	20	0.00	0.00	0.00	0.00	0.06	0.00	1006.93	959.00	0.42	0.00	5.59	0
no13	80	30	16	0.00	0.00	0.00	0.00	0.08	0.00	1114.80	1123.00	0.41	0.00	6.15	0
no14	80	10	3	0.00	0.00	0.59	0.55	0.77	0.65	1087.57	1057.00	0.22	0.00	6.01	0
no15	80	25	5	0.00	0.00	0.08	0.00	0.18	0.10	1098.63	1064.50	0.14	0.00	6.02	0
no16	80	27	9	0.00	0.00	0.02	0.00	0.18	0.10	971.37	941.50	0.31	0.00	5.38	0
no17	80	8	4	0.00	0.00	0.62	0.55	0.73	0.65	1064.17	1107.50	0.12	0.00	5.86	0
no18	80	18	8	0.00	0.00	0.29	0.00	0.39	0.20	842.50	760.50	0.10	0.00	4.63	0
no19	80	29	13	0.00	0.00	0.00	0.00	0.08	0.10	1164.00	1130.50	0.24	0.00	6.42	0
no20	80	17	7	0.00	0.00	0.31	0.00	0.40	0.20	1178.23	1212.50	0.24	0.00	6.38	0
no21	80	1	0	1.10	1.30	6671.48	4.45	6671.74	4.60	707.30	652.00	0.01	0.00	3.84	0
no22	80	7	1	0.00	0.00	0.85	0.60	1.09	0.85	1144.57	1137.50	0.25	0.00	6.24	0
no23	80	29	13	0.00	0.00	0.04	0.00	0.18	0.10	1181.20	1121.00	0.53	0.00	6.40	0

no24	80	7	2	0.00	0.00	0.98	0.60	1.16	0.75	835.43	815.00	0.29	0.00	4.51	0
no25	80	7	2	0.00	0.00	1.04	1.10	1.43	1.40	776.83	709.50	0.31	0.00	4.57	0
no26	80	0	0	3.30	3.40	5.81	5.55	6.51	5.95	531.07	508.50	0.00	0.00	3.27	0
no27	80	1	0	0.00	0.20	2.92	2.65	3.37	3.30	979.33	905.50	0.12	0.00	5.75	0
no28	80	24	12	0.00	0.00	0.15	0.00	0.34	0.10	1146.77	1176.50	0.10	0.00	6.71	0
no29	80	21	7	0.00	0.00	0.18	0.00	0.31	0.15	989.97	974.50	0.12	0.00	5.50	0
no30	80	14	3	0.00	0.00	0.30	0.10	0.58	0.35	965.67	980.00	0.22	0.01	5.35	0
μ		16	7	0.15	0.17							0.20			
$\tilde{x}$		19	7	0.00	0.00							0.20			

References

[1] M. Andronescu, A. P. Fejes, F. Hutter, H. H. Hoos, and A. Condon. A new algorithm for RNA secondary structure design. *J Mol Biol*, 336(3):607–624, Feb 2004.

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

[3] A. Busch and R. Backofen. INFO-RNA – a fast approach to inverse RNA folding. *Bioinformatics*, 22(15), Aug. 2006.

[4] K. Darty, A. Denise, and Y. Ponty. VARNA: Interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, 25(15):1974–1975, Aug 2009.

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

[6] A. Esmaili-Taheri, M. Ganjtabesh, and M. Mohammad-Noori. Evolutionary solution for the RNA design problem. *Bioinformatics*, 30(9):1250–1258, Jan 2014.

[7] A. Espah-Borujeni, D. M. Mishler, J. Wang, W. Huso, and H. M. Salis. Automated physics-based design of synthetic riboswitches from diverse RNA aptamers. *Nucleic Acids Research*, page gkv1289, Nov 2015.

[8] C. Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7(2):254–265, Feb. 2001.

[9] J. A. Garcia-Martin, P. Clote, and I. Dotu. RNAiFOLD: a constraint programming algorithm for rna inverse folding and molecular design. *J Bioinform Comput Biol*, 11(2):1350001, Apr 2013.

[10] J. A. Garcia-Martin, P. Clote, and I. Dotu. RNAiFold: a web server for rna inverse folding and molecular design. *Nucleic Acids Res*, 41(Web Server issue):W465–W470, Jul 2013.

[11] J. A. Garcia-Martin, I. Dotu, and P. Clote. RNAiFold 2.0: a web server and software to design custom and rfam-based RNA molecules. *Nucleic Acids Research*, 43(W1):W513–W521, may 2015.

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

[13] C. Höner zu Siederdissen, S. Hammer, I. Abfalter, I. L. Hofacker, C. Flamm, and P. F. Stadler. Computational design of RNAs with complex energy landscapes. *Biopolymers*, 99(12):1124–1136, 2013.

[14] R. Kleinkauf, T. Houwaart, R. Backofen, and M. Mann. antaRNA – Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. *BMC Bioinformatics*, 16, 2015.

[15] R. Kleinkauf, M. Mann, and R. Backofen. antaRNA: ant colony-based RNA sequence design. *Bioinformatics*, 31(19):31143121, May 2015.

[16] J. Lee, W. Kladwang, M. Lee, D. Cantu, M. Azizyan, H. Kim, A. Limpaecher, S. Yoon, A. Treuille, and R. Das. RNA design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, pages 2122–2127, Jan 2014.

[17] A. Levin, M. Lis, Y. Ponty, C. W. O'Donnell, S. Devadas, B. Berger, and J. Waldispühl. A global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Res*, 40(20):10041–10052, Nov 2012.

[18] R. B. Lyngso, J. W. Anderson, E. Sizikova, A. Badugu, T. Hyland, and J. Hein. Frnakenstein: multiple target inverse RNA folding. *BMC Bioinformatics*, 13(1):260, 2012.

[19] M. C. Matthies, S. Bienert, and A. E. Torda. Dynamics in sequence space for RNA secondary structure design. *Journal of Chemical Theory and Computation*, 8(10):3663–3670, Oct 2012.

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

[21] G. Rodrigo and A. Jaramillo. RiboMaker: computational design of conformation-based riboregulation. *Bioinformatics*, 30(17):2508–2510, may 2014.

[22] W. Shu, M. Liu, H. Chen, X. Bo, and S. Wang. ARDesigner: a web-based system for allosteric RNA design. *J Biotechnol*, 150(4):466–473, Dec 2010.

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

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

[25] B. R. Wolfe and N. A. Pierce. Sequence Design for a Test Tube of Interacting Nucleic Acid Strands. *ACS Synthetic Biology*, 4(10):1086–1100, Oct. 2015.

[26] J. N. Zadeh, B. R. Wolfe, and N. A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2011.

8

DESIGN PIPELINE FOR A LIGAND TRIGGERED RNA SWITCH

Sven Findeiß, Stefan Hammer, Michael T. Wolfinger, Felix Kühnl, Christoph Flamm, and Ivo L. Hofacker.
"In silico design of ligand triggered RNA switches.", 2018 in Elsevier Methods, accepted
doi:10.1101/245464

SUMMARY

This contribution focuses on the development and implementation of a work flow to design a ligand triggered RNA switch. Given the initial idea to design a switching molecule that is able to adapt two structural conformations in a ligand dependent way, the necessary steps of developing a valid objective function, implementing software that solves the optimization problem and analyzing the obtained candidates *in silico* are explained in detail. `RNAblueprint`, `RNAsketch` and several programs of the `ViennaRNA` package are utilized successfully to derive an RNA device able to respond to theophylline presence. Furthermore, an extensive *in silico* analysis pipeline evaluates important kinetic properties of the designed sequences. This protocol aims to be used as template for future design challenges as the developed software can be easily adapted and extended to fit novel design scenarios.

AUTHORS CONTRIBUTION

SF, SH, FK wrote the majority of the article. SF and SH developed the methods and implemented the relevant scripts, SF performed most of the calculations and analyses. FK and MW contributed with several analyses and valuable discussions. CF and IL contributed with essential scientific advise and ideas.

LICENSE

In silico design of ligand triggered RNA switches[☆]

Sven Findeiß[a,b,c,*], Stefan Hammer[a,b,c], Michael T. Wolfinger[c,d], Felix Kühnl[a], Christoph Flamm[c], Ivo L. Hofacker[b,c]

[a] *Bioinformatics, Institute of Computer Science, and Interdisciplinary Center for Bioinformatics, Leipzig University, Härtelstraße 16–18, 04107 Leipzig, Germany*
[b] *University of Vienna, Faculty of Computer Science, Research Group Bioinformatics and Computational Biology, Währingerstraße 29, 1090 Vienna, Austria*
[c] *University of Vienna, Faculty of Chemistry, Department of Theoretical Chemistry, Währingerstraße 17, 1090 Vienna, Austria*
[d] *Medical University of Vienna, Center for Anatomy and Cell Biology, Währingerstraße 13, 1090 Vienna, Austria*

Abstract

This contribution sketches a work flow to design an RNA switch that is able to adapt two structural conformations in a ligand-dependent way. A well characterized RNA aptamer, i. e., knowing its K_d and adaptive structural features, is an essential ingredient of the described design process. We exemplify the principles using the well-known theophylline aptamer throughout this work. The aptamer in its ligand-binding competent structure represents one structural conformation of the switch while an alternative fold that disrupts the binding-competent structure forms the other conformation. To keep it simple we do not incorporate any regulatory mechanism to control transcription or translation. We elucidate a commonly used design process by explicitly dissecting and explaining the necessary steps in detail. We developed a novel objective function which specifies the mechanistics of this simple, ligand-triggered riboswitch and describe an extensive *in silico* analysis pipeline to evaluate important kinetic properties of the designed sequences. This protocol and the developed software can be easily extended or adapted to fit novel design scenarios and thus can serve as a template for future needs.

Keywords: inverse folding, multi state design, RNA design, riboswitch, objective function, RNA kinetics

[☆] Issue title: Methods and Advances in RNA characterization and design
Guest Editors: Lydia M. Contreras
[*] Corresponding Author
Email address: sven@bioinf.uni-leipzig.de (Sven Findeiß)

1. Introduction

Riboswitches are highly structured RNA sequences commonly found in the 5'-untranslated region (UTR) of prokaryotic messenger RNAs (mRNAs). Within this regulatory domain, they are responsible for altering gene expression on the transcriptional or translational level in response to environmental changes, which is typically the concentration of a small ligand [1]. A riboswitch consists of two components: i) a sensory domain and ii) a regulatory domain. While the former specifically senses the environmental change, the latter is responsible for influencing the expression level of the downstream gene. Beside those switches that follow this commonly assumed two-component model, examples are known where a sensory domain alone, i. e., an aptamer, is able to alter gene expression [2, 3, 4, 5]. The possibility to encode effective sensors at RNA level makes riboswitches valuable gadgets that can directly interfere with the complex process of gene expression without the need of additional co-factors such as proteins. Here we elucidate the complex process of designing such a ligand-sensing riboswitch that, for simplicity, does not implement a specific regulation mechanism at transcriptional or translational level. The RNA sequence should "simply" adapt two alternative conformations depending on the presence or absence of a ligand. Therefore, we aim to extend an aptamer such that an alternative structural conformation is formed in the absence of the ligand.

Successful design approaches show that the problem of generating an artificial RNA sequence exhibiting a prescribed functionality needs to be formulated as a multi-step approach, including computational and experimental, analytic and constructive methods [6, 7]. Early design publications already followed such a multi-step scheme but included manual steps instead of computational methods, as there were just no computational tools available that implemented the features actually needed by the experimentalists. However, this changed over time and recently the common trend is to perform as many steps as possible with the support of advanced *in silico* methods [8].

As a first step, it is important to analyze the underlying biological system, the cellular environment and, most importantly, all the properties of the building blocks to use. With this information, it is then possible to design a model describing the functionality of the novel RNA molecule in its environment. To determine a sequence with the requested characteristics, usually an optimization problem is formalized, where the objectives are specified as constraints and a mathematical function describing various biophysical properties of the system. Obtaining a sequence compatible with constraints such as specific target structures and sequence motifs is a quite tricky task, which was solved with various methods ranging from manual design [9] to graph-theoretical coloring algorithms as implemented in *RNAblueprint* [10]. A variety of well-established optimization methods such as the Metropolis–Hastings algorithm [11, 12] or genetic algorithm based approaches [13] were used to find optimal solutions by traversing through the constrained solution space [14]. However, until now, only little effort was made to find proper objective functions. Often only some static properties of the molecule's energy landscape are used instead of more directly

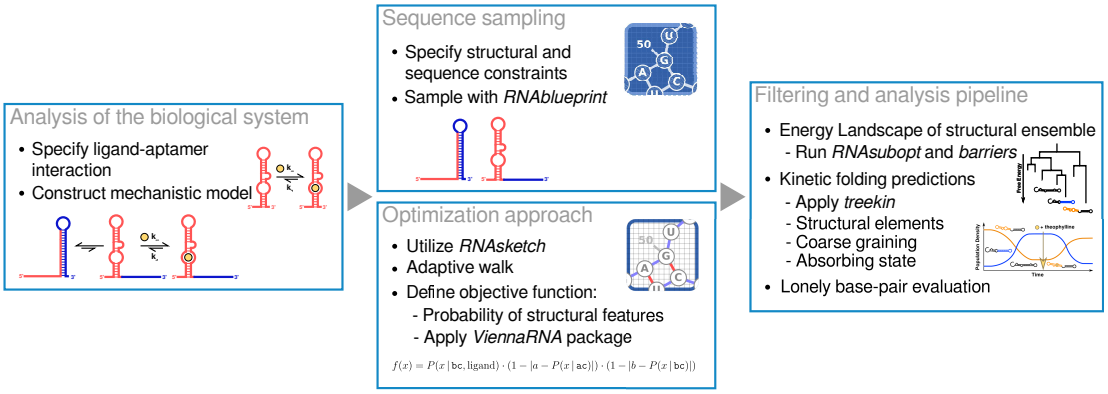


Figure 1: Graphical summary of the applied design approach. The work flow is depicted from left to right and consists of three major steps. For the individual parts important keywords and utilized software is listed and we refer to the individual sections of this contribution for more detailed information.

characterizing the mechanism of the artificial device. Only recently, some published design programs started to allow to compile an objective function from a catalog of predefined functions [15, 13]. *RNAblueprint* [10] went one step further and allowed to formulate the objective utilizing a scripting interface, which gives the user complete control over the optimization procedure.

To narrow down the number of obtained RNA sequences, a subsequent step to analyze and filter the obtained solutions was almost always performed. Essentially, the differences and advantages of various solutions are explored. The generation of proper visualizations or the evaluation of additional properties that could not be incorporated into the objective function help to perform this selection process.

Finally, it is crucial to biologically test for the desired functionality of the designed molecule as many biologically relevant aspects cannot be easily included in the objective of the optimization approach.

In this contribution, we aim to closely follow the described design steps to generate a simple, ligand-triggered riboswitch, see Figure 1. Therefore, we combine the previously published RNA design software *RNAblueprint* with analysis tools like the coarse graining program *barriers* and the kinetics simulator *treekin*, see Table 1. To achieve our goal, we propose a functional model, specify valid constraints, and develop an objective function, which directly describes the functionality of this riboswitch at an abstract level. To compute the quality measures used by our objective, we resort to the well-established thermodynamic RNA folding model implemented in the *ViennaRNA* package. An extensive *in silico* analysis pipeline evaluates important properties of the designed sequences and thus helps to narrow down and filter the list of obtained sequences that might be sent to the laboratory for biological testing. Although we describe a work flow for the purpose of generating a *specific* riboswitch, the overall result comprises the developed protocol and software, which can easily be extended or adapted to fit novel design scenarios and thus could serve as a design strategy

scaffold.

2. Materials and Methods

2.1. Specifying the design constraints

Given a model describing a desired riboswitch or functional RNA, it now needs to be converted into a machine-readable format in order to computationally generate valid sequences. Thus, the desired properties and the functionality can be expressed as a combination of constraints such as structural requirements and various properties specifying the energy landscape, and the kinetic folding properties. We specified these constraints of the functional states in the file `design_input.txt`:

```
# alternative conformation:
.......................((((((((((.....  ))))))))))...........
# binding competent conformation:
(((((...(((((((.....)))))...)))...)))))..  ......................
# sequence constraint:
AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCA NNNNNNNNNNNNNNNNNNNNN
```

These sequence and structure constraints are represented in IUPACK and dot–bracket notation, respectively. In the sequence constraint, `A`, `U`, `G` and `C` correspond to the nucleotides adenine, uracil, guanine and cytosine, respectively. To positions marked with `N`, any nucleotide can be assigned as long as they are compatible to the structural constraints, where "`.`" represents an unconstrained position and matching brackets "`()`" two positions paired with each other. Please note, constraints resulting from or overlapping with the chosen aptamer are separated by a space that needs to be removed when constraints are used as input for *RNAblueprint*, the utilized sequence sampler. We collected all applied tools in Table 1 including a short summary, the download link, citation and further remarks. *RNAblueprint* can be invoked by executing the following command:

```
$ RNAblueprint -v < design_input.txt > design_output.txt
```

This returns how many compatible sequences exist ($1.342\,18 \times 10^8$ for the given example) and, by default, ten randomly generated sequences, which are written into file `design_output.txt`. In principle, each of these sequences can fold into both specified structures, the most stable structure typically being a hybrid. Please note, that the obtained sequences are randomly generated and thus vary on every call.

For the demonstration of our analysis workflow, we selected a sequence that exhibits interesting properties, although it was not the best design generated during the applied optimization procedure. During such an optimization run thousands of compatible sequences are evaluated with respect to an objective function.

2.2. Prediction of minimum free energy structures

A transcribed RNA molecule immediately forms intra-molecular base pairs, folding into a structural conformation known as its (secondary) structure. The structure, in turn, often determines the RNA's biological function, e. g., in our case, the binding affinity for a given ligand. Any structure of a given RNA sequence can be assigned an energy value—the Gibbs free energy—and the structure expressed most likely is the one having the lowest possible energy. It is therefore called the minimum free energy (MFE) structure.

To predict the MFE of a given sequence and its associated secondary structure, we use the tool *RNAfold* included in the *ViennaRNA* package (cf. Table 1). We first store an example sequence in a text file `exa.txt` and subsequently apply *RNAfold* to it.

```
$ echo "AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGCUGGGGGAAUGUUUUUGU" \
    > exa.txt
$ cat exa.txt | RNAfold

AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGCUGGGGGAAUGUUUUUGU
...........(((((.....)))))((((((((((.....))))))))))..... (-19.70)
```

The above invocation of *RNAfold* returns, beside the input sequence, its most stable structure in dot–bracket notation and the corresponding MFE. Energies are given in $\mathrm{kcal\,mol^{-1}}$.

2.3. Modeling ligand binding with soft constraints

To incorporate the ability of binding a specific ligand into an *in silico* RNA design process, a model aware of the stabilizing contributions of this dimerization on the resulting RNA–ligand complex is required. For the work presented here, the recently implemented soft constraint framework of the *ViennaRNA* package [16] has been applied. Among other things, it allows to add an energy bonus to structural states that exhibit a certain motif. This enables for a direct integration of the effects of ligand binging into the RNA structure prediction process [16]. When evaluating the structure ensemble of a given molecule containing the theophylline aptamer sequence, an energy bonus of $\Delta G = -9.22\,\mathrm{kcal\,mol^{-1}}$ is added to every secondary structure that contains the correctly folded binding pocket. This value is obtained from the relation $\Delta G = R \times T \times \ln K_d$ for the gas constant $R = 1.987\,17\,\mathrm{cal\,mol^{-1}}$, the temperature $T = 310.15\,\mathrm{K}$, and the experimentally measured dissociation constant $K_d = 0.32\,\mu\mathrm{M}$ [17]. Using the example sequence and the `--motif` option of *RNAfold*,

```
$ cat exa.txt | RNAfold -p \
  --motif="GAUACCAG&CCCUUGGCAGC,(...((((&)...)))...),-9.22"

AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGCUGGGGGAAUGUUUUUGU
.((((...(((((((((.....)))))...)))...))))(((((...)))).............. (-21.92)
,((((...(((((((((,...,)))))...)))...))))|(((...})),.............. [-23.32]
.((((...(((((((((.....)))))...)))...)))).(((...))).............. {-12.20 d=4.04}
 frequency of mfe structure in ensemble 0.103202; ensemble diversity 6.30
```

the MFE structure now contains the binding-competent aptamer fold with a corrected energy value (cf. first row after the sequence). As a result from using the `-p` option, a condensed representation of the base pair probabilities of each nucleotide in the ensemble with the Gibbs free energy of the soft-constrained ensemble $G(x \mid s)$ (second row) as well as the centroid structure, i. e., the consensus structure of all base pairs with a probability higher than 50% in the ensemble [18], and its free energy (third row) are printed.

2.4. Obtain the probability of structural features

In a design process, it is usually desired to enforce the presence or absence of certain structural motifs, or even requires a certain sub-structure to be present with a specific probability. This requires a method that can determine the fraction of structures of a given RNA sequence that contain a given motif. An objective function can then use this information to compute probabilities of motifs and accordingly select sequences suitable for the design goal.

Hard constraints of the *ViennaRNA* package are well suited for such tasks. They allow to restrict the conformations of an RNA to states containing a combination of unconstrained bases "`.`", bases that have to be unpaired "`x`", bases that have to be paired no matter to which binding partner "`|`" and base pairs indicated by matching brackets "`()`". It is furthermore possible to specify if a base has to be paired with a binding partner up- or downstream by "`<`" and "`>`", respectively. Note, that structures lacking some constraints are also counted as long as no base pair conflicts with the constraint. To only include structures possessing all specified base pairs, use the `--enforceConstraint` option. To calculate the probability of the alternative conformation, one can use the following constraint and command:

```
constraint.txt
AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGCUGGGGGAAUGUUUUUGU
.......................((((((((((.....))))))))))...........


$ cat constraint.txt | RNAfold -C -p --canonicalBPonly\
  --motif="GAUACCAG&CCCUUGGCAGC,(...((((&)...)))...),-9.22"

AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGCUGGGGGAAUGUUUUUGU
............((((.....))))((((((((((.....))))))))))........... (-19.50)
....,.......((((|...,))))((((((((((.....))))))))))........... [-20.19]
............((((.....))))((((((((((.....))))))))))........... {-19.50 d=3.89}
 frequency of mfe structure in ensemble 0.32628; ensemble diversity 6.02
```

This performs a constrained (`-C`) partition function (`-p`) fold simulating ligand binding (`--motif`). The `--canonicalBPonly` option removes non-canonical base pairs, e. g., U-U, from the structure constrain if they where erroneously added. Here, all structures of a sequence x containing only base pairs compatible to the hard constraint h and the soft constraint s are considered during the calculation. The Gibbs free energy $G(x \mid h, s)$ of those structures can then

be used to calculate the frequency of the constrained sub-structure within the complete ensemble, which is denoted as

$$P(x \mid h, s) = \exp\left(-\frac{G(x \mid h, s) - G(x \mid s)}{RT}\right). \quad (1)$$

For more background information on the relation of these energies and the probabilities, please refer to subsection 2.6. Re-running the last command without the `-C` option yields the Gibbs free energy $G(x \mid s)$ without the hard constraint h. This will include all suboptimal structures in the calculation, but still uses the soft constraint option to model ligand binding (cf. subsection 2.3). For the example above, $G(x \mid s)$ and $G(x \mid h, s)$ are $-23.32\,\mathrm{kcal\,mol^{-1}}$ and $-20.19\,\mathrm{kcal\,mol^{-1}}$, respectively, and the resulting probability is

$$P(x \mid h, s) = \exp\left(-\frac{-20.19 + 23.32}{310.15 \times 1.98717}\right) \approx 0.01.$$

The frequency of a structural motif in the absence of any ligand can be obtained by running both commands without the `--motif` option. Thus, we can also calculate $P(x \mid h)$ as denoted in Equation 1.

2.5. Enumerating suboptimal structures of an RNA molecule

To analyze the *kinetics* of an RNA molecule, at least a part of its structural ensemble needs to be explicitly constructed. This is a challenging task as the number of structures even for small RNAs is enormous. The tool *RNAsubopt* (cf. Table 1) can be applied to generate all structures a given sequence can adopt up to a given energy threshold. Consider the following example:

```
seq1.txt
AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGUUGAGGGGGCUCAAUGAC

$ cat seq1.txt | RNAsubopt -e 1.2 -s

AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGUUGAGGGGGCUCAAUGAC -21.60   1.20
..............(((((((.((((((((((.....)))))))))).).)))).))) -21.60
....(((.....)))(((((((.((((((((((.....)))))))))).).)))).))) -21.50
(((((...(((((((.....)))))...)))...)))))..(((((((((....))))))))) -21.10
(((((...(((((((...)))))))...)))...)))))..(((((((((....))))))))) -20.80
.((((...(((((((.....)))))...)))...)))))...(((((((((....))))))))) -20.80
..(((.......))).(((((((.((((((((((.....)))))))))).).)))).))) -20.60
.((((...(((((((...)))))))...)))...)))))...(((((((((....))))))))) -20.50
..((.......))...(((((((.((((((((((.....)))))))))).).)))).))) -20.50
...(((..((.(((((.....)))))(((((((((.....)))))))))..)))..... -20.40
```

where all suboptimal structures with an energy at most $1.2\,\mathrm{kcal\,mol^{-1}}$ above the MFE are generated. Note that the number of generated structures grows exponentially with both the sequence length and the size of the selected energy band. Thus, larger instances of these calculations do not only consume CPU time, but also generate files of several gigabytes in size. To reduce the number of generated sequences, it is possible to skip all structures containing *lonely base*

pairs, i. e., helices of length one, by applying the `--noLP` option of *RNAsubopt*, cf. subsection 2.6.

