

MASTERARBEIT / MASTER'S THESIS

Titel der Masterarbeit / Title of the Master's Thesis

„Kinetics of RNA Structure Formation - Folding Pathways
and Evolution“

verfasst von / submitted by

Maria Waldl BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien, 2017 / Vienna 2017

Studienkennzahl lt. Studienblatt /
degree programme code as it appears on
the student record sheet:

A 066 862

Studienrichtung lt. Studienblatt /
degree programme as it appears on
the student record sheet:

Masterstudium Chemie

Betreut von / Supervisor:

Univ.-Prof. Dipl.-Phys. Dr. Ivo Hofacker

Mitbetreut von / Co-Supervisor:

Ing. Mag. Dr. Andrea Tanzer

ACKNOWLEDGMENTS

First of all, I'd like to thank Ivo Hofacker who was my supervisor for this thesis. He made this work possible, provided guidance and always made time to discuss research questions.

My co-supervisor Andrea Tanzer supported me in navigating the whole thesis process – from idea to finished, printed product – and even joined me in late night writing sessions to provide feedback and encouragement.

Thanks to the whole RNA Lands team for welcoming me to their project especially Ronny Lorenz, Mireille Régnier, Yann Ponty and Gregor Entzian. Ronny was always there to answer technical and programming questions and quickly implemented new features in ViennaRNA package when needed.

Mireille Régnier and Yann Ponty welcomed me in Paris for two months, during which the trajectory alignment and background model chapters took shape. Yann introduced me to analytic combinatorics and supervised the development of the trajectory alignment method. Mireille was the first to suggest the mathematical approach used in the Background Model chapter and was available for long, helpful discussions about its implementation.

Thanks to Judith Ivansits for her assistance with administrative matters and navigating the bureaucracy associated with this thesis and my employment.

Richard Neuböck provided the infrastructure that made this work possible and helped me solve many a tech support issue.

I also want to thank Christoph Flamm, Sebastian Will, Florian Eggenhofer and Stefan Badelt for the discussions and the helpful feedback. Creating this work at TBI was an engaging and enjoyable experience, also thanks to my current and former co-workers Bernhard, Christina, Daniel, Dominik, Fabian, Jörg, Mariam, Michael, Peter, Roman, Stefan, Sven and Veerendra.

My parents Susanne and Christian always inspired my curiosity and encouraged me to follow my interests. Without their support my study and this thesis would not have been possible.

I am also thankful to my sister Teresa and my brother Simon, my extended family and friends that are not mentioned here for their moral support over the years.

Last but not least I want to thank my partner Fin for his patience and encouragement during the last months. Our time together gave me the energy to finish this work.

CONTENTS

1	INTRODUCTION	1
1.1	RNA	1
1.1.1	Chemical background	1
1.2	RNA structure	2
1.2.1	Formal definition of RNA secondary structures	4
1.2.2	RNA secondary structure representation	5
1.3	Thermodynamic RNA structure prediction	7
1.3.1	Decomposition of nested secondary structures	7
1.3.2	Nussinov algorithm	7
1.3.3	Nearest neighbor model energy parameters	10
1.3.4	Zuker's algorithm	10
1.4	Dynamics of RNA structures	11
1.4.1	Kinetic prediction methods	11
1.5	RNA evolution and conservation	15
1.5.1	Transat	16
1.6	Motivation	18
2	DATASET	21
2.1	Introduction	21
2.1.1	Rfam families	22
2.2	Methods	29
2.2.1	Retrieving the raw datasets	29
2.2.2	Alignment extensions	32
2.2.3	Filtering and Clustering	32
2.2.4	Prediction of transient structures	33
2.2.5	Visualization of the results	36
2.3	Results and Discussion	36
2.3.1	Kinwalker transcription rate test	36
2.3.2	Improved Kinwalker parameters	37
2.3.3	Trp operon leader	37
2.3.4	Levivirus maturation gene 5'UTR	42
2.3.5	SAM riboswitch	44
2.3.6	HDV ribozyme	47
2.3.7	Small bacterial SRP RNA	47
2.4	Outlook	51
3	ALIGNING KINETIC FOLDING TRAJECTORIES	53
3.1	Introduction	53
3.2	Methods	54
3.2.1	Dynamic Programming	54
3.2.2	Implementation of the Folding Trajectory Alignment	59
3.3	Examples for trajectory alignments	61
3.3.1	TRP trajectories	61
3.3.2	SAM	64
3.4	Discussion and Outlook	66
4	BACKGROUND MODEL	69

4.1	Introduction	69
4.1.1	Investigated structure pattern	72
4.2	Stochastic models	72
4.2.1	Methods	73
4.2.2	Results and Discussion	74
4.2.3	Summary	76
4.3	Mathematical Background of the Analytic Combinatorics Model	78
4.3.1	Formal Grammars	78
4.3.2	Combinatorial Classes and Generating Functions	81
4.4	Analytic Combinatorics Model	84
4.4.1	Language equation on initial set	84
4.4.2	Generating function of initial set	85
4.4.3	Implementation	85
4.5	Outlook	86
5	DISCUSSION	89
	BIBLIOGRAPHY	93

ACRONYMS

MFE	minimum free energy
RNA	ribonucleic acid
rRNA	ribosomal RNA
tRNA	transfer RNA
mRNA	messenger RNA
DNA	deoxy-ribonucleic acid
UTR	untranslated region
nt	nucleotide
a.u.	arbitrary time units
RNase P	ribonuclease P
SAM	S-Adenosyl methionine
SRP	signal recognition particle
NMR	nuclear magnetic resonance
bp	base pair
ssRNA	single stranded RNA
ORF	open reading frame
RTase	reverse transcriptase
HDAg	hepatitis delta antigen
HBV	hepatitis beta virus
HDV	hepatitis delta virus
NCBI	National Center for Biotechnology Information
TSS	transcription start side
Trp	tryptophan
SD	Shine-Dalgarno

INTRODUCTION

1.1 RNA

Ribonucleic acids (RNAs) are linear or circular bio-polymers that consist of nucleotides connected by a phosphodiester bond. Along with DNA, proteins and carbohydrates, ribonucleic acid (RNA) is an essential building block in all known life forms. It is best known for its function as messenger RNA (messenger RNA (mRNA)) in protein synthesis [5, 9]. However, in the last decades more diverse roles of RNAs were discovered: Apart from facilitating protein synthesis as carrier of information, ribosomal RNA (ribosomal RNA (rRNA)) or transfer RNA (transfer RNA (tRNA)), it can show enzymatic activity (ribozyme) and play an important role in gene regulation as well as maturation.

Most of these functions strongly rely on the ability of RNAs to interact with their environment and to change their conformation upon external stimuli. Thus, the function of an RNA cannot be understood when only looking at one single structure. Viewing an RNA as a dynamic polymer that is represented by an ensemble of structures is much better suited for investigating the purpose of an RNA.

1.1.1 Chemical background

A detailed understanding of the chemical composition of RNAs helps understand their folding behavior and function.

Its building blocks, the nucleotides, are highly similar to those of deoxyribonucleic acid (deoxy-ribonucleic acid (DNA)). Every nucleotide consists of three organic molecules: a ribose (a five-carbon sugar), an organic base and a phosphate group. The base is attached to the 1' position of the ribose sugar while the phosphate group is esterified with the 5' OH group of the ribose (fig. 1).

The main difference between DNA and RNA is in the ribose sugar. While RNA is based on ribose, DNA is based on 2-deoxyribose. The additional hydroxyl group in the 2' position of the sugar makes RNA more reactive and less mechanically flexible than DNA.

There are four nitrogenous bases that occur in RNA: guanine, uracil, adenine, and cytosine. The two double ringed bases, adenine and guanine, are purines while cytosine and uracil are pyrimidine bases. In DNA another pyrimidine base, thymine, replaces uracil. Nitrogenous bases can be modified during the maturation of the RNA (RNAmodi-

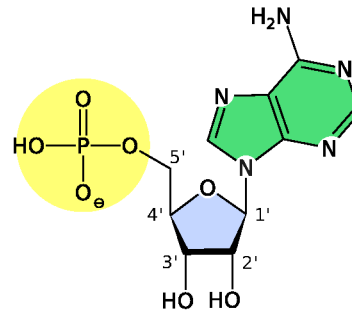


Figure 1: Chemical structure of a nucleotide: A nucleotide is the monomer subunit of an RNA. It consists of a ribose (blue), a nitrogenous base (green) and a phosphate group (yellow). The organic base is connected to the 1' C of the ribose, while the phosphate group is attached to the 5' C of the ribose through an ester bond. In RNAs, four nitrogenous bases (and modifications thereof) occur: guanine, uracil, adenine and cytosine.

fication). Therefore, a range of derivatives of these bases can be found in cells. Each phosphate group carries a negative charge, making the RNA a polyanion.

In nature the template based polymerization process is catalyzed by polymerase. The nucleotides are connected through a covalent phosphodiester bond between the 3' carbon of one ribose to the 5' carbon of the next ribose. This results in a directional sugar-phosphate backbone (fig. 2).

RNA bases form pairs with nucleotides from inverse RNA (sub-) strands, leading to well defined secondary structure motives [56]. The three most frequent base pair types are shown in figure 2. They consist of a purine and a pyrimidine base: In the case of canonical Watson-Crick base pairs between guanine (G) and cytosine (C) as well as between adenine (A) and uracil (U) [49]. In addition to these Watson-Crick base pairs, the wobble pair guanine-uracil is prevalent in RNA structure formation [55]. The selective pairing is facilitated by hydrogen bonds, three in the case of GC pairs and two in the case of AU and GU. Together these three base pair types account for about 76% of pairs found in RNAs [52]. Less common base pair types can be looked up in the classification scheme by Leontis and Westhof [30].

Similar to base pairs in DNA helices, neighboring base pairs in RNA structures are stabilized by π - π -stacking [68]. Delocalized electrons in p-orbitals of the aromatic rings lead to attractive interactions that play a major role in the free energy of RNA structure.

1.2 RNA STRUCTURE

Non-coding RNAs often form complex structures that strongly determine their function.

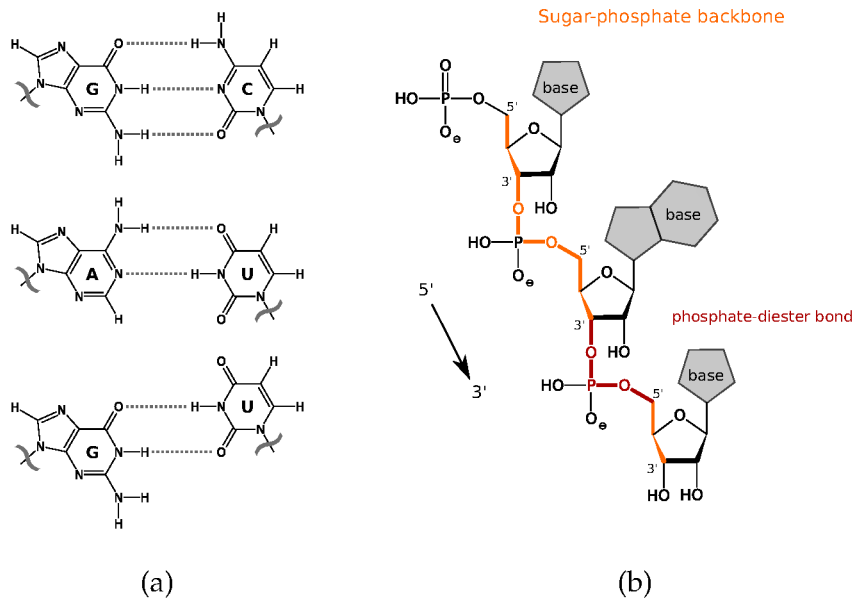


Figure 2: (a) Canonical base pairs and wobble base pair: This figure introduces the structure of the two purine bases adenine(A) and guanine(G) and the two pyrimidine bases cytosine(C) and uracil(U). The base pair AU and the base pair GU form two hydrogen bonds, while GC forms three hydrogen bonds. All three base pair types are considered cis Watson-Crick Watson-Crick base pairs with respect to the nomenclature introduced by Leontis and Westhof[30]. (b) Sugar phosphate backbone: During transcription an RNA is extended by esterification of the phosphate group of a new nucleotide to the free 3' OH group of the last nucleotide in the polymer. Thus two nucleotides are connected through a phosphate-diester bond(red), resulting in a 5' $\rightarrow$ 3' directional sugar-phosphate backbone(orange).

Transfer-RNAs illustrate the importance of RNA structures: They have to be structured in a way that ensures that the anti-codon is accessible for interactions with the ribosome and the mRNA.

Other RNAs like ribosomes or ribonuclease P (RNase P) have catalytic functions. Protein based enzymes often rely on forming a catalytic center. Similarly, ribozymes have to fold into specific structures that allow them to interact with their substrate. Regulatory RNAs can function through their structure, too. For instance, riboswitches are regulatory elements in the 5' end of mRNAs that control transcription and/or translation by changing their conformation. The conformation change can be initiated by the absence or presence of some ligand or temperature changes. Another example are protein recognition sites. Some proteins bind specifically to single stranded or double stranded regions. Thus, forming local structures that keep the interaction site single or double stranded is essential for proper function. Such protein-RNA interactions can for example facilitate RNA editing.

RNA structures can be organized in four different classes: The **primary** structure is a linear sequence of nucleotides. RNA sequences are written from the 5' to 3' end. The **secondary** structure is the set of base pairs between nucleotides of the same molecule. The **tertiary** structure refers to the three dimensional structure of a single RNA strand. The **quaternary** structure describes the higher ordering of RNA molecules, the ordering of more than one RNA molecule and the structure with respect to other molecules like proteins.

Sequencing experiments provide ample data on the primary (sequence) level. However, determining RNA structures through methods like chemical probing, x-ray crystallography and nuclear magnetic resonance (NMR) is still a complex and time-consuming process[61]. Particularly, tracking dynamic folding processes is still challenging. Thus, computational methods for predicting RNA structures and RNA folding are valuable tools for exploring the functions of RNA in the context of their structure.

1.2.1 Formal definition of RNA secondary structures

Formally, an RNA sequence with n nucleotides can be defined as an ordered sequence $S = (S_1, \dots, S_n)$ with S_i representing the type of the i^{th} nucleotide and $S_i \in \{A, C, G, U\} = \Sigma$ with Σ being the alphabet. In other words, an RNA sequence is a sequence $S = \{A, C, G, U\}^*$ that consists of multiple occurrences of the letters A, C, G and U with a total length of n .

An RNA structure can be defined as a set of base pairs $R \subseteq (i, j) | 1 \leq i < j \leq n$, with S_i and S_j being complementary, where (i, j) represents the formation of hydrogen bonds between sequence position i and j . In this work the set of base pairs $\mathcal{B}$ with the following nucleotide combinations are considered: $\mathcal{B} = \{AU, UA, CG, GC, GU, UG\}$.

In accordance with the notation in previous publications on RNA secondary structures, the structure formed by the (sub-)sequence $S[i : j]$ is noted as $s[i : j]$.

In the case of secondary structures each nucleotide can only be involved in at most one base pair. Therefore, the degree of R is at most one. For two base pairs (i, j) and (k, l) this also means $i = k$ if and only if $j = l$.

Most programs only consider nested structures. This means all base pairs are non-crossing. Formally this restricts the set of base pairs to a set, in which for any two base pairs (i, j) and (k, l) with $i < k < j$ the condition $i < l < j$ holds.

For steric reasons bases that are less than three positions apart cannot form stable hydrogen bonds. Therefore a minimum hairpin loop length m is defined such that for any base pair (i, j) $j - i - 1 \geq m$, with $m \geq 3$.

Some programs use alternative definitions of secondary structures that, for example, can involve additional non-canonical base pairs, base triplets or structural features with crossing base pairs, called pseudoknots.

An RNA secondary structure can also be viewed as planar graph $G = V, E$ with vertex set $V = 1, \dots, n$, that represents the nucleotides or their position in the sequence, and edge set $E = (i, j)$. Two types of bonds contribute to the edge set: The covalent bonds between consecutive nucleotides in the backbone $((i, i + 1))$ and the hydrogen bonds between nucleotides in base pairs $((i, j))$.

1.2.2 RNA secondary structure representation

A compact way of writing structures is the dot-bracket notation (fig. 3(c)). The structure of a sequence with length n can be depicted as a string composed of $“.”$, $“(“$ and $“)”$ of the same length. Unpaired positions are represented by a $“.”$. A base pair (i, j) is represented by matching parentheses $“(“$ and $“)”$ in position i and j , respectively. This notation is an unambiguous 1D representation, it can be adapted for alignments and it is easily machine- and human readable.

In the previous section secondary structures were defined as planar graphs, which allows for an easy 2D-visualization.

The most widely used depictions based on drawing the planar graph are radial drawings. In these drawings the length of H-bonds as well as the distance between consecutive nucleotides are fixed, resulting in drawings like figure 3(b). These graphics give a 2-D estimate of the actual 3D-structure.

In this work, some results are presented as arc-plots: Nucleotides are positioned equidistant along a straight line and base pairs are represented as arcs that connect two pairing nucleotides (fig. 3(a)). While

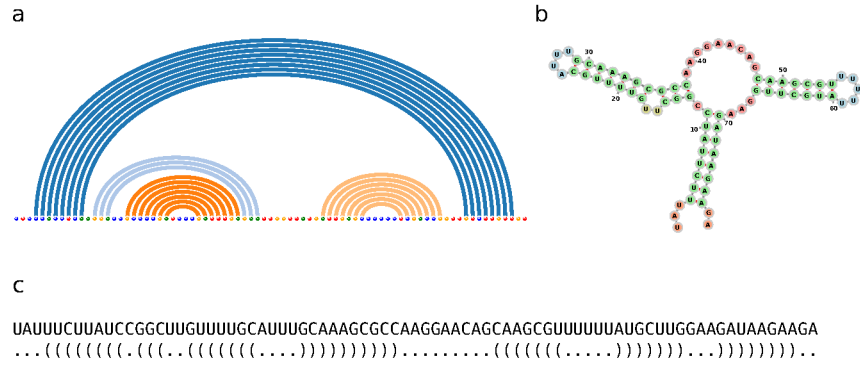


Figure 3: Representations of RNA secondary structures: (a) RNA secondary structures can be drawn as planar graphs. In arcplots the RNA backbone is drawn as the sequence of nucleotides on a straight line, while the base pairs are depicted as arcs that connect the respective pairing nucleotides. (b) Another way of drawing the planar graph that represents the RNA secondary structure is as a radial drawing. In contrast to arcplots not only the distance between two consecutive base pairs but also the distance between two pairing nucleotides is kept constant. (c) A compact way of visualizing a structure is as a dot-bracket string that has the same length as the sequence. Here unpaired nucleotides are represented by "." and base pairs are represented by "(" and ")" in the position of the first and the second nucleotide that form the base pair.

this type of plot might put too high visual weight on long distance base pairs, it is useful for comparing multiple structures (using different colors and line weights or placing arcs above and below the line of nucleotides) and can be combined with sequence alignments.

1.3 THERMODYNAMIC RNA STRUCTURE PREDICTION

1.3.1 *Decomposition of nested secondary structures*

To come up with a description for all structures that can be formed by a sequence S , one can start by analyzing any structure that can be formed by a subsequence $S[i : j]$, where i is the first nucleotide of the subsequence and j is the last nucleotide of the subsequence.

For any structure formed by a subsequence $S[i : j]$, there are two ways in which the nucleotide in position i can contribute to the structure of the subsequence. Either it is unpaired or it is paired with another nucleotide k with $i < k \leq j$. In the first case the structure, defined as a set of base pairs, is the same as the structure for the sequence $S[i + 1 : j]$. In the other case the base pair (i, k) splits the sequence in two independent parts $S[i + 1 : k - 1]$ and $S[k + 1 : j]$. These subsequences can be treated like the initial subsequence $S[i : j]$. Thus, a recursive decomposition of nested secondary RNA structures can be reached. Figure 4 is a visualization of this recursive decomposition.

It is possible to write this decomposition as a grammar (An introduction to grammars can be found in chapter 4):

$$S \rightarrow X$$

$$X \rightarrow xX|uXaX|aXuX|gXcX|cXgX|gXuX|uXgX$$

$$X \rightarrow \epsilon$$

Where X represents any nested structure, x represents any unpaired nucleotide, and a, c, g and u represents base pairing nucleotides.

Based on this grammar an algorithm for finding the structure with most base pairs, the number of possible structures or the structure with the minimum free energy can be derived.

As an example, $N_{i,j}$ can be defined as the number of possible structures of a given sequence S .

$$N_{i,j} = N_{i+1,j} + \sum_{k \in \text{possible } (i,k) \text{ pairs}} (N_{i+1,k-1} \cdot N_{k+1,j})$$

1.3.2 *Nussinov algorithm*

In 1980, Ruth Nussinov published an algorithm for predicting optimal secondary structures based on this decomposition. It is a dynamic programming algorithm that maximizes the number of base pairs [41].

It uses following decomposition:

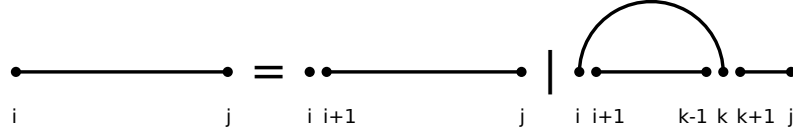


Figure 4: Decomposition of nested RNA secondary structures: Any secondary (sub-)structure $s[i : j]$ can be decomposed into smaller substructures by considering the possible contributions of the i^{th} nucleotide. Either the nucleotide is unpaired and the structure $s[i : j]$ is equivalent to the structure $s[i + 1 : j]$ or the nucleotide is paired with some other nucleotide in position k . Thus, the structure $s[i : j]$ is the base pair (i, k) plus the structures $s[i + 1 : k - 1]$ and $s[k + 1 : j]$.

$$E_{i,j} = \max \begin{cases} E_{i+1,j}, \\ \max_{\substack{i < k \leq j \\ (S[i]S[k]) \in B}} \{ E_{i+1,k-1} + E_{k+1,j} + W(i,k) \} \end{cases}$$

with $W(i, k) = 1$

where $E_{i,j}$ is the maximum number of non-crossing base pairs in the interval from i to j and $W(i, k)$ represents the score added for every base pair that is introduced into the structure. As the algorithm is supposed to find the maximum number of base pairs, it should increase the base pair count by one for every base pair found. Therefore $W(i, k)$ is set to 1. Implicitly, the algorithm checks all possible nested RNA secondary structures and returns highest possible number of base pairs that can occur within one structure.

$E_{i,j}$ can also be interpreted as an energy that is maximized by maximizing the number of base pairs. The model can be extended by introducing a base type specific scoring [40]:

$$W(i, k) = \begin{cases} 3 & \text{if } S[i]S[k] \in (GC, CG) \\ 2 & \text{if } S[i]S[k] \in (AU, UA) \\ 1 & \text{if } S[i]S[k] \in (GU, UG) \end{cases}$$

However, these maximum matching algorithms lead to poor results, mainly because the main contribution to free energy is not by the formation of base pairs through hydrogen bonds but by the stacking of π -electron systems. A major improvement to the algorithm was achieved by defining the energy function with respect to the loops that are enclosed by base pairs instead of the base pairs themselves [59].

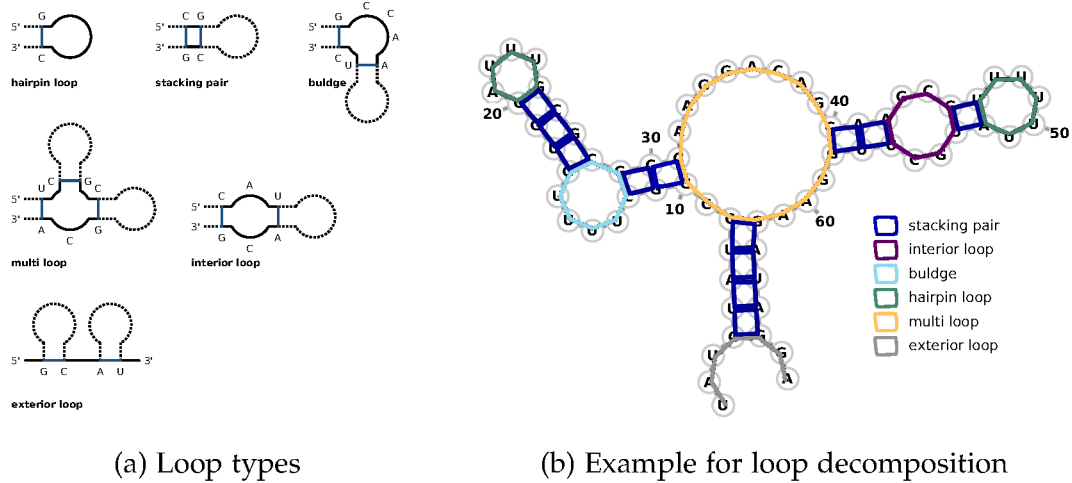


Figure 5: (a) Loop types according to the nearest neighbor model are: hairpin loops that are enclosed by exactly one base pair, interior loops that are formed by two base pairs, bulges that are completely unsymmetrical interior loops with unpaired nucleotides only on one side, stacking pairs that are interior loops without any enclosed unpaired nucleotides, multi loops that are enclosed by minimum three base pairs and exterior loops that are not closed off by a base pair, but can include unpaired nucleotides and base pairs. (b) This example for loop decomposition features all loop types distinguishes by the nearest neighbor model.

1.3.2.1 Loop decomposition

A loop is a part of an RNA structure that is enclosed by base pairs (and backbone) but does not contain any base pairs within. The important influence of base pair stacking and entropic effects are much better captured by using such loops instead of base pairs as the smallest building blocks of a structure. Figure 5(b) provides an example for how a RNA structure is split into its loops. The loops differ on how many helices branch from them and how many unpaired bases they enclose. With respect to the number of branching helices three main loop types can be defined: hairpin loops, the origin of one single helix, interior loops, have two helices branching from them and multi loops which have more than two helices branching from them. Two special cases of interior loops can be identified based on the enclosed unpaired bases: If only one side of the interior loop includes unpaired bases, a bulge is formed. If two base pairs are directly adjoint, they form a stacking pair. A graphic representation of these loop types can be found in figure 5(a). The number of helices branching from a loop is equivalent to the number of base pairs that enclose a loop.

1.3.3 Nearest neighbor model energy parameters

The possibility of decomposing a structure R into its loops suggests that its energy $E(R)$ can be computed from the sum over its loops:

$$E(R) = \sum_{L \in R} E_L$$

While the energy contribution of some small hairpin and interior loops were determined in UV-melting experiments, most energies are derived from mathematical models through interpolation between experimental data points.

Particularly, estimating the energy contribution E_M of a multi loop is challenging, because it depends on the structures it encloses. Although polymer theory suggests a logarithmic dependency on the size of the loop, a linear model is usually used for the sake of complexity.

$$E_M = a + b \cdot \# \text{ of branches} + c \cdot \# \text{ of unpaired nucleotides}$$

where a is a penalty for closing a multi loop, b is a score added for every stem (branch) connected to the multi loop and c being the contribution added for every unpaired nucleotide directly in the multi loop.

These energy parameters can be obtained from the Nearest Neighbor Data Base (NNDB) [54].

1.3.4 Zuker's algorithm

In 1981 Michael Zuker and Patrick Stiegler published a dynamic programming algorithm based on Nussinov's maximum matching algorithm that distinguished between hairpin, interior and multi loops. Thus, it is capable of computing the minimum free energy structure based on loop energies [71].

The original Zuker decomposition is not unique. Therefore, some structures are checked multiple times. While this does not influence the result of minimum free energy predictions, it cannot be used if properties of the structure ensemble, like the partition function [33], are of interest. Furthermore, while it takes multi loops into account in the decomposition, it lacks a model for computing their energy contribution.

This was solved by adding the multi loop energy model introduced in the Nearest Neighbor parameters section and by extending the recursion by another dynamic programming matrix (M^1) [72].