As the sorting routine applied by *RNAsubopt* (`-s` option) might fail on huge instances even on high memory machines with giga- or even terabytes of RAM, a workaround is to pipe the *RNAsubopt* output to Unix's *sort* tool. The latter scales much better with the memory consumption of the typically huge ensemble sizes. The following example generates the full *RNAsubopt* outputs. The execution of this command may take some time as the *RNAsubopt* output of the example sequence is approximately 16 GB in size.

```
$ cat seq1.txt | RNAsubopt -e 22.60 | sort -k2,2n -k1,1r -S20G > seq1.sub
```

Here, the main memory buffer allocated by *sort* is set to 20 GB. Above this threshold, *sort* will dump data to temporary files on the hard drive. Assuming enough disk space is available, this still implies performance loss but makes it possible to process even huge *RNAsubopt* output. We estimated the energy band width to use for the `-e` option by folding the sequence with *RNAfold* (cf. subsection 2.2), setting its value to $-1 \times \mathrm{MFE} + 1$ to convert the minimum free energy (MFE) into a positive value and also take a few structures with positive energies into account. The obtained file `seq1.sub` contains a list of all possible structures within 22.60 kcal mol^{-1}, sorted by ascending energy values.

2.6. Assessing the impact of avoiding lonely pairs

The `--noLP` option of *RNAsubopt* (and *barriers*, cf. subsection 2.7) achieves a considerable speed-up by neglecting structures containing so-called *lonely base pairs*, i. e., base pairs which are not directly surrounded by another base pair. Put differently, this option enforces a minimal helix length of two base pairs. The biological motivation of this optimization is that lonely base pairs usually destabilize a secondary structure and thus would open up again quickly. Structures not containing lonely pairs are called *canonical* structures.

Using `--noLP` significantly reduces the resources required for conducting the analysis, but may also bias its results. Therefore, when analyzing a newly designed sequence, the question arises whether applying this heuristics will, in this specific case, yield accurate results or not. Here, we derive a measure that helps to answer this question for individual sequences.

The probabilities of the secondary structures for a given RNA sequence x follow a Boltzmann distribution, i. e., the probability of a secondary structure ϕ is proportional to $B(x \,|\, \phi) := \exp(-\frac{G(x \,|\, \phi)}{RT})$. Here, R is the universal gas constant, T is the absolute temperature, and $G(x \,|\, \phi)$ is the Gibbs free energy of the RNA x folded into the structure ϕ. The term $B(x \,|\, \phi)$ is referred to as the *Boltzmann weight* of ϕ. If Φ is the entire structure ensemble of x, then the *partition function* of x is given by $Z = \sum_{\phi \in \Phi} B(x \,|\, \phi)$, and the probability of structure ϕ in the ensemble is

$$P(x \,|\, \phi) = \frac{B(x \,|\, \phi)}{Z}.$$

Note that $Z = B(x) := \exp(-\frac{G(x)}{RT})$, i. e., the partition function is the Boltzmann weight of the ensemble energy $G(x)$.

It is reasonable to assume that leaving out extremely unlikely structures will not significantly change the results of the analysis to be performed, so one way to assess the impact of the heuristics is to enumerate structures up to a certain energy threshold and compute the fraction of structures that contain lonely pairs and will therefore be excluded from the simplified analysis. Furthermore, instead of simply taking the fraction of *counts* of structures with and without lonely pairs, one can get more profound results by comparing the *sums of their Boltzmann weights* corresponding to the probabilities of the respective sets of structures.

To achieve this for a given sequence x, first calculate the partition of the full ensemble $Z = B(x)$ using the ensemble energy $G(x)$ that can be computed by running `RNAfold -p`. Now, `RNAsubopt -e` can be used to enumerate structures within a given energy band above the MFE. Initialize variables $Z^{(0)} \leftarrow 0$ and $Z^{(0)}_{\text{can}} \leftarrow 0$, which will be used to store the approximations of the partition functions of the full and the canonical ensemble, respectively. For the t-th output structure ϕ of *RNAsubopt*, compute its Boltzmann weight $B(x \,|\, \phi)$ and set $Z^{(t)} \leftarrow Z^{t-1} + B(x \,|\, \phi)$. Then, verify whether ϕ is canonical and, if this is the case, set $Z^{(t)}_{\text{can}} \leftarrow Z^{(t-1)}_{\text{can}} + B(x \,|\, \phi)$, otherwise, leave it unchanged by setting $Z^{(t)}_{\text{can}} \leftarrow Z^{(t-1)}_{\text{can}}$. Finally, compute the fractions $Z^{(t^*)}/Z$ and $Z^{(t^*)}_{\text{can}}/Z$, where t^* is the final value of t. The first fraction measures the structure coverage, i. e., which Boltzmann-weighted fraction of structures has been analyzed. It should be close to 1 for reliable results and can be improved by increasing the width of the energy band that limits the structure enumeration. The second fraction approximates the ratio of canonical structures in the ensemble.

As a side node, it is arguable that instead of the more complex enumeration process just described, the ensemble energy of the canonical ensemble could be directly computed using `RNAfold -p --noPS`. However, due to current technical limitations, the returned ensemble energy is only an upper bound of the actual value and may dramatically over-predict the fraction of canonical structures.

2.7. Analyzing the high-dimensional structure landscape

As the number of structures for a given RNA sequence grows exponentially with the sequence length, the folding process cannot be simulated with every single secondary structure even for small RNAs. Therefore, the number of simulation states needs to be reduced to a feasible number, ideally without biasing the outcome. This can be achieved by applying a coarse graining approach, which reduces the size of the high-dimensional structure landscape the sequence spans to a much smaller set of *macro-states*, each of which represents a set of multiple structures.

The tool *barriers* (cf. Table 1) implements a flooding algorithm that effectively coarse grains an energy landscape to macro-states or *basins*, each represented by a local minimum of the folding landscape. Each basin contains all the structures connected to its representative local minimum by the folding path

of steepest descend. For each two macro-states, the tool also computes *barrier height*, i. e., the highest intermediate structure (with respect to its energy) that has to be overcome in order to refold from one state to the other. It can be used to visualize the RNA landscape by drawing a *barrier tree*.

As input *barriers* requires a list of all suboptimal structures within a certain energy range, sorted by ascending energy value. How to obtain such a list is explained in subsection 2.5. To obtain correct simulation results, the energy range has to be large enough to connect all generated macro-states. If this is not the case, the width of the energy band has to be increased. Alternatively, heuristic approaches such as *findPath* [19] may be used to connect formerly disconnected states. In order to handle the huge amount of structural states generated by our design example, it is mandatory to configure *barriers* using the option `--with-hash-bits=29` and to run *make* with the argument `AM_CFLAGS=-mcmodel=large`.

Once the input file has been generated, *barriers* can be applied to it by executing

```
$ barriers --max=500 -G RNA -M noShift --bsize --rates < seq1.sub > seq1.bar
```

The `--max=500` option specifies the number of macro-states to be generated, `-G`, specifying the graph type, is set to RNA, `-M noShift` disallows so-called shift moves (i. e., a move changing exactly one of the two indices of an existing base pair), and `--bsize` and `--rates` enable the output of the size of each basin, and to compute transition rates between these macro-states. The results of *barriers* are then piped into the file `seq1.bar`. A graphical representation of the barrier tree in the PostScript format is by default saved to a file named `tree.ps`, whereas the rates are stored in file `rates.out`.

Note that *barriers* needs to be run with the `-G RNA-noLP` option when predicting an ensemble without lonely pairs.

2.8. Simulating kinetic folding using macro-states

When relying solely on thermodynamic criteria—e. g., probabilities of given conformations—during an RNA design process, one may miss important traits of the candidate sequences. Transcriptional riboswitches, for example, interact with the RNA polymerase in a time-critical manner, and information about the presence or absence of specific sub-structures within certain time frames are necessary to ensure correct switching behavior Such knowledge can be obtained by running a kinetics simulation for the given RNA sequence. As this type of analysis is too time-consuming to be included into the design process directly, it should be performed on a small set of promising candidates to verify their functionality.

The program *treekin* can be used to simulate single-molecule kinetics, which solves a continuous-time Markov process by numerical integration with the infinitesimal generator being a rate matrix. The latter is obtained by running *barriers*, which estimates the transition rates from each macro-state to all the other ones and stores them as a matrix in the file `rates.out` (subsection 2.7). The computation is performed using *treekin* as follows:

```
$ treekin --p0 1=1 -m I -f rates.out --t8=1E12 < seq1.bar > seq1.tkin
```

Here `-m I` tells *treekin* to parse the file specified by `-f` as *barriers* output, `--t8` sets the maximum simulation time to 1×10^{12} arbitrary time units (AU) and `--p0` sets the initial population size of the selected minimum of the barrier tree. Here, we set the global minimum of the barrier tree (i. e., macro-state 1) to be 100%. The output can then be visualized by using the program *xmgrace* with the following command:

```
$ xmgrace -log x -nxy seq1.tkin
```

2.9. Coarse grain visualization to emphasize structural features

Kinetic folding plots (cf. subsection 2.8) usually produce a big amount of independent curves (500 in our example), one for each macro-state of the barrier tree. However, we optimized the RNA to exhibit specific structural features and thus want to visualize how often we observe this sub-structure in the ensemble of structures and the kinetic plots. Thus, we are collecting states that exhibit our structural features, i. e., ligand-binding stem or alternative stem, and summarize them into combined density curves. We implemented a *Perl* script called `coarsify_bmap.pl`[1] that performs this task. It can be applied to `seq1.bar` and `seq1.tkin` output as follows:

```
coarsify_regex.txt

# ?25((((((((((.....))))))))))) | ?26((((((((((.....))))))))))
^.{25}\({11}\.{6}\){11}[\.\(\)]{11}|^.{26}\({10}\.{6}\){10}[\.\(\)]{11}

# ?2(((...(((((((((.....)))))...)))...))) | ?2(((...(((((((((.....))))...))))...)))"
^.{2}\({3}\.{3}\({8}\.{5}\){5}\.{3}\){3}\.{3}\){3}|^.{2}\({3}\.{3}\({8}\.{5}\){4}\.{3}\){4}\.{3}\){3}

$ perl coarsify_bmap.pl -regs coarsify_regex.txt -minh 30 \
  -outdir coarse_30 seq1.bar seq1.tkin
```

This script merges macro-states of a given barrier tree in two ways: i) if the barrier height of a state is below the selected `--minh` value, it is merged to its neighbor and the population density of this neighbor is increased accordingly, and ii) if states contain similar structural elements, specified as regular expressions (`coarsify_regex.txt`), they are merged. Note that macro-states containing a different set of these structural elements are never merged although i) would be applicable. In the above example, all states are merged as `--minh` is larger than the energy band generated by *RNAsubopt*. However, the two specified regular expressions combine states that are compatible with the initial structural constraints of the design and keep the remaining landscape separate. The coarse-grained *barriers* and *treekin* output is written to the specified `coarse_30/` subdirectory.

[1] `https://github.com/ViennaRNA/BarMap/`

2.10. Kinetic simulation of an RNA with ligand interaction

Analyzing the influence of a ligand on the folding kinetics of a potentially binding-competent RNA molecule can, in its most general form, be a difficult problem. However, under certain conditions discussed below it is possible to use *treekin* (cf. Table 1) for this task. The effect of ligand addition can be simulated by declaring the binding-competent state *absorbing*, i. e., prevent any transitions out of it. This can be achieved by starting *treekin* with the population density of the last time point in `seq1.tkin`—i. e., the equilibrium distribution—and setting the `-a` option to the most stable binding-competent state:

```
$ grep -v "#" seq1.tkin | tail -n 1 | \
  perl -ae '{for($i=1; $i<scalar(@F); $i++){print "--p0 $i=$F[$i] "}}' > t
$ treekin -m I `cat t` -f rates.out --t8=1E12 -a 3 < seq1.bar > seq1_absorb.tkin
$ coarsify_bmap.pl -regs coarsify_regex.txt -minh 30 \
  -outdir coarse_30absorb seq1.bar seq1_absorb.tkin
$ rm t
```

First, the last time point in `seq1.tkin` is extracted and converted such that the output saved in `t` can be used as repeated `--p0` parameter of *treekin*. Then, *treekin* is called and its output is stored in `seq1_absorb.tkin`, which is subsequently coarse grained. Finally, the temporary file `t` is removed. Visualization of the coarse-grained absorbing landscape is possible with the graph plotting tool *xmgrace* (cf. Table 1) by running:

```
$ xmgrace -log x -nxy coarse_30absorb/seq1_absorb.tkin
```

Using an absorbing state to model the ligand interaction is an approximation that is only reasonable under certain conditions. Irrespective of the properties of a specific ligand, a high ligand concentration as compared to the RNA concentration is assumed. In fact, absorbing states may be interpreted as an *infinite* ligand concentration leading to an immediate dimerization with the binding-competent RNA. Since ligands are usually much smaller molecules than their respective target RNA, this assumption is reasonable and ligand concentrations in the order of 1 mM are realistic in practice, though care has to be taken when dealing with toxic ligands like antibiotics.

The absorbing state assumption implies that the RNA–ligand complex has a rather low dissociation constant K_d, or put differently, the dissociation rate coefficient k_{off} is low compared to its association rate coefficient k_{on}. As pointed out by Wolfinger et al. in this special issue [20], in a working *(co-)transcriptional* RNA switch, the coefficients of the dimerization and the dissociation rate obey the criteria $k_{\text{on}} > 1/t_{\text{apt}}$ and $k_{\text{off}} \ll 1/t_{\text{elong}}$, where t_{apt} is the duration during which the aptamer senses the ligand during transcription, and t_{elong} is the time required for the transcription of a single nucleotide. Under such conditions, the usage of an absorbing state to model the dimerization seems to be adequate. For *thermodynamic* switching behavior, K_d translates into a (negative) energy bonus $\theta = -RT \ln K_d$ [21] awarded to all binding-competent structures. Let E_{MFE} be the MFE of the given sequence, and $E_{\mathtt{bc}}$ the free energy of the binding-competent state. Then, for the switch to work properly, $E_{\mathtt{bc}} + \theta \leq E_{\text{MFE}} \leq E_{\mathtt{bc}}$

must hold [22]. If the left-hand side of the inequality is significantly smaller, then the absorbing state model is a well-suited approach.

For many ligands of several different classes, aptamers with low K_d values in the order of 1 µM have been characterized, see [23] for a comprehensive summary. Though a precise K_d threshold cannot be given due to the concentration dependence, we hypothesize that such ligands behave in a way suited for this approximative approach.

Table 1: Summary of the utilized software. RNA related software tools are either standalone or part of the *ViennaRNA* package. The installation procedure is documented on the web pages listed. Standard Unix tools are tagged as "Other" and are typically included in or easy to install with the package manager of any distribution.

	Software	**Description and URL**	**Ref**
RNA related	*RNAblueprint* v1.2	Fair sampling approach that generates sequences compatible to sequence and to one or more structural constraints. You need to install the boost library first. Note that we compiled it with `--disable-perl` `https://github.com/ViennaRNA/RNAblueprint`	[10]
	RNAsketch v1.2	Python library to design nucleic acid sequences using *RNAblueprint*. It offers convenient functions to interact with the software packages of *ViennaRNA* and *NUPACK*. Furthermore, predefined methods, e. g., for sequence optimization, help to standardize the design process. `https://github.com/ViennaRNA/RNAsketch`	[10]
	barriers v1.6.0	Generates a coarse-grained energy landscape given an energy-sorted list of suboptimal RNA secondary structures. Note that we configured with `--with-hash-bits=29` and ran *make* with argument `AM_CFLAGS=-mcmodel=large` in order to handle upto 2^{29} structures. `http://www.tbi.univie.ac.at/RNA/Barriers/`	[24]
	treekin v0.3.1	Calculates folding kinetics on a coarse-grained energy landscape. One problem that often occurs during *treekin* installation is its dependency on the *blas* and *lapack* packages. Try to install them first. Note that we compiled an older version of *treekin* as v0.4.1 does not support the `-a` option. `http://www.tbi.univie.ac.at/RNA/Treekin/`	[25]
	ViennaRNA v2.4.0	Library containing the ViennaRNA tools. `http://www.tbi.univie.ac.at/RNA/`	[16, 26]
	RNAfold	Calculates minimum free energy secondary structures and partition function of nucleic acid sequences.	
	RNAsubopt	Calculates suboptimal secondary structures a nucleic acid sequence can fold into.	
Other	*sort*	As part of the gnu core utils this program takes a text file and sorts it in the specified order. `http://www.gnu.org/software/coreutils/sort`	
	xmgrace	*xmgrace* is a full-featured graphical user interface of *grace* to make two-dimensional plots. `http://plasma-gate.weizmann.ac.il/Grace/`	

3. Results

At the very beginning of a design process, necessary building blocks should be analyzed and evaluated to elucidate their properties. One of these blocks are RNA aptamers, as they cannot be generated by simply applying computational design methods. Here, we use an experimentally characterized RNA aptamer with known dissociation constant K_d and adaptive structural features, namely the well-known theophylline aptamer [17, 27, 28]. Alternatively, a novel aptamer could—at least in principle—be selected by performing an experimental protocol such as Systematic Evolution of Ligands by EXponential enrichment (SELEX) for the ligand of interest.

Next, a precise idea of how the desired RNA regulation mechanism should work is required. If it resembles a naturally occurring regulation mechanism, it is advisable to investigate its biological counterpart in detail before transferring the concept to a novel design. Figure 2 sketches the idea used to carry out the design step of this contribution. A given aptamer is extended in a way that an alternative structural conformation (`ac`) is formed in absence of the ligand. As the ligand is added, it reacts with the binding-competent conformation (`bc`) to form the ligand-bound conformation (`lc`), thereby stabilizing it and sequestering the alternative conformation (`ac`).

The mentioned downstream extension of the aptamer is necessary to introduce some degrees of freedom for the sequence sampler since the sequence of the aptamer itself is fixed. The insert needs to be long enough to sequester significant parts of the aptamer's binding-competent structure. On the other hand, *short* inserts are preferable to avoid unforeseen interactions with the surrounding sequence context. Experiments by Ceres et al. [29] suggest that the ability of many aptamers to bind their respective ligand may be disrupted by solely opening their P1 (i. e., outermost) stem. However, we decided to introduce 22 nucleotides, corresponding to about half the length of the aptamer, to allow for an adequate thermodynamic stability and the complete sequestering of the aptamer structure by the alternative conformation.

We converted this model into a sequence and two structural constraints that represent `ac` and `bc`. If its structure had been resolved, the ligand-bound conformation could be taken into account as a third structural constraint. This would allow *RNAblueprint* to only generate sequences compatible to the structure of the dimer conformation. However, upon ligand binding, aptamers typically adapt complex tertiary interactions going beyond the scope of the classical secondary structure model. In case of theophylline, extensive stacking as well as the formation of base triples during ligand recognition have been observed [27, 28]. Such interactions cannot be handled by currently available secondary structure prediction and RNA design tools. A structural constraint modeling conformation `lc` is therefore omitted.

The functional model can be expressed as a combination of constraints such as structural requirements and various properties specifying the energy landscape, and the kinetic folding properties. An RNA sequence meeting these requirements as close as possible can be obtained by performing a local opti-

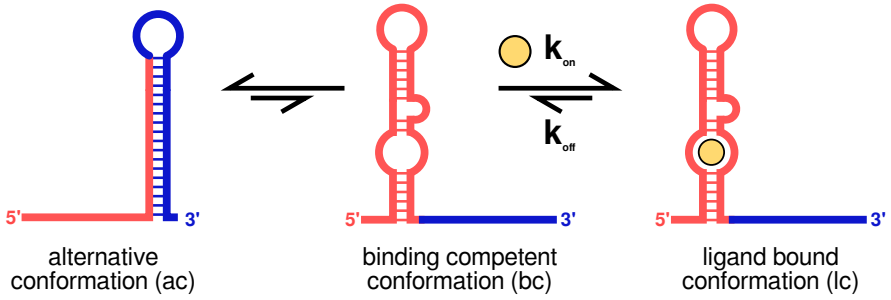


Figure 2: Graphical representation of the design idea. The system consists of two parts. In absence of the ligand, two conformations should dominate the structural ensemble. Depending on the design parameter, the `alternative conformation (ac)` should be higher populated than the `binding competent (bc)` one. Refolding rates between the two structural conformations depend on the energy barrier that separates them. Upon ligand addition, the `bc` gets trapped and the system should be shifted towards the `ligand bound conformation (lc)`.

mization approach. Such an approach includes i) the sampling of sequences with respect to a set of prescribed constraints, ii) the definition of a quality criterion through a proper objective function, and iii) an optimization method that decides whether to keep or reject a proposed solution.

We applied *RNAblueprint* to uniformly sample sequences that are compatible to the given sequence and structure constraints of the proposed design model, cf. subsection 2.1. The returned sequences need to be scored according to the design goal. Clearly, this design goal should include the evaluation of kinetic processes driving the implemented switching dynamics. However, these predictions are usually too demanding to be evaluated many times during optimization. Thus, we only use reasonable fast thermodynamic measures to ensure mandatory properties of the resulting kinetic processes [30].

On the thermodynamic level, we need to guarantee that the conformations of our model are exclusively present at least in the equilibrium. Given a sequence x and a compatible structure ϕ, one can calculate the corresponding Gibbs free energy $G(x \,|\, \phi)$ using the nearest neighbor model [31, 32]. The sequence–structure mapping is a one-to-many relation. Hence, one sequence can adapt a huge set of possible structures Φ called this sequence's structure ensemble. In the equilibrium, the Boltzmann weight $B(x \,|\, \phi) := \exp(-\frac{G(x|\phi)}{RT})$ of a structure ϕ is proportional to its probability. Summing over all structures of the ensemble gives rise to the partition function $Z = \sum_{\phi \in \Phi} B(x \,|\, \phi)$ of x. From that we can calculate the probability of ϕ with respect to the ensemble as

$$P(x \,|\, \phi) = \frac{B(x \,|\, \phi)}{Z}.$$

We utilize these properties to develop a novel objective function for the proposed model, cf. Figure 2. When adding the ligand to the system, we want to maximize the number of bound molecules, i. e., the probability of `lc` should ideally be one. As we do not have an explicit structural constraint of this state, we maximize the number of binding-competent structures in presence of the

ligand, assuming the ligand is available in excess and immediately bound. In case of the theophylline aptamer, this precondition is fulfilled as its association rate constant k_{on} has been determined to be much higher than the dissociation rate constant k_{off} [28]. This is in accordance to its independently measured K_d of about 0.32 µM [17]. We therefore add an energy bonus of $-9.22\,\text{kcal}\,\text{mol}^{-1}$ to every secondary structure in the ensemble that contains the correctly folded—i. e., binding-competent—theophylline aptamer.

By maximizing the probability of `bc` in the presence of the ligand, we favor the conversion to the ligand-bound conformation `lc`. In contrast, `ac` should be highly populated in absence of the ligand. However, no ligand binding is possible if the RNA molecule exclusively adapts `ac` as only `bc` induces a high binding affinity of the ligand for the RNA molecule. It is therefore necessary to establish a balance between `ac` and `bc` where `bc` must always be present. We combined all these assumptions into the novel objective function

$$f(x) = P(x \,|\, \texttt{bc}, \text{ligand}) \cdot (1 - |a - P(x \,|\, \texttt{ac})|) \cdot (1 - |b - P(x \,|\, \texttt{bc})|) \qquad (2)$$

where $a, b \in (0, 1),\ a + b \leq 1$ are the target probabilities of the alternative conformation and binding-competent conformation, respectively. This function is maximized as all terms tend to one. We set $a = 0.7$ and $b = 0.3$ for the discussed example. We describe the details on how to calculate the individual terms of the objective function given above utilizing the constraint framework of the *ViennaRNA* package in subsection 2.3 and subsection 2.4.

To perform a local optimization procedure searching for sequences optimal with respect to the derived objective function (2), we chose to harness a scripting library called *RNAsketch*, which is available as interface to the sequence sampler *RNAblueprint* (cf. Table 1). *RNAsketch* offers ready-to-use implementations of several well-known optimization strategies. To tackle the presented design problem, we implemented a *Python* script that performs adaptive walks with randomly chosen steps of varying size—ranging from point mutations to full resampling of the sequence—until the score evaluated with the designed objective function (2) stays minimal. This approach has been found to converge relatively fast towards reasonable results for other objectives [10]. Our implementation[2] and the corresponding commands including the inputs are available online to serve as an example of use for *RNAsketch*.