Nowadays, the following recursive decomposition is widely used, amongst others in the RNAfold program of the ViennaRNA package [23, 31]:

$$\begin{aligned}
F_{i,j} &= \min \begin{cases} F_{i,j-1}, \\ \min_{i < k \leq j} F_{i,k-1} + C_{k,j} \end{cases} \\
C_{i,j} &= \min \begin{cases} H_{i,j}, \\ \min_{i < k < l < j} I_{i,j;k,l} + C_{k,l}, \\ \min_{i+1 < k < j-1} M_{i+1,k-1} + M_{k,j-1}^1 + a + b \end{cases} \\
M_{i,j} &= \min \begin{cases} M_{i,j-1} + c, \\ \min_{i < k < j} (k-i)c + C_{k,j} + b, \\ \min_{i < k < j} M_{i,k-1} + C_{k,j} + b \end{cases} \\
M_{i,j}^1 &= \min \begin{cases} M_{i,j-1}^1 + c, \\ C_{i,j} + b \end{cases}
\end{aligned}$$

A graphic representation of this equations can be found in Figure 6. The parameters a , b and c are introduced in the nearest neighbor parameters section.

1.4 DYNAMICS OF RNA STRUCTURES

The structure of an RNA is not a static property. In fact, the folding process already starts while the RNA is transcribed. While nucleotides are added, the RNA keeps folding and refolding towards more stable structures. This folding pathway is not strictly deterministic and continues after transcription. Some RNAs never reach their most stable structure within their lifetime. Therefore, an RNA is better represented by an ensemble of structures than by the structure with the lowest energy.

1.4.1 Kinetic prediction methods

In theory an RNA of length n can form about $n^{\frac{3}{2}} \alpha^n$ structures [24, 60, 72]. The majority of this structures have a positive enthalpic energy in comparison to the open chain. Though this means only a small part of the possible structures is actually formed this still leaves us with a huge number of structures to investigate.

So far we mainly discussed methods that take a thermodynamic view. They investigate the energy distance between structures and can predict the likelihood of certain structures in the equilibrium. However, this steady state image does not hold for RNAs out of the equilibrium. Given the way an RNA is build, it does not start its existence from an

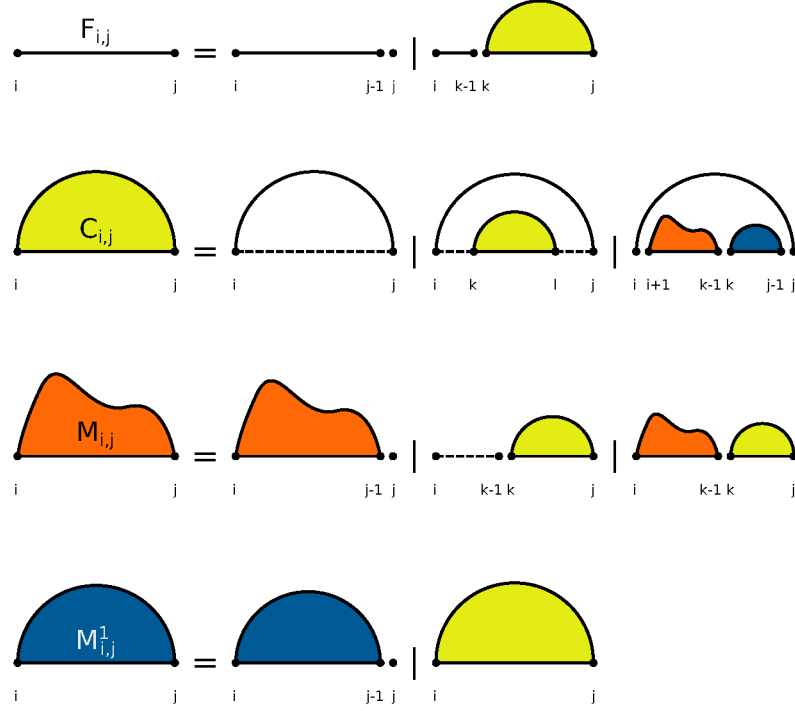


Figure 6: A graphic representation of the loop decomposition based on Bompfünnewerer et al. [4]: The $F_{i,j}$ matrix is equivalent too $N_{i,j}$ in the Nussinov decomposition and contains the possible structures of the (sub-)sequence $S[i : j]$. The last base j in the (sub)sequence can either be paired or unpaired. $C_{i,j}$ contains structures enclosed by a base pair (i, j) . These enclosed structures can be hairpin loops, interior loops and multi loops. $M_{i,j}$ and $M^1_{i,j}$ are needed to decompose structural parts of multi loops: $M_{i,j}$ for parts that include at least one stem and $M^1_{i,j}$ for parts that include exactly one stem. Single dots represent an unpaired nucleotide, dashed lines represent multiple unpaired nucleotides.

equilibrium state. Whenever a nucleotide is added during transcription, the current structure probably is not the minimum free energy structure anymore. Even after transcription ends, the RNA might not reach its thermodynamical most stable structure within its lifetime. In addition changes of the environment, like temperature changes or fluctuations in ligand concentration, might shift the thermodynamical system. Given the limited lifetime and the interactions an RNA is involved in during its folding, the structure formed is not only defined by the energy (model) of the structure alone but also by the path that an RNA has to take to fold from one structure into another.

If we want to understand the dynamic properties of an RNA we need to understand the process by which it changes its structure: On a microscopic level a change in structure is based on a change of the base pair set, by opening an existing base pair or introducing a new base pair. Using this **move set** we can also define the neighborhood relations of the whole set of possible structures. A direct neighbor is any structure that can be reached by opening or closing one base pair. The distance between any two structures can be defined as the minimum number of base pairs that have to be opened and closed to convert the first structure into the second one.

Knowing this neighborhood relations, we can view the set of structures as a **energy landscape**. The energy landscape L is defined by the set of RNA secondary structures Ω , the move set M and E , the energy function that assigns an energy score to every structure. The folding and refolding of an RNA sequence can be pictured as a walk on this energy landscape. RNA folding is a stochastic process and structures might be visited multiple times leading to cycles in the folding path. Efficient algorithms usually evaluate how fast an RNA folds from one structure into another based on cycle-free shortest paths on this energy landscape.

The occupancy $P_i(t)$ of a particular structure s_i at time t in the folding process can be computed using the master equation [16]

$$\frac{dP_i(t)}{dt} = \sum_{j \neq i} (P_j(t)k_{ji} - P_i(t)k_{ij})$$

The change of the occupancy of s_i is the sum over the influx into s_i ($P_j(t) \cdot k_{ji}$), minus the outflux from state s_i ($P_i(t) \cdot k_{ij}$).

Similar to chemical reactions, the folding rate k_{ij} is determined by the energy barrier between two structures. The exact folding barrier $\Delta G^\ddagger$ is the energy difference between the structure s_i and the highest energy along the path with the minimum barrier.

For single base pair moves the folding reaction rate can be computed based on the Metropolis rule [35].

$$k_{ij} = \begin{cases} k_0 & \text{if } \Delta G^\ddagger \leq 0, \\ k_0 e^{-\frac{\Delta G^\ddagger}{RT}} & \text{otherwise,} \end{cases}$$

where k_0 is a scaling factor, T is the thermodynamic temperature and R is the universal gas constant.

The folding kinetics of RNA structures are often investigated on coarse grained landscapes leading to larger structural changes in one move. The folding rates of such transitions are usually computed based on the Arrhenius equation:

$$k = k_0 \cdot e^{\frac{-\Delta G^\ddagger}{RT}} \text{ for } \Delta G^\ddagger > 0$$

Using these transition rates, folding trajectories on the energy landscape can be sampled with a Markov Chain Monte Carlo method [16, 25]. To cover a relevant portion energy landscape, a large amount of trajectories have to be sampled.

One example for programs that produce such stochastic trajectories is *Kinefold*.

1.4.1.1 *Kinefold*

Kinefold was published by Isambert and Siggia in 2000 [25, 67]. Its move set consists of full helices. This means, *Kinefold* trajectories can also include pseudoknots. An addition to the energy model that captures the entropy of the total structure facilitates the description of pseudoknotted structures.

The energy barrier between two steps is mainly determined by the stacking process of the single changing helix. The folding rates are described as Arrhenius rates that reflect the local stacking energies and helix mutations.

By clustering recently visited structures *Kinefold* can avoid getting trapped in circular paths and achieves a significant speed up.

1.4.1.2 *Kinwalker*

Another heuristic approach to kinetic folding is deterministic *Kinwalker* [17]. It constructs transient stable structures from thermodynamically optimal fragments. The transition rates can be computed by the Morgan-Higgs heuristic [37] that uses the energy barrier of the shortest path or by the *findpath* method [16] that tries to find the best path between two structures and uses the energy barrier of this optimized path. The optimal structures of subsequences $S[i : j]$ that form the basic building blocks of the transient structures can be obtained from the algorithms introduced in Section 1.3.4. Thus, the structures that *Kinwalker* considers are defined analog to the RNA secondary structure section in this thesis.

Kinwalker cannot predict pseudoknotted structures. After every transcription step it checks if some paths of the current sequence have not yet reached their optimal structure. If any such parts exist it tries to replace them with the optimal structure for these parts. Whether this

replacement is successful is determined by the energy needed to re-fold. Only if the transcription rate is high enough for the sequence to re-fold, the new structure is accepted. If the barriers are too high for any refolding to happen, the unused time from the current transcription step is added to the next one. Thereby, in the next step the sequence has more time to re-fold and can pass higher barriers.

1.5 RNA EVOLUTION AND CONSERVATION

The function of noncoding RNAs and regulatory RNA elements is strongly defined by their structure. Therefore, RNAs of a so called RNA family, which consist of homologous sequences that share a common ancestor, may vary greatly in sequence but still fold into a common structure.

Even though homologs acquire random mutations, evolutionary pressure selects for sequences that can adopt the functional structure. In the protein world homologous coding sequences may vary as long as their codons still encode the functional necessary amino acids. Likewise, noncoding RNAs may vary in their sequence as long as their functional structure is preserved.

The evolutionary time that passed since two homologous RNAs differentiated can be estimated by the number of mutations that they accumulated over time. Thus, also their evolutionary distance can be measured in mutations. Assuming, the shared ancestor of two homologous RNAs has a structure that is essential for its function, every mutation of a pairing nucleotide potentially destroys the functional structure, resulting in a less viable organism. To regain function and be evolutionary competitive, also the second nucleotide involved in the base pair has to mutate in a way that recovers the pairing potential. Therefore, whenever homologous sequences feature mutations that exchange two pairing nucleotides by two other pairing bases, it indicates that this base pair is important for the function of the RNA. Such double mutations are called compensatory mutations and are often used to evaluate the evolutionary pressure on the formation of a certain structure. The more compensatory mutations are found in an alignment of homologous RNAs, the stronger the underlying structure is conserved (see figure 7).

A phylogenetic tree is a way of representing the evolutionary distance computed from conservation patterns in the compensatory mutations between multiple homologous sequences in a tree structure.

While base pairing regions in RNAs often show some degree of conservation, loop regions (unpaired regions) might vary greatly. If strong evolutionary conservation is detected for unstructured regions, this might indicate that this region is involved in some other interaction, for example with ligands, proteins or other RNAs.

Based on these patterns of sequence and structure conservation a model that describes the properties of the individual RNA families

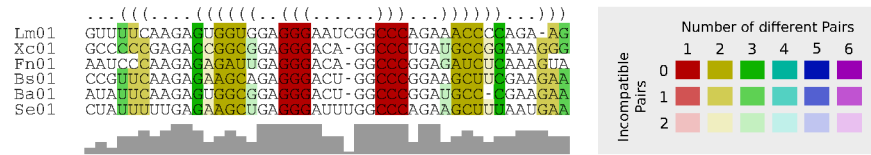


Figure 7: Conservation of RNA secondary structures: On the left side an artificial example for a sequence alignment of homologous RNAs is displayed: Every line represents one sequence. Corresponding nucleotides of the sequences are placed in the same column. On top of the alignment the shared functional structure of the sequences is depicted as dot-bracket string, that indicates which columns form base pairs. The conservation of the base pair is visualized by the coloring of the columns as depicted by the legend on the right side. The brightness of the color indicates how many sequences cannot form the respective base pair. The hue indicates the number of different base pair types that are found in the two pairing columns. This is also the minimum number of compensatory mutations that occur at this base pairing location.

can be developed. Eddy and Durbin [11] as well as Sakakibara Y et al. [50] developed such covariance models (CM) based on stochastic context free grammars.

The **Rfam data base** is the most comprehensive collection of RNA families available to-date [19]. As of June 2017, Rfam lists 2687 RNA families. For each family, covariance models were manually built from representative sequences, the seed alignment. Using these models, additional sequences that show similar features were identified [38].

1.5.1 *Transat*

Transat detects evolutionary conserved helices in alignments of homologous sequences based on their evolutionary distance.

In contrast to the previously introduced programs Kinwalker and Kinefold it does not model the in vivo co-transcriptional folding process. Instead, it relies on a purely statistical method that models evolution.

The following description of the program is based on a publication by Wiebe and Meyer 2010 [62].

In the first step, Transat defines a set of putative helices: It detects helices in the individual ungapped sequences in the input alignment. A helix is defined as a minimum of four consecutive base pairs of the type GC, GU and AU. These helices are mapped back on the alignment, resulting in a set of alignment wide helices that have to be scored based on their evolutionary conservation.

The evolution of RNA sequences can be described as a Markov process. Nucleotides that are not involved in base pairs have different

mutation rates than nucleotides that are part of a functional structure and therefore under stronger evolutionary pressure.

The evolutionary distance between two homologous sequences corresponds to the time that they had to accumulate different mutations. The Markov process that describes this evolutionary process models the substitution of nucleotides along the tree based on a mutation rate matrix.

Using the Felsenstein [14] algorithm one can compute the probability of a given alignment column to be produced by such a Markov process.

In the case of *Transat*, this probability is computed twice: once based on the mutation rate matrix for unpaired nucleotides and once based on the rate matrix of paired nucleotides.

The probability of the composition of the alignment column x and y appearing under the hypothesis θ_{unpaired} that x and y are unpaired is the product of the probabilities of column x and y being produced by the Markov process that described the evolutionary model ψ_{unpaired} for unpaired columns.

$$P(x, y | \theta_{\text{unpaired}}) = P(x | \psi_{\text{unpaired}}) * P(y | \psi_{\text{unpaired}})$$

The probability of two paired columns x and y is not independent. Their evolution is modeled using a mutation rate matrix that describes the mutation rates of both columns. The probability of

$$P(x, y | \theta_{\text{paired}}) = P(x, y | \psi_{\text{paired}})$$

The probability that all base pairs of a helix h result from the evolutionary model for unpaired columns can be computed as the product of the probabilities of the individual columns:

$$P(h | \theta_{\text{unpaired}}) = \prod_{i=1}^L P(x_i | \theta_{\text{unpaired}}) * P(y_i | \theta_{\text{unpaired}})$$

with L being the length of the helix h in base pairs.

Analogously, the probability of the columns involved in the matrix being produced by a model describing paired columns is:

$$P(h | \theta_{\text{paired}}) = \prod_{i=1}^L P(x_i y_i | \theta_{\text{paired}})$$

To estimate if the columns of the helix are more likely to be produced by a model that describes paired columns rather than unpaired columns, a length corrected log-likelihood score Λ is defined:

$$\Lambda(h) = \log_2 \frac{P(h | \theta_{\text{paired}})}{P(h | \theta_{\text{unpaired}})} \frac{1}{L}$$

The ability of RNAs to form random helices is highly dependent on factors like their length, nucleotide and di-nucleotide composition.

Thus, a log-likelihood value that is significant for one RNA family might not be significant for another. To correct for this, Wiebe and Meyer [62] defined a p-value for the log-likelihood value:

They realigned the sequences in the investigated alignment to remove structure information and shuffled it using the RNAz alignment shuffling method [58].

The log-likelihood values for helices in 500 different shuffled alignments were obtained and combined in a histogram. The log-likelihood value of every helix in the non-randomized alignment can be compared to this histogram. The fraction of log-likelihood scores in the histogram that are higher than the log-likelihood of the helix in question can be used as a p-value. Based on this p-value, conserved helices can be identified.

1.6 MOTIVATION

To identify structures that are vital for the function of an RNA, it is important to look for evolutionary conservation in the folding process. Evolutionary pressure acts on parts of the folding pathway that are essential for the correct function of an RNA. Thus, we look for similarities in folding pathways of homologous RNAs that can point to functionally important processes.

In theory, an RNA molecule can adopt a vast number of conformations. However, only a small subset of the possible conformations are actually formed in vivo. This specific set of structures is defined by the chemical properties of RNAs, the composition of the respective RNA as well as by external stimuli that the RNA is exposed to.

Similarly to the way that the shape of the extremities of an animal allows us to draw conclusions about what the extremities are used for (flying, running, swimming,...), the structure of an RNA encodes functional information. Thus, programs that can predict the structures formed by RNAs are essential for investigating their functions.

Due to their limited lifetime and dependence on changes in their environment, RNA structures in vivo are subjected to dynamic changes. Therefore, a correct prediction of RNA structures does not only require a correct model of the chemical properties and thermodynamics of RNAs, but also has to include dynamics.

One of the prominent stimuli on RNA structure formation is the sequence extension during transcription. The structures formed during transcription can exert strong control over the conformations that the RNA adapts within its life time. The co-transcriptional folding process can sometimes even control the transcription itself, as it is the case with transcriptional riboswitches.

Given these properties, the goal of this thesis is to get a better understanding of RNA structures and function using a range of structure

prediction methods that can simulate co-transcriptional, kinetic folding and detect evolutionary conservation within the folding process.

To get a better understanding of co-transcriptional kinetic folding and to test existing methods for predicting transient structures we want to investigate a large and diverse set of RNA sequences. So far most kinetic folding programs have only been tested on relatively small example sets. Additionally, most of these datasets are not available in a machine readable format and have not been kept up to date. Thus, we decided to compile a kinetic folding data set that is based on an automated and well defined process. This makes it easy to update and extend the dataset and guarantees more reproducible results than hand compiled datasets. Apart from sequence data on homologous RNAs for studies on structure conservation the dataset includes experimentally found structures that can be used for testing prediction programs.

For benchmarking purposes a stochastic folding simulation program (Kinefold), a deterministic folding simulation program (Kinwalker) and a program for detecting evolutionary conserved structures (Transat) were chosen. They were applied on homologous sequences, tested against experimental data and compared to one another. Not only the results from these but all intermediate steps and folding trajectories were added to the dataset. Thus, now the dataset can be used for benchmarking other new simulation approaches against existing methods.

To facilitate building this dataset a Python library was built. It serves as a unified interface to the kinetic structure prediction programs, can be used to automate kinetic simulations on large datasets and supports some analysis steps on the prediction results.

In the next step I developed a method for detecting evolutionary conserved structure formation patterns in folding trajectories. The main question that we wanted to answer is, if homologous RNAs reach their shared functional structure through the same folding pathway. The method is based on alignment algorithms for sequences. Instead of nucleotides in sequences it aligns structures in co-transcriptional folding time series. If a certain structure or time-series of structures appears in the folding trajectories of a large set of homologous RNAs, we can identify them as putative functional transient structures.

To assess the significance of the predictions, a statistic background model was built. It computes how likely a given structure is to appear in a sequence with given length by chance. This likelihood can be used to score the relevance of found structures in folding trajectories.

DATASET

2.1 INTRODUCTION

The most comprehensive dataset on RNA kinetics was published by the Meyer group in 2013 [70]. To reliably test RNA folding programs and to get a better understanding of the RNA structure formation process, we need an extensive and diverse dataset. Having a dataset like the one published by the Meyer group available in a machine readable format, is a valuable resource when developing new programs. While the Meyer group published figures for comparison of different programs, the detailed results are not included.

The first step in this work was to rebuild their dataset, based on the most recent Rfam database (12.3). So far it includes five RNA families featured in the Meyer group's dataset: the bacteria small signal particle RNA, the S-adenosyl methionine riboswitch, the 5'untranslated region (UTR) of the Levivirus maturation gene, the tryptophan operon leader and the hepatitis delta virus ribozyme.

One key consideration was the definition of a reproducible process for generating such a dataset that makes it easy to extend and update it. To facilitate this, I developed a Python library that serves as a unified interface to Kinefold, Kinwalker and Transat. It abstracts away the diverse and specific program calls and converts between their different input and output formats. Therefore, it allows users to run simulations on large datasets in an automated fashion. In addition it provides means to summarize, analyze and visualize the output.

Apart from simulation results for a number of homologous sequences of each RNA family, it includes some experimentally found functional and transient structures for at least one sequence in each RNA family. Based on these simulation results and experimental data, new approaches to kinetic folding can be benchmarked.

While at least some of the functions of the RNA families are known, there are still many open questions about their exact mechanisms. Based on the dataset we are able to identify candidate structures other than the final functional structure that might be essential to these functional mechanisms. Investigating simulated folding trajectories of many homologous sequences allows us to pinpoint transient structures that cover large phylogenetic distances. We expect that such structures are conserved because they are important for the function of the respective regulatory system. The homologous RNAs in RNA families feature highly similar final functional structures. In this chapter, we want to find indicators on whether co-transcriptional structure

folding is important to the function of RNAs by investigating the conservation patterns in their folding trajectories.

2.1.1 *Rfam families*

This section introduces the function and known structures of the RNA families found in the dataset.

2.1.1.1 *Bacterial small signal recognition particle RNA (RF00169) - SRP RNA*

The signal recognition particle RNA is part of the signal recognition particle (signal recognition particle (SRP)). In bacteria SRP usually consist of an 4.5S RNA and the Ffh protein, a homolog of the eukaryote SRP54 protein. Its main function is in transporting proteins to the endoplasmatic reticulum (ER). Therefore it is important for protein segregation [18, 57]. The small part of the SRP recognizes the signal recognition peptide as it is transcribed by a ribosome and pauses the transcription until the large subunit interacts with the signal recognition receptor in the ER membrane. Upon that interaction, the SRP disassembles, the signal recognition peptide is removed from the transcript and the rest of the protein is transcribed and transported into the ER through a translocation site.

The SRP RNA is conserved throughout eukaryotes. In bacteria its functional structure consists of five helices. Figure 8(c) displays this functional structure and labels the loops and helices according to Lentzen *et al.* 1996 [29]. The combination of sequence and structure motifs ensures that the bacterial SRP RNA can form the interactions that are critical for its function: The tetraloop A interacts with FFh and the signal peptide. After binding to the signal receptor on the membrane of the ER, FFh changes its conformation and interacts with the 3' side of loop E and helix f. Ataide *et al.* found evidence for an interaction between the Lys²⁷⁸ and Glu²⁷⁷ with C110 (in reference sequence, *E. coli*), in the conserved sequence motive GUGCCG (base 107 to 112) [2]. Loop B is important for conformational changes that support an interaction between the signal peptide and the SRP RNA. Lentzen *et al.* propose a bent in the otherwise rod like structure at loop B [29]. Their enzymatic studies show possibilities for additional base pairs and structures. Wong *et al.* investigated possible polymerase pausing sites, that could ensure that the folding process does not get trapped and facilitate the formation of the long range helices d, e and f in γ -proteobacteria [66] (fig. 8). They found a prominent pausing site at U106 and U108 of the *E. coli* SRP RNA and suggest a transient helix form base 25 to 43 with a hairpin at G34.

As a reference sequence for this family we chose the SRP RNA in *E. coli* (X01074.1/138-267). The functional and transient structure as well as the pausing and interaction site can be found in Figure 8.

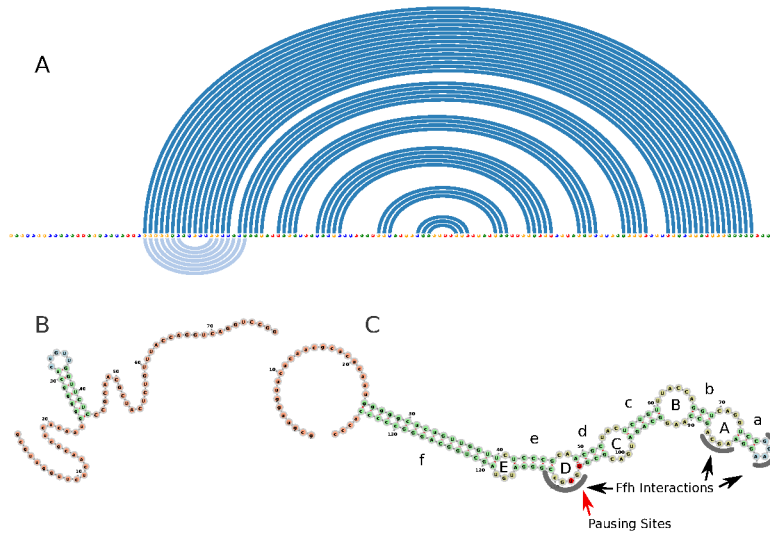


Figure 8: Functional and transient structure of the bacterial SRP RNA (Xo1074.1/138-267): The overview (A) shows where the base pairs of the functional and transient structure are located with respect to one another. The upper arcs represent the functional structure. The lower arcs represent the transient structure. The functional structure of the SRP RNA is rode-like. Important interaction sites and pausing sites are marked in (C). In (B) a transient structure that guides the folding into the functional structure is depicted.

2.1.1.2 *S-adenosyl methionine riboswitch - SAM riboswitch*

A riboswitch is a RNA control segment that can form at least two distinct structural states in. Riboswitches are often found in untranslated regions of messenger RNAs that respond to ligand concentrations. The presented riboswitch features an aptamer sequence in the 5'UTR (untranslated region) that can selectively bind S-adenosyl methionine (S-Adenosyl methionine (SAM)). It was proposed that interaction with SAM triggers the formation of a transcription terminating hairpin downstream from the aptamer sequence [21]. Winkler *et al.* [65] investigated the terminator and anti-terminator structure of the riboswitch using inline-probing and mutant studies. Based on their studies and the SAM riboswitch alignment published by Grundy *et al.* the two functional structures depicted in Figure 9 were identified.

The following mechanism can explain the function of these two structures: In the presence of SAM, the riboswitch forms a multi loop that serves as a binding site of the co-enzyme. The closing stem of the multi loop is in competition with the anti-terminator helix. In the absence of SAM the closing stem is energetically less favorable and might take longer to form. Instead, the anti-terminator structure is formed. This anti-terminator helix and the terminator hairpin are in turn mutually exclusive. Thus, to form the more stable terminator hairpin, the anti-terminator has to unfold. This delay is enough to make the formation of the terminator hairpin ineffective. Transcrip-

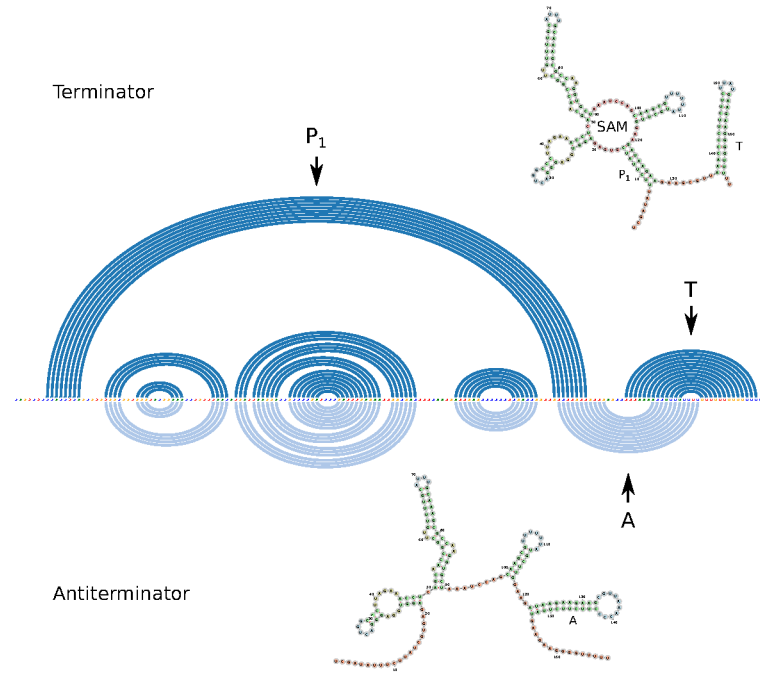


Figure 9: Terminator and anti-terminator structure of the SAM riboswitch: This figure shows a SAM riboswitch found in *Bacillus subtilis* (AL009126.3/1258297-1258464) and the experimentally found structures [65]. The upper arcs of the arcplot represent the terminator structure of the SAM riboswitch, the lower arcs the anti-terminator structure. In the presence of SAM, the aptamer domain forms a multi loop. It is closed by helix P_1 . In the absence of SAM, helix P_1 is less stable and an alternative anti-terminator helix A is formed. Helix A and helix P_1 are mutually exclusive. How fast the terminator helix T is formed, depends on whether the anti-terminator helix A is present and has to be unfolded first. If the formation of the terminator helix T is delayed, transcription continues.

tion is terminated if the closing stem of the multi loop inhibits the the formation of the anti-terminator, leading to the direct formation of the terminator hairpin. Driven by the strong (GC) base pairs, the terminator hairpin zips up and stops transcription by pulling the RNA out of the transcription complex. The poly-U stretch that follows the terminator hairpin weakens the RNA-DNA interaction and further promotes the disintegration of the transcription complex.

The SAM riboswitch is strongly conserved within gram-positive bacteria. A riboswitch found in *Bacillus subtilis subsp. subtilis* (AL009126.3/1258297-1258464) is used as a representative of this RNA family. In *Bacillus subtilis*, the SAM riboswitch exists in 11 transcription units, controlling the sulfur metabolism, methionine biosynthesis, cystein biosynthesis and SAM biosynthesis [65].

2.1.1.3 *Levivirus maturation gene 5'UTR*

Leviviruses are positive sense RNA viruses (ssRNA(+)), the most prominent species is Enterobacteria phage MS2. Natural hosts are

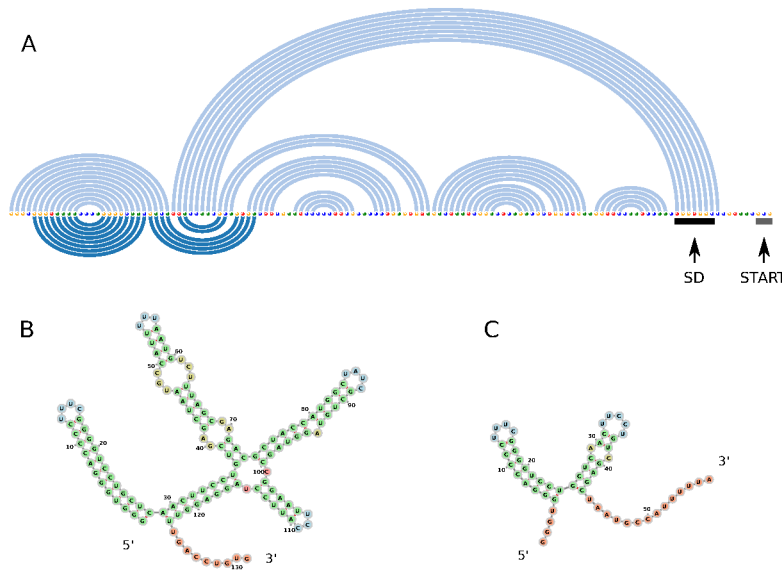


Figure 10: Equilibrium structure and transient structure of the maturation 5'UTR of the Levivirus (GQ153927.1/1-132): Subfigure (A) provides a comparison of the the equilibrium structure (upper arcs) and the transient structure (lower arcs). The Shine-Dalgarno sequence (SD) is sequestered by the long range helix of the equilibrium structure and therefore not accessible to the ribosome. The second helix in the transient structure is believed to delay the formation of the equilibrium structure and thereby allowing the translation complex to form. (B) represents the equilibrium structure and (C) represents the transient structure.

Enterobacteria, Caulobacter, Pseudomonas, and Acinetobacteria. The genome is about 4,000 nucleotides long and codes for 4 proteins: maturation protein (protein A), coat protein, lysis protein and replicase. The maturation protein A is needed to attach to bacterial pili and infect the cells. The synthesis of the maturation protein is controlled by the co-transcriptional structure formation in the 5'UTR of the gene. In the equilibrium, the 5'UTR forms a cloverleaf structure that hides the Shine-Dalgarno (SD) sequence in a long range interaction, inhibiting the translation of the maturation protein [20]. Thus, the translation is most likely initiated during the co-transcriptional folding of the 5'UTR when the SD sequence is accessible. The long range interaction that buries the SD sequence can be formed before the start codon is transcribed. It was proposed that a kinetic folding trap delays the formation of the long range interaction and thereby allows the ribosome to bind to the RNA [44]. Van Meerten *et al.* [34] performed extensive mutant experiments and identified a short helix in the beginning of the 5'UTR that is believed to be critical for this folding delay. Figure 10 introduces this transient helix as well as the equilibrium structure in the context of the reference sequence for this RNA family (GQ153927.1/1-132).

2.1.1.4 Tryptophan operon leader (RFoo513) - TRP operon leader

The Tryptophan operon contains a set of genes encoding components of tryptophan (tryptophan (Trp)) biosynthesis in bacteria (fig. 11). It has multiple layers of regulation to ensure that the proteins encoded are only expressed when needed, i.e. at low Trp concentration in the cell.

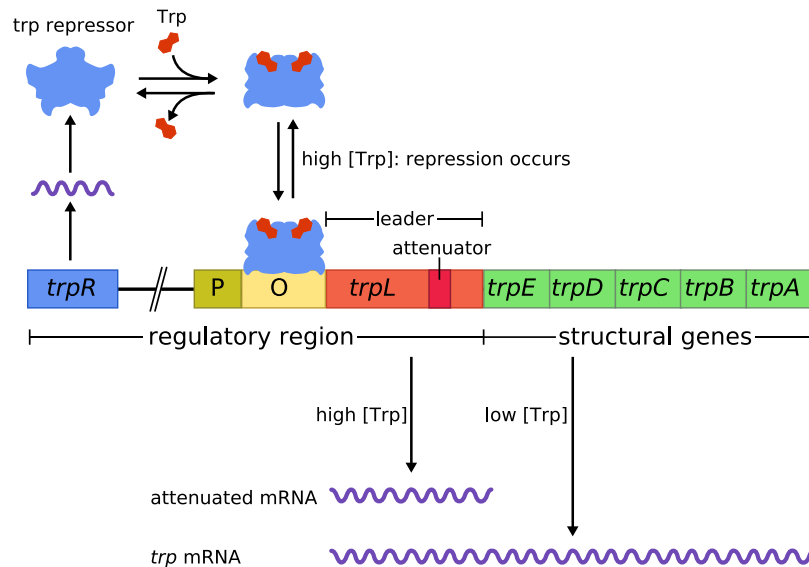


Figure 11: The gene structure of the Trp operon locus: The expression of the structural genes for tryptophan biosynthesis (*trpE*, *trpD*, *trpC*, *trpB*, *trpA*) is regulated by two mechanisms acting on the regulatory region. Both the protein (*trpR*) and the RNA based one (*trpL*/attenuator) sense Trp levels in the cell and activate gene expression only at low Trp concentrations. (figure taken from Wikipedia/Author Histidine under CC-BY-SA 3.0 https://en.wikipedia.org/wiki/Trp_operon)

The first level of regulation is protein based. Trp repressor protein (*trpR*) binds free Trp, thereby changing its conformation and binding to the Trp operator (*trpO*), a region upstream of the structural genes. This prevents initiation of transcription of the entire *trp* operon (fig. 11) with high efficiency.

A second RNA-based mechanism called attenuator, is activated by tRNAs already loaded with Trp. It uses an environmental signal, i.e. tRNA-Trp concentration, to regulate translation speed, and thereby stopping transcription. This regulatory system works in bacteria, because there transcription and translation take place simultaneously and the ribosome tightly follows the RNA polymerase.

The *trp* leader sequence (*trpL*) (fig. 11) contains several elements: a short open reading frame (ORF) rich in *trp* codons, the attenuator consisting of four regions (fig. 12) that can fold into two different RNA structures (fig. 13), and a U-rich region immediately downstream of the attenuator.

If tRNA-Trp levels are low, the ribosomes get stalled at the short ORF (fig. 12, yellow), because it takes long for a Trp-loaded tRNA

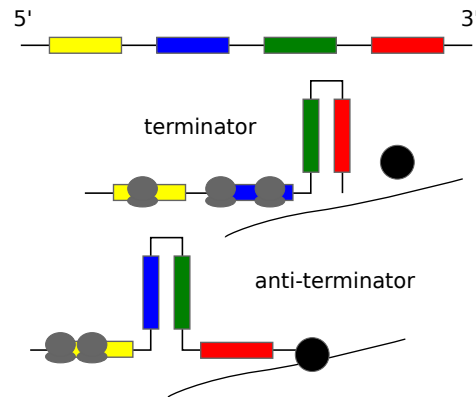


Figure 12: The repressive negative feedback mechanism of the Trp operon lead sequence: The four parts of the lead sequence, marked in yellow, blue, green and red, can form two alternative structures: The first being the more stable terminator structure formed by the green and red subsequences in the end of the lead sequence and the second being a competing anti-terminator structure that is formed by the blue and the green subsequence. Whether the terminator or the anti-terminator structure is formed depends on the Trp concentration: If there is sufficient Trp and the ribosomes are not stalled at Trp codons in the short ORF (yellow), they reach the subsequence represented in blue before the green one is fully transcribed. The anti-terminator (blue-green) cannot be formed, because the upstream strand (blue) is occupied. Thus the green part stays free to form the terminator with the red part. Backed by many GC pairs, this terminator helix zips up fast and pull the RNA out of the transcription complex and terminates transcription.

to reach the ribosomes. This allows regions 2 and 3 (fig. 12, blue and green) to form the anti-terminator hairpin. Once the ribosome continues translation and passes through the short ORF, it can easily open up the anti-terminator and proceed into the structural genes.

At high Trp concentration, however, the ribosome quickly translates the short ORF, such that region 2 (fig. 12, blue) is occupied. In this case the terminator forms (fig. 12, green and red). At this point, the RNA polymerase is about to transcribe the U-rich region immediately downstream of the attenuator. The formation of the terminator hairpin pulls the nascent RNA out of the transcription complex which then disassembles. Yanofsky *et al.* [69] also identified a polymerase pausing site upstream from the TRP codon stretch which might improve the synchronization of the ribosome attachment.

2.1.1.5 Hepatitis delta virus ribozyme (RF00094) - HDV ribozyme

Hepatitis delta virus (hepatitis delta virus (HDV)) is an enveloped RNA virus. It has a single stranded, negative strand (single stranded RNA (ssRNA)-) closed circular genome. Interestingly, the genome is highly self-complementary (74%) and thus forms a double stranded

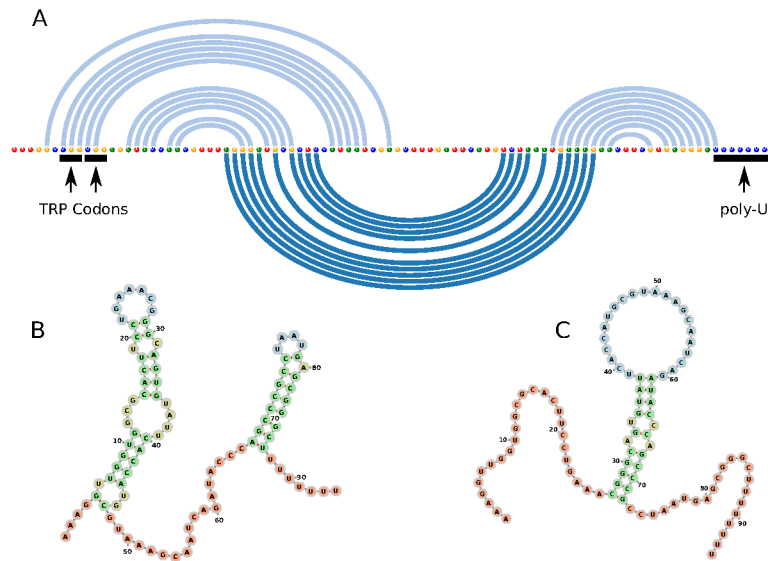


Figure 13: The terminator and anti-terminator structure of the Trp operon leader:

(A) The positions of the terminator and anti-terminator sequence with respect to one another on the reference sequence in *E. coli* (AE005174.2/2263095-2263188). The upper arcs represent the terminator structure, the lower arcs represent the anti-terminator structure. (B) The terminator structure. (C) The anti-terminator structure.

rod-like structure. The HDV genome is only 1,670 nt long and encodes a single protein, Hepatitis delta antigen (hepatitis delta antigen (HDAg)), which results in two products, small and large HDAg. HDV is a so called subviral satellite and needs hepatitis beta virus (HBV) co-infection of its host cell, because it requires HBV envelope proteins for packaging its genome.

The viral particle consists of multiple layers. The outer most coat is build of HBV envelope proteins and three HBV surface antigens (large, medium, small). The inner lining of this coat is made of host lipids. The nucleocapsid surrounds the RNA genome and is made of small and large HDAg, both of which are encoded by HDV itself. HDV is the smallest animal infecting virus. It is somewhat similar to viroids known in plants.

Production of small HDAg during the reproductive phase leads to editing of a stop codon, which then gives rise to the longer form of HDAg. This in turn induces a switch in the life cycle of HDV and triggers packaging of viral ssRNA- genomes.

The HDV genome consists of a negative RNA strand. In order to produce a mRNA for translation of HDAg, it needs to be reverse-transcribed. Unlike other ssRNA- viruses, HDV does not require a reverse transcriptase (reverse transcriptase (RTase)), but uses host polymerase polI and/or polII for production of sense RNA. The sense RNA exists in two forms, the circular anti-genome and the linear mR-

NAs. The mRNAs are made in the nucleus, most likely by host polII, are processed and polyadenylated. They are shorter than the virus genome and only serve for translation of HDAg. The anti-genome (ssRNA+) is made in the nucleolus by polI in two steps: first, a multimer is transcribed through a rolling-circle mechanism, which is then, in a second step, cleaved into monomers (ssRNA+). This cleavage step is performed by a ribozyme encoded by HDV itself. The resulting monomers are circularized and serve as templates for genomic RNA (ssRNA-). Again, the genomic RNA is produced by a rolling circle mechanism followed by self-cleavage of the HDV ribozyme. This strategy only works because (i) the virus has high self-complementary and thus (ii) both genomic and anti-genomic RNA are able to fold into the RNA ribozyme structure required for self cleavage. An overview over the life cycle can be found in Taylor 2006 [53] and Pascarella and Negro 2010 [42].

The HDV genome encodes one open reading frame (ORF) and two ribozymes, of which one is active in the genome (ssRNA-) and one is active in the anti-genome (ssRNA+). Due to the high complementarity within the HDV genome, the two ribozymes feature a similar sequence and structure. Both ribozymes can adopt an active (fig .14C) and inactive (fig .14B) structure. If the cleavage site is buried by interaction of strands upstream from the cleavage strand with strands downstream of the cleavage site, no reaction occurs. If however the upstream region forms a stable helix in itself, the cleavage site gets exposed and the RNA gets processed. In wild type cells both structures occur. The ratio of the two structures is determined by their stability and balances the production of mRNA and the synthesis of antigenome.

2.2 METHODS

2.2.1 Retrieving the raw datasets

The initial sequence alignments and reference structures for the five RNA families that are included into the dataset for now, are obtained from the sources described by Zhu et al. [70]. Table 1 list the alignment source, the alignment length, the accession number and genome coordinates of the reference sequence and literature that describe structures that are formed by the reference sequence.

The original dataset was based on Rfam 9.0. For this study I updated to the current version Rfam 12.3.

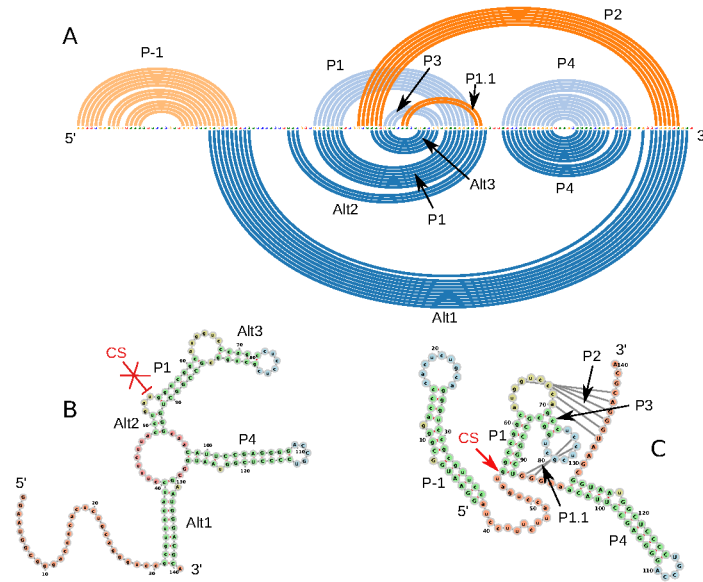


Figure 14: The reactive and non reactive structure of the HDV ribozyme:
The reference sequence for HDV ribozyme is obtained from genomic ribozyme sequence M28267.1/635-775.
The alternative helices found in the two structures can be identified in (A): In the reactive structure the region upstream from the cleavage site (CS) forms a hairpin structure (P_{-1}). The ribozyme sequence forms a complex pseudoknotted structure.
The non reactive structure features an alternative helices to P_{-1} and P_2 : The downstream stem of P_{-1} forms a long range interaction with the downstream stem of P_2 (labeled Alt1). P_1 is extended in both directions leading the the disassembly of the pseudknots P_2 and $P_{1.1}$ (Alt2 and Alt3). The cleavage sites is enclosed by on of this extensions (Alt2).
The sequence of the region upstream is essential for the formation of the Alt2 structure. In contrast to previous publications this means that sequence upstream from the cleavage site can have a high influence on the cleavage process.
Subfigure (B) represents the inactive structure while subfigure (C) represents the active structure.

SRP RNA	
Source	Rfam 12.3 seed alignment (RF00169)
Alignment length	261 sequences
Reference sequence	X01074.1/138-267 <i>Escherichia coli</i>
Literature	[2, 29, 66]
SAM riboswitch	
Source	Winkler Wade C et al. [65]
Alignment length	107 sequences excluding the reference
Reference sequence	AL009126/1258276-1258464 <i>Bacillus subtilis</i>
Literature	[65]
5'UTR of the Levivirus maturation gene	
Source	National Center for Biotechnology Information (NCBI) database (https://www.ncbi.nlm.nih.gov/) [10]
Alignment length	12 sequences
Reference sequence	GQ153927/1-132
Literature	[34]
TRP operon leader	
Source	Rfam 12.3 seed alignment (RF00513)
Alignment length	22 sequences excluding reference
Reference sequence	AE005174.2/2263095-2263188 <i>Escherichia coli</i>
Literature	[69]
HDV ribozyme	
Source	Rfam 12.3 seed alignment (RF00094)
Alignment length	33 sequences
Reference sequence	M28267.1/635-775
Literature	[6]

Table 1: Overview over the datasources used to investigate the RNAfamilies in this dataset

2.2.2 Alignment extensions

As some of the Rfam alignments miss upstream parts that are involved in regulatory structures, the first step was to extend the sequences and to find an alignment for these extensions.

As suggested by Zhu et al. [70] the alignments of the SRP RNA family was extended to the transcription start side (transcription start side (TSS)) that is located 32 nts upstream from the Rfam alignment. In the case of the TRP operon leader the Rfam alignment already includes the TSS. The HDV ribozyme is extended by 52nts in 5' direction to include both transient upstream helices.

It was assumed that the transcription start site is situated at the same position for all homologs and thus the length of the extensions is based on the reference sequence.

This extensions were obtained from Genbank (www.ncbi.nlm.nih.gov), aligned with the M-Coffee mode of T-Coffee [39], and subsequently concatenated onto the original Rfam alignment.

The alignment program muscle [12, 13] that was used in this step by the Meyer group, would be ideal for the realignment of the extensions, because it is able to align long alignments in short time. However, in the current version there is no option for turning off free endgaps, which is essential if we want to concatenate the extensions onto the existing alignments.

2.2.3 Filtering and Clustering

To investigate the conservation of folding pathways, we are mainly interested in comparing sequences that form the same functional structure.

Thus, we filtered the homologous sequences based on their ability to fold into the functional structure of the reference sequence. Only those sequence that can form a significant percentage of the base pairs in the reference structure are considered further.

If the reference sequence was not already part of the Rfam alignment, it was aligned to it. In the next step I mapped the functional structure of the reference onto the alignment. Every homologous sequence was scored based on how many of the reference base pairs it can form: every confirmed base pair scored 1, a deleted base pair scored 0. Cases where one nucleotide of a pair was missing or the nucleotides could not form a base pair scored -1. This is done to represent the fact, that a one-sided gap would lead to a bulge which destabilizes a structure, while a two sided gap means that the structure is shorter but can still feature a consecutive helix.

While the Meyer Lab decided to work with the 10 to 25 top ranked sequences, I found that in many cases this leads to a set of very closely

related sequences that only represent a small part of the known homologs. To avoid this, I clustered the sequences based on their phylogenetic distance. Running analysis within single clusters provides information on the conservation of co-transcriptional folding behavior within this subset of sequences, while analysis on cluster representatives can be used to identify folding steps that are preserved within a larger set of the known homologous sequences. The phylogenetic distance between homologous sequences was obtained using FastTree 2.1.10 [45, 46]. Based on this distance score a clustering is performed using the affinity propagation method provided by scikit-learn 0.19.0 [43].

For each RNAfamily different subsets of homologous sequences were defined based on this filtering and clustering methods. Some of these sets are better suited for describing the total set of homologous RNAs, others better for describing individual groups in the set. To test the influence of the alignment quality, new alignments of some of the subsets were generated with muscle 3.8.31 [12, 13] and LocARNA 1.9.2 [63, 64].

2.2.4 Prediction of transient structures

The goal of this work is to investigate the evolutionary conservation of transient structures. Candidate structures, or more precisely candidate base pairs, were obtained from folding trajectories computed by Kinfold and Kinwalker. In a second step the candidate base pairs are scored based on what fraction of known homologous sequence can form them. The results are then compared to the conserved, possibly transient, helices detected by Transsat.

2.2.4.1 Kinwalker

Kinwalker takes a single RNA sequence as input and returns a single deterministic folding trajectory. When simulating co-transcriptional folding, following parameters are used:

- First one can choose between two different methods for estimating the energy barrier that has to be crossed to refold from one state to the next state: The default Morgan-Higgs heuristic, only considered the direct path between two states. If any path exists that has a lower barrier but contains more base pair moves, it is not detected by this method and the barrier is overestimated. In contrast, the `findpath` method performs a breadth first search and is thereby able to find paths with lower energy barriers. The breadth of the breadth first search can be specified.
- Using the `dangling` parameter one can decide whether the energy contribution of unpaired bases that can stack onto the end of a helix is considered.
- Finally the transcription rate can be specified.

RNAfamily	experimentally found transcription rate
SRP RNA	22.5 nts/s
SAM riboswitch	75 nts/s
LEVI virus	30 nts/s
TRP operon leader	22.5 nts/s
HDV ribozyme	20 nts/s

Table 2: Transcription rate found in experiments [70]

Except for the transcription rate, the Meyer group used default parameters, which means the energy barrier is computed by the Morgan-Higgs heuristic and the stacking energies of dangling nucleotides are not considered. The transcription speed was set according to the experimentally found transcription speed (see Table 2 for a summary) [70].

This could lead to some issues because the wall clock time of `Kinwalker` does not directly translate into seconds. To test whether the experimentally found transcription rates also lead to reasonable results when used as `Kinwalker` input parameter, different transcription rates were tested. Based on this test we can estimate how the arbitrary time units used in the simulation program translate to real time. Apart from that it ensures that we use reasonable input parameters..

We defined an alternative parameter set that extends the energy model by including dangling bases (`--dangle 2`) and optimizes the refolding path between two structures by applying the `findpath` method with a breadth of 20 (`--barrier_heuristic B --maxkeep 20`).

`Kinwalker` was run on all sequences in the unfiltered dataset. To check if changing the parameters influences the results in any significant way, folding trajectories were simulated with both parameter sets.

TESTING DIFFERENT TRANSCRIPTION RATES: For `Kinwalker` simulations a transcription speed has to be set. There exists experimental data on the transcription speed of a range of polymerases [70]. However the time units of the simulations are unknown. Therefore, the experimentally found transcription speeds cannot be directly translated to the transcription speed in the simulations. Up to today time resolved experimental data on RNA structure formation is rare. To map the time units one requires time resolved series of structures that are formed during the folding process. I decided to use the known functional and transient structures to test whether or not assuming that the experimental and computational time units are equivalent leads to plausible results. `Kinwalker` runs with various transcription rates were performed. Then I tested if and in what range of transcription speeds the experimentally found transient and functional structures are recovered in the folding trajectory. `Kinwalker` was run on the reference sequences with transcription rates from 0.0001 to 1000000 nts/s. The percentage of base pairs of the known transient

and functional that is recovered dependent on the transcription rate is returned. Based on this data we found that using the experimentally found transcription rates presented in Table 2 are within a reasonable range. The detailed results of this transcription rate test is presented in the Results section.

EVALUATION OF THE SIMULATED TRAJECTORIES: The conservation of transient structures in the folding trajectory is analyzed on a base pair level. The base for each evaluation is the alignment of the investigated set of homologous sequences. For each sequence the set of base pairs that appear in the simulated trajectory is extracted and mapped onto the alignment. Each of this mapped base pairs is then assigned a score that represents the fraction of the trajectories that it is found in. The analysis returns a list of scored base pairs.

2.2.4.2 *Kinefold*

Kinefold performs stochastic folding simulations of single sequences. To capture the most relevant transient structures, for each sequence 200 trajectories were computed in the co-transcriptional folding mode with included pseudoknots.

Compared to Kinwalker, changing the transcription speed had less influence on whether or not the experimentally found structures were detected. While it did change the likelihood of finding a given structure in a single trajectory, to fully understand this factor a more in depth investigation than possible within the scope of this work has to be performed. Thus, we decided to use the experimentally found transcription rates also used in the publication by the Meyer group [70] and summarized in Table 2.

EVALUATION OF THE SIMULATED TRAJECTORIES: Analog to the evaluation of Kinwalker trajectories, the conservation in Kinefold trajectories is analyzed on a base pair level based on alignments.

For each of the 200 runs on a single sequence the list of base pairs that occur in the folding trajectory is extracted. The base pair lists of the individual runs are merged and each base pair is scored based on in how many trajectories it appears. The scored base pairs of the individual sequences are then mapped onto the alignment. The alignment wide score is obtained by summing up the scores from the individual sequences and dividing by the number of sequences in the investigated alignment.

2.2.4.3 *Transat*

Transat uses the alignment of the investigated sequence set and a phylogenetic tree that describes the evolutionary distance between the sequences in the alignment. The phylogenetic tree was prepared with FastTree 2.1.10 [45, 46] for nucleotide alignments.