The described local optimization procedure is capable of producing many potential solutions in a relatively short amount of time. As the returned scores contain no additional information but the three probabilities, we developed an *in silico* analysis pipeline to visualize additional properties of the obtained sequences, facilitating a consecutive ranking and filtering step. First and foremost, we need to verify the kinetic properties of our obtained solutions, a usually very expensive and time-consuming task. In the following, we discuss this process for an example sequence[3].

[2] `design-ligandswitch.py`

[3] `AAGUGAUACCAGCAUCGUCUUGAUGCCCUUGGCAGCACUUCAGUUGUUGAGGGGGCUCAAUGAC`

For a more complete picture of the energy landscape, we need to investigate the structural states our example sequence will likely fold into. *RNAsubopt* is applied to generate all suboptimal structures up to 22.6 kcal mol^{-1} above the sequence's minimum free energy, cf. subsection 2.5. The number of possible structures grows exponentially with sequence length and is approximately 225 million for the chosen energy range of 22.6 kcal mol^{-1}, resulting in a 16 GB large file. To reduce the number of generated suboptimal structures, and thereby speed up all subsequent steps, it is possible to skip all structures containing lonely pairs, i. e., helices of length one, generating only so-called canonical structures. This reduces the number of states to approximately 6.7 million, and the file size to 459 MB. However, for the shown example, this also excludes the predicted MFE structure, cf. most populated structures in the equilibrium in Figure 3. The previous ground state containing the alternative structural element is only the fourth-stable state while the MFE structure contains the binding-competent aptamer. Of course, this has a dramatic impact on the simulated kinetics, cf. Figure 3.

To assess the impact of the "no lonely pairs" (`--noLP`) heuristics more profoundly, the procedure described in subsection 2.6 has been applied to the example sequence. By enumerating all structures up to 10 kcal mol^{-1} above the MFE, one obtains a structure coverage of 99.9%. Here, the identified fraction of canonical structures is only 43%, so the vast majority of structures that would likely be encountered in the simulation are removed when applying the `--noLP` heuristics. It is clear that in this case, the reduction to the canonical structure ensemble leads to a strong bias.

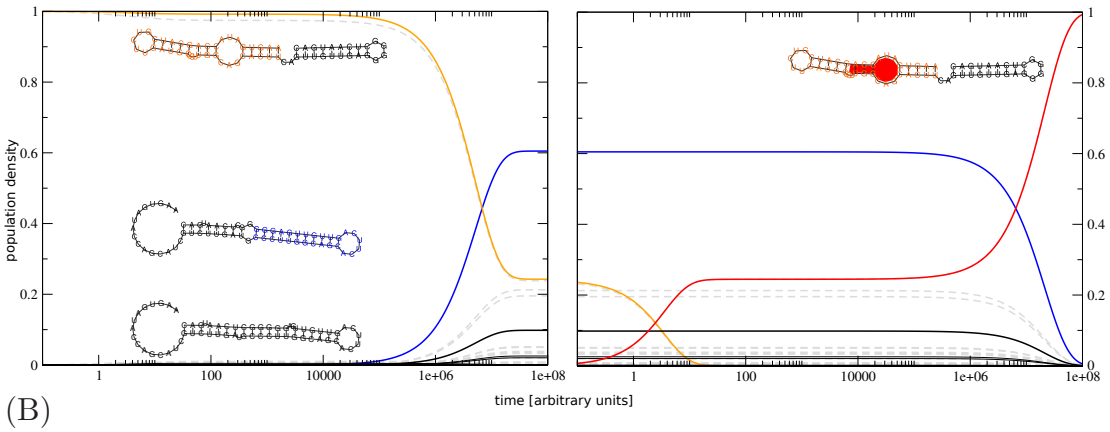
In contrast, other sequences have much higher fractions of canonical structures. The example sequence[4] has the same length (64 nt) and GC content (51%) as the previous one, but exhibits a predominantly canonical ensemble (96% of the structures).

In any case, further coarse graining of the structure landscape is mandatory. We apply the program *barriers* which implements a flooding algorithm and abstracts the structure landscape to a selected number of macro-states, each represented by a local minimum of the landscape (subsection 2.7). Transition rates from each of these macro-states to all other ones are then estimated and subsequently used to predict the folding kinetics.

It is possible that multiple macro-states exhibit structural features such as the structure of `bc` or the stem of `ac`. Thus, for better visualization we merged states that exhibit certain structural features by implementing *coarsify_bmap.pl*, cf. subsection 2.9. Based on the resulting landscape and the processed transition rates, *treekin* has been invoked to simulate the single-molecule folding kinetics, cf. subsection 2.8. A visualization of the output shows the expected population density of the two designed structural states `ac` and `bc` after the equilibrium has been reached, cf. Figure 3A. This way, we verified that the estimate based on partition function folds—as used during the optimization process—matches

[4] `GUAAGAGAGGCCGCGCACAACUUUCCUACUGUUCGAAAGGUAGGAGCGCUGUCAACUUACAUGG`

(A)



(B)

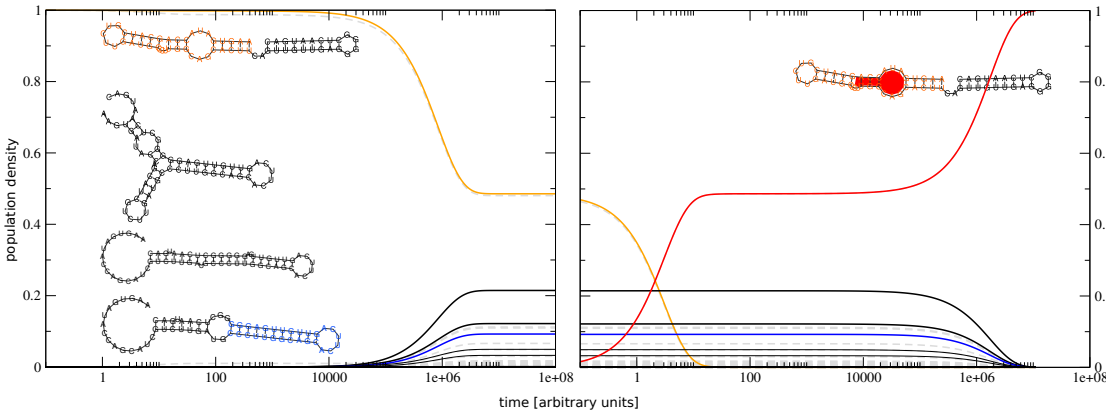


Figure 3: Simulated kinetics using (A) the complete and (B) a reduced structure ensemble by avoiding lonely pairs. In both cases, the simulation is started with the complete population in a structural state that contains the binding competent aptamer structure (orange). The left part of each plot shows the dynamics until the system is equilibrated, whereas the right part depicts the simulated systems kinetics after ligand addition. Dashed gray lines indicate the system's kinetics without coarse graining. By design, the population density in the equilibrium of all structures containing the alternative (blue) and the binding-competent (orange) structural element should be 0.7 and 0.3, respectively. Colored lines display the coarse-grained kinetics where states containing specific structural elements are merged. For the most prominent states, the secondary structure corresponding to their respective stable representative are shown using the same color. Despite the similarities of the representatives of the blue and the thick black curves, they were not merged because the alternative structural element was required to have a perfectly stacked stem of length at least 11 nt.

the results of the kinetic simulation even in absence of the ligand and when the full *RNAsubopt* output, including non-canonical structures, is used.

When sketching the design (cf. Figure 2), we assumed that the RNA–ligand complex has a rather low dissociation rate coefficient k_{off} compared to its association rate coefficient k_{on}. For the theophylline aptamer, this is in accordance with published rates of $(0.07 \pm 0.02)\,\text{sec}^{-1}$ and $(1.7 \pm 0.2) \times 10^5\,\text{M}^{-1}\,\text{sec}^{-1}$ at 25 °C for k_{off} and k_{on}, respectively [28]. We therefore modeled the effect of ligand addition by starting *treekin* with the population density of the equilibrium and making the binding-competent state absorbing, cf. subsection 2.10. Visualization of the coarse-grained absorbing landscape shows that after about 9×10^6 AU, which can be mapped to approximately 45 sec [33], 50% of the RNA molecules are in the ligand-bound state, cf. Figure 3.

4. Discussion

During the development of our software pipeline we realized that, until recently, there mainly existed two kinds of publications. One created by wet lab researchers, focused on an experimental testing setup as well as functional and analytical tests. They frequently missed the possible advances of *in silico* tools and their valuable predictive power. In contrast, publications written by researchers mainly working on computational biology often comprised sophisticated biophysical methods, great computational details, and a huge variety of mathematical and algorithmic tricks, but were frequently neglecting the aspect of biological applicability.

A main reason for this situation is that most of the RNA design programs available use predefined terms in the objective function as well as a fixed optimization procedure [22], and thus are inflexible and not customizable enough to be applied and adopted to the huge amount of considerably varying design scenarios. The *RNAblueprint* approach[10] decouples sampling of sequences compatible to one or more structural constraints from the subsequent optimization procedure, which gives the user the full flexibility to implement novel and innovative objectives.

Computational design studies are often missing the bigger context, such as the initial analysis of the system, suggestions for experimental testing or the design of proper controls. However, experimental validation is not a straightforward task and needs to be carefully planned already during the design process. This includes extensive *in vitro* or *in vivo* studies, or preferable both. To really gain knowledge about the device's mechanism and about potential mistakes or pitfalls in case of dysfunctionality, a purely qualitative answer will not be sufficient. Therefore, a complete testing pipeline should include the determination of structures, binding affinities, or elucidate kinetic properties. Smartly designed positive and negative controls are also helpful to reveal important properties of the newly generated RNA device. Ideally, these controls will unveil quantitative answers about the mechanistic details, the actual structures of the RNA or even about kinetic aspects like co-transcriptional dependency or ligand affinity.

In this contribution, we described in detail the *de novo* design of a ligand-sensing riboswitch that adapts two alternative conformations. Depending on the presence of the ligand, either a binding competent state, or a specified alternative structural conformation is dominating the ensemble. This riboswitch design can easily be extended, e. g., to perform regulatory tasks in a host cell such as translational or transcriptional regulation of a downstream target gene. A translational riboswitch for instance will probably contain a Ribosome Binding Site (RBS) which is sequestered in the inactive state. This can be included easily by specifying the appropriate sequence constraints and further objectives such as the accessibility of the RBS in both conformations.

For such a purpose, it is important to distinguish between two types of switching behaviors One type of riboswitches which are capable to switch on and off during the entire lifetime of the molecule. The other switches are fixed after a certain time of sensing disregarding future changes in the ligand concentration. For the latter, switching is only possible through RNA decay and repeated transcription.

If switching is possible at any time, fast response times to ligand changes are obtained. However, the individual states of the molecules remain fuzzy as not all of them will adapt the desired structure, leading to the observation of background activity. In our example, we provoked such a behavior by targeting a 70:30 ratio of alternative to binding-competent conformation, and indeed observed a quick refolding process upon addition or removal of the ligand.

Alternatively, we could generate a "one way switch" where the state decision is only possible during a specific window. Thereafter, the chosen state is stabilized, either by a kinetic folding trap or by ongoing molecular processes such as translation. Once decided, individual molecules cannot revert their choice within reasonable time, even if the ligand dissociates or is removed from the system [34]. Therefore, it must be ensured that the competing states are populated during the decision window. This method has the advantage of obtaining distinct states with very little noise. However, the response times to ligand changes are quite long as they depend on RNA decay and the transcription speed.

At a first glance, the design model we proposed in Figure 2 seems to be rather easy. However, it is not straight forward to develop an experimental setup that is able to determine if the target ratio of 70:30 of the two conformations is reached in the equilibrium or not. Sophisticated approaches such as single-molecule FRET and NMR have been applied to determine the structure and energy landscape of natural riboswitches[35, 36]. Both referenced studies revealed that more than the presumably two dominating states are adapted depending on environmental conditions, i. e., Mg^{2+} concentration and temperature. This might be the case for our designs as well although we optimized them towards two alternative states only. Furthermore, it is important to note that the presented *in silico* results are estimates. For instance, the target ratio of 70:30 might be achieved perfectly by the optimization procedure and predicted by kinetics simulations. However, the results are extremely sensitive to the underlying energy parameters. Those are measured under specific experimental

conditions for rather short structural elements. For more complex structures, i. e., those containing large or even multi loops, estimates are utilized to determine a structure's energy [31]. If the experimental conditions used to determine the energy parameters and those used in the *in vitro* or *in vivo* testing environment vary significantly, discrepancies of prediction and measurement are an inevitable effect.

Depending on the research question, neglecting structures with lonely pairs can give valuable insights into the studied system while dramatically speeding up the prediction process. The reason the analysis fails for the exemplary sequence presented in this work is that many of its low-energy structures contain lonely pairs and are therefore excluded when enabling the heuristics. The effect is dramatic here as even the MFE structure is not canonical.

A method to assess the importance of lonely pairs for the simulations has been developed and shown to correctly predict the consequences of noLP heuristics. In general, it is advisable to always consider the fraction of canonical base pairs before resorting to the heuristic. Another advantage of conducting this additional analysis is the proper estimation of the energy band width required to achieve a high coverage of the structure ensemble during the enumeration. This information is useful to improve both, the performance of the simulation as well as the quality of the results, even when not utilizing the noLP heuristics. Nevertheless, re-running the analysis of promising candidates with a full structure ensemble is advisable to assure correct results if the required resources are available.

Many of the techniques used in this work implicitly make simplifying assumptions about the processes involved in RNA switching. For example, the soft constraint framework is a considerable abstraction of the binding process in at least two ways. Firstly, it models a binary binding behavior in that the ligand either perfectly fits an RNA structure and gets the full binding energy bonus, or it does not bind to the structure at all. In reality, small variations in the binding domain may lead to an altered binding energy instead. Secondly, any structure exhibiting the binding site receives the full stabilizing energy contribution, neglecting the effect of the ligand concentration and assuming infinite reaction rates. During the kinetics simulation, a similar behavior is achieved by declaring the binding-competent macro-state absorbing, i. e., the dissociation of the ligand is not possible at all.

While these may be adequate assumptions for ligands with a high binding affinity present in an excessive concentration, it may lead to over-estimation of the fraction of RNA–ligand complexes in situations where the association rate becomes the bottleneck of the dimerization reaction. In such cases, one should resort to more sophisticated models considering these rates as well as the ligand's concentration [37]. An efficient implementation of this approach that can readily be applied to ligand-aware co-transcriptional folding is published in this special issue [20].

When analyzing our designed switch *in silico*, we started the kinetic simulation of the ligand-free environment with all molecules in the binding-competent state. Thereby, we ensured that even in this worst scenario possible, the system

quickly recovers to the defined ratio of alternative state and binding-competent state. To obtain better estimates of the switching times, starting with various other distributions depending on the application might be preferable.

In case of an *in vitro* experiment, the protocol would probably envisage to first heat up the solution to completely untangle the RNA structures and then quickly put the solution on ice until the ligand is added. A similar cooling experiment could be performed *in silico* by performing Boltzmann-weighted structure sampling from an ensemble at high temperature and using the resulting distribution of states as starting point for a subsequent kinetic simulation. In contrast, when using the generated riboswitch *in vivo*, it is likely co-transcriptionally folded within the cell. Therefore, it is advisable to obtain the initial distribution by applying a co-transcriptional folding approach which simulates the RNA's elongation process until the binding-competent part of the structure is fully transcribed. A software capable of this type of analysis is, for example, *BarMap* [38].

In this contribution, we aimed to generate a general ligand-triggered riboswitch which can be extended to control regulation mechanisms, such as transcriptional or translational control of a downstream target gene. We devised a functional model of such a riboswitch and successfully implemented a design approach to *de novo* generate RNA sequences that fulfill the prescribed properties. The proposed pipeline consists of several modular pieces which can be easily adopted or exchanged in case of varying needs. This includes the flexible sequence sampling engine *RNAblueprint*, a novel objective function to thermodynamically describe important features of the mechanism, the optimization approach and, finally, the *in silico* analysis pipeline to verify kinetic properties of the system.

5. Acknowledgments

This work has been supported by the European Commission under the Environment Theme of the 7th Framework Program for Research and Technological Development (Grant agreement number 323987), the Austrian science fund FWF project F43 "RNA regulation of the transcriptome", the German Network for Bioinformatics Infrastructure (de.NBI) by the German Federal Ministry of Education and Research (BMBF; support code 031A538B), and by the German Research Foundation (DFG; grant STA 850/15-2)

We thank Christina Wagner for fruitful discussion, Manuela Geiß for assistance with mathematical issues and our private experimental testing help desk Mario Mörl.

Abbreviations

UTR untranslated region

mRNA messenger RNA

SELEX Systematic Evolution of Ligands by EXponential enrichment

MFE minimum free energy

RBS Ribosome Binding Site

References

[1] R. R. Breaker, Prospects for Riboswitch Discovery and Analysis, Molecular Cell 43 (6) (2011) 867–879. doi:10.1016/j.molcel.2011.08.024.

[2] C. C. Fowler, E. D. Brown, Y. Li, A FACS-based approach to engineering artificial riboswitches., Chembiochem 9 (12) (2008) 1906–1911. doi:10.1002/cbic.200700713.

[3] J. E. Weigand, M. Sanchez, E.-B. Gunnesch, S. Zeiher, R. Schroeder, B. Suess, Screening for engineered neomycin riboswitches that control translation initiation, RNA 14 (1) (2008) 89–97. doi:10.1261/rna.772408.

[4] J. E. Weigand, B. Suess, Tetracycline aptamer-controlled regulation of pre-mRNA splicing in yeast, Nucleic Acids Research 35 (12) (2007) 4179–4185. doi:10.1093/nar/gkm425.

[5] C. Schneider, B. Suess, Identification of RNA aptamers with riboswitching properties, Methods 97 (2016) 44 – 50. doi:10.1016/j.ymeth.2015.12.001.

[6] A. A. Green, P. A. Silver, J. J. Collins, P. Yin, Toehold Switches: De-Novo-Designed Regulators of Gene Expression, Cell 159 (4) (2014) 925–939. doi:10.1016/j.cell.2014.10.002.

[7] M. Etzel, M. Mörl, Synthetic Riboswitches: From Plug and Pray toward Plug and Play, Biochemistry 56 (9) (2017) 1181–1198. doi:10.1021/acs.biochem.6b01218.

[8] A. Churkin, M. D. Retwitzer, V. Reinharz, Y. Ponty, J. Waldispühl, D. Barash, Design of RNAs: Comparing programs for inverse RNA folding, Briefings in Bioinformatics (2017) bbw120doi:10.1093/bib/bbw120.

[9] F. J. Isaacs, D. J. Dwyer, C. Ding, D. D. Pervouchine, C. R. Cantor, J. J. Collins, Engineered riboregulators enable post-transcriptional control of gene expression, Nature Biotechnology 22 (7) (2004) 841–847. doi:10.1038/nbt986.

[10] S. Hammer, B. Tschiatschek, C. Flamm, I. L. Hofacker, S. Findeiß, RNAblueprint: flexible multiple target nucleic acid sequence design, Bioinformatics 33 (18) (2017) 2850–2858. doi:10.1093/bioinformatics/btx263.

[11] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, E. Teller, Equation of State Calculations by Fast Computing Machines, The Journal of Chemical Physics 21 (6) (1953) 1087–1092. doi:10.1063/1.1699114.

[12] W. K. Hastings, Monte Carlo sampling methods using Markov chains and their applications, Biometrika 57 (1) (1970) 97–109. doi:10.1093/biomet/57.1.97.

[13] A. Taneda, Multi-objective optimization for RNA design with multiple target secondary structures, BMC Bioinformatics 16 (1) (2015) 280. doi:10.1186/s12859-015-0706-x.

[14] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, P. Schuster, Fast folding and comparison of RNA secondary structures, Monatshefte für Chemie / Chemical Monthly 125 (2) (1994) 167–188. doi:10.1007/BF00818163.

[15] C. Höner zu Siederdissen, S. Hammer, I. Abfalter, I. L. Hofacker, C. Flamm, P. F. Stadler, Computational design of RNAs with complex energy landscapes, Biopolymers 99 (12) (2013) 1124–1136. doi:10.1002/bip.22337.

[16] R. Lorenz, I. L. Hofacker, P. F. Stadler, RNA folding with hard and soft constraints, Algorithms for Molecular Biology 11 (8) (2016) . doi:10.1186/s13015-016-0070-z.

[17] R. D. Jenison, S. C. Gill, A. Pardi, B. Polisky, High-resolution molecular discrimination by RNA, Science 263 (5152) (1994) 1425–1429.

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

[19] C. Flamm, I. Hofacker, Beyond energy minimization: approaches to the kinetic folding of RNA, Chemical Monthly 139 (2008) 447–457.

[20] M. T. Wolfinger, C. Flamm, I. L. Hofacker, Efficient computation of co-transcriptional RNA-ligand interaction dynamics, MethodsSubmitted.

[21] M. Wachsmuth, S. Findeiß, N. Weissheimer, P. F. Stadler, M. Mörl, De novo design of a synthetic riboswitch that regulates transcription termination, Nucleic Acids Research 41 (4) (2013) 2541–2551. doi:10.1093/nar/gks1330.

[22] S. Findeiß, M. Wachsmuth, M. Mörl, P. F. Stadler, Design of transcription regulating riboswitches, Methods Enzymol 550 (2015) 1–22. doi:10.1016/bs.mie.2014.10.029.

144

[23] S. Klussmann (Ed.), The Aptamer Handbook: Functional Oligonucleotides and Their Applications, Wiley-VCH, 2006. doi:10.1002/3527608192.

[24] C. Flamm, I. L. Hofacker, P. F. Stadler, M. T. Wolfinger, Barrier trees of degenerate landscapes, Zeitschrift für Physikalische Chemie 216 (2/2002) (2002) . doi:10.1524/zpch.2002.216.2.155.

[25] M. T. Wolfinger, W. A. Svrcek-Seiler, C. Flamm, I. L. Hofacker, P. F. Stadler, Efficient computation of RNA folding dynamics, Journal of Physics A: Mathematical and General 37 (17) (2004) 4731–4741. doi:10.1088/0305-4470/37/17/005.

[26] R. Lorenz, S. H. Bernhart, C. H. z. Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, I. L. Hofacker, ViennaRNA Package 2.0, Algorithms for Molecular Biology 6 (1) (2011) 26. doi:10.1186/1748-7188-6-26.

[27] G. R. Zimmermann, R. D. Jenison, C. L. Wick, J.-P. Simorre, A. Pardi, Interlocking structural motifs mediate molecular discrimination by a theophylline-binding RNA, Nature Structural Biology 4 (8) (1997) 644–649. doi:10.1038/nsb0897-644.

[28] F. M. Jucker, R. M. Phillips, S. A. McCallum, A. Pardi, Role of a heterogeneous free state in the formation of a specific RNA-theophylline complex., Biochemistry 42 (9) (2003) 2560–2567. doi:10.1021/bi027103+.

[29] P. Ceres, J. J. Trausch, R. T. Batey, Engineering modular ON RNA switches using biological components, Nucleic Acids Research 41 (22) (2013) 10449–10461. doi:10.1093/nar/gkt787.

[30] S. Badelt, S. Hammer, C. Flamm, I. L. Hofacker, Chapter Eight - Thermodynamic and Kinetic Folding of Riboswitches, in: S.-J. C. Burke-Aguero, D. H. (Eds.), Methods in Enzymology, Vol. 553 of Computational Methods for Understanding Riboswitches, Academic Press, 2015, pp. 193–213.

[31] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, D. H. Turner, Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure., Proc Natl Acad Sci U S A 101 (19) (2004) 7287–7292. doi:10.1073/pnas.0401799101.

[32] D. H. Turner, D. H. Mathews, NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure., Nucleic Acids Res 38 (Database issue) (2010) D280–D282. doi:10.1093/nar/gkp892.

[33] B. Sauerwine, M. Widom, Folding kinetics of riboswitch transcriptional terminators and sequesterers, Entropy 15 (8) (2013) 3088–3099. doi:10.3390/e15083088.

[34] G. Quarta, K. Sin, T. Schlick, Dynamic energy landscapes of riboswitches help interpret conformational rearrangements and function., PLoS Comput Biol 8 (2) (2012) e1002368. doi:10.1371/journal.pcbi.1002368.

[35] A. Reining, S. Nozinovic, K. Schlepckow, F. Buhr, B. Fürtig, H. Schwalbe, Three-state mechanism couples ligand and temperature sensing in riboswitches, Nature 499 (7458) (2013) 355–359. doi:10.1038/nature12378.

[36] C. Manz, A. Y. Kobitski, A. Samanta, B. G. Keller, A. Jschke, G. U. Nienhaus, Single-molecule FRET reveals the energy landscape of the full-length SAM-i riboswitch, Nature Chemical Biology 13 (11) (2017) 1172–1178. doi:10.1038/nchembio.2476.

[37] F. Kühnl, P. F. Stadler, S. Will, Tractable RNA–ligand interaction kinetics, BMC Bioinformatics 18 (S12) (2017) . doi:10.1186/s12859-017-1823-5.