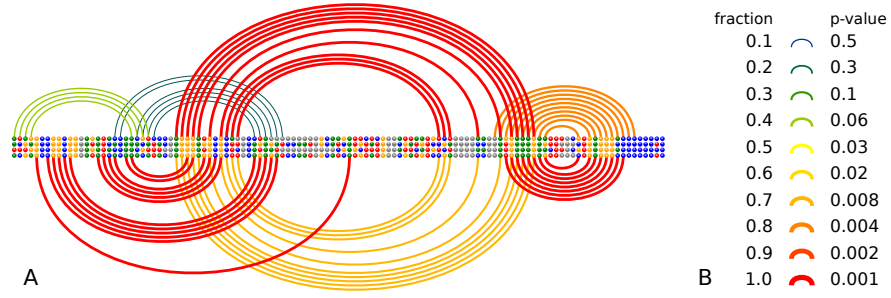


Figure 15: Color scheme used for arcplots of Kinwalker, Kinefold and Transat results: (A) The center part of the arc-plot represents the alignment of the investigated sequences. The nucleotides are colored based on their bases, gaps are depicted in Grey. The lower arcs represent the base pairs of the experimentally found structures that were introduced in Section 2.1.1. Base pairs that belong to different structures are depicted in different colors. The base pairs found with Transat and in the folding trajectories from Kinwalker and Kinefold are represented by the upper arcs. The found base pairs are colored based on their significance: The scores obtained from Kinwalker and Kinefold represent the fraction of trajectories in which a given base pair was found. Table (B) shows what color correspond to what fraction of trajectories. The base pairs found with Transat are scored with the p-value of the corresponding helix. P-values of 0.001 or below are depicted in Red, higher p-values are colored according to Table (B).

EVALUATION OF THE DETECTED HELICES: Transat returns a list of helices that are assigned a log-likelihood score and a p-value. The helices are converted to base pair lists that are then merged into one single list. Each base pair gets score based on p-value of the corresponding helix with the lowest p-value.

2.2.5 Visualization of the results

The base pairs found using Kinwalker, Kinefold and Transat were compared to one another and experimentally found structures using arc-plots (see Section 1.2.2). If not stated otherwise the color scheme presented in Figure 15 is used.

2.3 RESULTS AND DISCUSSION

2.3.1 Kinwalker transcription rate test

To check whether using the experimentally found transcription rates are also a reasonable choice for the Kinwalker simulations, we tested how many base pairs of the known structures are recovered within the folding trajectories dependent on the transcription rate. Figure 16 represents the result of this study.

Some examples like the HDV ribozyme are more susceptible to changes in the transcription rate, while other families have very robust folding trajectories. An extreme example for such robust trajectories is the levivirus maturation gene 5'UTR. The sequence of the reference RNA follows the same folding pathway over the whole range of investigated transcription rates. Only when the the transcription rate is extremely high it changes its behavior: At about $1.000 \frac{\text{nt}}{\text{s}}$ the sequence is already fully transcribed before structure formation starts. At this limit the simulation actually represents the kinetic folding from the fully transcribed open chain conformation towards the minimum free energy (MFE) structure. Interestingly in this case the 5'UTR actually forms the delay inducing hairpin the was not detected in the co-transcriptional Kinwalker folding trajectories (Section 2.3.4).

The SRP RNA and the Sam riboswitch recover the full known final and transient structures. Kinwalker can not model pseudoknots, therefore it can not recover all base pairs of the HDV ribozyme structures. However, the experimentally found transcription rate of $20 \frac{\text{nt}}{\text{s}}$ recovers the maximal possible number of base pairs. The simulated TRP operon leader trajectory at the experimentally found $22.5 \frac{\text{nt}}{\text{s}}$ does not form the anti-terminator structure. As this, is what we expect in the absence of ribosomes, the experimentally found transcription rate might be a good choice never the less.

The transient known hairpin of the levivirus maturation gener 5'UTR does not appear in any of the simulated co-transcriptional folding trajectories. Thus, the formation of this delay inducing hairpin is controlled by factors other than transcription speed.

2.3.2 Improved Kinwalker parameters

In our Kinwalker simulations we used an improved energy model that can model the energy contribution of dangling ends and a more sophisticated method for evaluating the energy barriers along the folding path, `findpath`. Figure 17 is an example for how using the improved parameters influences the folding simulation. When computing the folding trajectories using the improved model experimentally known functional structures are similarly or higher conserved than when performing the analysis using default parameters. At the same time structures that do not seem to be conserved, do not appear when using the improved model. This might be due to the fact that the `findpath` provides a better estimation of folding barriers and folding trajectory find the best refolding path faster.

2.3.3 Trp operon leader

According to what is known about the function of the TRP operon leader, the sequence forms a pre-terminator hairpin that suppresses the formation of the anti-terminator if no ribosomes attach to it. Kin-

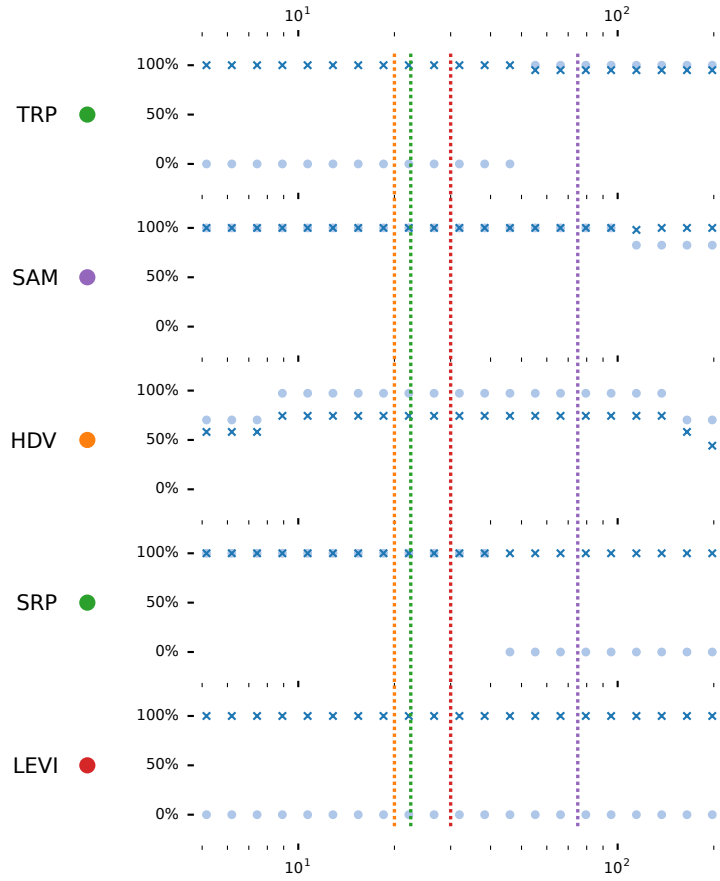


Figure 16: Influence of the transcription rate on the recovery of known final and transient structures:

This figure contains one plot per investigated RNA family. The y-axis represents the percentage of base pair of the final (dark blue x) or transient (light blue o) structure that is found in simulated Kinwalker trajectories at a given transcription rate (x-axis). The experimentally found transcription rates for the individual RNA families are depicted in as vertical lines: $20 \frac{\text{nt}}{\text{s}}$ for the HDV ribozyme, $22.5 \frac{\text{nt}}{\text{s}}$ for the TRP operon leader and the bacterial SRP RNA in *Escherichia coli*, $30 \frac{\text{nt}}{\text{s}}$ for the levivirus and $70 \frac{\text{nt}}{\text{s}}$ for the SAM ribositch in *Bacillus subtilis*.

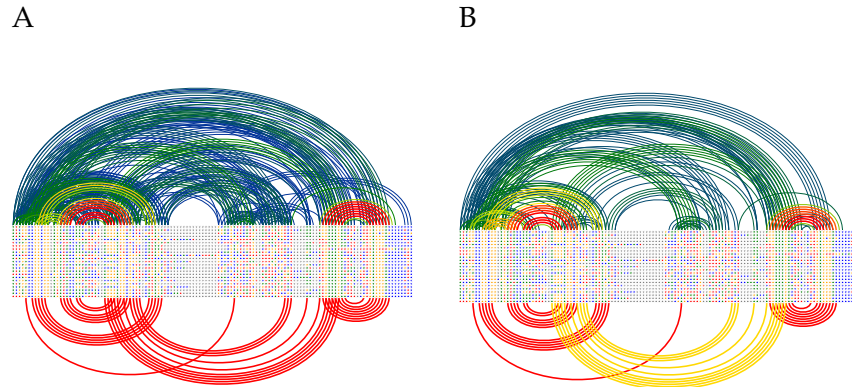


Figure 17: Kinwalker results for TRP operon leader alignment using default Kinwalker settings (A) and improved settings (dangle 2 and findpath) (B)

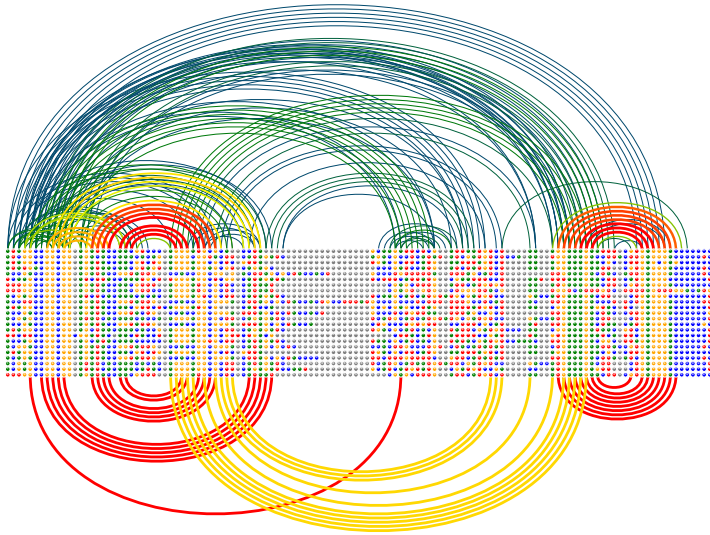


Figure 18: Kinwalker results on the full Rfam seed alignment of the TRP operon leader: The upper arcs show that the two terminator helices are found in the folding trajectories of all homologs, while most sequences do not form the anti-terminator during co-transcriptional folding. In addition to the two terminator hairpins a small helix is found upstream of the pre-terminator in a significant percentage of the sequences. Especially, in the paring regions of the known structures the sequences are well aligned. Six sequences seem to follow a slightly different sequence pattern. They can be identified by the UUCUUGGUUU sequence in the downstream part of the pre-terminator hairpin. A description of the used color scheme is given in Figure 15.

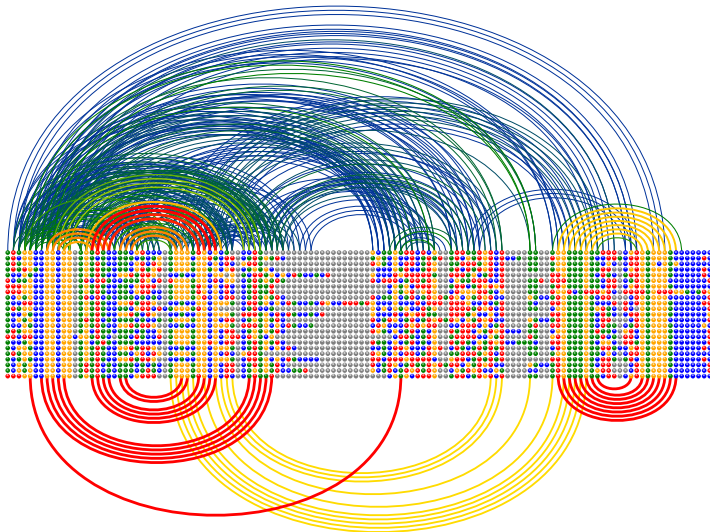


Figure 19: Kinefold results on the full Rfam seed alignment of the TRP operon leader: The Kinefold simulations show a similar base pair pattern as the Kinwalker results. The pre-terminator is found in every sequence and also the small helix upstream from the pre-terminator is well conserved. In cotrast to the Kinwalker results, the terminator helix itself is only found about 50% of the trajectories. A description of the used color scheme is given in Figure 15.

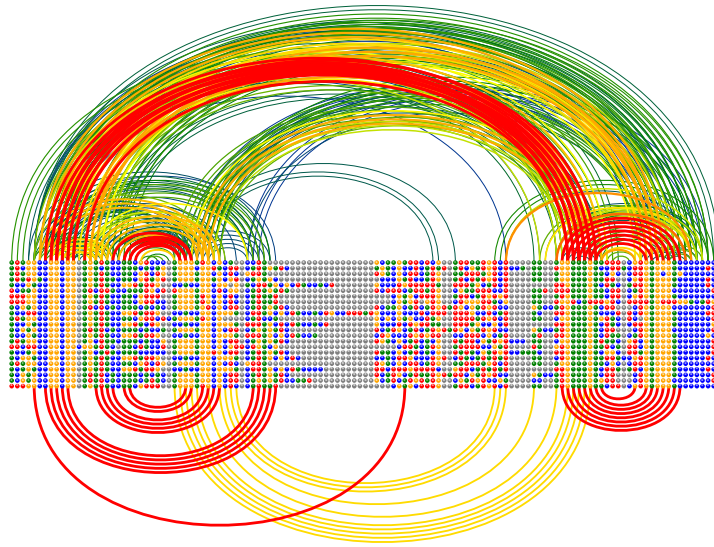


Figure 20: Conserved helices in the TRP operon leader family identified by Transat: Transat recovers the base pairs involved in the pre-terminator and terminator hairpins with a significant p-value. Even though the p-value is higher, also the anti-terminator is recovered. Three additional helices can be identified: The small hairpin in the 5' part of the pre-terminator that was also identified in the Kinefold and Kinwalker trajectories; base pairs between the upstream region of the pre-terminator and the downstream region of the terminator that can extend the anti-terminator and a long range interaction (L) between the upstream part of the pre-terminator and the upstream part of the terminator. (for the color scheme see Fig. 15)

walker and Kinefold cannot simulate interactions with ribosomes or any other biomolecules. Thus, we expect the folding trajectories to directly fold into the pre-terminator and terminator structure, while the anti-terminator does not appear. In deed, the kinetic folding trajectory of the reference sequence obtained from Kinwalker does not form the anti-terminator. Only the core of the anti-terminator sequence that is not in competition with the two terminator hairpins appears.

In Figure 18 each base pair that appears in the Kinwalker trajectory of the homologous TRP operon leader sequences is shown. While both terminator helices appear in the folding trajectory of all homologous sequences, the anti-terminator structure appears in less than one in ten homologs.

Analog to the Kinwalker results, the Kinefold results do not indicate that the anti-terminator is formed in the folding trajectories. Interestingly also the terminator is only formed by about half of the trajectories.

Even though we cannot find the anti-terminator in the simulated folding trajectories, we expect the homologs to be able to form them in the presence of ribosomes. Transat should be able to detect conserved helices independent of energy model and other factors like interaction partners, transcription speed and temperature. Thus, we would expect it to be able to identify the anti-terminator hairpin. The results of the Transat-based analysis in Figure 20 show that there is significant conservation of all three experimentally found structures. However, the signal for the anti-terminator structure is not as strong as the one for the two terminator helices.

By clustering based on the phylogenetic tree we were able to distinguish two different groups of sequences in the Rfam alignment, that we already identified in the Kinwalker results figure (Fig. 18). Transat was applied on both groups individually and the results are shown in Figure 21. By splitting the sequences provides some explanation on why the anti-terminator is found with less significance than the two terminator helices: The used Rfam alignment is optimized to support the terminator structure. However the position of the anti-terminator structure is slightly shifted between the two groups. To be able to identify the anti-terminator in both groups a realignment is necessary.

To my surprise Transat does assign a comparably low significance to helices that are supported by all sequences in the alignment, like the anti-terminator and pre-terminator helix in Figure 21(B). One possible explanation for the lower significance of the known structures in group (A) is based on the used evolutionary model: Transat scores pairing alignment columns based on an evolutionary model. In this model it might be unlikely that two pairing columns do not have any compensatory mutations at all. Thus areas with a fully conserved sequence might be penalized. Furthermore, Transat does not consider the fact that many of the bases are under exceptional constraint by having to maintain pairing with more than one partner in alterna-

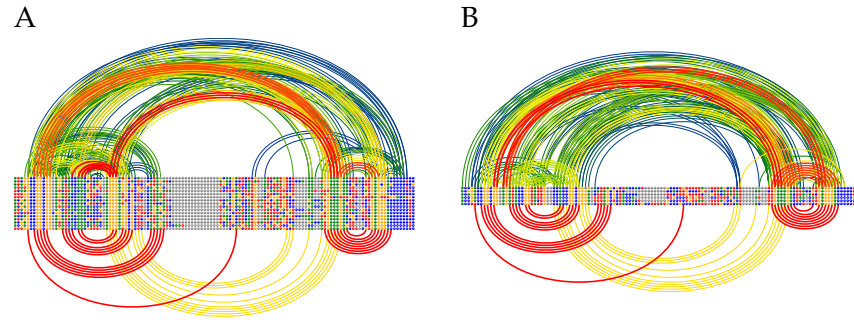


Figure 21: Transat results of the two TRP operon leader sub-groups:
 Subgroup (A) includes the reference sequence (AE005174.2 - E.coli): After removing the 6 slightly different sequences, Transat assigns higher p-values to the base pairs of the anti-terminator. The long range interaction between the upstream part of the pre-terminator and the upstream end of the terminator L already identified in Figure 20 appears with high significance.
 (B) The outliers found in Figure 18 form a set of their own. Within this set out of the experimentally found structures only terminator hairpin is found with a p-value below 10^{-3} . While base pairs of the pre-terminator and anti-terminator are still recovered with a p-value of about 0.05, the long range interaction (l) already identified in (A) is clearly more significant. The base-pairs of the anti-terminator appear slightly shifted with respect to the reference.

tive structure elements. This could in part explain why Transat has difficulties detecting the functional structures.

In summary the study on the TRP operon leader shows that the kinetic folding programs are able to reproduce the folding path that was suggested by literature through experiments. However, they are not able to recover the functional anti-terminator structure, because they can not simulate the interactions between the mRNA, the ribosome and the trp loaded t-RNA. Transat is able to identify the terminator structure as well anti-terminator structure without explicitly simulating the bio-molecular interactions. Problems arise when different steps in the folding process require a different alignment to support shared structures. In addition to the three known structures we identified two new transient structures that are conserved in the folding pathway: A small helix in the upstream region of the pre-terminator and a more long range interaction between the upstream region of the pre-terminator and the upstream region of the terminator.

2.3.4 *Levivirus maturation gene 5'UTR*

In its grounds state the Levivirus maturation gene 5'UTR forms a cloverleaf like structure that sequesters the Shine-Dalgarno (SD) sequence. Meerten, Girard G, and Duin [34] suggested that the translation of the coding sequence is initiated before the UTR reached its

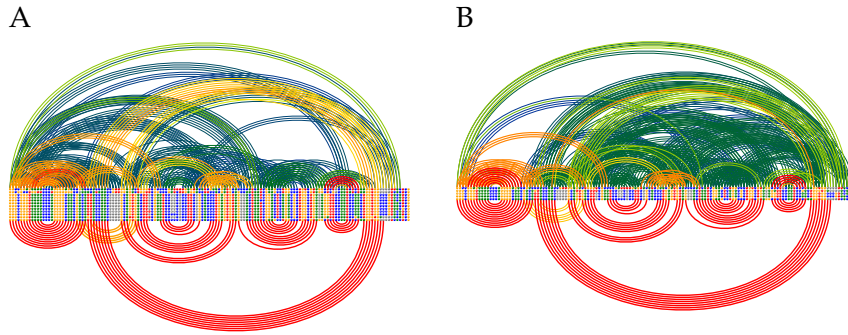


Figure 22: Kinefold results for the levivirus maturation gene 5'UTR based on the full sequence set (A) and cluster representatives(B): While the analysis on the full sequence set (A) suggests that at least two interior stems of the cloverleaf structure as well as the long range helix are conserved, studying the set of cluster representatives (B) shows not strong indicate for the formation of the cloverleaf structure within the simulated folding trajectories. Both subfigures suggest that the transient helix found by Meerten, Girard G, and Duin [34] as well as the hairpin upstream from the cloverleaf structure appear in most folding trajectories.

final conformation and that the formation of the long range helix that blocks the SD sequence is delayed by a short transient hairpin near the 5'end of the UTR.

The set of Levivirus maturation gene 5'UTRs that are found in the NCBI database mostly consists of RNAs with a high sequence identity. The problem with such sets is illustrated in the Kinefold results figure (see Fig. 22): We were able to differentiate between three main sequence groups. One can be represented by the KU1 phage and contains 2 sequences, one can be represented by the MS2 phage and contains 10 sequences while the last group is actually a single sequence that is placed with the KU1 group in the NCBI database, but actually the first two thirds of its sequence are identical to MS2 and only the last third is more similar to KU1. As the overwhelming majority of the sequence are highly similar to the MS2 phage and only differ by a handful of mutations, scoring base pairs based on how many the result mainly represents the behavior of the MS2 group. Choosing representatives for the groups To get a more balanced picture we can cluster the sequences based on their evolutionary distance and choose one representative from each cluster. Using such a set is much better suited to investigate conservation, because finding the same structure in two almost identical sequence often just means that the evolutionary distance is too small to accumulate any mutations. In the case of the levivirus example, we see that conservation of the long range helix that sequesters the SD sequence is overestimated in Kinefold trajectories when studying the full sequence set. After correcting for sequence identity we find that, the long range interaction is less conserved and might not be as essential to the function of the UTR as first indicated.

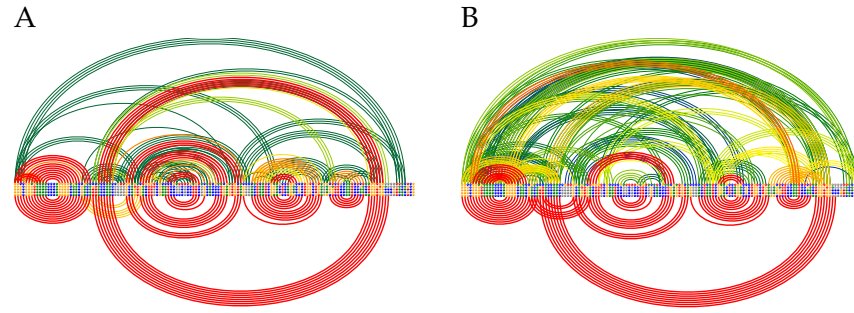


Figure 23: Kinwalker (A) and Transat (B) results for the levivirus maturation gene 5'UTR based on cluster representatives: (A): While all helices involved in the ground state structure of the 5'UTR are conserved in the simulated Kinwalker trajectories, only one single sequence forms the transient structure that is supposed to delay the formation of the inactive final structure. (B): While Transat indicates, with varying significance, that all helices involved in the ground state structure are conserved, while the delay inducing hairpin is not considered conserved.

In contrast to the Kinefold results (Fig. 22/B), the Kinwalker simulations (see Fig. 23/A) support the formation of the inactive cloverleaf structure while the delaying hairpin is not conserved. In fact when, looking at the individual Kinwalker trajectories we found that they directly fold into the inactive structure without any visible pausing. One explanation could be that pseudoknots that can only be modeled by Kinefold play an important role in stabilizing the delay inducing hairpin.

While the Kinefold trajectories prove that the delay inducing hairpin can be formed by most sequences, it is not considered evolutionary conserved by Transat. Studying the KU1 and MS2 group individually allows us to pull apart their contributions to the overall conservation score assigned to the structures found in the simulated trajectories, the KU1 set is too small to make any reliable predictions. This is shown for the Kinwalker results in Figure 24. Possibly, due to the high sequence similarity Transat barely detects any conserved structures within the set of MS2 group representatives (see Fig. 25).

While the structure of the delay inducing hairpin and its influence on the formation of the long range helix is well studied, literature does not explain how the 5'UTR exactly controls the life cycle of the levivirus. Thus, we can only speculate under what circumstances the formation of the delay inducing helix is induced. The Kinefold results hint that pseudoknots play an important role.

2.3.5 SAM riboswitch

The SAM riboswitch is a regulatory motive in 5'UTRs in gram-positive bacteria. It features a cloverleaf structure that is stabilized by a ligand

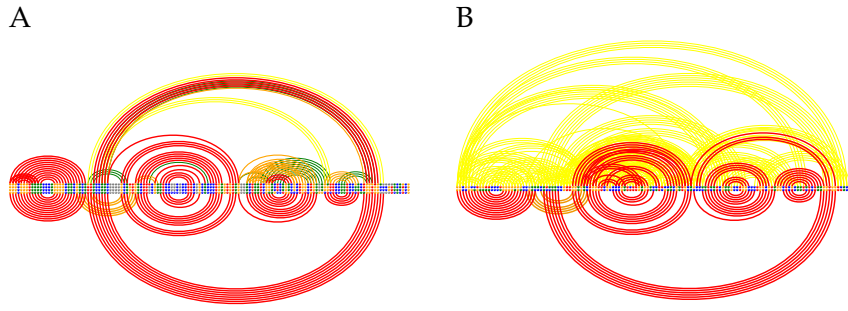


Figure 24: Kinwalker results for the levivirus maturation gene 5'UTR based on representatives of the MS2 group (A) and the KU1 group(B)

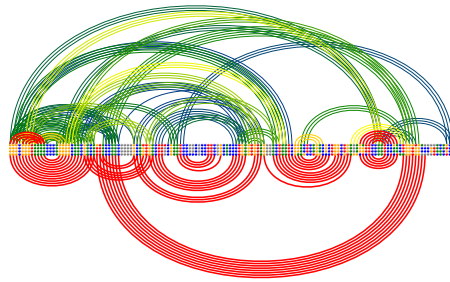


Figure 25: Transat results for the levivirus maturation gene 5'UTR based on representatives of the MS2 group

interaction with SAM. If the cloverleaf structure is stabilized the anti-terminator structure cannot form and transcription terminates.

Analog to the Levi virus example, we clustered the SAM riboswitch sequences and chose cluster representatives for investigating the family. Additionally, we investigated a set of sequences that is closely related to the reference sequence (reference cluster).

As the folding simulation programs cannot simulate the ligand interaction we expect the anti-terminator structure to be formed in the simulated trajectories. However, the Kinefold and Kinwalker results for the cluster representatives do not show significant conservation for the anti-terminator structure (Fig. 27/A and 26/A). Also, the Transat results feature a weak signal for the anti-terminator (Fig. 28). Even when analyzing only the reference cluster we only found an unexpectedly low conservation. Thus, we suspected that the alignment obtained from literature does not support the anti-terminator. We tested hypothesis this by realigning the sequences in the reference cluster, which indeed improved the conservation score of the anti-terminator (see Fig. 29).

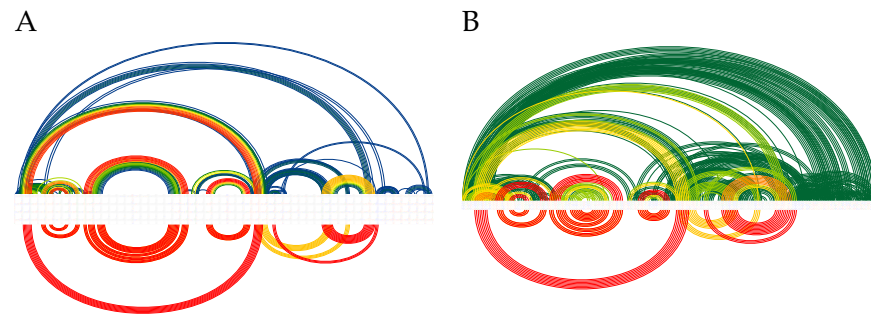


Figure 26: Kinwalker results for SAM riboswitch cluster representatives (A) and the reference cluster(B)

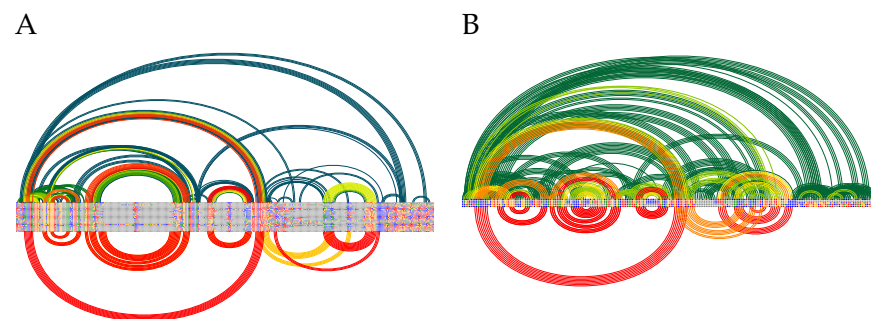


Figure 27: Kinefold results for SAM riboswitch cluster representatives (A) and the realigned reference cluster(B)

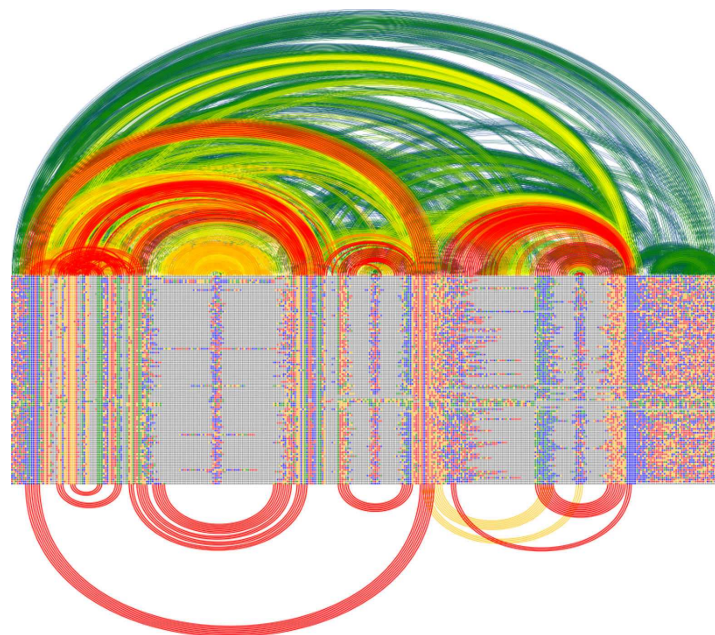


Figure 28: Transat results for the full SAM riboswitch alignment

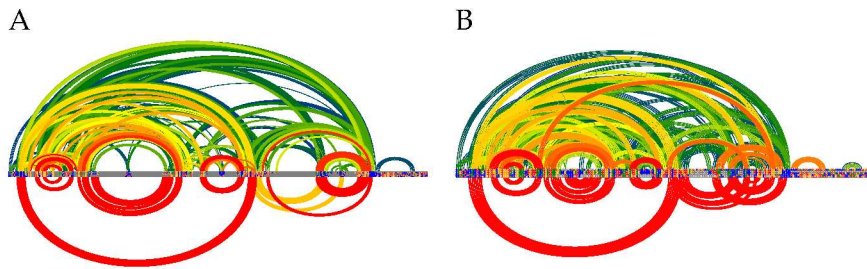


Figure 29: Transat results for the SAM riboswitch reference cluster based on the literature alignment (A) new muscle alignment (B)

2.3.6 HDV ribozyme

The Rfam alignment of the HDV ribozyme contains sequences of both the genome and the antigenome. While the genome and the antigenome feature a somewhat different sequence, according to literature they feature a shared active structure.

When analyzing the Kinwalker results, we mainly find evidence for two hairpins of the inactive structure. This is most likely because Kinwalker cannot model pseudoknots, which are an integral part of the active HDV ribozyme configuration (Fig. 31).

Kinefold should be able to model pseudoknotted structures, which shows in the results: the long-range pseudoknot was found in around half the sequences provided (Fig. 30). Interestingly, the P_1 helix which blocks the cleavage site does not appear to be conserved in the folding trajectories of Kinwalker as well as Kinefold. Thus, the cleavage site might be accessible even when essential parts of the active conformation are not formed. Only Kinefold shows a significant conservation of the P_{-1} helix that prevents the formation of the P_1 helix according to literature.

The Transat based analysis indicates a strong conservation of all helices involved in the pseudoknotted active conformation, while the P_{-1} is not considered conserved 32. In a further step, we investigated the sequences of the antigenome and the sequences of the genome separately using Transat (Fig. 33). In both instances, we could not find significant conservation of the P_{-1} helix. The short-range pseudoknots were significantly conserved within the antigenome set, while the long-range pseudoknot is well-conserved in the genomic sequences.

2.3.7 Small bacterial SRP RNA

The strongly conserved SRP structure is essential to a wide range of species. All three methods we used to detect conserved structures show that most sequences form the known functional structure (see Figures 35, 34 and 36). Wong, Sosnick, and Pan [66] suggested a transient hairpin that guides the formation of this functional structure.

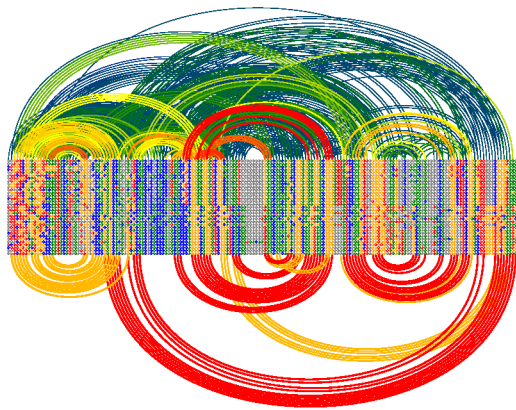


Figure 30: Kinefold results for the full HDV ribozyme Rfam family

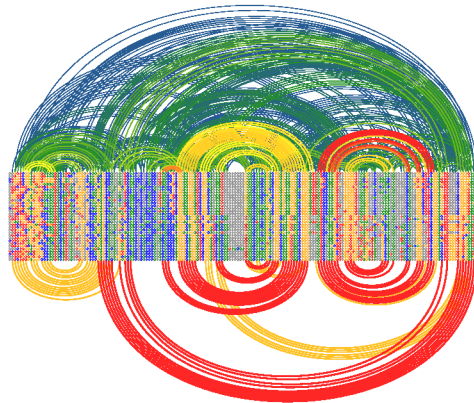


Figure 31: Kinwalker results for the full HDV ribozyme Rfam family

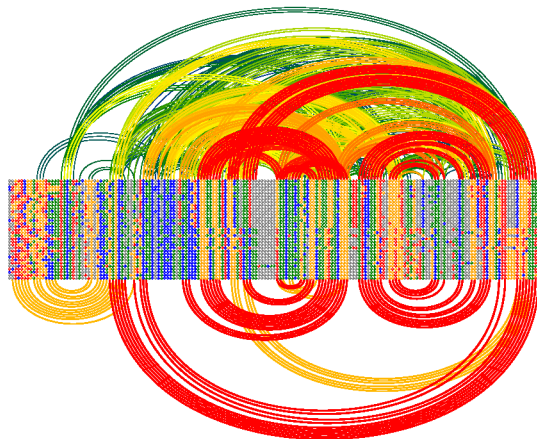


Figure 32: Transat results for the full HDV ribozyme Rfam family

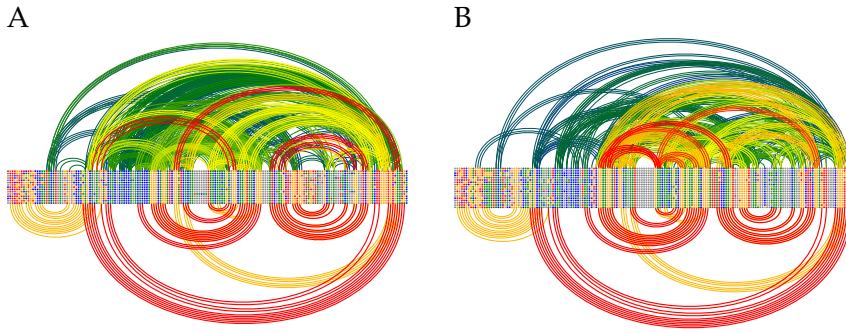


Figure 33: Transat results for the genome (A) and anti-genome(B) version of the HDV ribozyme.

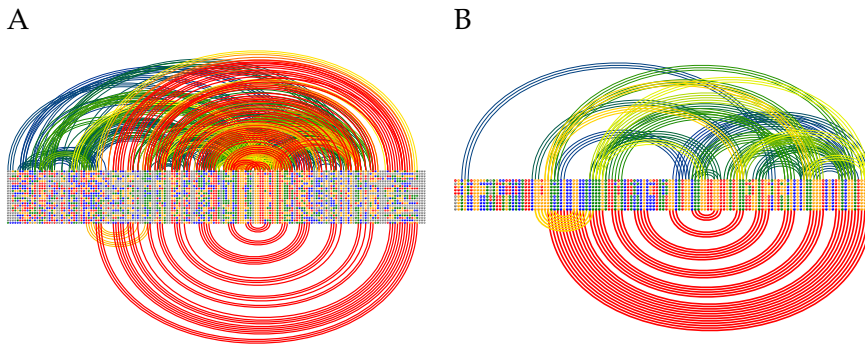


Figure 34: Transat results for SRP RNA cluster representatives (A) and the reference cluster(B)

However, we did not find any indication for the conservation of this hairpin in the set of cluster representatives. The reference cluster analysis shows that the hairpin is formed in some sequences that are closely related to the reference sequence as seen in Figure 35/B and Figure 36/B.

Still a significant fraction of the sequences in the reference cluster do not form the transient structure. They get trapped in a non-functional structure and do not reach the MFE structure. One explanation could be the influence of the Polymerase pausing site at U81 and U83 of the reference sequence that was described in literature. Kinefold and Kinwalker can not simulate pausing sites natively. However, the simulation can be set to end at U83. Running Kinwalker on this subsequence reveals that the transient guiding structure is part of the MFE structure of the subsequence and is instantly formed after the subsequence is fully transcribed. In the next step I started another kinetic folding simulation on the full sequence using the MFE structure of the subsequence as a starting structure. This folding simulation does not get trapped and reaches the MFE structure within reasonable time. Thus we were able to confirm the importance of the pausing site.

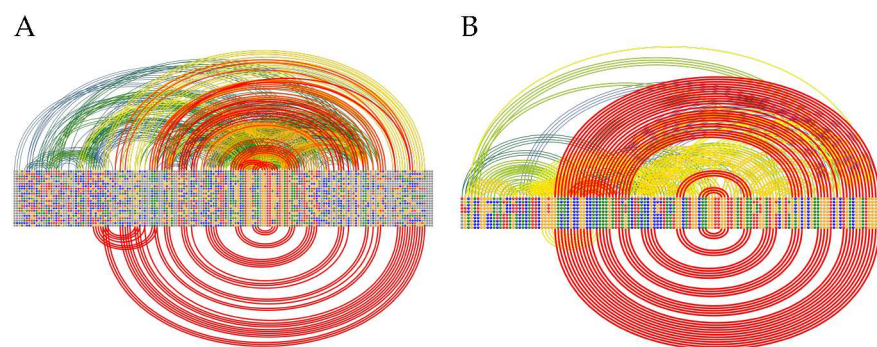


Figure 35: Kinwalker results for SRP RNA cluster representatives (A) and the reference cluster(B)

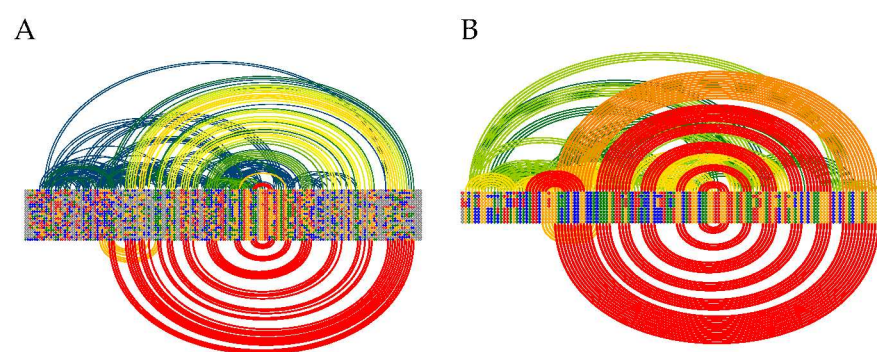


Figure 36: Kinefold results for SRP RNA cluster representatives (A) and the reference cluster(B)

2.4 OUTLOOK

Above work shows that the known experimentally found transient RNA structures can be predicted using kinetic folding programs. In some cases, previously-unknown potential functional structures were predicted using our comparative approach for detecting evolutionary conservation. We plan to publish the kinetic RNA folding dataset on our website so others can use it for benchmarking and further investigations. It contains detailed information on the simulated folding trajectories of all 437 sequences in the dataset as well as experimental data. Thanks to the automated pipeline we developed for building the dataset we can easily keep it updated and extend it in the future. The underlying python library is a valuable tool for automating work with simulated folding programs in general. It will be released as part of the ViennaRNA Python library.

ALIGNING KINETIC FOLDING TRAJECTORIES

3.1 INTRODUCTION

Evolutionary related RNAs often have highly similar sequences and a shared functional structure. In the Rfam database RNAs are grouped in families based on their homology. After investigating the folding process towards the conserved functional structure another question emerged: Do these homologous sequences reach their shared structure through the same folding process?

With the methods outlined in Chapter 2, I was able to identify base pairs that appear in the folding trajectories of a wide range of homologous sequences. These simulations support the assumption that the functional structures can be reached through a co-transcriptional kinetic folding process. Additionally, the method reveals conserved base pairs that are not part of the functional structure. Some of these base pairs correspond to experimentally found transient structures. To determine to which structures the other base pairs, that are not part of a known structure, belong I chose a time-resolved approach.

The focus of this chapter is to investigate the conservation of the folding pathway within related RNAs. How pathways converge and diverge over time can help to identify important functional transient structures. Points of convergence might indicate structures that have a function or are guiding the folding process, for example by helping to avoid folding traps.

A method that can identify such conserved structures or series of structures needs to tackle a range of challenges: Intuitively one would try to compare the structures that appear in two folding trajectories at the same time in the folding process. However, from a biological standpoint that is not expedient. Due to external influences like temperature the speed of the folding process can vary between different organisms. Furthermore, sequence variations between homologs can lead to variations in the points in time at which individual relevant steps appear. These variations do not necessarily change the function of the given structure.

Currently available tools for simulating the kinetic folding process use arbitrary time units (arbitrary time units (a.u.)) that can not be translated to real time. This means the time data is not comparable between different programs and often also between different sequences.

Thus, an alternative method for identifying possibly equivalent structures has to be developed. A sequence alignment of two homologous RNA sequences can be easily obtained from one of the many existing

programs. Based on such a sequence alignment, one could count how many base pairs of two structures in homologous folding trajectories match. This however does not account for the fact that a structure does not need to appear in exactly the same position in the homologous sequences to have the same function. Thus, a structure based alignment is preferable. In fact, an individual structure alignment for different steps in the folding trajectory may be necessary.

The structure alignment returns information on how similar two structures from different folding trajectories are. Based on this, the best matching structure from a homologous trajectory can be identified. Some structures might match well with multiple structures from the homologous trajectory.

While on a microscopic level the RNA folding process is not linear or deterministic, overall the sequence adopts more stable structures over time. Thus, most kinetic folding programs reliably sort the structures by time based on when they appear, even though the time units are arbitrary. Using this ordering of structures we developed a method that aligns steps from two individual trajectories based on their structural similarity, while keeping their ordering and maximizing the overall similarity. Based on this alignment, steps that are well supported by many homologous sequences can then be identified.

3.2 METHODS

The following approach relies on evaluating the similarity of individual steps in the folding trajectory and finding an optimal alignment based on this similarities that keeps the order of the steps. Part of the underlying mathematical problem is similar to the problem that has to be solved to align words (or RNA sequence). Instead of aligning steps structures (steps) in folding trajectories sequence alignment methods align letters (nucleotides). In the following section alignments of words are used to introduce dynamic programming, which the newly developed approach method relies on.

3.2.1 *Dynamic Programming*

Dynamic programming is a method for efficiently solving recursive problems that fulfill the Bellman equation [3]. The solution to a recursive problem is constructed from the solutions to one or more of its subsolutions. For example the n^{th} Fibonacci $F(n)$ number can be computed from following recursion:

$$F(n) = F(n - 1) + F(n - 2)$$

with the starting conditions

$$F(0) = 0 \text{ and } F(1) = 1$$

The solution for the 1002nd Fibonacci number can be obtained by adding the 1001st and 1000th. Those two numbers in turn are computed from their two respectively preceding numbers. If both numbers are computed independently every Fibonacci number below the 1001st is computed multiple times. Dynamic programming speeds up the computation by tabulating the Fibonacci numbers that are computed as subsolutions. Whenever a subsolution is needed more than once it can be read from memory instead of being recomputed. This way, the 1002nd Fibonacci number can be obtained by simply reading the 1001st and 1000th from memory and adding them. Thus the run time complexity is reduced from $O(2^n)$ to linear time ($O(n)$), while $O(n)$ memory is required for tabulating the subsolutions.

The following example introduces the Bellman principle: The shortest path between A and B is P_{AB} . Assume we know that X and Y are points on this shortest path. If path P_{AB} is indeed the shortest path, then the segment of P_{AB} between X and Y also has to be the shortest path between X and Y. If there were any shorter path between X and Y, then the path between A and B would not be the shortest path. In more general terms, all the subsolutions of an optimal solution need to be optimal solutions, to fulfill the Bellman principle. In turn, this means that the optimal solution to a recursive problem can be computed from its optimal subsolutions.

An application of dynamic programming is calculating the edit distance D between two words. The similarity of two words can be described as the number of edits needed to convert the first word into the second word. There are three operations for changing one word into the other: Insertion, deletion and replacement of a letter.

The following operations change the word *sunny* into the word *snowy*: The S remains, the U is deleted, the first N remains, the second N is replaced by an O, next a W is inserted and finally the Y remains. So three changes are necessary to transform *sunny* to *snowy*. Therefore, the edit distance is three. Based on this series of operations a diagram containing the alignment of the two words can be drawn:

```

sunn - y
|  |  |
s - nowy

```

An alignment connects two words by connecting characters that match and those where one was replaced with the other. Usually, it is visualized by writing the individual words in rows, organized in such a way that corresponding letters are placed in the same column. If a letter remains unchanged, both rows in its alignment column contain this letter. If a letter is replaced, then the first row contains the original letter and the second the letter that it is replaced by. If a letter is deleted, its corresponding column features a gap in the second word. An insertion leads to a gap in the corresponding column in the first word. The edit distance is equivalent to the number of gaps plus replacements in the alignment.

	ε	S	U	N	N	Y
ε	0					
S						
N						
O						
W						
Y						

Figure 37: The optimal alignment of two words can be found by extending the optimal alignments of their prefixes. If we want to align the words *sn* and *sun*, then their immediate prefixes (light blue) are *s* and *sn*. Thus, the possible alignment of prefixes that can be extended to the alignment of the full words (darker blue) with one single edit operation are (*s* - (*su*), (*s* - *sun*) and (*sn* - *sun*).

To find the ideal way of converting one word into the other, a dynamic programming approach can be used. The first step is defining the subproblems that need to be solved. The alignment of two words can be computed if the optimal alignments of their prefixes are known. For example the alignment of the prefix *sun* to the prefix *sn* can be computed from the minimum operations that are needed to align the immediately neighbored prefixes plus the score for the last change (Fig. 37).

Therefore, the recursive decomposition of the edit distance problem is:

$$D(i, j) = \min \begin{cases} D(i-1, j) + S_{\text{insert}} \\ D(i, j-1) + S_{\text{delete}} \\ D(i-1, j-1) + \delta(A(i-1), B(j-1)) \end{cases}$$

$$\delta(a, b) = S_{\text{remain}} \text{ if } a = b, S_{\text{replace}} \text{ otherwise}$$

$D(i, j)$ is the edit distance between the length- i prefix of the first word and the length- j prefix of the second word. S is the “cost” score for the individual operations. When computing edit distances S_{insert} , S_{delete} and S_{replace} are set to 1 and S_{remain} is set to 0.

In figure 37 the empty string ϵ is considered. Without the empty set no deletions or insertions in the first position of the string would be possible. This also means that the starting condition is $D(0, 0) = 0$, aligning the empty set on itself has an edit distance of 0.

A simple implementation of this recursive formula would compute a lot of subproblems multiple times. Figure 38 shows a part of the recursion tree. The grey nodes are computed multiple times. Saving the results of the subproblems reduces the run-time of the algorithm to $O(mn)$. In exchange $O(mn)$ memory is needed, where m is the length of the first word, n is the length of the second word.

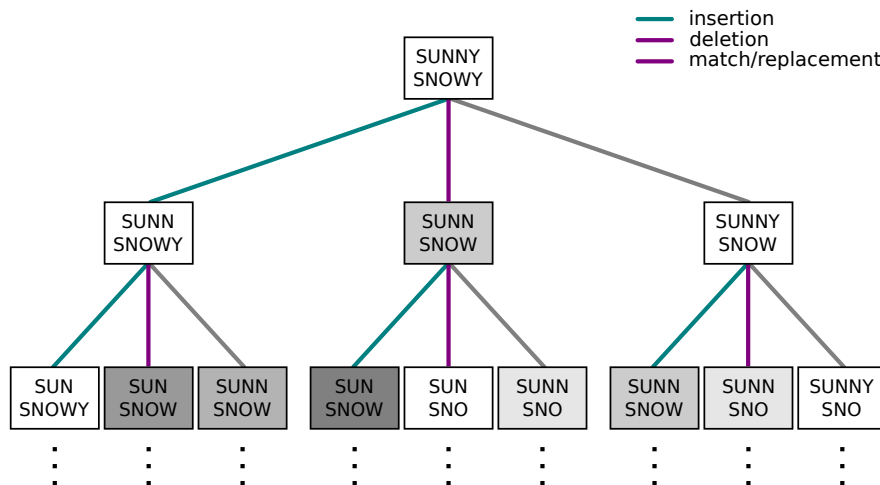


Figure 38: The optimal alignment of two words is computed from the optimal alignments of their prefixes. When recursively checking all possible previous alignments, many alignments are computed multiple times. The beginning of the recursion tree for the alignment of snow to sunny is shown here. Alignments that are computed multiple times are colored grey.

3.2.1.1 Backtracing

This dynamic programming algorithm returns the alignment score or edit distance. However, the score does not tell us which edit operations one has to perform to convert one word into another or which letter in the first word corresponds to which letter in the second word. To retrieve this alignment from the dynamic programming matrix one has to perform a backtrace:

Starting with the cell that contains the overall optimal alignment score one can look up the possible previous steps. In the case of the edit distance the possible previous steps for the cell (i, j) are the cells $(i-1, j)$, $(i-1, j-1)$ and $(i, j-1)$. By adding the cost for an insert, deletion or replacement/extension, respectively, to the score of the possible previous steps one can check which of the previous steps yields the optimal $D(i, j)$. To obtain an optimal alignment one of the previous steps that leads to the optimal score for (i, j) is chosen and the corresponding operation is added to the sequence of necessary edit operations, which also provides the alignment information.

The alignment can be obtained by back-tracing with a complexity of $O(m + n)$, where m and n are the lengths of the two words (see Fig. 39).

	ε	S	U	N	N	Y
ε	0	1	2	3	4	5
S	1	0	1	2	3	4
N	2	1	1	1	2	3
O	3	2	2	2	2	3
w	4	3	3	3	3	3
Y	5	4	4	4	4	3

Figure 39: Backtracing the optimal edit path:

The backtrace starts at the cell that contains the optimal score for the alignment of the full words. In this case this is the cell in the bottom right corner. The optimal alignment of snowy to sunny can be reached from the different previous optimal alignments.

snowy| -
 sunn-|y by inserting one letter
 $4 + 1 = 5$

snow|y
 sunn|y by a match
 $3 + 0 = 3$

snow-|y
 sunny|- by a deletion
 $3 + 1 = 4$

The operation that leads to the lowest total score is added to the list of edit operations. Based on this list, the best overall alignment can be drawn.

3.2.2 Implementation of the Folding Trajectory Alignment

The kinetic folding programs provide a time resolved list of structures that appear during the folding process.

The aim of my newly developed application is to find the best alignment between two or more such lists of structures. The alignment specifies which time step in a trajectory H fits best with which step in a reference trajectory R, in a way that maximizes a score over all steps and preserves the time order of the steps in the respective trajectories. Let $H(i)$ be the i -th step in trajectory H and $R(j)$ be the j -th step in trajectory R. If $H(i)$ and $R(j)$ are aligned, step $H(k)$ can not be aligned to any $R(l)$ with $k < i$ and $l > j$ or $k > i$ and $l < j$.

Aligning steps in folding trajectories is similar to building an alignment of words based on their edit distance. In the case of trajectories, time steps instead of characters have to be aligned. The main difference concerns the scoring: In the previous example the goal was to find the minimum number of edit operations that are needed to align two words. Now the steps should be aligned in a way that finds the best match for every step under the condition that all other steps can find a good match such that the overall dissimilarity over the whole alignment is minimal. On an algorithmic level this means instead of adding +1 for every edit operation, the new algorithm adds a distance score for every matched pair of steps. The better two steps match, the smaller the score gets. The algorithm chooses the match pairs in a way that minimizes the sum of the scores.

This leads to the following recursion:

$$D(i, j) = S(i, j) + \min \begin{cases} D(i-1, j) \\ D(i, j-1) \end{cases}$$

where $S(i, j)$ is the distance score between the i -th step in the first trajectory and the j -th step in the second trajectory. $D(i, j)$ is defined analogous to $D(i, j)$ in the edit distance algorithm. It represents the the overall score aligning the i first steps in H to the j first steps in R.

Using this recursion the back trace algorithm returns the alignment with the minimum overall distance. The minimum overall distance can answer the question, "How similar are the two trajectories?". In this context it is more interesting to find the single time steps that are conserved between homologous sequences. Therefore, the overall dissimilarity score is less relevant than the (dis-)similarity score between the individual steps.

We can compute the similarity of individual steps based on the structural similarity. If we also had reliable information on when which step occurs, we could use it to penalize matching two similar steps if they appear at a highly different time. However, as we do not have reliable and comparable time information we have to ensure that we do not falsely favor or penalize matches based on the number of steps in

the trajectory. To achieve this, we allow a step in on folding trajectory to match multiple steps in another folding trajectory. Including the term $D(i-1, j-1)$ used in the case of the edit distance would lead to a penalty for such multiple matches and therefor is not included in the recursion.

To complete the algorithm, a method for computing the distance between two single steps is necessary. As the main interest in this study is the conservation of structures, it makes sense to use a method that concentrates on the structure for scoring the similarity.

`RNAforester` can perform an alignment between two RNA sequences based on their structure and sequence. It uses a tree representation of RNA structures and performs the structure alignment by aligning the trees[22, 27]. The alignment score that it returns can be used in the trajectory alignment algorithm as $S(i, j)$.

`RNAforester` was run using the affine gap scoring model with gap opening cost set to 2 and gap extension cost set to 1 in the distance mode. The score scales approximately linearly with the sequence length. To correct for that, the score is normalized by the length of the aligned sequences. This ensures that steps in the end of the trajectories do not have a disproportionally high influence on the final alignment.

The available datasets usually include more than two homologous RNA sequences. It would be preferable to align every trajectory to every other trajectory. However, for larger datasets this is computationally too expensive.

The solution is to align all the homologs to one reference trajectory. As a reference I chose RNA sequences for which transient structures have been identified in experiments.

First every homologous trajectory is aligned to the reference. For every step the best match from every homologous trajectory is chosen. `RNAforester` returns a sequence structure alignment for every homologous step to the reference step. Using these alignments we can then build an alignment of all the matching steps with the reference step.

To identify conserved steps the length normalized distance score from `RNAforester` was used.

To test the capabilities of this trajectory program it was applied on the RNA families introduced in the Dataset Chapter 2. The folding trajectories were computed using `Kinwalker` with the same parameters as in the dataset chapter. To illustrate the capabilities of the programs two examples are shown in the results section: the TRR operon leader and the SAM riboswitch. The folding trajectories were obtained from `Kinwalker` using the optimized parameter set defined in the Dataset Chapter 2.