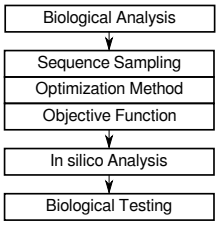
[38] I. L. Hofacker, C. Flamm, C. Heine, M. T. Wolfinger, G. Scheuermann, P. F. Stadler, BarMap: RNA folding on dynamic energy landscapes, RNA 16 (2010) 1308–1316.

Part III

UNPUBLISHED RESULTS AND DISCUSSION

9 SUMMARY AND STATE OF THE ART

Designing functional RiboNucleic Acid (RNA) molecules has a long history, starting from simple modifications such as mutational studies up to complete *de novo* designs of functional RNAs. Although the progress is slow, it is steady and includes many success stories and technical advances. Numerous great artificial devices were developed, summarized in Chapter 4. However, the applications are very divers and thus it is often hard to generalize and transfer the novelties and scientific findings from specific studies to other applications. This is especially true for experimental data as the performed experiments are often very specific to a particular use-case, are not standardized and thus the data is not comparable. Furthermore, many experiments are basic functional tests where not much can be learned in terms of mechanistic function of RNA molecules and riboswitches. Nevertheless, I collected the similarities of previous attempts and tried to extract important knowledge beneficial for future works. This includes many comparable technical and experimental methods. It was even possible to organize the performed tasks of any reviewed design study – theoretical or experimental – into a general multi-step design pipeline, see Chapter 5.



Most publications follow this common RNA design pipeline.

Until now, a noticeable gap can be observed between experimental studies that try to implement new RNA devices and theoretical methods which aim to utilize the predictive power of *in silico* calculations to accelerate the tedious laboratory work (see Section 4.4). *In silico* methods often do not solve the actual problems of experimentalists and are on top hard to adapt to changing needs. Moreover, many indispensable methods for a proper computational *de novo* design are still missing or need major improvement. This includes a solid high-throughput data acquisition and analysis pipeline for innate and designed RNA devices (see Section 5.1, Section 5.5), reliable methods to explore the sequence space (see Section 5.2), suitable optimization methods (see Section 5.3.1), and precise objective functions (see Section 5.3.2). During the last years I developed new programs and algorithms tackling some of these issues, see Part ii, with the focus on sequence sampling and applicability to further close the before mentioned gap and produce software that actually helps to *de novo* design context-sensitive RNA molecules with prescribed functionality.

10 CONTRIBUTIONS AND OUTLOOK

In this last part of my thesis I will highlight how my work contributed to the field of RNA design and how it solves crucial problems mentioned in Section 5.6. Furthermore, I will discuss how it might be possible to further advance the current approaches by improving many aspects of the outlined design pipeline (see Chapter 5), and how current scientific discoveries and techniques could be utilized in upcoming design studies. Therefore, I once more split up the process of designing RNA molecules into the various steps of the pipeline and will discuss them separately.

10.1 ANALYSIS OF BIOLOGICAL SYSTEMS

The ideally first step in an RNA design pipeline, the analysis of existing systems and building blocks, is unfortunately often missing or at least not clearly outlined in some publications. In these cases, it is unclear which prerequisites are necessary for the artificial RNA molecule to work and which assumptions were made during the design process. Other design studies are based on previous design attempts which can be interpreted as a detailed analysis and thus involve this step to some extent. An example for the latter would be Rodrigo et al. [154] with all its consecutive publications [153, 155, 156, 157, 158] which analyze previous attempts in detail to enhance the designs. Also the design by Wachsmuth et al. [191] was analyzed diligently in prospect to make sense of the experimental results [47, 192].

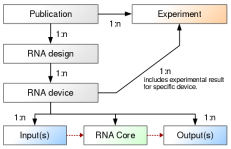
DETAILED RIBOSWITCH STUDIES How else can we increase our knowledge to understand the cellular processes and fundamental mechanisms involved in RNA based gene regulation? Recent findings and methods from research areas outside of the design field could contribute here. For example, Reining et al. [150] refined nuclear magnetic resonance spectroscopy (NMR) technology to examine ligand-RNA interactions in a time-resolved manner. Using this technique, they could analyze the adenine-sensing riboswitch in *Vibrio vulnificus* in astonishing detail and confirmed their results with a mathematical model describing the mechanistic function of this riboswitch. They found that the robust, temperature independent switching in this pathogenic bacterium is based on two ligand-free conformations. Thus, the authors discovered one of the first natural occurring three-state riboswitches [18, 79, 129, 150]. Similarly, Stagno et al. [175] used

"mix-and-inject" time-resolved serial crystallography to investigate the same adenine riboswitch and revealed that the reaction mechanism model even contains four states.

These and similar studies facilitate the design of artificial RNA devices, as they pave the way for finding the right objectives and constraints to construct novel molecules. Building upon this work by constructing a switch with the published model as objective would be a great future project to consider.

Moreover, the discovery of multi-state RNA switches highlights once more that sophisticated software able to deal with multi-structural constraints is necessary to be able to *de novo* design riboswitches of this kind. This motivated us to develop `RNAdesign` and its successor `RNAblueprint`, see Chapter 6 and Chapter 7.

RNADEVICES DATABASE In addition to advances in experimental data acquisition, also data analysis and storage contribute to such an initial knowledge acquisition step. Data is only valuable, if it is properly screened, annotated and standardized. Thus, I developed the *RNAdevices* database which attempts to collect, structure and characterize artificial RNA devices from previous design studies. The setup is similar to the `iGEM` database of biological parts[1] [174], but for complete devices made solely from RNA. By introducing standardized nomenclatures, collecting RNA sequences, experimental data, and mechanistic details in a structured and easy to understand way, such a database facilitates the process of learning from previous attempts. While designing the database structure, I introduced the distinction between an *RNAdesign* and an *RNAdevice*. An *RNAdesign* acts like a blueprint or plan for the artificial RNA molecule encompassing structure, mechanistic functionality, and the general design idea but not its actual sequence. On the other side, a *RNAdevice* is the actual implementation of this design and is determined by its RNA sequence.



The RNAdevices database introduces nomenclature and structures the elements of published artificial RNA devices.

In my opinion, this distinction is necessary in order to establish how RNA molecules should be designed. Instead of changing and mutating sequences which is an error prone process (see Section 4.1), researchers need to design RNA molecules *de novo*. A building plan which describes the riboswitch and contains necessary properties needs to be generated, followed by the modification of this plan to fit the new needs. In a real world analogy like houses, nobody would dare to make fundamental changes to a building without a plan encompassing static calculations.

Furthermore, the database structure splits each riboswitch into three distinct components: The *Input* which can be a ligand, temperature, or also a small RNA (sRNA), the *Core* which is the actual riboswitch usually located in the 5' untranslated region (5'UTR) and an *Output*. The latter indicates which signal is sent to a downstream pathway.

1 https://parts.igem.org/Catalog

Usually, gene regulation of a reporter gene such as green fluorescent protein (GFP) acts as am output signal of a RNA device. Complying strictly with this distinction adds significant improvements to the structure of such a database.

IN VIVO SELECTION FOR DATA ACQUISITION For an effective data acquisition it is helpful to not only have measurements from a couple of case studies but also from a collection of variants and mutants which might be working with different degrees of the desired function. This includes negative examples (which do not work properly) as they are of utter importance for the development of dedicated mechanistic models and even more due to the rise of novel deep learning approaches. However, the generation of numerous plasmids encoding for RNA molecules using cloning techniques and the successive analysis is often a tedious, time consuming task. Especially when performing experiments to reveal mechanistic properties of riboswitches, the work load strongly increases with the number of designs to test. Parallel measurement of a two-digit number of constructs is already hard to handle in the laboratory with a minimum of three technical and three biological replicates. Automated experimentation [96] could solve this problem, however acquiring such a machine is still a costly investment. Thus, in case of no or only little data available concerning a novel design idea, *in vivo* selection might be a good starting point for many design studies to acquire valuable data about the biological system and the involved processes.

Fowler et al. [53] established a fluorescence-activated cell sorting (FACS) based approach to select functional candidates from a library of devices with a random sequence part. Moreover, Schneider and Suess [164] showed that synthetic riboswitches can be obtained by *in vitro* selection with systematic evolution of ligands by exponential enrichment (SELEX) combined with a *in vivo* screening approach. Nevertheless, these studies terminate after obtaining operative RNA devices although such a selection pipeline could deliver valuable high-throughput biological data and thus knowledge about the properties of such devices. For example, after the sorting step by FACS, RNA sequencing (RNA-seq) could reveal the sequences of well and poorly performing candidates. Statistical analysis and correlations to predictive models will then tell the properties of e.g. a proper terminator, an aptamer or a switching region in this system. This information can be used as objectives in successive *de novo* designs.

10.2 DE NOVO SEQUENCE DESIGN OF MULTI-STATE RNA MOLECULES

Designing sequences *de novo* with prescribed properties is usually achieved by formulating an optimization approach consisting of *sequence sampling*, an *optimization method* and a relevant *objective function*

which describes the design in a mathematical way. Previous software solutions often offered these parts as a complete package. However, RNA design scenarios are very divers which requires an easy way to customize and adapt. Thus, these parts are decoupled and treated separately here.

10.2.1 *Sampling sequences for multi-state RNA devices*

The main focus of this work definitively lies on the sequence sampling part of the design approach. Initially, I began way down the pipeline and tried to develop an objective function with prescribed kinetic properties. However, it soon turned out that there were many prerequisites missing, so we needed to start further upstream. The proposed kinetic design was planned to have several structural constraints which will be reached sequentially in time. This required that we are able to sample RNA sequences that fulfill multiple structural constraints. However, the only available software packages able to achieve this, `Frnakenstein` [114] and later `MODENA` [182], were using ad hoc approaches for sequence sampling with an unspecified distribution. Thus, it was not guaranteed that we could efficiently explore the whole solution space, a feature necessary for optimization with a demanding objective function. This led to the development of `RNAdesign` and `RNAblueprint`.

RNADESIGN: UNIFORM SAMPLING RESPECTING MULTIPLE CONSTRAINTS In Chapter 6 we introduced an algorithm that allows to uniformly sample RNA sequences with respect to an arbitrary amount of target structures. It uses a dynamic programming approach on a graph structure. The dependency graph introduced by Flamm et al. [51] is decomposed by various graph algorithms such as the ear decomposition [145] in order to obtain a tree of subgraphs, where the leaves only contain fully connected subgraphs with degree of at most two, i.e. paths and cycles. By counting the number of possible sequence assignments for the leaves utilizing Fibonacci numbers and subsequentially concatenating these using simple combinatorics, it is possible to enumerate all possible sequence assignments for the complete dependency graph. Afterwards, dynamic programming (DP) allows for stochastic sampling to generate sequences in a fair way. The prototype implementation of `RNAdesign` [84] uses heuristic local optimization to obtain sequences whose energy landscapes are dominated by the prescribed target structures. We showed that the complexity of the algorithm heavily depends on the particular choice of the spanning tree used for the ear decomposition. However, finding the spanning tree with the least complexity for our DP algorithm is a hard problem. Thus, we solved a small optimization problem by randomly sampling spanning trees.

RNABLUEPRINT: FLEXIBLE SEQUENCE SAMPLING LIBRARY In Chapter 7 we extended the DP approach of RNAdesign to be valid for all decomposed components of the dependency graph and formulated the problem statement in a more generic manner. The prototype implementation of RNAdesign used lazy enumeration of all solutions to allow for uniform sampling. We found that it works sufficiently well for small problems but does not scale for many and complex constraints. RNAblueprint [74] is thus implemented exhaustively as a library in C++ with SWIG interfaces for Perl and Python. Utilizing a library allows to reuse the generated DP tables for successive sampling procedures which saves most of the computational work.

Furthermore, the library interface enables to incorporate any evaluation package in the objective function, which immediately extends the functionality to pseudoknotted structures e.g. by using the NUPACK package [215] in the objective function. Moreover, it opens doors to new possibilities for the development of diverse objective functions and to benchmark various optimization methods. Allowing an easy adaption of existing code to accommodate new scenarios makes the design process very flexible.

Furthermore, RNAblueprint introduced several new local move-steps based on the dependency graph. *P-local* and *C-local* allow small moves by only changing a subset of the current sequence. Another move to sample a completely new sequence independent of sampling history is called *global*. Extensive benchmarks compare these moves and show that uniform sampling brings significant advantages over ad hoc methods like MODENA or Frnakenstein as it allows to explore bigger parts of the sequence space in same amount of time.

RNAREDPRINT: MULTI-TARGET BOLTZMANN SAMPLING In the previous publications, the problem of how to obtain sequences from the solution space with an uniform distribution was solved. However, it would be advantageous to not only address the *specificity problem*, but also the *affinity problem* to some extent already in the sampling step. While specificity leads to a restriction of the solution space by fulfilling structural constraints, affinity will raise the probability of finding desired solutions, see Section 5.2 for more details. Sampling towards a Boltzmann distribution as introduced by IncaRNAtion [149] for single structural constraints would increase the probability of solutions that exhibit a low free energy of the target structures. Extending this approach to multiple structural constraints is achieved analogous to the counting in **RNAblueprint!** (**RNAblueprint!**). However, instead of memorizing counts in the DP tables, we are generalizing the approach by summing up partition functions over all sequences given the structural constraints. Note, that this partition function is different to the one used by folding prediction algorithm.

This requires the inclusion of an RNA energy model (see Section 3.1) into the algorithm. Depending on the choice of the energy model, this can bring more complex dependencies specified by the structural constraints, such as stacking, dangling ends or loop contributions. Thus, the dependency graph framework is not sufficient any more. In a subsequent publication not included in this thesis [75], we therefore introduced the RNA design network. It basically consists of tuples of a set of sequence positions and a set of functions specifying various energy contributions for these positions. Thus, a network of dependencies is created which can be efficiently resolved by a tree decomposition over the dependency graph. Finally, we can use DP to calculate the partition function of the target structures over all sequences which can then direct a stochastic backtracking to achieve Boltzmann sampling of sequences.

However, as Boltzmann sampling prefers solutions with low target energies, it also introduces a bias towards G-C rich sequences. To counteract this problem, we also introduce a way to control the G-C content similar to the approach by Reinharz et al. [149].

In summary, the new algorithm of `RNARedPrint` solves several open issues of the previous algorithm. It shows the #P-hardness of uniform sampling and introduces Boltzmann sampling and thus affinity towards desirable solutions. Moreover, utilizing a tree decomposition algorithm is equivalent to finding the most efficient graph decomposition, an approach that was initially found for the decomposition of the dependency graph. Therefore, our problem becomes fixed-parameter tractable with its complexity growing exponentially only with the treewith of the decomposition.

We are currently working on incorporating the novelties of the `RNARedPrint` into the `RNAblueprint` library in order to combine the advantages of both approaches. Due to the RNA design network, it is also possible to include more complex arbitrary constraints. Some ideas of current developments and future implementations are outlined hereafter.

SAMPLING WITH ADVANCED SEQUENCE CONSTRAINTS The current implementation of `RNAblueprint` allows to specify sequence constraints for each individual position in `IUPAC` notation [89]. This means that the choice for the assignment at each position can be restricted to one or several nucleotides, the latter with equal probability for each nucleotide involved (when disregarding structural constraints). However, there are many cases where the inclusion of more advanced sequence constraints in addition to the structural constraints might be very useful.

The simplest extension probably is constraints with prescribed probabilities which specify a sampling distribution for individual positions. Thus, it is possible to specify e.g. an A-U rich region, a G-

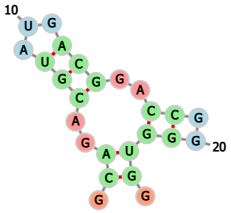
C rich stem structure or even distributions described by sequence logos [165]. It is much more difficult to allow for sequence constraints which depend on multiple positions at once, especially as structural constraints and uniform or Boltzmann sampling also need to be taken into account. The dependency graph (G) only allowed to specify base-pairing constraints between nucleotides or sequence constraints for single nucleotide positions. However, the new RNA design network concept is expressive enough to allow for such constraints and we are currently working on an implementation. In this network it is possible to define an arbitrary constraint for a collection of nucleotide positions by specifying a function describing this constraint. Such advanced constraints make it possible to sample from a library of specific sequence motifs or sequence patterns, e.g. from a library of ribosome binding site (RBS) sequences. Another application would be sampling of sequences with prescribed structural constraints within a coding region of an open reading frame (ORF). An application of such functionality could be to artificially design SECIS RNA hairpins in the selenoprotein coding frame [99] similar to the SECISdesigner by Backofen and Busch [8] [20]. This requires that various triplets of nucleotides encoding for the same amino acid sequence are sampled with respect to the given base-pairing pattern. Additionally, it is possible to choose codons with certain probabilities such as their species dependent codon frequency. Garcia-Martin et al. [59] suggested to take the similarity score of the `BLOSUM62` matrix into account to also allow the insertion of amino acids with comparable properties.

Equally important as positive sequence constraints are negative constraints. Currently, we are filtering out sampled sequences that include forbidden motifs such as restriction cut sites, binding sites or polynucleotide pattern, which is quite inefficient. The RNA design network framework now enables us to forbid certain sequence motifs directly in the sampling engine to save a lot of post-processing effort.

VARIABLE STRUCTURAL CONSTRAINTS Another highly desired extension to the current set of possible sampling inputs would be variable structural constraints. This means that the lengths of structural features such as stems, loops or unpaired regions can be specified within desired ranges. As these lengths infer the energy contribution of these substructures to the free energy of the specified state, changing their sizes could have dramatic impact on thermodynamic and especially on kinetic properties. More flexibility in this context also loosens the constraints, leading to a growing solution space and thus better solutions during the optimization approach. When thinking about automatic ways to detect the optimal lengths of structural features, multiple ways are possible to achieve this.

A straightforward approach is to exhaustively enumerate the specified structural inputs, generate the corresponding dependency graphs, and calculate the DP matrices for all of them. An introduced move-set between the dependency graphs, e.g. allowing the insertion/deletion of single base-pairs, relates these graphs to each other. During the sequence optimization, it is then possible to sample from any of the dependency graphs by moving between the various structural inputs with certain probabilities. Thus, in addition to an optimal sequence also the best combination of structural features will be found. We already applied this technique for instances with little structural variability, however due to the vast amount of combinations, enumerating all dependency graphs soon becomes infeasible.

Alternatively, it is also possible to alter the structural constraints in place with certain probabilities during the optimization. Currently, this would need a reevaluation of all the DP matrices, but with minimal changes it should be not too hard to detect the minimal amount of depending matrices which need to be recalculated on certain structural changes. However, this effort is correlated with the connectivity of the dependency graph, i.e., the higher the degree of connectedness, the more computational effort is needed.




```
.((..(((...)))
..((..)))).

(U)((U2)((U3)P3)
(U2)((U2)P2)P2)(U)

(((((H3)S3)
((H2)S2)M4)S2)E2)
```

Example of a structure and its according HIT representations, full (middle) and coarse grained (bottom).

Lastly, variable structural constraints could also be realized as an initial optimization approach on the structural level. By calculating the partition function over all sequences regarding the target structures, it is possible to obtain the Boltzmann weight of the current set of structures. Using this information, an optimization problem over the structures could be solved to maximize these Boltzmann weights. Thus, an estimate for the average energy of the given structure over all sequences can be calculated and the optimal combination of structural features for the desired problem can be found.

The required input notation for such a purpose could be easily adopted from the homeomorphically irreducible tree (HIT) representation specified by `RNAdistance` [80]. In this representation, the identifiers `(U)` and `(P)` for unpaired and paired bases are used. There also exist a coarse grained representations where hairpins, interior loops, bulges, multi-loops, stacks and external bases are represented by `(H)`, `(I)`, `(B)`, `(M)`, `(S)` and `(E)`, respectively. For both representations variable lengths could easily be specified next to the letter code.

NEGATIVE STRUCTURAL CONSTRAINTS Forbidding undesirable structures with hard constraints is a challenging task as potentially any base-pair in the pairing matrix can be formed along the whole sequence. There are simply just too many combinations of disallowed base-pairs and a too small alphabet of nucleotides to be able to achieve such hard constraints. Nevertheless, for some cases it is reasonable to use negative constraints, for instance in order to prevent the elongation of specified loop or stem lengths or to disallow specific

off-target binding sites between multiple RNA strands. In these cases, smart rules which enforce incompatible bases at specified positions ensure proper structure formation.

Preventing specific off-target stems or other substructures could of course also be achieved by hard constraints. However, this does not guarantee that this substructure will come up on a slightly different length or shifted position as we cannot prevent base-pairing with such a small alphabet of nucleotides and dense pairing matrix ($\mathcal{P}$) by using constraints only.

Thus, another method is to penalize any undesired structure in the objective function by utilizing a multi-structure version of the ensemble defect (see Section 5.3.2). In case of Boltzmann sampling it would be even possible to specify a corresponding sampling objective. Therefore, the Boltzmann distributed sampling would prefer sequences with low ensemble defect, meaning that sequences which mainly fold into structures with little distance to the targets are preferred. In order to enable such a sampling objective, it must be possible calculate the ensemble defect over all sequences given the target structures. It is an open question if this is possible by e.g. using DP.

RESOLVE CONFLICTING BASE-PAIRS A practical feature that is currently not included in any implementation is to automatically resolve the appearance of conflicting base-pairs in multi-state constraints. When specifying several structures that a sequence should be able to adopt, it may happen that incompatible base-pairs are introduced. This can be easily detected, as the resulting dependency graph is no longer bipartite. However, especially when automatically generating these constraints, e.g. when enumerating variable constraints, we need a way to resolve these conflicts. In Chapter 6 we suggest an objective for solving a graph editing problem. Alternatively, it might also be possible to remove the minimal set of conflicting edges by solving the NP-hard graph bipartization problem. Hüffner [86] examined several algorithms based on branch-and-bound or integer linear programming (ILP) and also present a new algorithm based on iterative compression (IC) which could be utilized for our application.

MORE FINE TUNED STRUCTURAL INPUTS Serganov and Nudler [167] found that the three major types of architectures in natural occurring riboswitches are the straight junctional fold, the inverse junctional fold, and pseudoknot fold, all of which contain important tertiary interactions. Moreover, non-Watson-Crick base pairs play a central role in important 3D motifs such as hairpins, internal and junction loops. However, these motifs appear single-stranded in standard RNA secondary structure predictions. Stombaugh et al. [177] put these statements into numbers and derived the base-pair occurrence frequencies for 12 geometric family motifs.

This indicates that non-Watson-Crick pairs and tertiary interactions should be included as design goals for functional RNA molecules. So how is it possible to include such information in RNA design? Recently, Lotfi et al. [112] developed a design strategy which implicitly includes such information. They increase their success rate by using selective 2'-hydroxyl acylation analyzed by primer extension (SHAPE) data to guide their designs, which implicitly contains information about non-canonical and tertiary interactions.

However, it would be even better to explicitly allow structural constraints with a more fine-grained structural model such as the Leontis-Westhof nomenclature for non-canonical base-pairs [104]. This nomenclature contains much more structural information about the reactive elements of functional motifs, which usually appear single stranded in conventional secondary structures. Miao and Westhof [128] reviewed these structural elements and showed how they contribute to the overall 3D structure of RNA. Bida and Das [13] found that including RNA motifs as loop types in design approaches helps to solve the negative design problem and thus to get rid of undesired alternative structures. This works as these 3D motifs support a specific 2D structure due to their non-canonical interactions and geometric shape. Thus, they tend to preserve their behavior even when included into a new context. In contrast, unexpected tertiary interactions and non-canonical base-pairs could significantly change the energy landscape. Especially in riboswitches, where two energetically similar conformations are designed, this could easily destroy the desired switching capabilities.

Although there already exist several algorithms and programs that are able to predict extended RNA secondary structures, e.g. `RNAwolf` [217] or `RMDetect` [32], accurate energy parameters and exhaustive experimental data sets are unfortunately missing for non-canonical interactions. Thus, it is still challenging to include such information in our design approach, despite all the advantages for RNA design. Nevertheless, to circumvent this obstacle it might be helpful to directly include 3D structure prediction at least as a *in silico* analysis step into the pipeline. Coarse-grained models of tertiary structures, e.g. those predicted by `Ernwin` [95], could for example be used to confirm the desired tertiary structure [184].

10.2.2 *Optimization Methods*

Although various different optimization methods were used in RNA design applications (see Section 5.3.1), an extensive study comparing their influence on the quality of the solutions and the efficiency of the optimization is to my knowledge still missing in the context of RNA design. Previous published programs are quite diverse, making it hard to narrow down the benefits or failures from a benchmark to

a specific detail of the procedure. Thus, benchmarking only the optimization method, sequence generation, move-step or objective function was simply not possible.

With `RNAblueprint` we made it possible to exchange any part of the optimization approach which enabled us to compare the run-time and solution quality of a Monte Carlo simulated annealing (MCSA) with a simple adaptive walk approach, see Chapter 7 and its supplement. Additionally, it would be interesting to test uniform sampling with further optimization strategies, such as genetic algorithms or multi-objective optimization.

Not only the optimization method, but also the chosen move-set can heavily influence the resulting solution. Several studies show the theoretical background on how sequence space and structure space are connected [69, 70, 148, 166] and how the sequence space can be traversed. However, a detailed investigation of the solution space from the RNA design perspective could bring new insights and might lead to the development of move-sets being more meaningful for design applications.

DEEP LEARNING Instead of following the process of gathering statistical data about riboswitches, developing a mathematical model and optimizing towards this goal manually, it might be advantageous to apply modern deep learning techniques [29]. Nevertheless, this does not spare the work of acquiring meaningful biological data, developing a learning and sampling technique and it probably brings analogous problems and pitfalls.

10.2.3 *New Objectives to design RNA sequences*

In this section, I want to present some ideas of building blocks that might be useful in future objective functions. This will not include ready-made complete functions which can be used out of the box, as for any design problem valid objective functions should be derived from biological data. For instance, from a statistical analysis of a big dataset with working and non-working devices, a initial *in vivo* or *in vitro* selection pipeline such as proposed in Section 10.3, or an iterative design process where the results of the last iteration will influence the objectives of the next step. It is always important to analyze existing systems first and then develop a mathematical model that exactly describes the properties and mechanism of the desired RNA molecule.

For reliably designing functional RNA molecules with a big variety of features, it must be possible to include versatile and modular objectives. The program `RNAdesign` already supported the inclusion of several functions and measurements from a defined catalog in order to achieve this. With `RNAblueprint` we went one step further and

now use a scripting interface which brings maximal flexibility in the choice of programs and prediction software. Thus, I was able to include several measurements for the first time in objective functions as outlined below.

PROBABILITIES OF STRUCTURAL ELEMENTS Instead of only using design goals concerning the full sequence, it is also possible to specify certain properties for particular parts or modules of the RNA molecule. The stability and thus the probability of a certain stem structure, e.g. a terminator, can be included into the objective function. Such measurements have already been calculated in previous studies but to my knowledge have never been used as terms in objective functions [65, 87, 143].

In Chapter 8 we developed the objective for a simple ligand-sensing riboswitch that is able to adapt two different conformations depending on the presence of the ligand. This design serves as a template and can be adapted to perform regulatory tasks in a host cell such as translational or transcriptional regulation of a downstream target gene. The objective consists of three terms, which describe the probability of structural features in the ensemble of structures. In the ligand negative state, the aptamer stem and the alternative stem structure must be present in a certain ratio, while in the ligand bound conformation, the probability of the aptamer must be maximized.

These probabilities are calculated using the hard constraint framework of the `ViennaRNA` package [109], as described in Equation 5, Section 3.2. Using this framework, it is also possible to predict the accessibility of a certain region to ensure that it is single-stranded, or if the contrary is desired, that it is captured in a stable structure. Furthermore, an interesting application would also be to calculate the probability of a certain region to only pair with upstream or downstream nucleotides. This could ensure that the coding region is not interfering with the switching capabilities of an 5'UTR encoded riboswitch.

LIGAND AND PROTEIN BINDING Similarly, ligand binding was already included in some calculations [143, 191, 192], but never included as an optimization goal. In such a case, a detailed understanding of the underlying features and mechanisms of the natural aptamer is needed. This includes knowledge about the binding energy, ligand competent structure or plasticity of the binding pocket. We need to measure and validate these properties of the system and in case they deviate from the applied model, the *in silico* system has to be adjusted [47, 48].

Given this data, it is possible to utilize the `ViennaRNA` soft constraints framework to model ligand binding by adding an energy contribution to a specific sequence/structure motif [109]. In our con-

tribution, see Chapter 8, we use this feature to model the binding of theophylline to the aptamer.

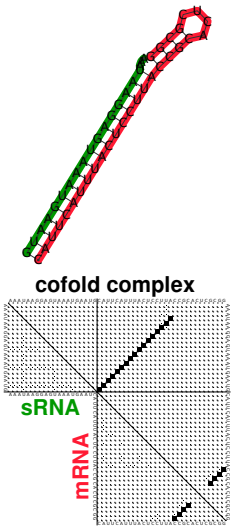
With the framework it should also be possible to model protein binding which allows to *de novo* design RiboNucleoProtein (RNP) switches, such as the one manually designed by Saito et al. [161]. This riboswitch works in such a way that the ribosomal protein L7Ae interacts with the box C/D kink-turn motif which was placed directly downstream of the reporter gene start codon. Binding of the protein then leads to a reduction of gene expression.

However, as we pointed out in Chapter 8, the soft-constraint framework enforces various simplifying assumptions in our model. In reality, the binding pocket exhibits plasticity to some extend which leads to varying binding energies for various structural conformations of the binding pocket. In contrast, our model only has one structural motif which is assigned a fixed energy contribution. Furthermore, we are assuming infinite reaction rates and also neglect the effects of varying ligand concentrations. Additionally, our knowledge about the binding properties in the natural system is often not sufficient for being able to model these interactions. This includes detailed information about the structure of the binding pocket, differences at specific salt conditions, pH values or temperatures, or exact measurements of the K_d values and interaction rates. A major problem for the latter is that *in vitro* aptamer testing happens with much higher ligand concentrations than those possible in cellular systems. Thus, the measurements are often of limited use.

KINETIC DESIGNS A *de novo* design is only successful if the design criteria are fully met and all nucleotides behave as intended, especially those in crucial structural regions such as terminator structures or RBS sites. This also includes that timely aspects need to be satisfied, kinetic properties and co-transcriptional folding support the design goals, and do not interfere with the designed mechanism. Nevertheless, kinetic and co-transcriptional effects were often missing during the design and analysis steps of previous studies.

As the mechanistic understanding of riboswitches grows due to more recent studies [73, 79, 144, 200, 211] I am confident that this situation will change. These and other findings suggest that there are two major types of switching behaviors [144]. One type is capable of switching during the entire lifetime of the molecule, which leads to fast response times but fuzzy individual states and background activity. The other type exhibits a fixed, stabilized state after a certain time of sensing disregarding any input change. The latter has the advantage of obtaining distinct states and very little background noise, in exchange for long response times as switching is only possible through RNA decay and anew transcription. For more information see Chapter 8.

Obviously, it would be ideal to include kinetic models describing such RNA folding mechanisms or also ligand interactions into the objective function. However, exactly these calculations are often very demanding and thus cannot be iteratively evaluated as it would be necessary in an optimization approach. In Chapter 8 we solved this issue by developing a thermodynamic design objective which optimizes for a specific kinetic behavior. The latter is finally verified in our extensive *in silico* analysis pipeline. Furthermore, we are currently working on a cotranscriptional folding approach which is sufficiently fast to be included in objective functions [9].



Dot-plot of a sRNA (green) triggered OFF switch (red). Upper triangle shows the complex, lower the individual molecules.

CONCENTRATIONS OF STRUCTURAL ELEMENTS In this paragraph I want to shortly present a new idea to design a translational OFF switch, where a sRNA triggers the direct sequestering of the RBS in the 5'UTR of a messenger RNA (mRNA). The objective function for such a translational OFF switch requires the inclusion of concentrations to be able to model this two molecule reaction. Findeiß et al. [49] recently suggested to use concentrations in objective functions calculated by `RNAcofold` [108]. This idea here goes one step further and proposes to use the relative abundance of structural elements in the ensemble of folding molecules as a design criterion. This means, we maximize the relative concentration of the complex structure $S_{complex}$ which resembles the binding sites of the sRNA and the 5'UTR in the example.

$$f(x) = [S_{complex}]/[mRNA_0] \tag{9}$$

It maximizes the amount of mRNA molecules being caught in a proper sRNA/mRNA complex with respect to $[mRNA_0]$, the input concentration of the mRNA. Here we assume that the sRNA is present in excess (100:1). Thus, the concentration of the complex only depends on the mRNA concentration. This assumption is comparable to high- and low-copy plasmid systems used to analyze regulatory effects of a sRNA and (potential) mRNA targets [188].

The concentration of $S_{complex}$ is equal to the concentration of all states exhibiting sRNA/mRNA interactions [complex] multiplied by the probability that the desired complex structure is indeed formed $P(S_{complex})$.

$$\begin{aligned} [S_{complex}] &= [complex] \cdot P(S_{complex}) \\ P(S_{complex}) &= \frac{Z_{S_{complex}}}{Z'} \\ Z' &= Z - Z_{mRNA} Z_{sRNA} \end{aligned}$$

While [complex] can be calculated by `RNAcofold` and the partition function of all states exhibiting the structural element $Z_{S_{complex}}$ by using the hard constraint framework of `ViennaRNA` [109] as outlined in

Section 3.2, Z' – the partition function of all structures actually forming inter-molecular base-pairs between the two molecules – cannot be directly obtained. Thus, we derive it by subtracting the individual partition functions Z_{mRNA} and Z_{sRNA} from the `RNAcofold` partition function Z, which includes all species.

By calculating the accessibility (see Section 3.2) we additionally ensure that the RBS is unstructured in the unbound state in order to allow ribosome binding and thus gene expression. As soon as the sRNA binds to the 5'UTR, this region is blocked and thereby translation is turned off.

The usage of concentrations for cofolding has the advantage that it comprises additional negative design goals. For example, it ensures that monomers and homo-dimers are efficiently avoided. Note that `NUPACK:TestTubeDesign` [205] also developed concentration dependent objectives where multiple states are achieved by modeling multiple RNA strands in various test tubes describing a single complex-state each. Thus, a thermodynamic switching behavior within one test tube would not be possible. However, the approach described in Equation 9 could be used for a true multi-state design application, where the riboswitch adopts more than two states.

EXTENSIVE MODELS IN OBJECTIVE FUNCTIONS In many cases not only the inclusion of concentrations is necessary for a sufficient description of the riboswitch mechanism. Mathematical modeling to explore the dynamics of riboswitch refolding, response curves, ligand concentrations or even describe cellular mechanisms like transcriptional termination, translational repression or RNA degradation might be advantageous. For instance, Beisel and Smolke [10] describe models for transcription, translation or ribozymal cleavage and explain in detail how ligand concentration influences protein output depending on various rates in the system.

Based on the work of Carothers et al. [23] we developed ordinary differential equation (ODE) models describing the effect of transcriptional and translational ON and OFF switches in order to use them in future objective functions. These are based on the core model depicted in Figure 8. These models are publicly available on the version control system Github[2] in various formats such as SBML, which allows for effortless extensions and future developments.

In this context, I want to point out that it would be very interesting to include the 3-state riboswitch model by Reining et al. [150] as an objective function as well for the *de novo* design of novel RNA devices of this type. This experimental publication already brings detailed kinetic measurements and thus serves as a great starting point for a successive design [18, 22, 79, 150]. Another linear model by Cambray et al. [21] was obtained by systematically measuring termi-

2 https://github.com/ribonets/rnadev-models

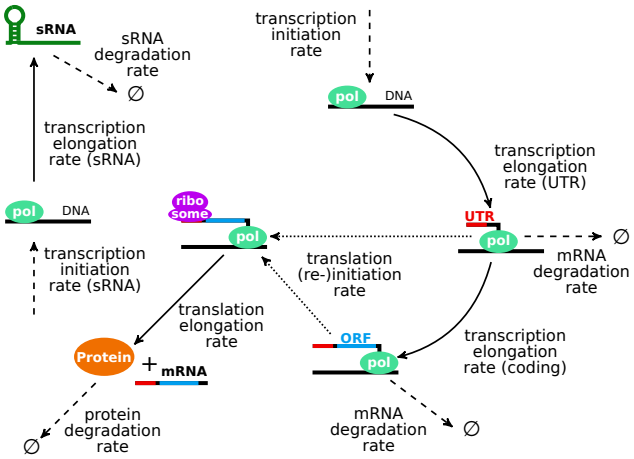


Figure 8: Graphical representation of the ODE core system to model transcription and translation of a reporter gene. After polymerase initiation (top right), transcription will produce the 5'UTR and later the full mRNA including its open reading frame (ORF). Both species will be degraded again. Translation initiation and elongation (middle) leads to the production of the reporter protein. Additionally, a regulatory sRNA is produces (top left), which is used to model sRNA triggered transcription or translation elongation in the extended models.

nation efficiencies of various intrinsic terminators. This model might also improve the engineering of synthetic systems containing intrinsic terminators by including the quality of the terminator into the design objective.

ENVIRONMENTAL FACTORS Folding prediction models and therefore the objectives of artificial riboswitches sometimes ignore important environmental aspects. This includes ion and salt concentrations influencing RNA folding, such as magnesium or sodium. Although a measured parameter set is available [187] and `NUPACK` [215] is already making use of it, `ViennaRNA` [108] is not able to deal with such environmental parameters. A different story is the inclusion of the pH value of the solution as there simply is no energy parameter set is available.

In *in vivo* systems, a big impact probably also comes from chaperons, such as HFQ/HF-I protein (HFQ), which might heavily influence RNA folding. Also the consequences of unspecific or targeted degradation of RNA due to ribonucleases (RNases) should not be underestimated and should be included in *in silico* models.

An environmental factor which is almost always taken care of is temperature. It can even be explicitly used as a trigger for switching molecules, which are then called thermoswitches. While the *de novo* design of bistable thermoswitches was possible before [51, 160, 196], using `RNAblueprint`, `ViennaRNA` and a multi-structure version of the temperature objective function by Flamm et al. [51] it is now trivial

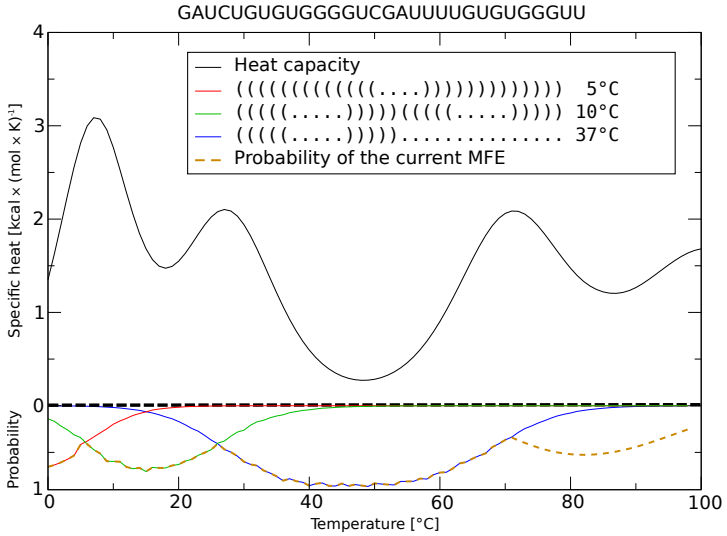


Figure 9: Multi-state thermoswitch designed with `RNAblueprint` and `ViennaRNA` package. The designed molecules exhibit the given MFE structures (red, green and blue line) at the specified temperatures (lower plot). Folding predictions at all temperatures from 0.0° to 100.0° Celsius show that the first structural change happens at ~ 7.0 degree Celsius and the second one at ~ 26.0° Celsius. Above ~ 72.0° Celsius, the sequence occurs only in the open chain conformation. The `RNAheat` heat capacity plot (upper) further confirms that the designed sequence is indeed a three-stable thermoswitch.

to generate sequences that even fold into multiple states at different temperatures, see Figure 9.

As you can see, there seems to be endless possibilities of formulating design goals. While some of them might be able to describe the natural system quite extensively, others will fail in doing so. Thus, it is necessary to come up with many novel ideas of models and correlate them to biological data in order to pick the right ones leading to high success rates of *de novo* design.

10.3 Analysis and Filtering of Potential Solutions

The previously solved optimization problems do not lead to perfect solutions, as not all aspects can be included in the objective function. Therefore, it is better to generate bigger amounts of design candidates and filter them for the desired features. This filtering step in the pipeline aims to narrow down this list to a number able to handle in the laboratories. Ideally, it should include sophisticated methods to cluster, sort and compare the designed devices in order to pick the best performing ones for biological analysis. Several studies already included many varying predictions, tools and calculations to achieve this task (see Section 5.4). In this section, I want to shortly introduce our approaches used to screen the generated artificial designs and how we selected the most promising candidates.

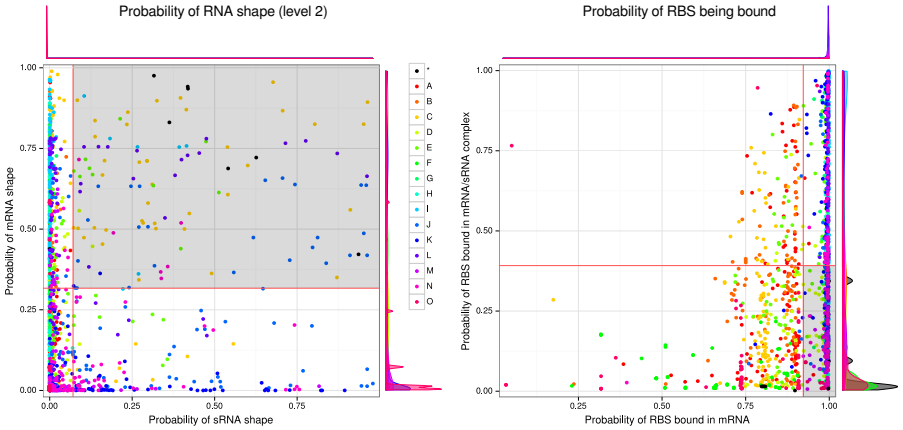


Figure 10: Plots showing essential features of design candidates, such as the probability of the target RNA shapes (left) and the probability of the RBS being accessible in presence and absense of the trigger sRNA (right). Ideally, the probabilities of both shapes, sRNA (x-axis) and mRNA (y-axis), should reach 1.00. The probability of the RBS being bound should be 1.00 in absence of the sRNA and 0.00 in the sRNA/mRNA complex. It is immediately evident which goals could not be effectively reached during the optimization procedure, e.g. that the RBS is bound in the complex. Thus, we removed candidates outside of the gray areas (gated by the median values, red lines) during this step.

KINETIC CALCULATIONS Sometimes, the methods to determine some design goals are too demanding to be included into the objective function. In such a case, it is necessary to come up with similar, feasible objectives and do a detailed *in silico* analysis afterwards to detect designs which do not exhibit the required properties. In Chapter 8 we suggested to use a thermodynamic model as the optimization goal to design a kinetic riboswitch mechanism. This required the development of a testing pipeline to confirm the kinetic properties of the produced candidates. We applied `treekin` [207] to detect undesired kinetic folding traps and to verify the process of ligand binding by using an absorbing states model, see Chapter 8. The latter could still be improved by extending ligand binding dynamics following Kühnl et al. [101].

FILTER FOR ESSENTIAL PROPERTIES For the design of a sRNA triggered translational OFF switch, we performed a simple filtering step to exclude designs which did not behave precisely as desired. We defined three requirements which need to be met by any working design. The RBS should be accessible on in the 5'UTR of the mRNA, binding of the sRNA should sequester the RBS in the OFF state, and the desired structures of sRNA and mRNA or at least abstractions of them [61] should dominate the equilibrium ensemble. Therefore we calculated the probability of the RBS being bound with and without sRNA as shown in Equation 5, Section 3.2. Furthermore, we predicted the probability of the desired RNA shapes [60, 90] for the sRNA as

well as the mRNA. Then, we filtered according to these criteria by using the median of all candidates as a threshold value, see Figure 10. These cut-offs can of course be adjusted to restrict the number of designs even more to only include the most promising candidates. As we used various different objective functions (A-O), the colors directly indicate which objectives performed better. As a positive control, we also plotted designs with a similar mechanism by Isaacs et al. [87] marked with an asterisk (black, Figure 10).

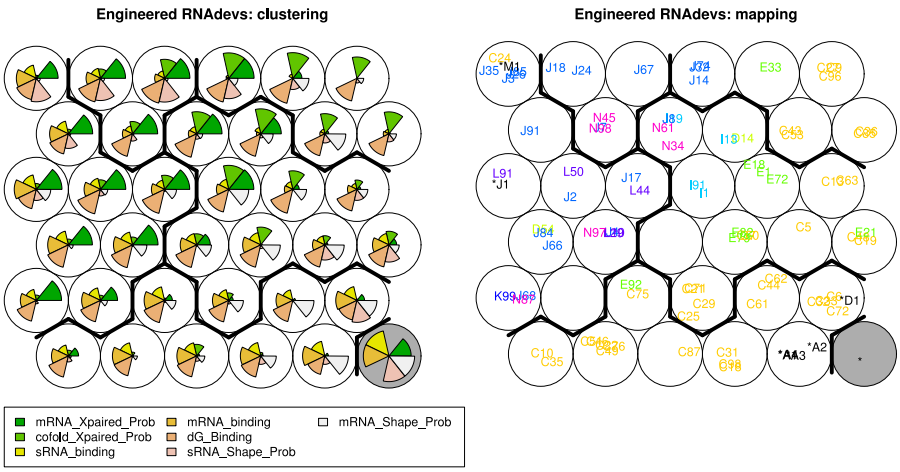


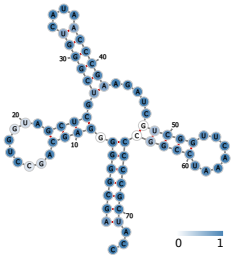
Figure 11: SOM representation of designs similar to the *taR12-crR12* system by Isaacs et al. [87]. The original designs are included in black, marked with an asterisk in the mapping plot (right). The cluster on the bottom right only contains the perfect feature vector which was included as reference (gray). The clustering representation (left) visualizes the composition of the feature vector of the according cluster and thus enables immediate comparison of numerous candidates. The feature vector contains numerous features, such as the probability of the RBS to be bound in the mRNA (mRNA_Xpaired_Prob), or a score over all positions of the sRNA binding site which indicates if they are bound to the correct positions of the mRNA (sRNA_binding).

MACHINE LEARNING AND CLUSTERING TECHNIQUES Filtering performs well for excluding undesired designs not exhibiting any essential properties. However, we also need methods to compare candidates concerning multiple optional features. Maybe we also want to detect candidates with similar features to be able to send a quite divers set of RNA devices to the laboratory for testing. Thus, we applied self-organized maps (SOMs) [201] to cluster similar designs according to their features. An artificial perfect feature vector, i.e. having the best achievable value of each feature, is included to obtain a gradient according to the desired features. The clustering of the SOM (Figure 11) immediately reveals which candidates exhibit a feature vector close to the desired goals. Similar designs are captured in one cluster with their feature vector visualized. This enabled us

to efficiently explore the properties of the design candidates and pick promising candidates for experimental analysis.

VISUALIZATION METHODS AND TOOLS FOR DATA EXPLORATION

Sophisticated visualization tools might facilitate this analysis and filtering step a lot. This includes the visualization of SOM clusters, Pareto fronts for multi-objective optimizations or also of RNA properties such as energy landscapes and structure ensemble distributions. I participated in the development of a web-based tool for displaying RNA secondary structures, called `forna` [94]. This tool creates interactive plots of sequences and secondary structures in a clean and concise way. It can handle pseudoknotted structures, allows for interactive editing and additional information can be overplayed by the highly customizable coloring layer.



Forna plot of the E. coli tRNAfMet A1-U72 secondary structure with base-pair probabilities in blue.

This tool led to valuable succeeding developments, such as an interactive dot-plot viewer [93] which is able to display many properties important for RNA design in one representative plot. This includes the ensemble of highly populated structures with their probability, structural features and their localization in the different structures, the accessibility of the individual positions and the base-pair probabilities as usual for dot-plot representations. A similar dot-plot representation was recently proposed by Wiegreffe et al. [202].

Another tool based on `forna` is the `DrTransformer` visualization [9, 93], see Figure 12. This interactive visualization of the structural ensemble during cotranscriptional folding is a great tool for the analysis of transcriptional riboswitches. Moreover, the introduced time-plot on the bottom if Figure 12 could be used for multiple other applications. For example, when plotting temperature instead of time on the x-axis, this plot transforms into a tool for exploring temperature sensitivity of an RNA molecule.

In conclusion, this filtering and comparison step of the design pipeline still requires many manual steps, from running all the calculations and predictions to assembling tables and plots in order to compare the data. Automatic analysis pipelines specialized to particular design scenarios would speed up this part and make it more efficient.

10.4 BIOLOGICAL TESTING

Analyzing and verifying the artificial RNA devices in biological experiments is expensive and time-consuming as it involves many manual tasks. Although high-throughput measurement methods are on the rise, including fluorescence measurements using flow cytometry or many methods utilizing next generation sequencing, the necessary initial synthesis of DesoxyriboNucleic Acid (DNA) strands serving as templates for the designed RNA devices is still expensive. Thus, testing bigger numbers of potential candidates for their functionality is

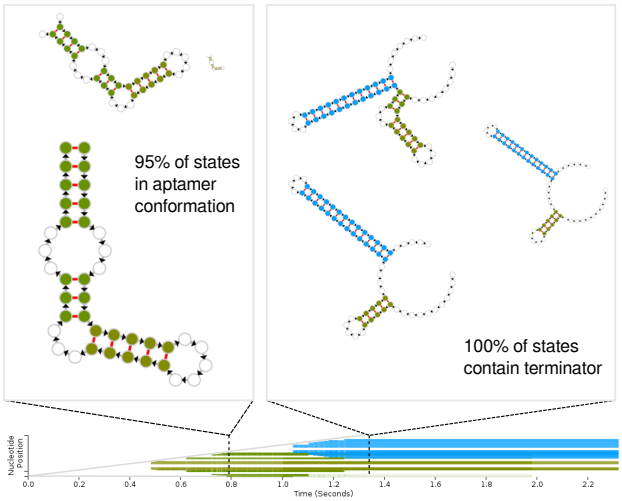


Figure 12: Interactive visualization of the structural ensemble during the cotranscriptional folding of a transcriptional riboswitch. Structural elements such as the aptamer (green) and the terminator stem (blue) are automatically highlighted. The progress plot on the bottom indicates at which time the structural elements appear and disappear in the ensemble. After an initial aptamer formation (left), this riboswitch will always switch into the terminator hairpin state (right) without the addition of the ligand.

just not feasible. The ideal solution to this problem would be to use sophisticated computational methods to *de novo* design these functional RNA molecules with high success rates. Unfortunately, these methods are not yet good enough to obtain such accurate predictions. At the same time, big amounts of standardized biological data of many varying devices is needed to enhance the predictive power of the *in silico* methods.