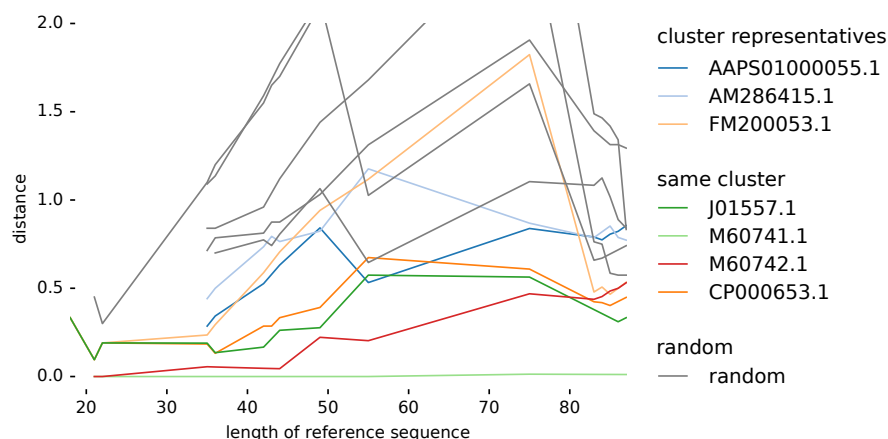


Figure 40: Trajectory alignment of the TRP operon leader family: AE005174.2/2263095-2263188 was chosen as a reference. The y-axis represents the distance of the best matched step of the homologous trajectory to the time step of the reference trajectory. The x-axis is the length of the reference at a given step. Through transcription speed it can be related to time units. A detailed interpretation can be found in the text.

3.3 EXAMPLES FOR TRAJECTORY ALIGNMENTS

3.3.1 *TRP trajectories*

In Figure 41 the individual steps of the folding trajectory of the reference sequence (AE005174.2/2263095-2263188) are depicted. Each step is labeled with the length of the reference sequence. The upper most sequence in the alignment is the reference sequence, the lower arcs represent the structure of the reference sequence at this step, the upper arcs how many of the aligned homologous steps form a given base pair. The color scheme is identical to the one described in the Dataset Chapter (see Fig. 15). One can identify four points in the trajectory that correspond to a significant change of the structure of the reference sequence. The first change is the replacement of a first short helix by the seed of the pre-terminator hairpin pre-T that is located slightly downstream. The pre-terminator further extends until the seed helix for the anti-terminator A is formed. In the next step a long range interaction L between the 5' end of the sequence and the upstream strand of the terminator helix T (=downstream strand of the anti-terminator) is formed. Finally, the last major change introduces the seed of the terminator hairpin T, that is extended further in the following steps.

This trajectory captures the functional mechanism described in literature well. In the absence of ribosome the pre-terminator forms and blocks the formation of large portions of the anti-terminator. This then leads to the formation of the terminator. Only the existence and purpose of the long range interaction introduced in pre-T has not been described previously.

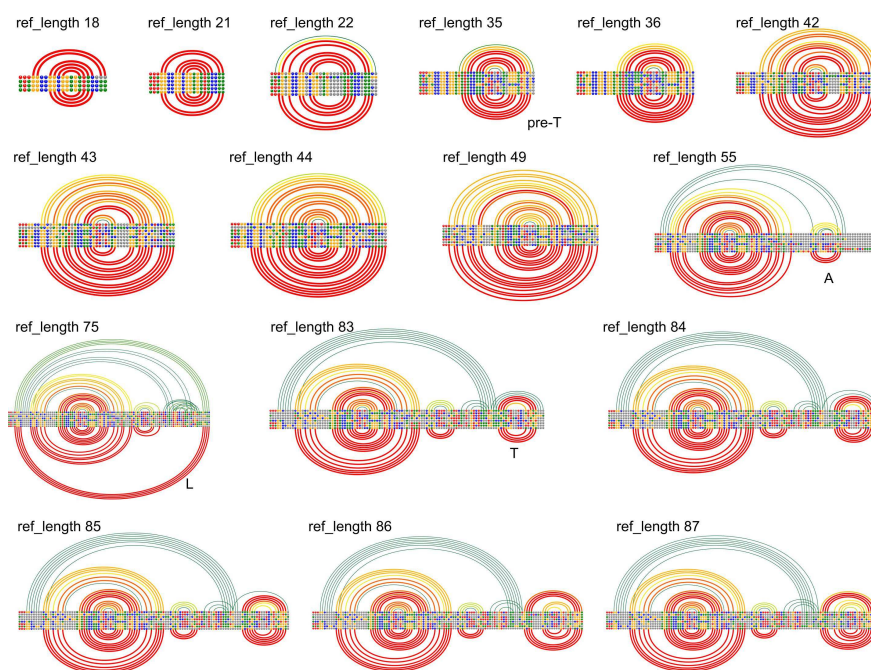


Figure 41: Trp operon leader folding trajectories. For details see text.

A trajectory alignment contains three different sequence sets: random sequences, sequences that represent the clusters defined in the dataset section and sequences that are closely related to the reference sequence and belong to the same cluster. Even though this alignment does not show as clear convergence points as other examples that I investigated, we can identify some interesting points:

The first one is at a reference length of 35 nucleotides when the pre-terminator helix is formed: Some homologs like J01557.1 and CP000653.1 form the pre-terminator pre-T in good agreement with the reference. Others like AM286415.1 feature an alternative helix in a similar area, even though their sequence features complementary base pairs in the pairing positions of the pre-terminator. The more the structure diverges from the reference pre-terminator, the higher the distance score plotted in figure 40.

The second point of interest is when the reference forms the seed for the anti-terminator A at a reference length of 55 nucleotides. The distance score of the sequences that form the same anti-terminator seed as the reference decreases at this point. Some sequences like AM286415.1 form a significantly longer anti-terminator seed, leading to a higher distance score. J01557.1 and CP000653.1 form an alternative helix that does not correspond to pairing positions of the anti-terminator seed.

The long range interaction L that is found in the trajectory of the reference sequence is only found in the trajectory of one other homolog, the FM200053.1. However, the step in the FM200053.1 trajectory that contains the long range interaction also already features the seed of the terminator hairpin. This leads to an overall different structure pattern that prevents the algorithm to align those two steps correctly. This mismatch leads to the exceptional high score of the FM200053.1 trajectory at this point. The other trajectories simply miss this interaction but do not feature an alternative structure, leading to only slightly increased distance scores.

At full length the reference sequence features the pre-terminator, the terminator and the seed of the anti-terminator. While all other homologs form a highly similar structure, AAPSo1000055.1 and AM286415.1 feature an extended version of the anti-terminator seed and do not form the terminator within their life time.

In summary, the alignment indicates that there are variations in the folding pathways of homologous TRP operon leader sequences. Nevertheless, most sequences form some version of the pre-terminator and terminator hairpin at some point in their trajectory.

Interestingly, the fully extended anti-terminator structure as well as the fully extended terminator structure have a similar free energy of about -33 kcal/mol. This means that in a thermodynamic picture they have a similar probability to appear. The decision which of the two is formed could be guided by the co-transcriptional folding. However, the variations in the folding trajectories of homologous

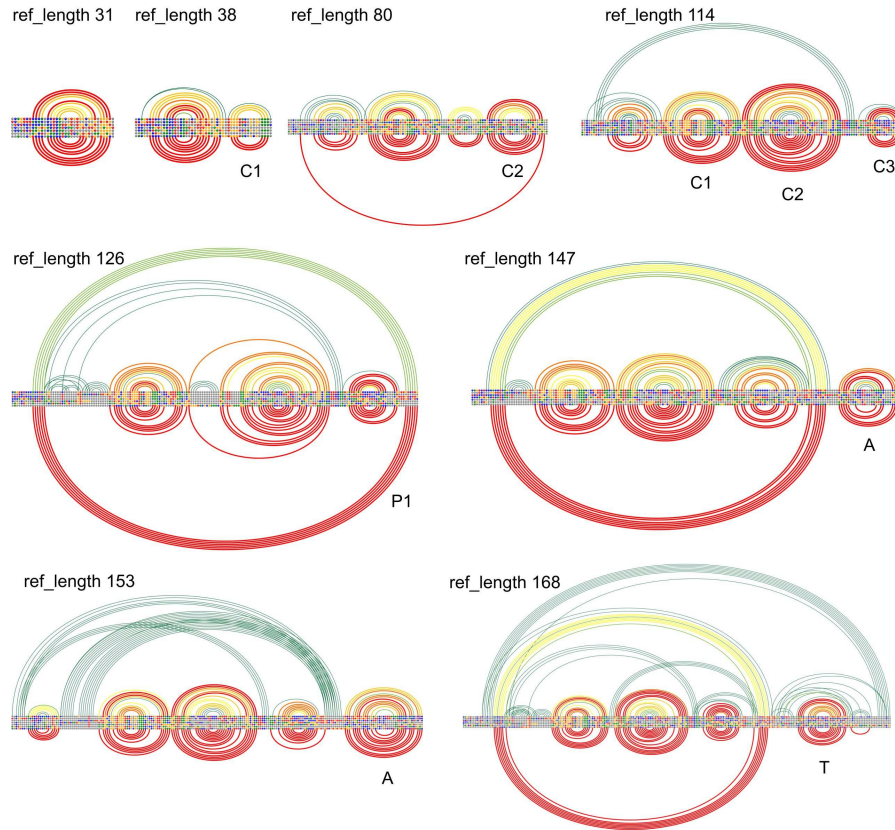


Figure 42: Important steps in the SAM riboswitch folding trajectories. For details see text.

sequences could also indicate that the direction is not controlled by the co-transcriptional folding process itself, but by interactions with biomolecules like ribosomes and polymerase.

3.3.2 SAM

Important steps of the trajectory of the SAM riboswitch reference sequence are depicted in Figure 42. It forms the first enclosed helix of the cloverleaf structure at a length of 38 nucleotides (C₁), the second helix of the cloverleaf (C₂) at a length of 80 nucleotides and the third (C₃) at a length of 114 nucleotides. At a length of 126 nucleotides it temporarily forms the P₁ helix. As soon as the anti-terminator sequence is fully transcribed it refolds into the anti-terminator A (reference length 147 nucleotides). At full length it then starts to refold into the terminator structure T.

In the summary of the trajectory alignment (see Fig. 43) we can see points of convergence at 38, 80 and 114 nucleotides. These steps in the trajectory correspond to the formation of the three enclosed helices of the cloverleaf structure. At a length of 126 nucleotides the reference sequence closes the multi loop that serves as a binding site for SAM. One can distinguish two different groups at this point: Two sequences (Ca05 and Cto1) too form the P₁ helix, all other sequences

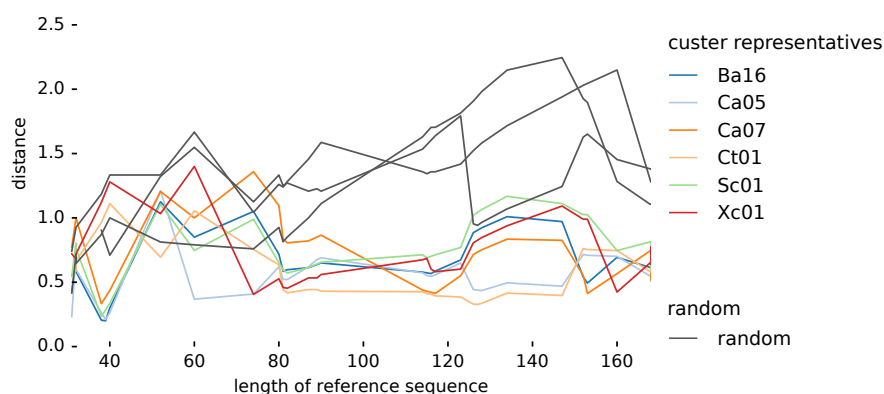


Figure 43: Trajectory alignment of the SAM riboswitch leader family: switch. AL009126.3/1258276-1258464 was chosen as a reference. The y-axis represents the distance of the best matched step of the homologous trajectory to the time step of the reference trajectory. The x-axis is the length of the reference at a given step. Through transcription speed it can be related to time units. A detailed interpretation can be found in the text.

directly form the anti-terminator helix, thus have a larger distance to the reference. At the folding steps of the trajectory that feature the anti-terminator (147 nucleotides) these trajectories have a smaller distance to the reference again. After the reference sequence is fully transcribed it starts refolding into the terminator. All homologous sequences reach this terminator structure after some delay.

Even though we can not translate Kinwalker time (a.u. - arbitrary time units) to actual time units (s - seconds), they provide some information on the delay the anti-terminator structure induces. In total the reference sequence needs about 3.5 a.u. to fully transcribe and fold into the final terminator structure. 1.5 a.u. out of these 3.5 a.u. are the refolding time from the anti-terminator into the terminator. This provides time for the polymerase to pass the poly-U strand before the terminator is formed.

Compared to the TRP operon leader, the SAM riboswitch family alignment shows comparably small and consistent distances. This suggests that the folding trajectory of SAM is strongly conserved. Furthermore, the switch between the terminator and anti-terminator seems to be mainly controlled by the stability of the P₁ helix and not by other external factors. All sequences form the anti-terminator before forming the terminator hairpin. This is what we expect to happen in the absence of SAM.

Longer sequences have a significantly higher number of possible structures they could form. Thus, it is a lot less likely for longer sequences to have similar folding trajectories than for shorter sequences. While the random sequences that were included in the trajectory alignments are not enough to provide a reasonable background, they support this

point. In general this makes it difficult to get reliable results for short sequences.

3.4 DISCUSSION AND OUTLOOK

The implemented approach provides a fast way of detecting important co-transcriptional structures based on their evolutionary conservation. It can easily be adapted to the output of various kinetic folding programs. It is already well capable of representing the similarities and differences in folding trajectories. Nevertheless, we are working on further extending its capabilities:

Preliminary tests show that a significant speed improvement of the alignment method can be achieved by merging similar steps in the trajectory, without negatively impacting the detection of conserved structures. This speedup even enables the creation of all-against-all trajectory alignments for fairly large datasets in reasonable runtime. The advantage of using all-against-all alignments is that they can not only score the conservation of structures present in one reference trajectory, but the conservation of all structures that appear in any of the homologous trajectories.

The current approach might miss some conserved transient structures because it relies on deterministic trajectories. Most importantly, when two possible folding paths are possible, such approaches are only able to recover one of them. An extension of the program is needed to deal with folding pathways of stochastic programs. Usually one has to look at multiple trajectories to identify important structures. The implementation that I am currently working on summarizes all base pairs that are found at a specific sequence length and weighs them based on the number of trajectories they appear in. Next, steps that are similar are merged to reduce the number of steps that have to be aligned. The individual steps can not be described by a single structure anymore. Thus *RNAforester* can not be used to assign a similarity score to steps in the trajectory of different sequences. However, the structure information can be saved to a the dot-plot format used in the *ViennaRNA* package [31]. The sequence-structure alignment program *CARNA* [36, 51] can align two steps based on these dot-plots and assign a score. This score can then be used in the program that is described in this chapter.

Another improvement can be achieved by using a similarity score instead of a distance score. A similarity score would allow the implementation of a local alignment algorithm that is better suited to identify parts of the folding trajectories that are conserved, even when other parts are not conserved at all.

Both suggested extensions to the described method could significantly improve the analysis of families like the TRP operon leader. Previous tests on the influence of different *Kinwalker* settings suggest that the outcome of the simulation is comparably sensitive to influences

like transcription speed and temperature. This suggests that there exist multiple possible folding paths that can not be recovered by Kinwalker. Using the extension that is currently in development can solve this problem.

The interpretation of the alignment graphs heavily relies on having specific information on the trajectories. While we were able to provide details on reference trajectories and a summary of the best matching steps of the homologous trajectories, also having the detailed information for all homologs would be interesting. Fortunately, this data is already included in the dataset presented in the previous chapter and will be published on the TBI website (<https://www.tbi.univie.ac.at>).

BACKGROUND MODEL

4.1 INTRODUCTION

In the last few chapters I identified potentially evolutionary conserved helices. To assess the significance of these predictions we need a background model to compare to.

There are two main questions we want to answer: (1) How likely is a certain structure to be found in a random sequence that is not under evolutionary pressure to form this structure? (2) How significant is a structural motive that is found in two or more homologous sequences?

Assuming an uniform frequency of nucleotide types and considering the set of possible base pairs $\mathcal{B} = \{AU, UA, CG, GC, GU, UG\}$ defined in Section 1.2.1, it is easy to compute the probability that a random sequence can form a given structure. The only restriction on the sequence is introduced by the pairing nucleotides. Thus, the probability of the structure is only dependent on the number of base pairs h . To be able to pair the two involved bases have to be complementary. If the opening base is an A or a C the closing base can only be one of the four nucleotide types. The base U and G each have two possible pairing partners. Therefore, the probability for a single fixed base pair in a random sequence is $\frac{1}{2} \times \frac{1}{4} + \frac{1}{2} \times \frac{1}{2} = \frac{3}{8}$. Based on this the probability p of any structure s consisting of h base pairs can be computed like this:

$$p(h) = \left(\frac{3}{8}\right)^h \quad (1)$$

This can also be extended to random sequences with non-uniform nucleotide composition:

$$\begin{aligned} p(h) = & (p(G) \times p(C) + p(G) \times p(U) + \\ & p(A) \times p(U) + \\ & p(U) \times p(G) + p(U) \times p(A) + \\ & p(C) \times p(G))^h \end{aligned} \quad (2)$$

where $p(A)$ is the fraction of adenines in the sequence, $p(U)$ is the fraction of uracils, $p(C)$ the fraction of cytosine and $p(G)$ the fraction of guanines.

According to Equation 1, the probability of a structure in a random sequence exponentially decays with the number of base pairs h . For example, the probability of a helix with three base pairs is $p(h = 3) =$

0.05 and the probability of a helix with eight base pairs is $p(h = 8) = 0.0004$. This means, that finding a sequence pattern that allows for the formation of a helix with $h = 8$ base pairs is about 100 times as significant as finding a pattern for a helix with $h = 3$ base pairs. However, even a short helix can be relevant if it is also supported by thermodynamics.

Regarding the second question: Finding the same structure within multiple homologous sequences indicates that it is evolutionary conserved and essential for the function of the RNA. To judge this we usually start from a multiple sequence alignment. These alignments will usually not be perfect. We therefore want to allow for shifts between structural elements in different sequences. Rather than dealing with exact structures, we will therefore use structure motives such as a simple helix of a given length with a variable hairpin loop.

In Chapter 3 two structures of homologous sequences are aligned. While the alignment score provides some information on how similar the structures are, knowing how likely their shared structure motive is to appear in two random sequences, can be used to assess the significance of the alignment score given the structure motive.

While finding an analytic expression for exact structures is easy, allowing shifted sequences makes that problem a lot more complex. As illustrated in Figure 44, the primary problem is that in a straightforward approach some sequences are counted multiple times. The next best approach we tried was to generate randomized sequences, checking them for the structure forming pattern and compute the percentage of sequences in the randomized set that form such a pattern. The issue with this approach is that the probability of a random sequence featuring a helix forming pattern is very small. For example based on Equation 1, only one in about 20.000 random sequences can form a structure with at least ten base pairs. If we want to derive the likelihood of an observation with a very small probability we need a huge amount of samples (e.g. 10^7). Particularly, when using more elaborate methods for generating randomized sequences this is computationally too costly (see Section 4.2). Moreover, scanning a sequence for a structure forming pattern is a lot more expensive than checking if it can form one specific structure.

As stochastic solutions are often not feasible we want to develop an analytic approach using combinatorial methods that can not only compute the likelihood of exact structures but also structure patterns over multiple sequences. Having such a model would help us assess whether or not a found helix can be explained without evolutionary pressure and how significant structures that are found in multiple homologs are.

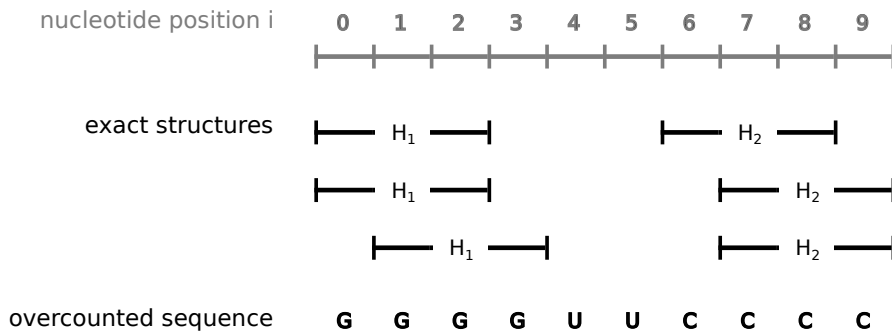


Figure 44: Over-counting the probability of structure motives:

The probability with which a random sequence can form a given exact structure can be easily computed using Equation 1. Each structure motive defines a set of exact structures that are compatible with its pattern. If one wants to compute the probability with which a random sequence features a given structure motive, one could start by summing up all the probabilities of the exact structures defined by the motive. However this leads to an overestimation:

The probability with which a random sequence with length n can form a given exact structure is computed by counting how many sequences out of all possible sequences of length n can form the structure. To obtain the probability of a random sequence being able to form at least one out of multiple exact structures one has to consider that one sequence may be able to form more than one of the exact structures. When summing the probabilities of the exact structures, any such sequence would be counted multiple times.

In this figure we investigate a structure motive that describes a helix of at least three consecutive base pairs within a sequence of length $n = 10$. Any matching sequence has to feature two complementary strands H_1 and H_2 that are separated by at least three nucleotides (= minimum hairpin length). Above figure lists all three exact structures that match this pattern. Each of the three exact structures consists of three base pairs. Thus, according to Equation 1 they have a probability of $p = 0.053$. Adding up the probabilities of these three exact structures yields an overall probability for the structure motive of $p = 0.16$. However, the set of sequences of length $n = 10$ contains sequences that can form multiple exact structures. The example “GGGGUCCCC”, which is shown in the bottom part of the figure, can form all three structures and it is counted into the probability of all three. When computing the probability of a random sequence featuring a given structure pattern, we need to count how many sequences feature at least one of the exact structures and must avoid counting sequences multiple times.

For the above example the correct probability of the structure pattern can be obtained by enumerating all sequences and checking them for all exact structures. This yields the correct probability $p = 0.14$ (instead of $p = 0.16$ when over-counting). The error introduced by over-counting increases with the number of exact structures that fulfill the structure pattern. Thus, it becomes even more relevant in longer examples.

This enumerative solution is not possible for longer sequences and less restricted structure patterns. To solve this problem, one has to derive analytic correction terms for over counting, which is non-trivial.



Figure 45: Helix forming pattern:

A sequence has to feature a subsequence H_1 of length h followed by a complementary subsequence H_2 of same length in a minimum distance of three nucleotides, to be able to form a helix consisting of at least h consecutive base pairs.

4.1.1 Investigated structure pattern

In the trajectory alignment section one major challenge is to identify equivalent structures in trajectories of homologous sequences, while not falsely matching structures that appear at random in another trajectory.

To perform this discrimination, it would be helpful to know how much a structure can be varied before a significant percentage of randomized sequences also matched. Combined with information on the phylogenetic distance, this can serve as a base for defining a cutoff threshold.

The building blocks of any RNA secondary structure are helices. Thus, they are the first type of structure motives that we want to be able to score. Variations on simple helical structures could be in length and position. To investigate the influence of allowing shifts, we can define the structural pattern of a helix with a minimum number of base pairs h . A sequence that can form a helix of length h has to feature two complementary strands of length h that are separated by at least three nucleotides (see Fig. 45).

One can test how many random sequences of a given length n can form a helix that conforms to this pattern. Increasing the length of the sequences within those the pattern is searched for allows the helix to shift over more positions.

The further helices are allowed to be shifted, while still being matched to the reference helix, the more helices from random sequences are matched.

Thus, the significance of a matched structure decreases with the number of random sequences that form the helix pattern.

4.2 STOCHASTIC MODELS

We did a small stochastic study on the different probabilities of exact structures and more flexible structure motives. This section highlights the influence of shifts on the significance of structures found in homologous sequences. Furthermore, it provides some insight in the feasibility of stochastic approaches on the problem.

4.2.1 Methods

As before, we want to know how likely the occurrence of a specific structure or structure motive is in a randomized sequence. Using this information we can then assign a significance score to predicted structures. This method relies on generating randomized sequences and then testing if they can form a structure pattern.

Three different methods for generating randomized sequences were used:

- One set of random sequences were generated using the Python 3.6 module `random` that relies on the Mersenne twister pseudo random number generator [32]. Specifically, the implementation is built on the `choice` function [47].
- In RNAs not only the nucleotide composition, but also the dinucleotide composition are important descriptors because stacking energies between consecutive bases are crucial energy contributions to the stability of an RNA structure [1]. To reproduce the doublet and triplet composition of the investigated sequence a k-let shuffling method based on `ushuffle` was implemented [26].
- Sometimes we need to randomize an alignment rather than a single sequence. This can be accomplished with the `rnazRandomizeAln.pl` script from the RNAz package that shuffles alignment columns while it tries to maintain local nucleotide conservation patterns [48, 58].

The used randomization methods are specific for the investigated sequence. An important helix that we discussed in the previous chapters is the putative transient helix in the 5'-UTR of the bacterial small SRP RNA. In this section it is used as a test sequence and structure to illustrate how the background model can be used to interpret prediction results.

From each of the methods a randomize sequence sets was generated: 10^6 sequences with Python 3.6 `random` (short: **random**), 10^6 sequences by mononucleotide shuffling of the test sequence (short: **1-let**), 10^6 sequences by dinucleotide shuffling (short: **2-let**), 10^6 sequences by trinucleotide shuffling (short: **3-let**) and 5000 sequences by the alignment shuffling script of RNAz (short: **RNAz**).

The RNAz alignment shuffling script is run on an alignment of 5 representative SRP RNAs that includes the test sequence. After shuffling alignment columns with a similar conservation score, the shuffled test sequence is extracted. This is repeated 5000 times to obtain the set of shuffled sequences.

In a first step, I tested how probable the exact structure of the proposed transient SRP helix is in the different randomized sets.

In the second step, the probabilities of finding at least one helix forming sequence pattern somewhere in a randomized sequences of

length n were computed. For this purpose all $H_1 - H_2$ pairs of length h were pre-generated. Then, for every infix H_1 of the randomized sequence, the randomized sequence was screened for occurrences of the pre-generated complementary H_2 sequences in a minimum distance of $m = 3$ base pairs. The probability of finding at least one helix with at least h base pairs within a window of size n in a random sequence can be computed by counting the sequences that feature at least one such H_1 - H_2 pair. This was done for helix lengths $h = 4$ and $h = 7$ on the sequence sets **random**, **2-let** and **RNAz**. More detailed analysis on the relation of helix length h and sequence window n were performed for the **random** sequence set.

4.2.2 Results and Discussion

Here we investigate the relationship between the length of the investigated helix h , the randomization method and the size n of the sequence window within which the helix occurs.

Figure 46 shows how different randomization methods lead to different p-values. While the Python 3.6 **random** method and the k-let shuffling method from **ushuffle** are fast enough for generating a significant sample set, more complex randomization methods like the alignment shuffling algorithm from **RNAz** have a longer run time, leading to a smaller sample set. In this example 1 million samples from the **random** sequence generator and the k-let shuffling method were used, while only 5000 sequences were obtained from the alignment shuffling algorithm. Given the overall low percentage of sequences that can form the investigated structure this set is too small to calculate a reliable p-value. Only one single sequence out of the 5000 generated sequences was able to form the structure of interest. As computing p-values for such models does not seem feasible from sampling methods, this is one of the main motivations for developing analytic methods that can compute p-value for more complex background models. The size of the random sequence set obtained from the Mersenne twister pseudo random number generator is large enough to approximate the probability $p(h = 7) = 0.001$, computed from Equation 1.

In Chapter 2 we investigated multiple SRP RNA homologs. Most of them did not form the full transient helix ($h = 7$) that was proposed, but only the four base pairs within the same area. A simplified way of examining the significance of this finding, is to compute the probability of finding a helix with 7 base pairs in the experimentally found range of 24 nucleotides within a random sequence and to compare it to finding a helix of 4 base pairs within the same range in a random sequence.

The minimum span of a 7 base pair helix pairs is 17 nucleotides. The likelihood of finding a 7 nucleotide long sequence window is about 4%. Thus, finding the same helix in two unrelated random sequences of length $n = 24\text{nt}$ is about 0.1%. For comparison, finding a helix of

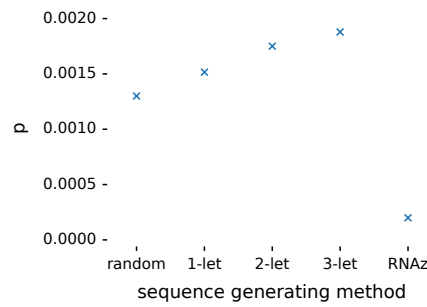


Figure 46: Likelihood of putative transient SRP RNA helix in randomized sequences - comparison of sequence generation methods:

The percentage of randomized sequences that can form the proposed transient helix (helix length $h = 7$) is computed and the p-value of this helix given the different random models is derived.

The randomized sequence sets are described in Section 4.2.1. The **random** set is generated based on a uniform nucleotide probability (10^6 sequences). **1-let**, **2-let** and **3-let** originate from a k-let shuffling algorithm (10^6 sequences). The **RNAz** set is obtained from an alignment shuffling algorithm (5000 sequences).

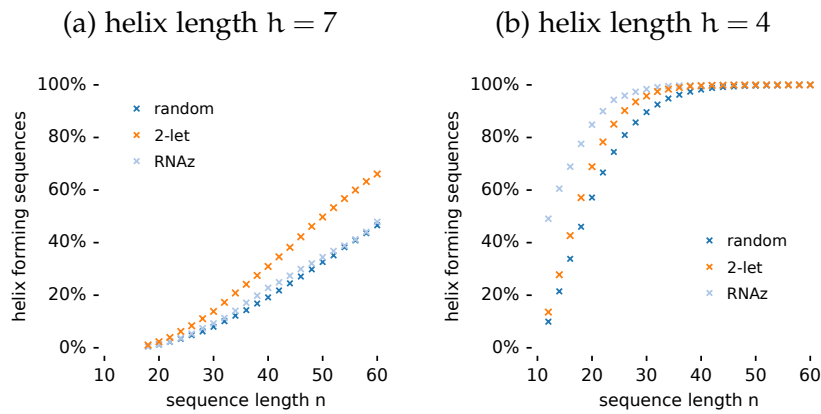


Figure 47: Likelihood of finding a helix of length $h = 4$ and $h = 7$ in a set of randomized sequences of length n : Three of the previously described randomized sets (**random**, **2-let**, **RNAz**) were searched for sequence pattern that allow the formation of helices with $h = 4$ and $h = 7$ base pairs. Subsequences of length $n = 12$ to $n = 59$ were chosen around the area of the putative transient SRP helix from each random sequence in the set and screened for helix patterns. The percentage of sequences of length n that can form one or more such helices is plotted on the y-axis.

	h = 4	h = 7
P(n = 24)	0.745	0.034
P(n = 12)	0.099	-

Table 3: Probability of a helix with h base pairs in a sequence with n base pairs

length $h = 4$ in a single sequence of length $n = 24$ is about 74% and finding it in two unrelated helices is about 55%. Thus, identifying a 7nt helix in two sequences within a 24nt window indicates that the helix might be functionally important and that the sequences are evolutionary related. Identifying a helix of length 4nt within the same window is likely to happen at random, at least if thermodynamics are neglected. For better overview the mentioned probabilities are summarized in Table 3.

Apart from highlighting the influence of the helix length on the significance of a prediction, Figure 47 shows, that sequence from the **random** set are less likely to form helices than randomized sequences. This might be due to the fact, that the **random** set includes sequences that have a nucleotide composition, that does not support structure formation as well as the nucleotide composition of the test sequence.

The probabilities of all computed combinations of sequence window sizes and helix lengths are summarized in Figure 48. Based on this data the maximum sequence window size n within which a found helix is significant can be derived (see Figure 49).

In this stochastic study all helix forming patterns that feature complementary nucleotides in the pairing positions are considered. In a preliminary study we restricts the set of helix patterns to thermodynamically stable helices. When taking thermodynamics into account we found a highly similar influence of the allowed shift on the probability of finding a given structure in a random sequence. It confirmed that particularly the significance of detecting short helices is low when shifts in their positions are allowed.

4.2.3 Summary

In summary this study shows that the significance of a found structure sharply decreases with the positional flexibility granted by the model. As this flexibility is for example necessary when identifying equivalent structures in homologous RNA sequences, it has to be accounted for in the background model.

Even when using the simplified model used above it is not possible to enumerate all possible sequences and structures and even stochastic methods are limited when rare events are investigated. To escape this limitations we developed a combinatorial background model, which we introduce in the following section.

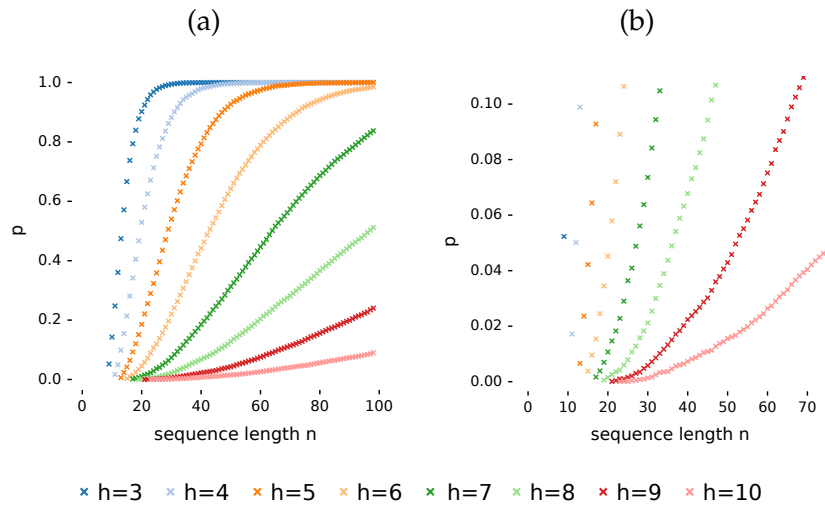


Figure 48: Relationship between helix length, sequence length and likelihood of helix forming pattern: Based on the **random** set, sequences windows of size $n = 9_{nt}$ to $n = 100_{nt}$ were screened for helix patterns with $h = 3_{bp}$ to $h = 10_{bp}$. The y-axis depicts the fraction of sequences that feature at least one possible helix of length h . Subfigure (b) is a detail of Subfigure (a) that depicts helix length and sequence window combinations that are unlikely in random sequences.

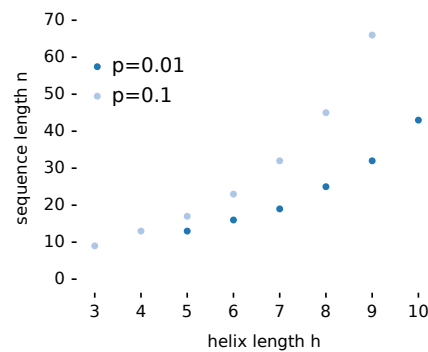


Figure 49: Based on Figure 48, we can identify combinations of helix lengths h and window sizes n that are unlikely to be found in randomized sequences. Any helix that is longer than h found in a sequence window smaller than n , has only a 1% ($p=0.01$) or 10% ($p=0.1$) probability to occur at random.

4.3 MATHEMATICAL BACKGROUND OF THE ANALYTIC COMBINATORICS MODEL

Any sentence in a natural language can be decomposed based on the underlying grammar. For example, the sentence “The sun shines.” is a grammatically correct sentence because it follows the composition D.N.V, where D is a determiner, N is a noun and V is a verb.

Similarly we can try to describe an RNA sequence that can form at least one helix: The sequence $S = AAAGGGAAAAACCCAAA$ can form a helix because it follows the pattern

$$R_1 H_1 \star^3 R_2 H_2 B \quad (3)$$

where $R_1 = AAA$ is a leading sequence,
 followed by the helix opening sequence $H_1 = GGG$,
 followed by the first three spacer nucleotides $\star^3 = AAA$
 and some additional spacer nucleotides $R_2 = AA$,
 followed by the complementary helix closing sequence $H_2 = CCC$,
 followed by some tailing nucleotides B .

The following section will introduce formal grammars and how they can be used to count sequences that form such a pattern.

4.3.1 Formal Grammars

One can define a set of symbols Σ , for example $\Sigma = \{a, b, c\}$ or in the context of RNA sequences $\Sigma = \{A, U, G, C\}$, called alphabet. Applying the Kleen star operator on the symbol set returns the set of words that can be build from the alphabet by concatenating symbols from the alphabet. This set of words also includes the empty word ϵ . For example, if $\Sigma = \{a, b\}$, then $\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$.

Formal grammars are mathematical models that can be used to describe and investigate subsets of these words that follow specific rules. The subset of words defined by a grammar G is called its corresponding language L_G , with $L_G \leq \Sigma^*$.

The derivation of every word in a formal language starts from a start symbol S that is transformed into a string of other symbols following predefined rules. This string can than be further rewritten according to the set of rules. Two subsets of symbols are used: terminal symbols Σ that can not be replaced by any other string, and the set of non-terminal symbols N that can be further expanded. The rules for such replacements/expansions are written as tuples (α, β) , or $\alpha \rightarrow \beta$, where the symbol or string α is replaced by the symbol or string β . If there are multiple rules that replace the string α , for example $\alpha \rightarrow \beta_1$, $\alpha \rightarrow \beta_2$ and $\alpha \rightarrow \beta_3$, it can also be written in the form $\alpha \rightarrow \beta_1 | \beta_2 | \beta_3$, where $|$ indicates a choice. Often more than one rule can be applied to the same string and one rule might be applicable to

multiple sites(infixes) of the string. In these cases the order in which the rules are applied is arbitrary. Thus, dependent on the rule application order different words, also called derivations, are obtained.

4.3.1.1 Definition of Formal Grammars

A formal grammar is defined as a tuple $G = (N, \Sigma, P, S)$, where

- Σ is a finite set of terminal symbols that is used to encode the formal language (usually depicted by lower case letters),
- N is a finite set of non terminal symbols, with $N \cap \Sigma = \emptyset$ (usually depicted as upper case letters),
- P is a finite set of production rules of the form $\alpha \rightarrow \beta$, with $\alpha, \beta \in \{\Sigma \cup N\}^*$ and
- $S \in N$ is the start symbol, also called axiom, from which the generation is initiated.

4.3.1.2 Chomsky Hierarchy

Grammars can be classified into four types according to the Chomsky hierarchy[7].

- **Type 0: Free or unrestricted grammars** can express any language that is accepted by a Turing machine. Its production rules are of the form $\alpha \rightarrow \beta$, where there are no restrictions on what symbols appear in the strings α and β , except for $\alpha \neq \epsilon$. Therefore it is the most general grammar class.
- **Type 1: Production rules of context sensitive grammars** replace exactly one non-terminal symbol X by a string $\beta \in \{\Sigma \cup N\}^*$. The non-terminal Symbol can be embedded in terminal symbols that define the context in which the replacement may occur. The production rules are of the type $uXw \rightarrow u\beta w$, where u, β and w are strings consisting of symbols in $\{\Sigma \cup N\}$ and X is a non-terminal that may be replaced by β , but only if it appears in the context $u w$.
- **Type 2: Context free grammars** have production rules of the type $X \rightarrow \alpha$, where $\alpha \in \{\Sigma \cup N\}^*$ and X is a single non-terminal symbol. Thus a non-terminal X can be replaced by α independent of the context in which it appears.
- **Type 3: In regular grammars** the left side of the production rules is defined analog to grammars of type 2. The right side is restricted to at most one non-terminal and zero or one terminal symbol followed by one non-terminal. Therefore rules of regular grammars are of the form $X \rightarrow \epsilon$, $X \rightarrow A$ and $X \rightarrow aA$.

The corresponding languages are inclusive in the following order : $L_3 \subset L_2 \subset L_1 \subset L_0$.

4.3.1.3 Chomsky Normal Form

A formal grammar is in its Chomsky normal form (CNF) if and only if all of its production rules are of the form $A \rightarrow BC$, $A \rightarrow t$ or $S \rightarrow \epsilon$, where A, B and C are non-terminal symbols and t is a terminal Symbol [8]. Any context free language G can be transformed into its equivalent grammar in Chomsky normal G_{CNF} form that produces the same language L_G .

Thus, the production rules of any grammar in its CNF have one of the following simple structures:

Product type rules: $A \rightarrow BC$

Union type rules: $A \rightarrow B|C$

Terminal type rules: $A \rightarrow t$

Epsilon type rules: $S \rightarrow \epsilon$

Any further investigations on a given context free grammar can then be based on this four rule types. Therefore, using the CNF simplifies any further investigations on the grammar of interest.

4.3.1.4 Formal Languages and Enumeration

If a grammar that describes helix forming sequence patterns is known, it can be used to generate all corresponding sequences of a certain length n . However, as we are not interested in the actual sequences but in the number of such sequences, doing so is not an efficient approach. Instead we want to derive an analytic expression for computing the number of sequences of length n that are generated by such a grammar.

For any unambiguous grammar in Chomsky normal form, one can easily characterize the set of words L_V generated from each non-terminal $V \in N$ and the number v_n of its elements of length n , depending on the type of rule [15]:

- **Product type:** $V \rightarrow A.B$ with $A, B \in N$

Each word in L_V can be decomposed into one element in L_A and one element in L_B which are concatenated:

$$L_V = \{x.y | x \in L_A \text{ and } y \in L_B\} = L_A \times L_B$$

The grammar is unambiguous, thus this decomposition is unique and v_n can be computed as

$$v_n = |\{x.y | x \in L_A, y \in L_B \text{ and } |x| + |y| = n\}| = \sum_{i=0}^n a_i \times b_{n-i}.$$

- **Union type:** $V \rightarrow A, V \rightarrow B$ with $A, B \in N$

Provided the grammar is unambiguous, the two sets of words L_A and L_B are disjoint.

$$\text{Therefore, } L_V = L_A \cup L_B$$

$$\text{and } v_n = a_n + b_n.$$

- **Terminal type:** $V \rightarrow t$ with $t \in \Sigma$

Any rule of the terminal type produces a single word $\{t\}$ of length 1.

Thus, $L_V = \{t\}$

$$\text{and } v_n = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases}$$

- **Epsilon type:** $V \rightarrow \epsilon$

A single word of length 0 is generated.

Thus, $L_V = \{\epsilon\}$

$$\text{and } v_n = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$$

4.3.2 Combinatorial Classes and Generating Functions

A **combinatorial class** is a possibly infinite countable set A , where every object $x \in A$ has a length $|x| \in \mathbb{N}_{\geq 0}$ and the number of objects in A with length n is finite [15]. Formal languages are one example of combinatorial classes.

The **counting sequence** a of a combinatorial class is the sequence $a = a_0, a_1, a_2, \dots$ of the numbers of elements of size $n = 0, 1, 2, \dots$

Like any other such sequence of numbers, it can be written as a **generating function**. A generating function is the sum of a power series $G = a_0 + a_1z + a_2z^2 + a_3z^3 + \dots = \sum_{n \geq 0} a_n z^n$, where z is a formal variable [28].

Using these generating functions opens up a whole range of analytic techniques for problem solving. For example they are helpful when the exact formula for computing the sequence is not known or in case statistic properties of the sequence are of interest.

4.3.2.1 Generating Functions and Enumeration

In Section 4.3.1.4 I defined expressions for the number v_n of words of length n that can be produced from a non-terminal $V \in N$ dependent on the type of production rule. By definition

$$V(z) = \sum_{w \in L_V} z^{|w|} = \sum_{n \geq 0} v_n z^n.$$

Using this knowledge, a generating function $V(z)$ for the language L_V derived from a non-terminal V can be defined [15]:

- Product type $V \rightarrow AB$: After inserting the previously derived term for v_n into the general form of a generating function, we have

$$V(z) = \sum_{n \geq 0} v_n z^n = \sum_{n \geq 0} \left(\sum_{i=0}^n a_i \times b_{n-i} \right) z^n = \sum_{n \geq 0} \left(\sum_{i=0}^n a_i z^i \times b_{n-i} z^{n-i} \right). \quad (4)$$

Furthermore,

$$A(z) \times B(z) = \left(\sum_{p \geq 0} a_p z^p \right) \left(\sum_{q \geq 0} b_q z^q \right) = \sum_{n \geq 0} \left(\sum_{p+q=n} a_p \times b_q \right) z^n. \quad (5)$$

After substituting $p \rightarrow i$ and $q \rightarrow n - i$ the right hand side of both equations are identical and we conclude:

$$V(z) = A(z) \times B(z)$$

- Union type $V \rightarrow A$ and $V \rightarrow B$: After splitting the terms contributing the z^n into the contribution from A and B , we have:

$$\begin{aligned} V(z) &= \sum_{n \geq 0} v_n z^n = \sum_{n \geq 0} (a_n + b_n) z^n \\ &= \sum_{n \geq 0} a_n z^n + \sum_{n \geq 0} b_n z^n \\ &= A(z) + B(z). \end{aligned}$$

- Terminal type $V \rightarrow t$: As the only word in $L_V = \{t\}$

$$V(z) = \sum_{w \in L_V} z^{|w|} = z^{|t|} = z$$

- Epsilon type $V \rightarrow \epsilon$: The only word in $L_V = \{\epsilon\}$, thus

$$V(z) = \sum_{w \in L_V} z^{|w|} = z^{|\epsilon|} = z^0 = 1$$

Any unambiguous context-free grammar in CNF can thus be transformed into a system of functional equations. Solving the system yields the generating function of the corresponding grammar. A summary of the constructs derived in this section can be found in Table 4.

4.3.2.2 Useful Expressions for Working with Generating Functions

THE n -TH COEFFICIENT OF A GENERATING FUNCTION A common notation for the n -th coefficient of $G(z)$ is $[z^n]G(z)$.

Rule type	Language	n-th coefficient	Generating function
Product $V \rightarrow A.B$	$L_V = L_A \times L_B$	$v_n = \sum_{i=0}^n a_i \times b_{n-i}$	$V(z) = A(z) \times B(z)$
Union $V \rightarrow A, V \rightarrow B$	$L_V = L_A \cup L_B$	$v_n = a_n + b_n$	$V(z) = A(z) + B(z)$
Terminal $V \rightarrow t$	$L_V = \{t\}$	$v_n = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases}$	$V(z) = z$
Epsilon $V \rightarrow \epsilon$	$L_V = \{\epsilon\}$	$v_n = \begin{cases} 1 & \text{if } n = 0 \\ 0 & \text{otherwise} \end{cases}$	$V(z) = 1$

Table 4: Summary of constructs

QUASI INVERSE A language construct we need for our approach is the Kleene star V^* of a non-terminal symbol V [15]. The corresponding language L_{V^*} is

$$L_{V^*} = \bigcup_{k \geq 0} \underbrace{L_V \times \cdots \times L_V}_{k \text{ times}}$$

All elements in L_{V^*} can be recursively derived from following production rules:

$$V^* \rightarrow VV^* | \epsilon$$

Thus, we have the following efficient recursion for v_n^* :

$$v_n^* = \begin{cases} 1 & \text{if } n = 0, \\ \sum_{i=1}^n v_i \times v_{n-i}^* & \text{otherwise.} \end{cases} \quad (6)$$

Based on this expression, the generating function $V^*(z)$ of L_{V^*} is

$$V^*(z) = \sum_{n \geq 0} v_n^* z^n = V(z) \times V^*(z) + 1$$

which can be rearranged to

$$V^*(z) = \frac{1}{1 - V(z)}$$

where $\frac{1}{1 - V(z)}$ is called the quasi inverse.

As

$$[z^n] \frac{1}{1 - V(z)} = v_n^*,$$

we can use the recursive scheme of Equation 6 to compute the n -th coefficient of a quasi inverse.

GENERATING FUNCTION OF UNRESTRICTED WORDS B is the set of all words that can be build from the alphabet by concatenating symbols from the alphabet. It can also be viewed as a concatenation of non-terminals V that produce exactly one terminal symbol. As shown in section 4.3.2.1, the corresponding generating function is $V(z) = z$. After inserting this into the definition of the quasi inverse, we have $B(z) = \frac{1}{1-z}$.

4.4 ANALYTIC COMBINATORICS MODEL

Our goal is to return the probability that a sequence of length n can form a helix of a given length h (h consecutive base pairs). To achieve this, we describe below all sequences L that contain two given words H_1 and H_2 . These two words have to be complementary with respect to the base pairs $\mathcal{B} = \{AU, UA, CG, GC, GU, UG\}$ and to have length h . Therefore, they have the ability to form a specific helix. The description relies on a non-ambiguous decomposition.

4.4.1 Language equation on initial set

Writing the decomposition of L relies on the use of so-called initial sets and overlap sets.

Definition 4.4.1 *Given a word H , the initial set for H is the set of words w such that wH contains a single occurrence of H . The initial set is denoted as R .*

Example 4.4.1 *Let H be ACATACA. Words CCGT and CTACAGTT are in R and TTACAT is not, because H occurs at position 3 in TTACAT · ACATACA.*

Definition 4.4.2 *Given a word H , the overlap set for H is the set of words w of length smaller than h such that H is a suffix of Hw . It is denoted A .*

Example 4.4.2 *Let H be ACATACA. Set A is $\{CATACA, TACA\}$.*

The set L satisfies the decomposition equation below

$$L = R_1 H_1 \star^m R_2 H_2 B^* \quad (7)$$

where R_i is the initial set for word H_i and notation $\star^m$ indicates any word of length m . Uniqueness condition for initial set guarantees here that valid sequences are counted only once. For instance, sequence TACATACAGGACATACACCCTGTATGT contains two occurrences of $H = ACATACA$ but is counted only once in this decomposition.

The most challenging part is to find a description for language R when a word H is given. The set BH is the set of all words ending with H . It satisfies

$$BH = RH \cdot BH | RH\epsilon | RHA\epsilon \quad (8)$$

To prove this decomposition of any word bH , one searches for the leftmost occurrence of H in bH . Such an occurrence necessarily exists because H is a suffix of bH . Let r be the prefix of b that is before that leftmost occurrence of H . Then, word bH rewrites rHw .

This leftmost occurrence may also be the suffix H of b , then w is ϵ and bH is in Re . If it overlaps the H_1 suffix of bH , then word w is a suffix of H_1 that is in A and bH is in RHA . Otherwise, bH rewrites $rHxH$ that is in $RHBH$.

It follows from this proof that this decomposition is unique and complete.

4.4.2 Generating function of initial set

Using Equations (4) and (5), we can derive a functional equation on generating functions.

$$B(z)H(z) = R(z)H(z)B(z)H(z) + R(z)H(z) + R(z)H(z)A(z)$$

It was shown above that $B(z) = \frac{1}{1-z}$. An expression for $R(z)$ follows.

$$R(z) = \frac{1}{H(z) + (1-z)A(z) + (1-z)} \quad (9)$$

Setting

$$T(z) = z + zA(z) - A(z) - H(z)$$

yields

$$R(z) = \frac{1}{1-T(z)}.$$

Observing that $T(0) = 0$ and using the quasi inverse (see Eq. 6) leads to

$$R(n) = \begin{cases} 1, & \text{if } n = 0 \\ \sum_{i=1}^n T(i)R(n-i), & \text{otherwise.} \end{cases} \quad (10)$$

4.4.3 Implementation

To obtain a function $f_L(n)$ that returns the total probability of words of length n in $L = R_1H_1 \star^m R_2H_2B^*$, we can multiply the probabilities of its components.

Based on the generating functions that describe the initial set we obtain the following recursive equations for computing $f_R(n)$:

$$f_R(n) = \begin{cases} 1, & \text{if } n = 0 \\ \sum_{i=1}^n f_T(i)f_R(n-i), & \text{otherwise} \end{cases}$$

$$f_T(n) = f_A(n-1) - f_A(n) + \delta + \gamma$$

with $\delta = 1$ if $n = 1$, and $\gamma = -f_P(H)$ if $n = |H|$.

$$f_A(n) = \begin{cases} 0, & \text{if } n = 0 \\ p(w), & \text{if } \exists v, w \text{ such that } vH_1 = H_1w \text{ and } |w| = n \\ 0, & \text{otherwise} \end{cases}$$

Additional components we need are B and $\star^m$. As both sets are unrestricted, $f_B(n) = 1$ and $f_{\star^m}(n) = 1$.

The current implementation uses specific H_1 and H_2 sequences that are complementary. Describing the relationship between H_1 and H_2 using analytic combinatorics could improve the model but is beyond the scope of this study.

To compute the probability of a specific H_1 - H_2 pattern we can use the probabilities $p(H_1)$ and $p(H_2)$. $L = R_1H_1\star^mR_2H_2B^*$ (or $L \rightarrow R_1H_1\star^mR_2H_2B^*$) is a product type rule. Due to the commutative property of products we can rearranged the expression:

$$L = H_1H_2\star^mR_1R_2B^*.$$

H_1 , H_2 and $\star^m$ have a fixed length, thus we can write

$$f_L(n) = p(H_1)p(H_2) \sum_{i=0}^{n-2h-m} f_{R_1}(i)f_{L_1}(n-2h-m-i)$$

and

$$L_1 = R_2B^* \quad f_{L_1}(n) = \sum_{i=0}^n f_{R_2}(i) \times 1.$$

This further simplifies to

$$f_L(n) = p(H_1)p(H_2) \sum_{i=0}^{n-2h-m} f_{R_1}(i)f_{R_2}(n-2h-m-i). \quad (11)$$

By memoizing (see Section 3.2.2) f_R , f_A and f_T we obtain an efficient implementation.

4.5 OUTLOOK

A range of methods for predicting thermodynamically and kinetically important structures exist. While one can easily compute the probability of a random sequence to be able to form one specific structure (see Eq. 1) and thereby evaluate the significance of detecting this exact structure, scoring the significance of finding similar structures in the same area is a much more complex problem.

For instance in the Trajectory Alignment Chapter 3 we evaluate the similarity of two steps of different folding trajectories using a structure alignment score. Even when steps are highly similar their structures and sequences vary slightly. Rather than forming the exact same

structure they form a similar structure motive. To assess the significance of two sequences forming structures that fit such a shared sequence motive, we have to compute the probability that a random sequence can form a structure that fits the structure motive. For our initial background model we limited ourselves to simple structure patterns containing a single helix.

While stochastic methods are an important tool for testing the significance of predicted structures, they are limited when events with a low probability are of interest. Generating a large enough set of random data is only feasible for short structure motives. Using a stochastic model we showed that the significance of a detected structure sharply decreases with the flexibility that a structure motive allows. Thus, accounting for this flexibility in the background model is essential.

To be able to cover longer structure motives, we decided to introduce a background model based on analytic combinatorics: To correctly count the number of random sequence that can form a given structure motif (pattern) we have to find an unambiguous grammar that describes such a motif. Only if there is a one single way of generating each sequence that forms the helix pattern from the grammar, counting sequences multiple times can be avoided.

In the case of simple helix patterns a sequence needs to feature two complementary strands H_1 and H_2 that are separated by at least three nucleotides. We managed to find such an unambiguous description for when H_1 and H_2 have a fixed sequence. Future work will be to extend the model to all possible H_1 - H_2 combinations. As sequence might fit multiple of the fixed H_1 - H_2 patterns, the remaining challenge is to find a correction term for the resulting overcounting.