A solution to this chicken-and-egg dilemma to obtain data of many varying devices in a fast and cheap way might be to generate or mutate sequences randomly and experimentally select those with the right sequence or function from this pool. In my opinion, such *in vivo* or *in vitro* selection pipelines are probably an indispensable tool in this field until we are able to *de novo* design RNA molecules with prescribed properties *in silico* with high accuracy.

Moreover, we need to focus more on detailed analyses which reveal mechanistic details of the riboswitch function, the structure to function relationship, and their interaction with ligands, chaperons, RNA biding proteins and other components of the cellular environment. This valuable information is required to build or enhance our models of the involved mechanisms and thus lead to better predictions in future design studies. The often performed experiments that just verify a functional switching behavior and the presence of RNA or reporter protein, will unfortunately not contribute to this process. However, in many cases it would be possible to increase the knowledge gained by quite simple experiments. For example, in Section 5.5.3 I mentioned

how to perform valuable kinetic measurements and gain mechanistic insights by basic experiments and I am confident that with a smart experimental setup and well-designed controls much more information can be obtained.

Nevertheless, it is obvious that experimental work is complex and demanding starting from planning the setup until the analysis of the obtained data. Even for a rather easy design model as the one proposed in Chapter 8 it is challenging to develop an experimental setup which is able to verify the desired target ratio between the two structural conformations in the equilibrium. Approaches such as Förster resonance energy transfer (FRET) and NMR were used before to determine structural states and the energy landscape of riboswitches. However, these are quite extensive and elaborate techniques. Thus, I am looking forward to experimental advances such as the technique by Watters et al. [200], which is able to get timely resolved structural information and even measure co-transcriptional folding kinetics using SHAPE-seq.

In any case, a stable experimental testing pipeline – either *in vitro*, *in vivo* or ideally both – which interacts with the *in silico* design models is not only desired, but necessary for efficient generation of artificial RNA devices, cellular logic gates or whole biological circuits.

11 CONCLUDING REMARKS

At this point I want to thank you, the reader, for staying with me for such a long essay and hope you enjoyed reading this thesis. It might easily be that you are from the field and knew most of the content already. However, I hope that you have also learned something new and are now able to see particular things differently. Maybe some parts of this thesis even helped you to gain important knowledge and advance your research. However, be warned that some statements might even be affirmatively wrong (no illusions, this really happens!) although I honestly did my best to avoid such obstacles. In any case, in my view this dissertation ended up being a quite extensive summary of the current state of the art regarding the *de novo* design of multi-state RNA molecules, such as riboswitches, thermoswitches or sRNA triggered RNA devices. Furthermore, I summarized my contribution to this field of research and highlighted the open questions and future tasks. Hopefully, some of these will be answered and accomplished soon, either by me, our research group, or anyone else from the wonderful RNA research community. Maybe even by you? I am definitely curious about all the new findings!

ACRONYMS

DNA	DesoxyriboNucleic Acid
dsDNA	double stranded DNA
cDNA	complementary DNA
RNA	RiboNucleic Acid
dsRNA	double stranded RNA
sRNA	small RNA
mRNA	messenger RNA
rRNA	ribosomal RNA
tRNA	transfer RNA
tmRNA	transfer-messenger RNA
snRNA	small nuclear RNA
UTR	untranslated region
5′UTR	5′ untranslated region
TIR	translation initiation region
RBS	ribosome binding site
SD	Shine-Dalgarno sequence
aSD	anti-Shine-Dalgarno sequence
ORF	open reading frame
RNApol	RNA polymerase
TF	transcription factor
SNP	single-nucleotide polymorphism
GFP	green fluorescent protein
RNase	ribonuclease
HFQ	HFQ/HF-I protein
RNP	RiboNucleoProtein
Θ	secondary structure

$\mathfrak{x}$	RNA sequence
Ω	sequence space
$\overline{\Omega}$	structure space
$\mathcal{A}$	alphabet of nucleotides
$\mathcal{B}$	set of base pairs
$\mathcal{P}$	pairing matrix
G	dependency graph
N	RNA design network
Q	partition function
MFE	minimum free energy
DP	dynamic programming
CP	constraint programming
ODE	ordinary differential equation
LNS	large neighborhood search
MCMC	Markov chain Monte Carlo
MCSA	Monte Carlo simulated annealing
EA	evolutionary algorithm
GA	genetic algorithm
MOGA	multi-objective genetic algorithm
ILP	integer linear programming
IC	iterative compression
HIT	homeomorphically irreducible tree
SOM	self-organized map
qPCR	quantitative real-time polymerase chain reaction
PCR	polymerase chain reaction
cPCR	competitive polymerase chain reaction
rcPCR	real competitive polymerase chain reaction
RNA-seq	RNA sequencing
MALDI-TOF	matrix-assisted laser desorption/ionization-time-of-flight

PAGE polyacrylamide gel electrophoresis

ELISA enzyme-linked immunosorbent assay

FACS fluorescence-activated cell sorting

ITC isothermal titration calorimetry

NMR nuclear magnetic resonance spectroscopy

FRET Förster resonance energy transfer

AFM atomic force microscopy

SHAPE selective 2′-hydroxyl acylation analyzed by primer extension

EMSA electrophoretic mobility shift assay

SELEX systematic evolution of ligands by exponential enrichment

BIBLIOGRAPHY

[1] Ingrid Abfalter, Christoph Flamm, and Peter F. Stadler. Design of Multi-Stable Nucleic Acid Sequences. volume 1, pages 1–7, Muenchen, 2003. LNCS.

[2] Ingrid G. Abfalter. *Nucleic Acid Sequence Design As A Graph Colouring Problem*. Dissertation, University of Vienna, November 2005.

[3] Bree B. Aldridge, Julio Saez-Rodriguez, Jeremy L. Muhlich, Peter K. Sorger, and Douglas A. Lauffenburger. Fuzzy Logic Analysis of Kinase Pathway Crosstalk in TNF/EGF/Insulin-Induced Signaling. *PLOS Computational Biology*, 5(4):e1000340, April 2009. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000340.

[4] Uri Alon. Network motifs: Theory and experimental approaches. *Nature Reviews Genetics*, 8(6):450–461, June 2007. ISSN 1471-0056. doi: 10.1038/nrg2102.

[5] J C Alwine, D J Kemp, and G R Stark. Method for detection of specific RNAs in agarose gels by transfer to diazobenzyloxymethyl-paper and hybridization with DNA probes. *Proceedings of the National Academy of Sciences of the United States of America*, 74(12):5350–5354, December 1977. ISSN 0027-8424.

[6] Mirela Andronescu, Anthony P. Fejes, Frank Hutter, Holger H. Hoos, and Anne Condon. A New Algorithm for RNA Secondary Structure Design. *Journal of Molecular Biology*, 336(3): 607–624, February 2004. ISSN 0022-2836. doi: 10.1016/j.jmb.2003.12.041.

[7] Assaf Avihoo, Alexander Churkin, and Danny Barash. RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences. *BMC Bioinformatics*, 12(1):319, August 2011. ISSN 1471-2105. doi: 10.1186/1471-2105-12-319.

[8] Rolf Backofen and Anke Busch. Computational Design of New and Recombinant Selenoproteins. In *Combinatorial Pattern Matching*, Lecture Notes in Computer Science, pages 270–284. Springer, Berlin, Heidelberg, July 2004. ISBN 978-3-540-22341-2 978-3-540-27801-6. doi: 10.1007/978-3-540-27801-6_20.

[9] Stefan Badelt. *Control of RNA Function by Conformational Design*. PhD Thesis, University of Vienna, Wien, 2016.

[10] Chase L. Beisel and Christina D. Smolke. Design Principles for Riboswitch Function. *PLoS Comput Biol*, 5(4):e1000363, April 2009. doi: 10.1371/journal.pcbi.1000363.

[11] Christian Berens and Beatrix Suess. Riboswitch engineering — making the all-important second and third steps. *Current Opinion in Biotechnology*, 31(Supplement C):10–15, February 2015. ISSN 0958-1669. doi: 10.1016/j.copbio.2014.07.014.

[12] Stephan H. Bernhart, Hakim Tafer, Ulrike Mückstein, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms for Molecular Biology*, 1(1):3, March 2006. ISSN 1748-7188. doi: 10.1186/1748-7188-1-3.

[13] JP Bida and R Das. Squaring theory with practice in RNA design. *Current Opinion in Structural Biology*, 22(4):457–466, August 2012. ISSN 0959-440X. doi: 10.1016/j.sbi.2012.06.003.

[14] Kenneth F. Blount and Ronald R. Breaker. Riboswitches as antibacterial drug targets. *Nature Biotechnology*, 24(12):1558–1564, December 2006. ISSN 1087-0156. doi: 10.1038/nbt1268.

[15] Sergei Borukhov and Evgeny Nudler. RNA polymerase: The vehicle of transcription. *Trends in Microbiology*, 16(3):126–134, March 2008. ISSN 0966-842X. doi: 10.1016/j.tim.2007.12.006.

[16] Ronald R. Breaker. Natural and engineered nucleic acids as tools to explore biology. *Nature*, 432(7019):838–845, December 2004. ISSN 0028-0836. doi: 10.1038/nature03195.

[17] Douglas F. Browning and Stephen J. W. Busby. The regulation of bacterial transcription initiation. *Nature Reviews Microbiology*, 2(1):57–65, January 2004. ISSN 1740-1526. doi: 10.1038/nrmicro787.

[18] Janina Buck, Boris Fürtig, Jonas Noeske, Jens Wöhnert, and Harald Schwalbe. Time-resolved NMR methods resolving ligand-induced RNA folding at atomic resolution. *Proceedings of the National Academy of Sciences*, 104(40):15699–15704, February 2007. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0703182104.

[19] Anke Busch and Rolf Backofen. INFO-RNA—a fast approach to inverse RNA folding. *Bioinformatics*, 22(15):1823–1831, January 2006. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btl194.

[20] Anke Busch, Sebastian Will, and Rolf Backofen. SECISDesign: A server to design SECIS-elements within the coding sequence.

Bioinformatics, 21(15):3312–3313, August 2005. ISSN 1367-4803. doi: 10.1093/bioinformatics/bti507.

[21] Guillaume Cambray, Joao C. Guimaraes, Vivek K. Mutalik, Colin Lam, Quynh-Anh Mai, Tim Thimmaiah, James M. Carothers, Adam P. Arkin, and Drew Endy. Measurement and modeling of intrinsic transcription terminators. *Nucleic Acids Research*, page gkt163, March 2013. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkt163.

[22] Song Cao, Boris Fürtig, Harald Schwalbe, and Shi-Jie Chen. Folding Kinetics for the Conformational Switch between Alternative RNA Structures. *The Journal of Physical Chemistry B*, 114(42):13609–13615, October 2010. ISSN 1520-6106. doi: 10.1021/jp107912s.

[23] James M. Carothers, Jonathan A. Goler, Darmawi Juminaga, and Jay D. Keasling. Model-Driven Engineering of RNA Devices to Quantitatively Program Gene Expression. *Science*, 334 (6063):1716–1719, December 2011. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1212209.

[24] Thomas R. Cech. The RNA Worlds in Context. *Cold Spring Harbor Perspectives in Biology*, 4(7):a006742, January 2012. ISSN , 1943-0264. doi: 10.1101/cshperspect.a006742.

[25] Pablo Ceres, Jeremiah J. Trausch, and Robert T. Batey. Engineering modular 'ON' RNA switches using biological components. *Nucleic Acids Research*, 41(22):10449–10461, January 2013. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkt787.

[26] James Chappell, Melissa K. Takahashi, and Julius B. Lucks. Creating small transcription activating RNAs. *Nature Chemical Biology*, 11(3):214–220, March 2015. ISSN 1552-4450. doi: 10.1038/nchembio.1737.

[27] James Chappell, Kyle E Watters, Melissa K Takahashi, and Julius B Lucks. A renaissance in RNA synthetic biology: New mechanisms, applications and tools for the future. *Current Opinion in Chemical Biology*, 28:47–56, October 2015. ISSN 1367-5931. doi: 10.1016/j.cbpa.2015.05.018.

[28] Feng Chen, Allyson Evans, Elizabeth Gaskell, John Pham, and Miao-Chih Tsai. Regulatory RNA: The New Age. *Molecular Cell*, 43(6):851–852, September 2011. ISSN 1097-2765. doi: 10.1016/j.molcel.2011.09.001.

[29] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque,

Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. Wide & Deep Learning for Recommender Systems. *arXiv:1606.07792 [cs, stat]*, June 2016.

[30] Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, and Danny Barash. Design of RNAs: Comparing programs for inverse RNA folding. *Briefings in Bioinformatics*, page bbw120, January 2017. ISSN 1467-5463, 1477-4054. doi: 10.1093/bib/bbw120.

[31] Peter Clote. Expected degree for RNA secondary structure networks. *Journal of Computational Chemistry*, 36(2):103–117, January 2015. ISSN 1096-987X. doi: 10.1002/jcc.23776.

[32] José Almeida Cruz and Eric Westhof. Sequence-based identification of 3D structural modules in RNA with RMDetect. *Nature Methods*, 8(6):513–519, June 2011. ISSN 1548-7105. doi: 10.1038/nmeth.1603.

[33] Gerrit M Daubner, Antoine Cléry, and Frédéric H-T Allain. RRM–RNA recognition: NMR or crystallography...and new findings. *Current Opinion in Structural Biology*, 23(1):100–108, February 2013. ISSN 0959-440X. doi: 10.1016/j.sbi.2012.11.006.

[34] Alexandre Dawid, Bastien Cayrol, and Hervé Isambert. RNA synthetic biology inspired from bacteria: Construction of transcription attenuators under antisense regulation. *Physical Biology*, 6(2):025007, June 2009. ISSN 1478-3975. doi: 10.1088/1478-3975/6/2/025007.

[35] Katherine E. Deigan, Tian W. Li, David H. Mathews, and Kevin M. Weeks. Accurate SHAPE-directed RNA structure determination. *Proceedings of the National Academy of Sciences of the United States of America*, 106(1):97–102, January 2009. ISSN 0027-8424. doi: 10.1073/pnas.0806929106.

[36] Murray P. Deutscher. Degradation of RNA in bacteria: Comparison of mRNA and stable RNA. *Nucleic Acids Research*, 34(2):659–666, 2006. ISSN 0305-1048. doi: 10.1093/nar/gkj472.

[37] Chunming Ding and Charles R. Cantor. A high-throughput gene expression analysis technique using competitive PCR and matrix-assisted laser desorption ionization time-of-flight MS. *Proceedings of the National Academy of Sciences*, 100(6):3059–3064, March 2003. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0630494100.

[38] Robert M. Dirks, Milo Lin, Erik Winfree, and Niles A. Pierce. Paradigms for computational nucleic acid design. *Nucleic Acids Research*, 32(4):1392–1403, January 2004. ISSN 0305-1048. doi: 10.1093/nar/gkh291.

[39] Ivan Dotu, William A. Lorenz, Pascal Van Hentenryck, and Peter Clote. Computing folding pathways between RNA secondary structures. *Nucleic Acids Research*, 38(5):1711–1722, March 2010. ISSN 0305-1048. doi: 10.1093/nar/gkp1054.

[40] Ivan Dotu, Juan Antonio Garcia-Martin, Betty L. Slinger, Vinodh Mechery, Michelle M. Meyer, and Peter Clote. Complete RNA inverse folding: Computational design of functional hammerhead ribozymes. *Nucleic Acids Research*, 42(18): 11752–11762, October 2014. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gku740.

[41] Pan Du, Jian Gong, E. S. Wurtele, and J. A. Dickerson. Modeling gene expression networks using fuzzy logic. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1351–1359, December 2005. ISSN 1083-4419. doi: 10.1109/TSMCB.2005.855590.

[42] A. Esmaili-Taheri, M. Ganjtabesh, and M. Mohammad-Noori. Evolutionary Solution for the RNA Design Problem. *Bioinformatics*, page btu001, January 2014. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu001.

[43] Ali Esmaili-Taheri and Mohammad Ganjtabesh. ERD: A fast and reliable tool for RNA design including constraints. *BMC Bioinformatics*, 16:20, 2015. ISSN 1471-2105. doi: 10.1186/s12859-014-0444-5.

[44] Amin Espah Borujeni and Howard M Salis. Translation Initiation is Controlled by RNA Folding Kinetics via a Ribosome Drafting Mechanism. *Journal of the American Chemical Society*, 138(22):7016–7023, June 2016. ISSN 0002-7863. doi: 10.1021/jacs.6b01453.

[45] Amin Espah Borujeni, Anirudh S. Channarasappa, and Howard M. Salis. Translation rate is controlled by coupled trade-offs between site accessibility, selective RNA unfolding and sliding at upstream standby sites. *Nucleic Acids Research*, 42(4):2646–2659, February 2014. ISSN 0305-1048. doi: 10.1093/nar/gkt1139.

[46] Amin Espah Borujeni, Dennis M. Mishler, Jingzhi Wang, Walker Huso, and Howard M. Salis. Automated physics-based design of synthetic riboswitches from diverse RNA aptamers. *Nucleic Acids Research*, 44(1):1–13, January 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv1289.

[47] Maja Etzel and Mario Mörl. Synthetic Riboswitches: From Plug and Pray toward Plug and Play. *Biochemistry*, 56(9):1181–

1198, March 2017. ISSN 0006-2960. doi: 10.1021/acs.biochem.6b01218.

[48] Sven Findeiß, Manja Wachsmuth, Mario Mörl, and Peter F. Stadler. Chapter One - Design of Transcription Regulating Riboswitches. In Donald H. Burke-Aguero, editor, *Methods in Enzymology*, volume 550 of *Riboswitches as Targets and Tools*, pages 1–22. Academic Press, 2015.

[49] Sven Findeiß, Maja Etzel, Sebastian Will, Mario Mörl, and Peter F. Stadler. Design of Artificial Riboswitches as Biosensors. *Sensors*, 17(9):1990, August 2017. doi: 10.3390/s17091990.

[50] Christoph Flamm, W Fontana, I L Hofacker, and P Schuster. RNA folding at elementary step resolution. *RNA*, 6(3):325–338, March 2000. ISSN 1355-8382.

[51] Christoph Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. *RNA*, 7(2): 254–265, January 2001. ISSN 1355-8382, 1469-9001.

[52] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier Trees of Degenerate Landscapes. *Zeitschrift für Physikalische Chemie*, 216(2-2002):155, February 2002. ISSN 0942-9352. doi: 10.1524/zpch.2002.216.2.155.

[53] Casey C. Fowler, Eric D. Brown, and Yingfu Li. A FACS-Based Approach to Engineering Artificial Riboswitches. *ChemBioChem*, 9(12):1906–1911, 2008. ISSN 1439-7633. doi: 10.1002/cbic.200700713.

[54] M. J. Fulwyler. Electronic Separation of Biological Cells by Volume. *Science*, 150(3698):910–911, November 1965. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.150.3698.910.

[55] Boris Fürtig, Christian Richter, Jens Wöhnert, and Harald Schwalbe. NMR Spectroscopy of RNA. *ChemBioChem*, 4(10): 936–962, 2003. ISSN 1439-7633. doi: 10.1002/cbic.200300700.

[56] Justin P Gallivan. Toward reprogramming bacteria with small molecules and RNA. *Current Opinion in Chemical Biology*, 11(6): 612–619, December 2007. ISSN 1367-5931. doi: 10.1016/j.cbpa.2007.10.004.

[57] James ZM Gao, Linda YM Li, and Christian M. Reidys. Inverse folding of RNA pseudoknot structures. *Algorithms for Molecular Biology*, 5(1):27, June 2010. ISSN 1748-7188. doi: 10.1186/1748-7188-5-27.

[58] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. RNAiFold: A constraint programming algorithm for RNA inverse folding and molecular design. *Journal of Bioinformatics and Computational Biology*, 11(02):1350001, November 2012. ISSN 0219-7200. doi: 10.1142/S0219720013500017.

[59] Juan Antonio Garcia-Martin, Ivan Dotu, and Peter Clote. RNAiFold 2.0: A web server and software to design custom and Rfam-based RNA molecules. *Nucleic Acids Research*, 43(W1): W513–W521, January 2015. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkv460.

[60] Robert Giegerich and Björn Voß. RNA Secondary Structure Analysis Using Abstract Shapes. In Roland K. Hartmann, Albrecht Bindereif, Astrid Schön, and Eric Westhof, editors, *Handbook of RNA Biochemistry*, pages 579–594. Wiley-VCH Verlag GmbH & Co. KGaA, 2014. ISBN 978-3-527-64706-4.

[61] Robert Giegerich, Björn Voß, and Marc Rehmsmeier. Abstract shapes of RNA. *Nucleic Acids Research*, 32(16):4843–4851, January 2004. ISSN 0305-1048. doi: 10.1093/nar/gkh779.

[62] Michael Gilman. Ribonuclease Protection Assay. In *Current Protocols in Molecular Biology*. John Wiley & Sons, Inc., 2001. ISBN 978-0-471-14272-0. doi: 10.1002/0471142727.mb0407s24.

[63] Susan Gottesman. Micros for microbes: Non-coding regulatory RNAs in bacteria. *Trends in Genetics*, 21(7):399–404, July 2005. ISSN 0168-9525. doi: 10.1016/j.tig.2005.05.008.

[64] Susan Gottesman, Colleen McCullen, Maude Guillier, Carin Vanderpool, Nadim Majdalani, Jihane Benhammou, Karl Thompson, Peter FitzGerald, Nathaniel Sowa, and David FitzGerald. Small RNA Regulators and the Bacterial Response to Stress. *Cold Spring Harbor symposia on quantitative biology*, 71: 1–11, 2006. ISSN 0091-7451. doi: 10.1101/sqb.2006.71.016.

[65] Alexander A. Green, Pamela A. Silver, James J. Collins, and Peng Yin. Toehold Switches: De-Novo-Designed Regulators of Gene Expression. *Cell*, 159(4):925–939, November 2014. ISSN 0092-8674. doi: 10.1016/j.cell.2014.10.002.

[66] Alexander A. Green, Jongmin Kim, Duo Ma, Pamela A. Silver, James J. Collins, and Peng Yin. Complex cellular logic computation using ribocomputing devices. *Nature*, 548(7665):117, August 2017. ISSN 1476-4687. doi: 10.1038/nature23271.

[67] Sandra J. Greive and Peter H. von Hippel. Thinking quantitatively about transcriptional regulation. *Nature Reviews Molecular Cell Biology*, 6(3):221–232, March 2005. ISSN 1471-0072. doi: 10.1038/nrm1588.

[68] Florian Groher and Beatrix Suess. Synthetic riboswitches — A tool comes of age. *Biochimica et Biophysica Acta (BBA) - Gene Regulatory Mechanisms*, 2014. ISSN 1874-9399. doi: 10.1016/j.bbagrm.2014.05.005.

[69] W. Grüner, R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster. Analysis of RNA sequence structure maps by exhaustive enumeration I. Neutral networks. *Monatshefte für Chemie / Chemical Monthly*, 127(4):355–374, April 1996. ISSN 0026-9247, 1434-4475. doi: 10.1007/BF00810881.

[70] W. Grüner, R. Giegerich, D. Strothmann, C. Reidys, J. Weber, I. L. Hofacker, P. F. Stadler, and P. Schuster. Analysis of RNA sequence structure maps by exhaustive enumeration II. Structures of neutral networks and shape space covering. *Monatshefte für Chemie / Chemical Monthly*, 127(4):375–389, April 1996. ISSN 0026-9247, 1434-4475. doi: 10.1007/BF00810882.

[71] Claudio O. Gualerzi and Cynthia L. Pon. Initiation of mRNA translation in bacteria: Structural and dynamic aspects. *Cellular and Molecular Life Sciences*, 72:4341–4367, 2015. ISSN 1420-682X. doi: 10.1007/s00018-015-2010-3.

[72] Christine E. Hajdin, Stanislav Bellaousov, Wayne Huggins, Christopher W. Leonard, David H. Mathews, and Kevin M. Weeks. Accurate SHAPE-directed RNA secondary structure modeling, including pseudoknots. *Proceedings of the National Academy of Sciences*, 110(14):5498–5503, February 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1219988110.

[73] Andrea Haller, Marie F. Soulière, and Ronald Micura. The Dynamic Nature of RNA as Key to Understanding Riboswitch Mechanisms. *Accounts of Chemical Research*, 44(12):1339–1348, December 2011. ISSN 0001-4842. doi: 10.1021/ar200035g.

[74] Stefan Hammer, Birgit Tschiatschek, Christoph Flamm, Ivo L. Hofacker, and Sven Findeiß. RNAblueprint: Flexible multiple target nucleic acid sequence design. *Bioinformatics*, 33(18):2850–2858, September 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx263.

[75] Stefan Hammer, Yann Ponty, Wei Wang, and Sebastian Will. Fixed-Parameter Tractable Sampling for RNA Design with Multiple Target Structures. April 2018.

[76] Helen G Hansma, Kenichi Kasuya, and Emin Oroudjev. Atomic force microscopy imaging and pulling of nucleic acids. *Current Opinion in Structural Biology*, 14(3):380–385, June 2004. ISSN 0959-440X. doi: 10.1016/j.sbi.2004.05.005.

[77] Dieter Hartz, David S. McPheeters, Robert Traut, and Larry Gold. Extension inhibition analysis of translation initiation complexes. In *Methods in Enzymology*, volume 164 of *Ribosomes*, pages 419–425. Academic Press, January 1988. doi: 10.1016/S0076-6879(88)64058-4.

[78] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970. ISSN 0006-3444, 1464-3510. doi: 10.1093/biomet/57.1.97.

[79] Christina Helmling, Anna Wacker, Michael T. Wolfinger, Ivo L. Hofacker, Martin Hengesbach, Boris Fürtig, and Harald Schwalbe. NMR Structural Profiling of Transcriptional Intermediates Reveals Riboswitch Regulation by Metastable RNA Conformations. *Journal of the American Chemical Society*, 139(7):2647–2656, February 2017. ISSN 0002-7863. doi: 10.1021/jacs.6b10429.

[80] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie / Chemical Monthly*, 125(2):167–188, February 1994. ISSN 0026-9247, 1434-4475. doi: 10.1007/BF00818163.

[81] Ivo L. Hofacker, Christoph Flamm, Christian Heine, Michael T. Wolfinger, Gerik Scheuermann, and Peter F. Stadler. BarMap: RNA folding on dynamic energy landscapes. *RNA*, 16(7):1308–1316, January 2010. ISSN 1355-8382, 1469-9001. doi: 10.1261/rna.2093310.

[82] S. R. Holbrook, E. L. Holbrook, and H. E. Walukiewicz. Crystallization of RNA. *Cellular and Molecular Life Sciences CMLS*, 58(2):234–243, February 2001. ISSN 1420-682X, 1420-9071. doi: 10.1007/PL00000851.

[83] Kerry Hollands, Sergey Proshkin, Svetlana Sklyarova, Vitaly Epshtein, Alexander Mironov, Evgeny Nudler, and Eduardo A. Groisman. Riboswitch control of Rho-dependent transcription termination. *Proceedings of the National Academy of Sciences*, 109 (14):5376–5381, March 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1112211109.

[84] Christian Höner zu Siederdissen, Stefan Hammer, Ingrid Abfalter, Ivo L. Hofacker, Christoph Flamm, and Peter F. Stadler. Computational design of RNAs with complex energy landscapes. *Biopolymers*, 99(12):1124–1136, 2013. ISSN 1097-0282. doi: 10.1002/bip.22337.

[85] Wilf T. Horn, Máire A. Convery, Nicola J. Stonehouse, Chris J. Adams, Lars Liljas, Simon E. V. Phillips, and Peter G. Stock-

ley. The crystal structure of a high affinity RNA stem-loop complexed with the bacteriophage MS2 capsid: Further challenges in the modeling of ligand–RNA interactions. *RNA*, 10 (11):1776–1782, January 2004. ISSN 1355-8382, 1469-9001. doi: 10.1261/rna.7710304.

[86] Falk Hüffner. Algorithm Engineering for Optimal Graph Bipartization. *Journal of Graph Algorithms and Applications*, 13(2): 77–98, 2009. ISSN 1526-1719. doi: 10.7155/jgaa.00177.

[87] Farren J. Isaacs, Daniel J. Dwyer, Chunming Ding, Dmitri D. Pervouchine, Charles R. Cantor, and James J. Collins. Engineered riboregulators enable post-transcriptional control of gene expression. *Nature Biotechnology*, 22(7):841–847, July 2004. ISSN 1087-0156. doi: 10.1038/nbt986.

[88] Farren J. Isaacs, Daniel J. Dwyer, and James J. Collins. RNA synthetic biology. *Nature Biotechnology*, 24(5):545–554, May 2006. ISSN 1087-0156. doi: 10.1038/nbt1208.

[89] IUPAC-IUB Comm. on Biochem. Nomenclature (CBN). Abbreviations and symbols for nucleic acids, polynucleotides, and their constituents. *Biochemistry*, 9(20):4022–4027, September 1970. ISSN 0006-2960. doi: 10.1021/bi00822a023.

[90] Stefan Janssen and Robert Giegerich. The RNA shapes studio. *Bioinformatics*, 31(3):423–425, January 2015. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu649.

[91] H. Joos, B. Timmerman, M. Van Montagu, and J. Schell. Genetic analysis of transfer and stabilization of Agrobacterium DNA in plant cells. *The EMBO Journal*, 2(12):2151–2160, 1983. ISSN 0261-4189.

[92] Fiona M. Jucker, Rebecca M. Phillips, Scott A. McCallum, and Arthur Pardi. Role of a Heterogeneous Free State in the Formation of a Specific RNA-Theophylline Complex†. *Biochemistry*, 42(9):2560–2567, March 2003. ISSN 0006-2960. doi: 10.1021/bi027103+.

[93] Peter Kerpedjiev. *Seeing Secondary, Sampling Tertiary*. PhD Thesis, University of Vienna, Wien, 2016.

[94] Peter Kerpedjiev, Stefan Hammer, and Ivo L. Hofacker. Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. *Bioinformatics*, 31(20):3377–3379, June 2015. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btv372.

[95] Peter Kerpedjiev, Christian Höner Zu Siederdissen, and Ivo L. Hofacker. Predicting RNA 3D structure using a coarse-grain

helix-centered model. *RNA*, April 2015. ISSN 1355-8382, 1469-9001. doi: 10.1261/rna.047522.114.

[96] Ross D. King, Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N. Soldatova, Andrew Sparkes, Kenneth E. Whelan, and Amanda Clare. The Automation of Science. *Science*, 324(5923):85–89, April 2009. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1165620.

[97] Robert Kleinkauf, Torsten Houwaart, Rolf Backofen, and Martin Mann. antaRNA – Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. *BMC Bioinformatics*, 16:389, 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0815-6.

[98] Jens Kortmann and Franz Narberhaus. Bacterial RNA thermometers: Molecular zippers and switches. *Nature Reviews Microbiology*, 10(4):255–265, April 2012. ISSN 1740-1526. doi: 10.1038/nrmicro2730.

[99] Alain Krol. Evolutionarily different RNA motifs and RNA–protein complexes to achieve selenoprotein synthesis. *Biochimie*, 84(8):765–774, August 2002. ISSN 0300-9084. doi: 10.1016/S0300-9084(02)01405-0.

[100] Jens Kroll, Stefan Klinter, Cornelia Schneider, Isabella Voß, and Alexander Steinbüchel. Plasmid addiction systems: Perspectives and applications in biotechnology. *Microbial biotechnology*, 3(6):634–657, November 2010. ISSN 1751-7915. doi: 10.1111/j.1751-7915.2010.00170.x.

[101] Felix Kühnl, Peter F Stadler, and Sebastian Will. Tractable RNA–ligand interaction kinetics. 2017.

[102] Heidi Ledford. CRISPR: Gene editing is just the beginning. *Nature News*, 531(7593):156, March 2016. doi: 10.1038/531156a.

[103] Jeehyung Lee, Wipapat Kladwang, Minjae Lee, Daniel Cantu, Martin Azizyan, Hanjoo Kim, Alex Limpaecher, Sungroh Yoon, Adrien Treuille, and Rhiju Das. RNA design rules from a massive open laboratory. *Proceedings of the National Academy of Sciences*, page 201313039, January 2014. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1313039111.

[104] N. B. Leontis and E. Westhof. Geometric nomenclature and classification of RNA base pairs. *RNA*, 7(4):499–512, January 2001. ISSN 1355-8382, 1469-9001.

[105] Alex Levin, Mieszko Lis, Yann Ponty, Charles W. O'Donnell, Srinivas Devadas, Bonnie Berger, and Jérôme Waldispühl. A

global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Research*, 40(20):10041–10052, January 2012. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gks768.

[106] Efthimia Lioliou, Cédric Romilly, Pascale Romby, and Pierre Fechter. RNA-mediated regulation in bacteria: From natural to artificial systems. *New Biotechnology*, 27(3):222–235, July 2010. ISSN 1871-6784. doi: 10.1016/j.nbt.2010.03.002.

[107] Ronny Lorenz. *RNA Secondary Structure Thermodynamics and Kinetics*. Doctoral Thesis, University of Vienna, Vienna, 2014.

[108] Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26, November 2011. ISSN 1748-7188. doi: 10.1186/1748-7188-6-26.

[109] Ronny Lorenz, Ivo L. Hofacker, and Peter F. Stadler. RNA folding with hard and soft constraints. *Algorithms for Molecular Biology*, 11:8, 2016. ISSN 1748-7188. doi: 10.1186/s13015-016-0070-z.

[110] Ronny Lorenz, Dominik Luntzer, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. SHAPE directed RNA folding. *Bioinformatics*, 32(1):145–147, January 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btv523.

[111] Ronny Lorenz, Michael T. Wolfinger, Andrea Tanzer, and Ivo L. Hofacker. Predicting RNA secondary structures from sequence and probing data. *Methods*, 103:86–98, July 2016. ISSN 1046-2023. doi: 10.1016/j.ymeth.2016.04.004.

[112] Mohadeseh Lotfi, Fatemeh Zare-Mirakabad, and Soheila Montaseri. RNA design using simulated SHAPE data. *Genes & Genetic Systems*, advpub, 2017. doi: 10.1266/ggs.16-00067.

[113] Julius B. Lucks, Lei Qi, Vivek K. Mutalik, Denise Wang, and Adam P. Arkin. Versatile RNA-sensing transcriptional regulators for engineering genetic networks. *Proceedings of the National Academy of Sciences*, 108(21):8617–8622, May 2011. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1015741108.

[114] Rune B. Lyngsø, James WJ Anderson, Elena Sizikova, Amarendra Badugu, Tomas Hyland, and Jotun Hein. Frnakenstein: Multiple target inverse RNA folding. *BMC Bioinformatics*, 13(1):260, October 2012. ISSN 1471-2105. doi: 10.1186/1471-2105-13-260.

[115] Javier Macía, Francesc Posas, and Ricard V. Solé. Distributed computation: The new wave of synthetic biology devices. *Trends in Biotechnology*, 30(6):342–349, June 2012. ISSN 0167-7799. doi: 10.1016/j.tibtech.2012.03.006.

[116] Nadim Majdalani, Christofer Cunning, Darren Sledjeski, Tom Elliott, and Susan Gottesman. DsrA RNA regulates translation of RpoS message by an anti-antisense mechanism, independent of its action as an antisilencer of transcription. *Proceedings of the National Academy of Sciences*, 95(21):12462–12467, October 1998. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.95.21.12462.

[117] Ewelina M. Małecka, Joanna Stróżecka, Daria Sobańska, and Mikołaj Olejniczak. Structure of Bacterial Regulatory RNAs Determines Their Performance in Competition for the Chaperone Protein Hfq. *Biochemistry*, 54(5):1157–1170, February 2015. ISSN 0006-2960. doi: 10.1021/bi500741d.

[118] Shuai Man, Rubin Cheng, Cuicui Miao, Qianhong Gong, Yuchao Gu, Xinzhi Lu, Feng Han, and Wengong Yu. Artificial trans-encoded small non-coding RNAs specifically silence the selected gene expression in bacteria. *Nucleic Acids Research*, 39(8):e50–e50, January 2011. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkr034.

[119] Maumita Mandal and Ronald R. Breaker. Gene regulation by riboswitches. *Nature Reviews Molecular Cell Biology*, 5(6):451–463, June 2004. ISSN 1471-0072. doi: 10.1038/nrm1403.

[120] Martin Mann, Marcel Kucharík, Christoph Flamm, and Michael T. Wolfinger. Memory-efficient RNA energy landscape exploration. *Bioinformatics*, 30(18):2584–2591, September 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu337.

[121] Yael Maon, Baruch Schieber, and Uzi Vishkin. Parallel ear decomposition search (EDS) and st-numbering in graphs. *Theoretical Computer Science*, 47:277–298, 1986. ISSN 0304-3975. doi: 10.1016/0304-3975(86)90153-2.

[122] NicholasR. Markham and Michael Zuker. UNAFold: Software for Nucleic Acid Folding and Hybridization. In JonathanM. Keith, editor, *Bioinformatics*, number 453 in Methods in Molecular Biology™, pages 3–31. Humana Press, Totowa, NJ, January 2008. ISBN 978-1-60327-428-9. doi: 10.1007/978-1-60327-429-6_1.

[123] Agustino Martıńez-Antonio and Julio Collado-Vides. Identifying global regulators in transcriptional regulatory networks in bacteria. *Current Opinion in Microbiology*, 6(5):482–489, October 2003. ISSN 1369-5274. doi: 10.1016/j.mib.2003.09.002.

[124] David H. Mathews, Matthew D. Disney, Jessica L. Childs, Susan J. Schroeder, Michael Zuker, and Douglas H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(19):7287–7292, November 2004. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0401799101.

[125] Marco C. Matthies, Stefan Bienert, and Andrew E. Torda. Dynamics in Sequence Space for RNA Secondary Structure Design. *Journal of Chemical Theory and Computation*, 8(10):3663–3670, October 2012. ISSN 1549-9618. doi: 10.1021/ct300267j.

[126] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, May 1990. ISSN 1097-0282. doi: 10.1002/bip.360290621.

[127] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *The Journal of Chemical Physics*, 21(6):1087–1092, June 1953. ISSN 0021-9606. doi: 10.1063/1.1699114.

[128] Zhichao Miao and Eric Westhof. RNA Structure: Advances and Assessment of 3D Structure Prediction. *Annual Review of Biophysics*, 46(1):483–503, 2017. doi: 10.1146/annurev-biophys-070816-034125.

[129] Ronald Micura. RNA biophysics: A three-state balancing act. *Nature*, 499(7458):289–290, July 2013. ISSN 0028-0836. doi: 10.1038/nature12410.

[130] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, October 2002. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.298.5594.824.

[131] Isabella Moll and Hanna Engelberg-Kulka. Selective translation during stress in Escherichia coli. *Trends in biochemical sciences*, 37(11):493–498, November 2012. ISSN 0968-0004. doi: 10.1016/j.tibs.2012.07.007.

[132] Isabella Moll, Sonja Grill, Claudio O. Gualerzi, and Udo Bläsi. Leaderless mRNAs in bacteria: Surprises in ribosomal recruitment and translational control. *Molecular Microbiology*, 43 (1):239–246, January 2002. ISSN 1365-2958. doi: 10.1046/j.1365-2958.2002.02739.x.

[133] Auriane Monestier, Alexey Aleksandrov, Pierre-Damien Coureux, Michel Panvert, Yves Mechulam, and Emmanuelle Schmitt. The structure of an E. coli tRNAfMet A1–U72 variant shows an unusual conformation of the A1–U72 base pair. *RNA*, 23(5):673–682, January 2017. ISSN 1355-8382, 1469-9001. doi: 10.1261/rna.057877.116.

[134] Steven R. Morgan and Paul G. Higgs. Barrier heights between ground states in a model of RNA secondary structure. *Journal of Physics A: Mathematical and General*, 31(14):3153, April 1998. ISSN 0305-4470. doi: 10.1088/0305-4470/31/14/005.

[135] Kevin V. Morris and John S. Mattick. The rise of regulatory RNA. *Nature Reviews Genetics*, 15(6):423, June 2014. ISSN 1471-0064. doi: 10.1038/nrg3722.

[136] Ulrike Mückstein, Hakim Tafer, Jörg Hackermüller, Stephan H. Bernhart, Peter F. Stadler, and Ivo L. Hofacker. Thermodynamics of RNA–RNA binding. *Bioinformatics*, 22(10):1177–1182, May 2006. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl024.

[137] Ulrike Mückstein, Hakim Tafer, Stephan H. Bernhart, Maribel Hernandez-Rosales, Jörg Vogel, Peter F. Stadler, and Ivo L. Hofacker. Translational Control by RNA-RNA Interaction: Improved Computation of RNA-RNA Binding Thermodynamics. In Mourad Elloumi, Josef Küng, Michal Linial, Robert F. Murphy, Kristan Schneider, and Cristian Toma, editors, *Bioinformatics Research and Development*, number 13 in Communications in Computer and Information Science, pages 114–127. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-70598-7 978-3-540-70600-7.

[138] Franz Narberhaus, Torsten Waldminghaus, and Saheli Chowdhury. RNA thermometers. *FEMS Microbiology Reviews*, 30(1): 3–16, January 2006. ISSN 1574-6976. doi: 10.1111/j.1574-6976.2005.004.x.

[139] Juliane Neupert and Ralph Bock. Designing and using synthetic RNA thermometers for temperature-controlled gene expression in bacteria. *Nature Protocols*, 4(9):1262–1273, August 2009. ISSN 1754-2189. doi: 10.1038/nprot.2009.112.

[140] Juliane Neupert, Daniel Karcher, and Ralph Bock. Design of simple synthetic RNA thermometers for temperature-controlled gene expression in Escherichia coli. *Nucleic Acids Research*, 36(19):e124–e124, January 2008. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkn545.

[141] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. Algorithms for Loop Matchings. *SIAM Jour-*

nal on Applied Mathematics, 35(1):68–82, July 1978. ISSN 0036-1399, 1095-712X. doi: 10.1137/0135006.

[142] Prof Luiz O. F. Penalva. mRNA Degradation. In Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota, editors, *Encyclopedia of Systems Biology*, pages 1460–1460. Springer New York, 2013. ISBN 978-1-4419-9862-0 978-1-4419-9863-7. doi: 10.1007/978-1-4419-9863-7_318.

[143] Lei Qi, Julius B. Lucks, Chang C. Liu, Vivek K. Mutalik, and Adam P. Arkin. Engineering naturally occurring trans-acting non-coding RNAs to sense molecular signals. *Nucleic Acids Research*, 40(12):5775–5786, January 2012. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gks168.

[144] Giulio Quarta, Ken Sin, and Tamar Schlick. Dynamic Energy Landscapes of Riboswitches Help Interpret Conformational Rearrangements and Function. *PLOS Computational Biology*, 8(2):e1002368, February 2012. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002368.

[145] Vijaya Ramachandran. Parallel Open Ear Decomposition with Applications to Graph Biconnectivity and Triconnectivity. In *Synthesis of Parallel Algorithms*, pages 275–340. Morgan-Kaufmann, 1992.

[146] Effirul I. Ramlan and Klaus-Peter Zauner. Design of interacting multi-stable nucleic acids for molecular information processing. *Biosystems*, 105(1):14–24, July 2011. ISSN 0303-2647. doi: 10.1016/j.biosystems.2011.02.006.

[147] Elizabeth E. Regulski and Ronald R. Breaker. In-Line Probing Analysis of Riboswitches. In *Post-Transcriptional Gene Regulation*, Methods In Molecular Biology™, pages 53–67. Humana Press, 2008. ISBN 978-1-58829-783-9 978-1-59745-033-1. doi: 10.1007/978-1-59745-033-1_4.

[148] Christian Reidys, Peter F. Stadler, and Peter Schuster. Generic properties of combinatory maps: Neutral networks of RNA secondary structures. *Bulletin of Mathematical Biology*, 59(2):339–397, March 1997. ISSN 0092-8240, 1522-9602. doi: 10.1007/BF02462007.

[149] Vladimir Reinharz, Yann Ponty, and Jérôme Waldispühl. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, 29(13):i308–i315, January 2013. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btt217.

[150] Anke Reining, Senada Nozinovic, Kai Schlepckow, Florian Buhr, Boris Fürtig, and Harald Schwalbe. Three-state mechanism couples ligand and temperature sensing in riboswitches. *Nature*, 499(7458):355–359, July 2013. ISSN 0028-0836. doi: 10.1038/nature12378.

[151] Francis E. Reyes, Andrew D. Garst, and Robert T. Batey. Chapter 6 - Strategies in RNA Crystallography. In *Methods in Enzymology*, volume 469 of *Biophysical, Chemical, and Functional Probes of RNA Structure, Interactions and Folding: Part B*, pages 119–139. Academic Press, January 2009. doi: 10.1016/S0076-6879(09)69006-6.

[152] Michael P. Robertson and Gerald F. Joyce. The Origins of the RNA World. *Cold Spring Harbor Perspectives in Biology*, 4 (5):a003608, January 2012. ISSN , 1943-0264. doi: 10.1101/cshperspect.a003608.

[153] Guillermo Rodrigo and Alfonso Jaramillo. RiboMaker: Computational design of conformation-based riboregulation. *Bioinformatics*, page btu335, May 2014. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btu335.

[154] Guillermo Rodrigo, Thomas E. Landrain, and Alfonso Jaramillo. De novo automated design of small RNA circuits for engineering synthetic riboregulation in living cells. *Proceedings of the National Academy of Sciences*, 109(38):15271–15276, September 2012. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1203831109.

[155] 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. *PLoS Comput Biol*, 9(8):e1003172, August 2013. doi: 10.1371/journal.pcbi.1003172.

[156] Guillermo Rodrigo, Thomas E. Landrain, Shensi Shen, and Alfonso Jaramillo. A new frontier in synthetic biology: Automated design of small RNA devices in bacteria. *Trends in Genetics*, 29(9):529–536, September 2013. ISSN 0168-9525. doi: 10.1016/j.tig.2013.06.005.

[157] Guillermo Rodrigo, Satya Prakash, Teresa Cordero, Manish Kushwaha, and Alfonso Jaramillo. Functionalization of an Antisense Small RNA. *Journal of Molecular Biology*, 428(5, Part B): 889–892, February 2016. ISSN 0022-2836. doi: 10.1016/j.jmb.2015.12.022.

[158] Guillermo Rodrigo, Satya Prakash, Shensi Shen, Eszter Majer, José-Antonio Daròs, and Alfonso Jaramillo. Model-based de-

sign of RNA hybridization networks implemented in living cells. *Nucleic Acids Research*, August 2017. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkx698.

[159] Nitzan Rosenfeld, Jonathan W. Young, Uri Alon, Peter S. Swain, and Michael B. Elowitz. Gene Regulation at the Single-Cell Level. *Science*, 307(5717):1962–1965, March 2005. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1106914.

[160] Johanna Roßmanith and Franz Narberhaus. Exploring the modular nature of riboswitches and RNA thermometers. *Nucleic Acids Research*, page gkw232, April 2016. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkw232.

[161] Hirohide Saito, Tetsuhiro Kobayashi, Tomoaki Hara, Yoshihiko Fujita, Karin Hayashi, Rie Furushima, and Tan Inoue. Synthetic translational regulation by an L7Ae–kink-turn RNP switch. *Nature Chemical Biology*, 6(1):71–78, January 2010. ISSN 1552-4450. doi: 10.1038/nchembio.273.

[162] Howard M. Salis, Ethan A. Mirsky, and Christopher A. Voigt. Automated design of synthetic ribosome binding sites to control protein expression. *Nature Biotechnology*, 27(10):946–950, October 2009. ISSN 1087-0156. doi: 10.1038/nbt.1568.

[163] Cornelius Schmidtke, Sven Findeiß, Cynthia M. Sharma, Juliane Kuhfuß, Steve Hoffmann, Jörg Vogel, Peter F. Stadler, and Ulla Bonas. Genome-wide transcriptome analysis of the plant pathogen Xanthomonas identifies sRNAs with putative virulence functions. *Nucleic Acids Research*, 40(5):2020–2031, March 2012. ISSN 1362-4962, 0305-1048. doi: 10.1093/nar/gkr904.

[164] Christopher Schneider and Beatrix Suess. Identification of RNA aptamers with riboswitching properties. *Methods*, 97:44–50, March 2016. ISSN 1046-2023. doi: 10.1016/j.ymeth.2015.12.001.

[165] Thomas D. Schneider and R. Michael Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Research*, 18(20):6097–6100, October 1990. ISSN 0305-1048. doi: 10.1093/nar/18.20.6097.

[166] Peter Schuster, Walter Fontana, Peter F. Stadler, and Ivo L. Hofacker. From Sequences to Shapes and Back: A Case Study in RNA Secondary Structures. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 255(1344):279–284, March 1994. ISSN 0962-8452, 1471-2954. doi: 10.1098/rspb.1994.0040.

[167] Alexander Serganov and Evgeny Nudler. A Decade of Riboswitches. *Cell*, 152(1–2):17–24, January 2013. ISSN 0092-8674. doi: 10.1016/j.cell.2012.12.024.

[168] Anastasia Sevostyanova and Eduardo A. Groisman. An RNA motif advances transcription by preventing Rho-dependent termination. *Proceedings of the National Academy of Sciences*, 112(50): E6835–E6843, December 2015. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1515383112.

[169] Cynthia M. Sharma, Steve Hoffmann, Fabien Darfeuille, Jérémy Reignier, Sven Findeiß, Alexandra Sittka, Sandrine Chabas, Kristin Reiche, Jörg Hackermüller, Richard Reinhardt, Peter F. Stadler, and Jörg Vogel. The primary transcriptome of the major human pathogen Helicobacter pylori. *Nature*, 464(7286):250–255, March 2010. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature08756.

[170] Shai S. Shen-Orr, Ron Milo, Shmoolik Mangan, and Uri Alon. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–68, May 2002. ISSN 1061-4036. doi: 10.1038/ng881.

[171] Yishai Shimoni, Gilgi Friedlander, Guy Hetzroni, Gali Niv, Shoshy Altuvia, Ofer Biham, and Hanah Margalit. Regulation of gene expression by small non-coding RNAs: A quantitative view. *Molecular Systems Biology*, 3(1):138, January 2007. ISSN 1744-4292, 1744-4292. doi: 10.1038/msb4100181.

[172] Atsuko Shinhara, Motomu Matsui, Kiriko Hiraoka, Wataru Nomura, Reiko Hirano, Kenji Nakahigashi, Masaru Tomita, Hirotada Mori, and Akio Kanai. Deep sequencing reveals as-yet-undiscovered small RNAs in Escherichia coli. *BMC Genomics*, 12:428, 2011. ISSN 1471-2164. doi: 10.1186/1471-2164-12-428.

[173] Wenjie Shu, Ming Liu, Hebing Chen, Xiaochen Bo, and Shengqi Wang. ARDesigner: A web-based system for allosteric RNA design. *Journal of Biotechnology*, 150(4):466–473, December 2010. ISSN 0168-1656. doi: 10.1016/j.jbiotec.2010.10.067.

[174] Christina D Smolke. Building outside of the box: iGEM and the BioBricks Foundation. *Nature Biotechnology*, 27(12): 1099–1102, December 2009. ISSN 1087-0156, 1546-1696. doi: 10.1038/nbt1209-1099.

[175] J. R. Stagno, Y. Liu, Y. R. Bhandari, C. E. Conrad, S. Panja, M. Swain, L. Fan, G. Nelson, C. Li, D. R. Wendel, T. A. White, J. D. Coe, M. O. Wiedorn, J. Knoska, D. Oberthuer, R. A. Tuckey, P. Yu, M. Dyba, S. G. Tarasov, U. Weierstall, T. D. Grant, C. D. Schwieters, J. Zhang, A. R. Ferré-D'Amaré, P. Fromme, D. E. Draper, M. Liang, M. S. Hunter, S. Boutet, K. Tan, X. Zuo, X. Ji, A. Barty, N. A. Zatsepin, H. N. Chapman, J. C. H. Spence, S. A. Woodson, and Y.-X. Wang. Structures of riboswitch RNA

reaction states by mix-and-inject XFEL serial crystallography. *Nature*, 541(7636):242, November 2016. ISSN 1476-4687. doi: 10.1038/nature20599.

[176] James D. Stephenson, Julia C. Kenyon, Martyn F. Symmons, and Andrew M. L. Lever. Characterizing 3D RNA structure by single molecule FRET. *Methods*, 103:57–67, July 2016. ISSN 1046-2023. doi: 10.1016/j.ymeth.2016.02.004.

[177] Jesse Stombaugh, Craig L. Zirbel, Eric Westhof, and Neocles B. Leontis. Frequency and isostericity of RNA base pairs. *Nucleic Acids Research*, 37(7):2294–2312, April 2009. ISSN 0305-1048. doi: 10.1093/nar/gkp011.

[178] Gisela Storz, Shoshy Altuvia, and Karen M Wassarman. An Abundance of RNA Regulators. *Annual Review of Biochemistry*, 74(1):199–217, 2005. doi: 10.1146/annurev.biochem.74.082803.133136.

[179] Gisela Storz, Jörg Vogel, and Karen M. Wassarman. Regulation by Small RNAs in Bacteria: Expanding Frontiers. *Molecular Cell*, 43(6):880–891, September 2011. ISSN 1097-2765. doi: 10.1016/j.molcel.2011.08.022.

[180] Beatrix Suess and Julia E. Weigand. Engineered riboswitches: Overview, problems and trends. *RNA Biology*, 5(1):24–29, January 2008. ISSN 1547-6286. doi: 10.4161/rna.5.1.5955.

[181] Akito Taneda. MODENA: A multi-objective RNA inverse folding. *Advances and applications in bioinformatics and chemistry : AABC*, 4:1–12, December 2010. ISSN 1178-6949.

[182] Akito Taneda. Multi-objective optimization for RNA design with multiple target secondary structures. *BMC Bioinformatics*, 16(1):280, September 2015. ISSN 1471-2105. doi: 10.1186/s12859-015-0706-x.

[183] Valentina Tereshko, Eugene Skripkin, and Dinshaw J Patel. Encapsulating Streptomycin within a Small 40-mer RNA. *Chemistry & Biology*, 10(2):175–187, February 2003. ISSN 1074-5521. doi: 10.1016/S1074-5521(03)00024-3.

[184] Bernhard C. Thiel, Christoph Flamm, and Ivo L. Hofacker. RNA structure prediction: From 2D to 3D. *Emerging Topics in Life Sciences*, 1(3):275–285, November 2017. ISSN 2397-8554, 2397-8562. doi: 10.1042/ETLS20160027.

[185] Shana Topp and Justin P. Gallivan. Emerging Applications of Riboswitches in Chemical Biology. *ACS Chemical Biology*, 5(1):139–148, January 2010. ISSN 1554-8929. doi: 10.1021/cb900278x.

[186] Brian J Tucker and Ronald R Breaker. Riboswitches as versatile gene control elements. *Current Opinion in Structural Biology*, 15 (3):342–348, June 2005. ISSN 0959-440X. doi: 10.1016/j.sbi.2005.05.003.

[187] Douglas H. Turner and David H. Mathews. NNDB: The nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38(suppl 1): D280–D282, January 2010. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkp892.

[188] Johannes H. Urban and Jörg Vogel. Translational control and target recognition by Escherichia coli small RNAs in vivo. *Nucleic Acids Research*, 35(3):1018–1037, February 2007. ISSN 0305-1048. doi: 10.1093/nar/gkl1040.

[189] Gabriele Varani, Fareed Aboul-ela, and Frédéric H. T. Allain. NMR investigation of RNA structure. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 29(1):51–127, June 1996. ISSN 0079-6565. doi: 10.1016/0079-6565(96)01028-X.

[190] Oliver Vesper, Shahar Amitai, Maria Belitsky, Konstantin Byrgazov, Anna Chao Kaberdina, Hanna Engelberg-Kulka, and Isabella Moll. Selective Translation of Leaderless mRNAs by Specialized Ribosomes Generated by MazF in Escherichia coli. *Cell*, 147(1):147–157, September 2011. ISSN 0092-8674. doi: 10.1016/j.cell.2011.07.047.

[191] Manja Wachsmuth, Sven Findeiß, Nadine Weissheimer, Peter F. Stadler, and Mario Mörl. De novo design of a synthetic riboswitch that regulates transcription termination. *Nucleic Acids Research*, 41(4):2541–2551, January 2013. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gks1330.

[192] Manja Wachsmuth, Gesine Domin, Ronny Lorenz, Robert Serfling, Sven Findeiß, Peter F. Stadler, and Mario Mörl. Design criteria for synthetic riboswitches acting on transcription. *RNA Biology*, 12(2):221–231, February 2015. ISSN 1547-6286. doi: 10.1080/15476286.2015.1017235.

[193] Andreas Wachter, Meral Tunc-Ozdemir, Beth C. Grove, Pamela J. Green, David K. Shintani, and Ronald R. Breaker. Riboswitch Control of Gene Expression in Plants by Splicing and Alternative 3′ End Processing of mRNAs. *The Plant Cell*, 19(11):3437–3450, November 2007. ISSN 1040-4651. doi: 10.1105/tpc.107.053645.

[194] Jérôme Waldispühl and Yann Ponty. An Unbiased Adaptive Sampling Algorithm for the Exploration of RNA Mutational

Landscapes under Evolutionary Pressure. In *Research in Computational Molecular Biology*, pages 501–515. Springer, Berlin, Heidelberg, March 2011. doi: 10.1007/978-3-642-20036-6_45.

[195] Jérôme Waldispühl, Srinivas Devadas, Bonnie Berger, and Peter Clote. Efficient Algorithms for Probing the RNA Mutation Landscape. *PLOS Computational Biology*, 4(8):e1000124, August 2008. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1000124.

[196] Torsten Waldminghaus, Jens Kortmann, Stefan Gesing, and Franz Narberhaus. Generation of synthetic RNA-based thermosensors. *Biological Chemistry*, 389(10):1319–1326, 2008. ISSN 1437-4315. doi: 10.1515/BC.2008.150.

[197] Michael Waterman, Dedicated To, and John R. Kinney. Secondary Structure of Single-Stranded Nucleic Acids. In *Studies on Foundations and Combinatorics, Advances in Mathematics Supplementary Studies, Academic Press N.Y., 1:167 – 212*, pages 167–212, 1978.

[198] Lauren S. Waters and Gisela Storz. Regulatory RNAs in Bacteria. *Cell*, 136(4):615–628, February 2009. ISSN 0092-8674, 1097-4172. doi: 10.1016/j.cell.2009.01.043.

[199] Kyle E. Watters, Timothy R. Abbott, and Julius B. Lucks. Simultaneous characterization of cellular RNA structure and function with in-cell SHAPE-Seq. *Nucleic Acids Research*, 44(2):e12–e12, January 2016. ISSN 0305-1048. doi: 10.1093/nar/gkv879.

[200] Kyle E. Watters, Eric J. Strobel, Angela M. Yu, John T. Lis, and Julius B. Lucks. Cotranscriptional folding of a riboswitch at nucleotide resolution. *Nature Structural & Molecular Biology*, advance online publication, October 2016. ISSN 1545-9993. doi: 10.1038/nsmb.3316.

[201] Ron Wehrens and Lutgarde MC Buydens. Self-and super-organizing maps in R: The Kohonen package. *Journal of Statistical Software*, 21(5):1–19, 2007.

[202] Daniel Wiegreffe, Alrik Hausdorf, Sebastian Zänker, and Dirk Zeckzer. iDotter – an interactive dot plot viewer. May 2017.

[203] Wade C. Winkler and Ronald R. Breaker. Regulation of bacterial gene expression by riboswitches. *Annual Review of Microbiology*, 59(1):487–517, September 2005. ISSN 0066-4227. doi: 10.1146/annurev.micro.59.030804.121336.

[204] Alexander Wittmann and Beatrix Suess. Engineered riboswitches: Expanding researchers' toolbox with synthetic RNA regulators. *FEBS Letters*, 586(15):2076–2083, July 2012. ISSN 1873-3468. doi: 10.1016/j.febslet.2012.02.038.

[205] Brian R. Wolfe and Niles A. Pierce. Sequence Design for a Test Tube of Interacting Nucleic Acid Strands. *ACS Synthetic Biology*, 4(10):1086–1100, October 2015. doi: 10.1021/sb5002196.

[206] Brian R. Wolfe, Nicholas J. Porubsky, Joseph N. Zadeh, Robert M. Dirks, and Niles A. Pierce. Constrained Multistate Sequence Design for Nucleic Acid Reaction Pathway Engineering. *Journal of the American Chemical Society*, 139(8):3134–3144, March 2017. ISSN 0002-7863. doi: 10.1021/jacs.6b12693.

[207] Michael T Wolfinger, W Andreas Svrcek-Seiler, Christoph Flamm, Ivo L Hofacker, and Peter F Stadler. Efficient computation of RNA folding dynamics. *Journal of Physics A: Mathematical and General*, 37(17):4731–4741, April 2004. ISSN 0305-4470, 1361-6447. doi: 10.1088/0305-4470/37/17/005.

[208] Peter J. Woolf and Yixin Wang. A fuzzy logic approach to analyzing gene expression data. *Physiological Genomics*, 3(1):9–15, June 2000. ISSN 1094-8341. doi: 10.1152/physiolgenomics.2000.3.1.9.

[209] Stefan Wuchty, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999. ISSN 1097-0282. doi: 10.1002/(SICI)1097-0282(199902)49:2<145::AID-BIP4>3.0.CO;2-G.

[210] A. Xayaphoummine, T. Bucher, and H. Isambert. Kinefold web server for RNA/DNA folding path and structure prediction including pseudoknots and knots. *Nucleic Acids Research*, 33(suppl_2):W605–W610, July 2005. ISSN 0305-1048. doi: 10.1093/nar/gki447.

[211] A. Xayaphoummine, V. Viasnoff, S. Harlepp, and H. Isambert. Encoding folding paths of RNA switches. *Nucleic Acids Research*, 35(2):614–622, January 2007. ISSN 0305-1048. doi: 10.1093/nar/gkl1036.

[212] Zhenjiang Zech Xu and David H. Mathews. Secondary Structure Prediction of Single Sequences Using RNAstructure. *Methods in Molecular Biology (Clifton, N.J.)*, 1490:15–34, 2016. ISSN 1940-6029. doi: 10.1007/978-1-4939-6433-8_2.

[213] Rosalyn S. Yalow and Solomon A. Berson. Immunoassay of endogenous plasma insulin in man. *The Journal of Clinical Investigation*, 39(7):1157–1175, July 1960. ISSN 0021-9738. doi: 10.1172/JCI104130.

[214] Esti Yeger-Lotem, Shmuel Sattath, Nadav Kashtan, Shalev Itzkovitz, Ron Milo, Ron Y. Pinter, Uri Alon, and Hanah Mar-

galit. Network motifs in integrated cellular networks of transcription–regulation and protein–protein interaction. *Proceedings of the National Academy of Sciences of the United States of America*, 101(16):5934–5939, April 2004. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0306752101.

[215] Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. *Journal of Computational Chemistry*, 32(1):170–173, January 2011. ISSN 1096-987X. doi: 10.1002/jcc.21596.

[216] Joseph N. Zadeh, Brian R. Wolfe, and Niles A. Pierce. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of Computational Chemistry*, 32(3):439–452, 2011. ISSN 1096-987X. doi: 10.1002/jcc.21633.

[217] Christian Höner zu Siederdissen, Stephan H. Bernhart, Peter F. Stadler, and Ivo L. Hofacker. A folding algorithm for extended RNA secondary structures. *Bioinformatics*, 27(13): i129–i136, January 2011. ISSN 1367-4803, 1460-2059. doi: 10.1093/bioinformatics/btr220.

[218] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244(4900):48–52, April 1989. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.2468181.

[219] Michael Zuker and David Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984. ISSN 0092-8240. doi: 10.1016/S0092-8240(84)80062-2.

Part IV

APPENDIX

JOHN WILEY AND SONS LICENSE TERMS AND CONDITIONS

Oct 19, 2017

This Agreement between Stefan Hammer ("You") and John Wiley and Sons ("John Wiley and Sons") consists of your license details and the terms and conditions provided by John Wiley and Sons and Copyright Clearance Center.

License Number	4111261430968
License date	May 17, 2017
Licensed Content Publisher	John Wiley and Sons
Licensed Content Publication	Biopolymers
Licensed Content Title	Computational design of RNAs with complex energy landscapes
Licensed Content Author	Christian Höner zu Siederdissen,Stefan Hammer,Ingrid Abfalter,Ivo L. Hofacker,Christoph Flamm,Peter F. Stadler
Licensed Content Date	Sep 23, 2013
Licensed Content Pages	13
Type of use	Dissertation/Thesis
Requestor type	Author of this Wiley article
Format	Print and electronic
Portion	Full article
Will you be translating?	No
Title of your thesis / dissertation	Design of of context-sensitive, multi-stable RNA molecules
Expected completion date	Sep 2017
Expected size (number of pages)	140
Requestor Location	Stefan Hammer Währinger Straße 17 Room 309 Vienna, 1090 Austria Attn: Stefan Hammer
Publisher Tax ID	EU826007151
Billing Type	Invoice
Billing Address	Stefan Hammer Währinger Straße 17 Room 309 Vienna, Austria 1090 Attn: Stefan Hammer
Total	0.00 EUR
Terms and Conditions	

TERMS AND CONDITIONS

Wiley & Sons, Inc. or one of its group companies (each a"Wiley Company") or handled on behalf of a society with which a Wiley Company has exclusive publishing rights in relation to a particular work (collectively "WILEY"). By clicking "accept" in connection with completing this licensing transaction, you agree that the following terms and conditions apply to this transaction (along with the billing and payment terms and conditions established by the Copyright Clearance Center Inc., ("CCC's Billing and Payment terms and conditions"), at the time that you opened your RightsLink account (these are available at any time at http://myaccount.copyright.com).

Terms and Conditions

- The materials you have requested permission to reproduce or reuse (the "Wiley Materials") are protected by copyright.

- You are hereby granted a personal, non-exclusive, non-sub licensable (on a stand-alone basis), non-transferable, worldwide, limited license to reproduce the Wiley Materials for the purpose specified in the licensing process. This license, **and any CONTENT (PDF or image file) purchased as part of your order,** is for a one-time use only and limited to any maximum distribution number specified in the license. The first instance of republication or reuse granted by this license must be completed within two years of the date of the grant of this license (although copies prepared before the end date may be distributed thereafter). The Wiley Materials shall not be used in any other manner or for any other purpose, beyond what is granted in the license. Permission is granted subject to an appropriate acknowledgement given to the author, title of the material/book/journal and the publisher. You shall also duplicate the copyright notice that appears in the Wiley publication in your use of the Wiley Material. Permission is also granted on the understanding that nowhere in the text is a previously published source acknowledged for all or part of this Wiley Material. Any third party content is expressly excluded from this permission.

- With respect to the Wiley Materials, all rights are reserved. Except as expressly granted by the terms of the license, no part of the Wiley Materials may be copied, modified, adapted (except for minor reformatting required by the new Publication), translated, reproduced, transferred or distributed, in any form or by any means, and no derivative works may be made based on the Wiley Materials without the prior permission of the respective copyright owner.**For STM Signatory Publishers clearing permission under the terms of the STM Permissions Guidelines only, the terms of the license are extended to include subsequent editions and for editions in other languages, provided such editions are for the work as a whole in situ and does not involve the separate exploitation of the permitted figures or extracts,** You may not alter, remove or suppress in any manner any copyright, trademark or other notices displayed by the Wiley Materials. You may not license, rent, sell, loan, lease, pledge, offer

as security, transfer or assign the Wiley Materials on a stand-alone basis, or any of the rights granted to you hereunder to any other person.

- The Wiley Materials and all of the intellectual property rights therein shall at all times remain the exclusive property of John Wiley & Sons Inc, the Wiley Companies, or their respective licensors, and your interest therein is only that of having possession of and the right to reproduce the Wiley Materials pursuant to Section 2 herein during the continuance of this Agreement. You agree that you own no right, title or interest in or to the Wiley Materials or any of the intellectual property rights therein. You shall have no rights hereunder other than the license as provided for above in Section 2. No right, license or interest to any trademark, trade name, service mark or other branding ("Marks") of WILEY or its licensors is granted hereunder, and you agree that you shall not assert any such right, license or interest with respect thereto

- NEITHER WILEY NOR ITS LICENSORS MAKES ANY WARRANTY OR REPRESENTATION OF ANY KIND TO YOU OR ANY THIRD PARTY, EXPRESS, IMPLIED OR STATUTORY, WITH RESPECT TO THE MATERIALS OR THE ACCURACY OF ANY INFORMATION CONTAINED IN THE MATERIALS, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, ACCURACY, SATISFACTORY QUALITY, FITNESS FOR A PARTICULAR PURPOSE, USABILITY, INTEGRATION OR NON-INFRINGEMENT AND ALL SUCH WARRANTIES ARE HEREBY EXCLUDED BY WILEY AND ITS LICENSORS AND WAIVED BY YOU.

- WILEY shall have the right to terminate this Agreement immediately upon breach of this Agreement by you.

- You shall indemnify, defend and hold harmless WILEY, its Licensors and their respective directors, officers, agents and employees, from and against any actual or threatened claims, demands, causes of action or proceedings arising from any breach of this Agreement by you.

- IN NO EVENT SHALL WILEY OR ITS LICENSORS BE LIABLE TO YOU OR ANY OTHER PARTY OR ANY OTHER PERSON OR ENTITY FOR ANY SPECIAL, CONSEQUENTIAL, INCIDENTAL, INDIRECT, EXEMPLARY OR PUNITIVE DAMAGES, HOWEVER CAUSED, ARISING OUT OF OR IN CONNECTION WITH THE DOWNLOADING, PROVISIONING, VIEWING OR USE OF THE MATERIALS REGARDLESS OF THE FORM OF ACTION, WHETHER FOR BREACH OF CONTRACT, BREACH OF WARRANTY, TORT, NEGLIGENCE, INFRINGEMENT OR OTHERWISE (INCLUDING, WITHOUT LIMITATION, DAMAGES BASED ON LOSS OF PROFITS, DATA, FILES, USE, BUSINESS OPPORTUNITY OR CLAIMS OF THIRD PARTIES), AND WHETHER OR NOT THE PARTY HAS BEEN

ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. THIS LIMITATION SHALL APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY PROVIDED HEREIN.

- Should any provision of this Agreement be held by a court of competent jurisdiction to be illegal, invalid, or unenforceable, that provision shall be deemed amended to achieve as nearly as possible the same economic effect as the original provision, and the legality, validity and enforceability of the remaining provisions of this Agreement shall not be affected or impaired thereby.

- The failure of either party to enforce any term or condition of this Agreement shall not constitute a waiver of either party's right to enforce each and every term and condition of this Agreement. No breach under this agreement shall be deemed waived or excused by either party unless such waiver or consent is in writing signed by the party granting such waiver or consent. The waiver by or consent of a party to a breach of any provision of this Agreement shall not operate or be construed as a waiver of or consent to any other or subsequent breach by such other party.

- This Agreement may not be assigned (including by operation of law or otherwise) by you without WILEY's prior written consent.

- Any fee required for this permission shall be non-refundable after thirty (30) days from receipt by the CCC.

- These terms and conditions together with CCC's Billing and Payment terms and conditions (which are incorporated herein) form the entire agreement between you and WILEY concerning this licensing transaction and (in the absence of fraud) supersedes all prior agreements and representations of the parties, oral or written. This Agreement may not be amended except in writing signed by both parties. This Agreement shall be binding upon and inure to the benefit of the parties' successors, legal representatives, and authorized assigns.

- In the event of any conflict between your obligations established by these terms and conditions and those established by CCC's Billing and Payment terms and conditions, these terms and conditions shall prevail.

- WILEY expressly reserves all rights not specifically granted in the combination of (i) the license details provided by you and accepted in the course of this licensing transaction, (ii) these terms and conditions and (iii) CCC's Billing and Payment terms and conditions.

- This Agreement will be void if the Type of Use, Format, Circulation, or Requestor Type was misrepresented during the licensing process.

- This Agreement shall be governed by and construed in accordance with the laws of the State of New York, USA, without regards to such state's conflict of law rules. Any legal action, suit or proceeding arising out of or relating to these Terms and Conditions or the breach thereof shall be instituted in a court of competent jurisdiction in New York County in the State of New York in the United States of America and each party hereby consents and submits to the personal jurisdiction of such court, waives any objection to venue in such court and consents to service of process by registered or certified mail, return receipt requested, at the last known address of such party.

WILEY OPEN ACCESS TERMS AND CONDITIONS

Wiley Publishes Open Access Articles in fully Open Access Journals and in Subscription journals offering Online Open. Although most of the fully Open Access journals publish open access articles under the terms of the Creative Commons Attribution (CC BY) License only, the subscription journals and a few of the Open Access Journals offer a choice of Creative Commons Licenses. The license type is clearly identified on the article.

The Creative Commons Attribution License

The Creative Commons Attribution License (CC-BY) allows users to copy, distribute and transmit an article, adapt the article and make commercial use of the article. The CC-BY license permits commercial and non-

Creative Commons Attribution Non-Commercial License

The Creative Commons Attribution Non-Commercial (CC-BY-NC)License permits use, distribution and reproduction in any medium, provided the original work is properly cited and is not used for commercial purposes.(see below)

Creative Commons Attribution-Non-Commercial-NoDerivs License

The Creative Commons Attribution Non-Commercial-NoDerivs License (CC-BY-NC-ND) permits use, distribution and reproduction in any medium, provided the original work is properly cited, is not used for commercial purposes and no modifications or adaptations are made. (see below)

Use by commercial "for-profit" organizations

Use of Wiley Open Access articles for commercial, promotional, or marketing purposes requires further explicit permission from Wiley and will be subject to a fee.

Further details can be found on Wiley Online Library
http://olabout.wiley.com/WileyCDA/Section/id-410895.html

Other Terms and Conditions:

v1.10 Last updated September 2015