In the case of detecting the same structure motive in multiple homologous sequences, including phylogenetic information into the significance assessment. Finding a similar structure in evolutionary distant RNA sequences indicates that it might be conserved through evolutionary pressure due to its functional importance and not just randomly formed. While we can use existing methods to evaluate the evolutionary distance between two sequences of interest, including this information, for example by counting compensatory mutations, into the combinatorial background model has not been done yet and might prove challenging.

DISCUSSION

The function of non-coding RNAs is highly dependent on their structure. For example they might have to form a specific interaction site, as the SRP RNA does in order to interact with the SRP signal peptide and to stall the translation into protein until it can be released through a membrane translocon. Others might control genetic pathways by activating or terminating translation and transcription.

Experiments show that many RNAs can form multiple functional conformations. As RNA sequences start folding during their transcription, the elongation of the sequence is one of the major effects that controls structure formation. Some transient structures have been investigated experimentally and functions were suggested:

The TRP operon leader and the SAM riboswitch, for example, form transient structures based on ligand or Trp concentrations and can terminate transcription. Another instance are guiding structures that ensure that RNAs do not fold into a non-functional trapping structure that inhibits the formation of the functional structure or delay the formation of an inactive conformation, like the 5' UTR of the LEVI maturation gene.

While RNA sequences are subjected to mutations over the course of evolution, the structures are conserved to retain their function. While this is well studied for the final functional structure that is reached after full transcription, little is known about the conservation of transient structures. The Meyer research group published a tool, Transat, that can detect the conservation of helices in sequence alignments of homologous RNAs. However, this tool does not evaluate whether such a putative conserved structure actually forms during the co-transcriptional folding process or at any other point within the lifetime of an RNA.

By using Kinwalker and Kinefold, programs that can simulate co-transcriptional folding, we obtained structures that are actually formed during the folding process. Based on a sequence alignment of homologous RNAs we then investigated the conservation of structures in the folding pathway (see Chapter 2). Our simulations suggest that not only the final functional structure but also experimentally determined transient structures are indeed conserved. In addition we identified previously unknown conserved structures that might as well be functional.

In case of the TRP operon leader the terminator structure was found in the folding trajectories of nearly all homologous sequences. However, we were not able to find the anti-terminator structure. This result, however, was to a certain degree expected for this family because

the anti-terminator only forms when ribosomes attach to the TRP operon leader. As the folding programs cannot simulate the ribosome-mRNA interaction, the anti-terminator structure does not appear in the simulated trajectories.

In other cases where no additional partners influence the refolding, I was able to successfully predict the important transient structures. Even changes in transcription speed and polymerase pausing sites were successfully modeled: Wong, Sosnick, and Pan [66] suggested that the formation of the final functional structure of the SRP RNA is guided by a short range helix in the 5' end of the sequence and a pausing site. Even when using the optimized Kinwalker parameters the SRP RNA folding trajectory does not reach the final functional structure (= minimum free energy structure) within the lifetime of the RNA. While Kinwalker does not support the simulation of pausing sites natively, I was able to approximate it by simulating the co-transcriptional folding up to the pausing site, giving the sequence time to reach its MFE structure and then restarting the co-transcriptional folding simulation from the obtained structure and sequence. Doing so, I found that the subsequence folds into MFE structure almost immediately after the pausing site is reached and features the experimentally identified short range helix. Starting from the MFE structure of the subsequence the co-transcriptional folding process is directly guided into the functional MFE structure formed by the full sequence. Thus, the simulations support the importance of the pausing site. Even though the program does not support the simulation of pausing sites natively, I showed that by using more creative approaches putative pausing sites can be investigated computationally.

Transat uses a purely statistical method that analyses all possible helices in a sequence alignment without considering their thermodynamic stability or kinetic importance. In contrast to the kinetic folding simulation programs it does not rely on correctly modeling factors that influence the folding behavior. Thus, it is able to detect conserved helices that Kinefold and Kinwalker do not recover because they for example cannot model interaction with biomolecules like proteins, other RNAs, ligands, ribosomes or polymerase and depend on factors like transcription speed and temperature. Amongst other important structures Transat found the anti-terminator structure of the TRP operon leader that the folding simulation programs missed, because they cannot model the ribosome-mRNA interaction. As it scores all helices individually it also is able to recover pseudoknotted structures like the active configuration of the HDV ribozyme. One drawback is that Transat sometimes has difficulties when analyzing alignments with high sequence similarity. Especially when a helix is formed by areas that are fully conserved, for example due to a protein interaction site, it often assigns a low significance to the helix. Thus, Transat is a highly valuable tool for detecting structures that other programs miss because they only appear under conditions that the folding simulation programs can not model. However, a helix that is

not considered significant by Transat might appear in most folding trajectories and can be functional nevertheless.

Pseudoknots are essential parts of the functional structure of the HDV ribozyme. While Kinwalker cannot model those knotted helices, Kinefold is able to model some types of pseudoknots and therefore recovers more base pairs of the active HDV ribozyme structure.

While analysing the folding behavior of evolutionary related RNAs, I saw the need for finding representative subsets. To find subgroups in my dataset I clustered the RNA sequences based on their phylogenetic distance, because I found certain structural elements to be only present in subgroups of each RNA family. Zhu et al. [70] used a filtering step to avoid having sequences with an overly high sequence similarity in their dataset and thus their method often selected sequences that only represent a small subset of the known homologous RNAs. A general problem is that, to reliably detect conserved structures, the sequences need to show sufficient sequence variation while still allowing for a reliable alignment. Using my clustering approach I was able to investigate group specific structures within clusters and detect family wide conserved structures based on cluster representatives. Performing the family wide studies based on cluster representatives ensures that groups that feature a lot of highly similar sequences are not overrepresented.

In many cases the clustering revealed that different subgroups of sequences in RNA families featured a slightly different folding path to the shared functional structure. For example there are two different subgroups in the LEVI maturation gene 5'UTR set: One group can be represented by the MS2 phage, while the other group can be represented by the KU1 phage. While the KU1 phage actually forms a short hairpin that supposedly delays the formation of the inactive final structure, I did not obtain significant scores for it in the group represented by the MS2 phage. Based on sequence similarity and folding behavior of the 5'UTR sequence some phages of the MS2 group seem to be more closely related to KU1 than to MS2.

Most methods that can detect the evolutionary conservation of (transient) RNA structures strongly rely on correct sequence alignments of the RNAs in question. If the same structure appears in two sequences but they are not correctly aligned, these methods fail. Rfam alignments are built to represent the MFE structure. However, nucleotides that are not involved in the MFE structures are often poorly aligned. A way of improving areas of the alignments that are involved in alternative structures is to use programs like Carna [36, 51] that are capable of aligning sequences based on multiple structures. Even such improved alignments cannot solve one major problem: Different steps (structures) in the folding trajectories might require different alignments, because the structures are slightly shifted.

This problem, for instance, occurred in the detection of the SAM anti-terminator helix and the TRP operon anti-terminator helix. The conservation of both transient structures was not correctly detected be-

cause the transient structures require a different alignment than the final functional structure.

We developed a method for aligning kinetic folding trajectories. By computing a structure based alignment for the individual structures in the folding trajectory our approach solves the problem that one alignment sometimes does not support all transient structures. In addition it provides extensive information on the folding behavior in general.

As we can see in the example of the SAM riboswitch, the RNAs reach their functional structure through similar folding pathways. Steps that are important to the function are especially well conserved and lead to clear convergence points between the folding trajectories.

The last step in this work was to introduce a background model that computes the probability of structures and structure motives in random sequences.

When we investigate the conservation of structures in RNA folding trajectories we most likely find similar but not identical structures. To build a background model for such cases we need to compute the probability of a random structure to form such a structure motive.

Using stochastic models we showed that allowing even small differences between structures means that a much larger set of random sequences can form such a structure motive. Thus, detecting a structure that just fits the structure motive is a lot less significant than finding the exact same structure. Our background model, therefore, has to account for this structural flexibility.

The reference sequence of the LEVI virus maturation gene 5'UTR, for example, forms a small hairpin that is involved in delaying the formation of the inactive MFE structure. Ideally, the background model is able to show, whether or not the shifted hairpins that were found in the folding trajectories of homologous RNAs should be considered the same structure.

Taken together we showed that kinetic folding programs can correctly predict experimentally found transient RNA structures. Known transient structures that have a function described in literature are conserved in the folding pathways of homologous RNAs. On a more general level we found that homologous RNAs often reach their functional structures through similar pathways. We will publish the kinetic RNA folding dataset on our website for benchmarking purposes and further investigations. The Python library that was developed to build this dataset will be released as part of the ViennaRNA Python library and provides tools for working with kinetic folding simulation programs.

BIBLIOGRAPHY

- [1] S F Altschul and B W Erickson. "Significance of nucleotide sequence alignments: a method for random sequence permutation that preserves dinucleotide and codon usage." In: *Molecular Biology and Evolution* 2.6 (1985), pp. 526–538. DOI: [10.1093/oxfordjournals.molbev.a040370](https://doi.org/10.1093/oxfordjournals.molbev.a040370). eprint: [/oup/backfile/content/_public/journal/mbe/2/6/10.1093_oxfordjournals.molbev.a040370/1/8alts.pdf](http://oup/backfile/content/_public/journal/mbe/2/6/10.1093_oxfordjournals.molbev.a040370/1/8alts.pdf). URL: <http://dx.doi.org/10.1093/oxfordjournals.molbev.a040370>.
- [2] Ataide Sandro F, Schmitz Nikolaus, Shen Kuang, Ke Ailong, Shan Shu-ou, Doudna Jennifer A, and Ban Nenad. "The Crystal Structure of the Signal Recognition Particle in Complex with its Receptor." In: *Science (New York, N.Y.)* 331.6019 (2011), pp. 881–886. ISSN: 0036-8075 1095-9203. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3758919>.
- [3] Richard Bellman. "The theory of dynamic programming." In: *Bull. Amer. Math. Soc.* 60.6 (Nov. 1954), pp. 503–515. URL: <http://projecteuclid.org/euclid.bams/1183519147>.
- [4] Athanasius F. Bompfünnewerer, Rolf Backofen, Stephan H. Bernhart, Jana Hertel, Ivo L. Hofacker, Peter F. Stadler, and Sebastian Will. "Variations on RNA folding and alignment: lessons from Benasque." In: *Journal of Mathematical Biology* 56.1 (2008), pp. 129–144. ISSN: 1432-1416. DOI: [10.1007/s00285-007-0107-5](https://doi.org/10.1007/s00285-007-0107-5). URL: <http://dx.doi.org/10.1007/s00285-007-0107-5>.
- [5] CRICK FRANCIS. "Central Dogma of Molecular Biology." In: *Nature* 227.5258 (1970). 10.1038/227561a0, pp. 561–563. DOI: [10.1038/227561a0](https://doi.org/10.1038/227561a0).
- [6] Durga M. Chadalavada, Scott M. Knudsen, Shu-ichi Nakano, and Philip C. Bevilacqua. "A role for upstream {RNA} structure in facilitating the catalytic fold of the genomic hepatitis delta virus ribozyme1." In: *Journal of Molecular Biology* 301.2 (2000), pp. 349–367. ISSN: 0022-2836. DOI: [10.1006/jmbi.2000.3953](https://doi.org/10.1006/jmbi.2000.3953). URL: <http://www.sciencedirect.com/science/article/pii/S0022283600939536>.
- [7] N. Chomsky. "Three models for the description of language." In: *IRE Transactions on Information Theory* 2.3 (1956), pp. 113–124. ISSN: 0096-1000. DOI: [10.1109/TIT.1956.1056813](https://doi.org/10.1109/TIT.1956.1056813).
- [8] Noam Chomsky. "On certain formal properties of grammars." In: *Information and Control* 2.2 (1959), pp. 137–167. ISSN: 0019-9958. DOI: [10.1016/S0019-9958\(59\)90362-6](https://doi.org/10.1016/S0019-9958(59)90362-6). URL: <http://www.sciencedirect.com/science/article/pii/S0019995859903626>.
- [9] FHC Crick. *On protein synthesis. SEB Symposium, No. 12.* 1958.

- [10] "Database resources of the National Center for Biotechnology Information." In: *Nucleic Acids Research* 41.D1 (2013), pp. D8–D20. DOI: [10.1093/nar/gks1189](https://doi.org/10.1093/nar/gks1189). eprint: [/oup/backfile/content_public/journal/nar/41/d1/10.1093/nar/gks1189/2/gks1189.pdf](http://oup/backfile/content_public/journal/nar/41/d1/10.1093/nar/gks1189/2/gks1189.pdf). URL: <http://dx.doi.org/10.1093/nar/gks1189>.
- [11] Sean R. Eddy and Richard Durbin. "RNA sequence analysis using covariance models." In: *Nucleic Acids Research* 22.11 (1994), p. 2079. DOI: [10.1093/nar/22.11.2079](https://doi.org/10.1093/nar/22.11.2079). eprint: [/oup/backfile/content_public/journal/nar/22/11/10.1093_nar_22.11.2079/2/22-11-2079.pdf](http://oup/backfile/content_public/journal/nar/22/11/10.1093_nar_22.11.2079/2/22-11-2079.pdf). URL: <http://dx.doi.org/10.1093/nar/22.11.2079>.
- [12] Edgar Robert C. "MUSCLE: a multiple sequence alignment method with reduced time and space complexity." In: *BMC Bioinformatics* 5 (2004), pp. 113–113. ISSN: 1471-2105. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC517706>.
- [13] Edgar Robert C. "MUSCLE: multiple sequence alignment with high accuracy and high throughput." In: *Nucleic Acids Research* 32.5 (2004), pp. 1792–1797. ISSN: 0305-1048 1362-4962. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC390337>.
- [14] Joseph Felsenstein. "Evolutionary trees from DNA sequences: A maximum likelihood approach." In: *Journal of Molecular Evolution* 17.6 (Nov. 1981), pp. 368–376. ISSN: 0022-2844. DOI: [10.1007/BF01734359](https://doi.org/10.1007/BF01734359).
- [15] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Cambridge University Press, 2009. ISBN: 978-0-521-89806-5. URL: <http://www.cambridge.org/uk/catalogue/catalogue.asp?isbn=9780521898065>; <http://dblp.dagstuhl.de/rec/bib/books/daglib/0023751>.
- [16] Flamm C, Fontana W, Hofacker I L, and Schuster P. "RNA folding at elementary step resolution." In: *RNA* 6.3 (2000), pp. 325–338. ISSN: 1355-8382 1469-9001. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1369916>.
- [17] Michael Geis, Christoph Flamm, Michael T. Wolfinger, Andrea Tanzer, Ivo L. Hofacker, Martin Middendorf, Christian Mandl, Peter F. Stadler, and Caroline Thurner. "Folding Kinetics of Large {RNAs}." In: *Journal of Molecular Biology* 379.1 (2008), pp. 160–173. ISSN: 0022-2836. DOI: [10.1016/j.jmb.2008.02.064](https://doi.org/10.1016/j.jmb.2008.02.064). URL: <http://www.sciencedirect.com/science/article/pii/S002228360800274X>.
- [18] Reid Gilmore, Gunter Blobel, and Peter Walter. "Protein translocation across the endoplasmic reticulum. I. Detection in the microsomal membrane of a receptor for the signal recognition particle." In: *The Journal of Cell Biology* 95.2 (1982), pp. 463–469. ISSN: 0021-9525 1540-8140. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2112970>.

- [19] Sam Griffiths-Jones, Alex Bateman, Mhairi Marshall, Ajay Khanna, and Sean R. Eddy. "Rfam: an RNA family database." In: *Nucleic Acids Research* 31.1 (2003), p. 439. DOI: [10.1093/nar/gkg006](https://doi.org/10.1093/nar/gkg006). eprint: [/oup/backfile/content_public/journal/nar/31/1/10.1093_nar_gkg006/2/gkg006.pdf](http://oup/backfile/content_public/journal/nar/31/1/10.1093_nar_gkg006/2/gkg006.pdf). URL: <http://dx.doi.org/10.1093/nar/gkg006>.
- [20] Groeneveld H, Thimon K, and J van Duin. "Translational control of maturation-protein synthesis in phage MS2: a role for the kinetics of RNA folding?" In: *RNA* 1.1 (1995), pp. 79–88. ISSN: 1355-8382 1469-9001. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1369063>.
- [21] Frank J. Grundy and Tina M. Henkin. "The S box regulon: a new global transcription termination control system for methionine and cysteine biosynthesis genes in Gram-positive bacteria." In: *Molecular Microbiology* 30.4 (1998), pp. 737–749. ISSN: 1365-2958. DOI: [10.1046/j.1365-2958.1998.01105.x](https://doi.org/10.1046/j.1365-2958.1998.01105.x). URL: <http://dx.doi.org/10.1046/j.1365-2958.1998.01105.x>.
- [22] M. Hochsmann, T. Toller, R. Giegerich, and S. Kurtz. "Local similarity in RNA secondary structures." In: *Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003*. 2003, pp. 159–168. DOI: [10.1109/CSB.2003.1227315](https://doi.org/10.1109/CSB.2003.1227315).
- [23] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. "Fast folding and comparison of RNA secondary structures." In: *Monatshefte für Chemie / Chemical Monthly* 125.2 (1994), pp. 167–188. ISSN: 1434-4475. DOI: [10.1007/BF00818163](https://doi.org/10.1007/BF00818163). URL: <http://dx.doi.org/10.1007/BF00818163>.
- [24] Ivo L. Hofacker, Peter Schuster, and Peter F. Stadler. "Combinatorics of RNA secondary structures." In: *Discrete Applied Mathematics* 88.1 (1998). Computational Molecular Biology DAM - CMB Series, pp. 207–237. ISSN: 0166-218X. DOI: [10.1016/S0166-218X\(98\)00073-0](https://doi.org/10.1016/S0166-218X(98)00073-0). URL: <http://www.sciencedirect.com/science/article/pii/S0166218X98000730>.
- [25] Isambert Hervé and Siggia Eric D. "Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme." In: *Proceedings of the National Academy of Sciences of the United States of America* 97.12 (1999), pp. 6515–6520. ISSN: 0027-8424 1091-6490. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC18642>.
- [26] Minghui Jiang, James Anderson, Joel Gillespie, and Martin Mayne. "uShuffle: A useful tool for shuffling biological sequences while preserving the k-let counts." In: *BMC Bioinformatics* 9.1 (2008), p. 192. ISSN: 1471-2105. DOI: [10.1186/1471-2105-9-192](https://doi.org/10.1186/1471-2105-9-192). URL: <https://doi.org/10.1186/1471-2105-9-192>.
- [27] Tao Jiang, Lusheng Wang, and Kaizhong Zhang. "Alignment of trees — an alternative to tree edit." In: *Theoretical Computer Science* 143.1 (1995), pp. 137–148. ISSN: 0304-3975. DOI: [10.1016/](https://doi.org/10.1016/)

- 0304-3975(95)80029-9. URL: <http://www.sciencedirect.com/science/article/pii/0304397595800299>.
- [28] Donald E. Knuth. *The Art of Computer Programming, Volume 1 (3rd Ed.): Fundamental Algorithms*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1997. ISBN: 0-201-89683-4.
- [29] Lentzen G, Moine H, Ehresmann C, Ehresmann B, and Wintermeyer W. "Structure of 4.5S RNA in the signal recognition particle of Escherichia coli as studied by enzymatic and chemical probing." In: *RNA* 2.3 (1996), pp. 244–253. ISSN: 1355-8382 1469-9001. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1369367>.
- [30] Leontis N B and Westhof E. "Geometric nomenclature and classification of RNA base pairs." In: *RNA* 7.4 (2001), pp. 499–512. ISSN: 1355-8382 1469-9001. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1370104>.
- [31] Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. "ViennaRNA Package 2.0." In: *Algorithms for Molecular Biology* 6.1 (2011), p. 26. ISSN: 1748-7188. DOI: 10.1186/1748-7188-6-26. URL: <http://dx.doi.org/10.1186/1748-7188-6-26>.
- [32] Makoto Matsumoto and Takuji Nishimura. "Mersenne Twister: A 623-dimensionally Equidistributed Uniform Pseudo-random Number Generator." In: *ACM Trans. Model. Comput. Simul.* 8.1 (Jan. 1998), pp. 3–30. ISSN: 1049-3301. DOI: 10.1145/272991.272995. URL: <http://doi.acm.org/10.1145/272991.272995>.
- [33] J. S. McCaskill. "The equilibrium partition function and base pair binding probabilities for RNA secondary structure." In: *Biopolymers* 29.6-7 (1990), pp. 1105–1119. ISSN: 1097-0282. DOI: 10.1002/bip.360290621. URL: <http://dx.doi.org/10.1002/bip.360290621>.
- [34] D van Meerten, Girard G, and J van Duin. "Translational control by delayed RNA folding: identification of the kinetic trap." In: *RNA* 7.3 (Mar. 2001), pp. 483–494. ISSN: 1355-8382 1469-9001. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC1370103>.
- [35] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. "Equation of State Calculations by Fast Computing Machines." In: *The Journal of Chemical Physics* 21.6 (1953), pp. 1087–1092. DOI: 10.1063/1.1699114. URL: <http://scitation.aip.org/content/aip/journal/jcp/21/6/10.1063/1.1699114>.
- [36] Mathias Möhl, Raheleh Salari, Sebastian Will, Rolf Backofen, and S. Cenk Sahinalp. "Sparsification of RNA Structure Prediction Including Pseudoknots." In: *Proc. of the 10th Workshop on Algorithms in Bioinformatics (WABI)*. Ed. by Vincent Moulton and Mona Singh. Vol. 6293. Lecture Notes in Computer Science.

- Springer Berlin / Heidelberg, 2010, pp. 40–51. DOI: [10.1007/978-3-642-15294-8_4](https://doi.org/10.1007/978-3-642-15294-8_4).
- [37] Steven R Morgan and Paul G Higgs. “Barrier heights between ground states in a model of RNA secondary structure.” In: *Journal of Physics A: Mathematical and General* 31.14 (1998), p. 3153.
 - [38] Eric P. Nawrocki et al. “Rfam 12.0: updates to the RNA families database.” In: *Nucleic Acids Research* 43.D1 (2015), p. D130. DOI: [10.1093/nar/gku1063](https://doi.org/10.1093/nar/gku1063). eprint: [/oup/backfile/content_public/journal/nar/43/d1/10.1093_nar_gku1063/2/gku1063.pdf](http://oup/backfile/content_public/journal/nar/43/d1/10.1093_nar_gku1063/2/gku1063.pdf). URL: <http://dx.doi.org/10.1093/nar/gku1063>.
 - [39] Cédric Notredame, Desmond G Higgins, and Jaap Heringa. “T-coffee: a novel method for fast and accurate multiple sequence alignment.” Edited by J. Thornton. In: *Journal of Molecular Biology* 302.1 (2000), pp. 205–217. ISSN: 0022-2836. DOI: [10.1006/jmbi.2000.4042](https://doi.org/10.1006/jmbi.2000.4042). URL: <http://www.sciencedirect.com/science/article/pii/S0022283600940427>.
 - [40] Nussinov R and Jacobson A B. “Fast algorithm for predicting the secondary structure of single-stranded RNA.” In: *Proceedings of the National Academy of Sciences of the United States of America* 77.11 (1980), pp. 6309–6313. ISSN: 0027-8424 1091-6490. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC350273>.
 - [41] Ruth Nussinov, George Pieczenik, Jerrold R. Griggs, and Daniel J. Kleitman. “Algorithms for Loop Matchings.” In: *SIAM Journal on Applied Mathematics* 35.1 (1978), pp. 68–82. DOI: [10.1137/0135006](https://doi.org/10.1137/0135006). eprint: <https://doi.org/10.1137/0135006>. URL: <https://doi.org/10.1137/0135006>.
 - [42] Stéphanie Pascarella and Francesco Negro. “Hepatitis D virus: an update.” In: *Liver International* 31.1 (2011), pp. 7–21. ISSN: 1478-3231. DOI: [10.1111/j.1478-3231.2010.02320.x](https://doi.org/10.1111/j.1478-3231.2010.02320.x). URL: <http://dx.doi.org/10.1111/j.1478-3231.2010.02320.x>.
 - [43] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python.” In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
 - [44] Poot Raymond A, Tsareva Nina V, Boni Irina V, and van Duin Jan. “RNA folding kinetics regulates translation of phage MS2 maturation gene.” In: *Proceedings of the National Academy of Sciences of the United States of America* 94.19 (1997), pp. 10110–10115. ISSN: 0027-8424 1091-6490. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC23320>.
 - [45] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. “Fast-Tree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix.” In: *Molecular Biology and Evolution* 26.7 (2009), pp. 1641–1650. DOI: [10.1093/molbev/msp077](https://doi.org/10.1093/molbev/msp077). eprint: [/oup/backfile/content_public/journal/mbe/26/7/10.1093_molbev_msp077/2/msp077.pdf](http://oup/backfile/content_public/journal/mbe/26/7/10.1093_molbev_msp077/2/msp077.pdf). URL: <http://dx.doi.org/10.1093/molbev/msp077>.

- [46] Morgan N. Price, Paramvir S. Dehal, and Adam P. Arkin. "Fast-Tree 2 – Approximately Maximum-Likelihood Trees for Large Alignments." In: *PLOS ONE* 5.3 (Mar. 2010), pp. 1–10. DOI: [10.1371/journal.pone.0009490](https://doi.org/10.1371/journal.pone.0009490). URL: <https://doi.org/10.1371/journal.pone.0009490>.
- [47] *Python 3.6.2 documentation: 9.6. random — generate pseudo-random numbers*. Accessed: 2017-08-01. Dec. 2016. URL: <https://docs.python.org/3.6/library/random.html>.
- [48] Gruber Andreas R., Findeiß Sven, Washietl Stefan, Hofacker Ivo L., and Stadler Peter F. "RNAZ 2.0: Improved Noncoding RNA Detection." In: *WORLD SCIENTIFIC*, 2009, pp. 69–79. ISBN: 978-981-4299-47-3.
- [49] A Rich and JD Watson. "SOME RELATIONS BETWEEN DNA AND RNA." In: *Proceedings of the National Academy of Sciences of the United States of America* 40.8 (1954), pp. 759–764. ISSN: 0027-8424. DOI: [10.1073/pnas.40.8.759](https://doi.org/10.1073/pnas.40.8.759). URL: <http://europepmc.org/articles/PMC534159>.
- [50] Sakakibara Y, Brown M, Hughey R, Mian I S, Sjölander K, Underwood R C, and Haussler D. "Stochastic context-free grammars for tRNA modeling." In: *Nucleic Acids Research* 22.23 (1994), pp. 5112–5120. ISSN: 0305-1048 1362-4962. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC523785>.
- [51] Dragos A. Sorescu, Mathias Möhl, Martin Mann, Rolf Backofen, and Sebastian Will. "CARNA - alignment of RNA structure ensembles." In: *Nucleic Acids Res* 40.W1 (2012). DAS, MMö, and MMA contributed equally to this work., W49–W53. ISSN: 1362-4962. DOI: [10.1093/nar/gks491](https://doi.org/10.1093/nar/gks491).
- [52] Stombaugh Jesse, Zirbel Craig L, Westhof Eric, and Leontis Neocles B. "Frequency and isostericity of RNA base pairs." In: *Nucleic Acids Research* 37.7 (2009), pp. 2294–2312. ISSN: 0305-1048 1362-4962. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2673412>.
- [53] John M. Taylor. "Hepatitis delta virus." In: *Virology* 344.1 (2006). *Virology 50th Anniversary Issue*, pp. 71–76. ISSN: 0042-6822. DOI: [10.1016/j.virol.2005.09.033](https://doi.org/10.1016/j.virol.2005.09.033).
- [54] Turner Douglas H and Mathews David H. "NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure." In: *Nucleic Acids Research* 38.Database issue (2009), pp. D280–D282. ISSN: 0305-1048 1362-4962. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2808915>.
- [55] Gabriele Varani and William H McClain. "The G-U wobble base pair." In: *EMBO reports* 1.1 (2000), pp. 18–23. ISSN: 1469-221X. DOI: [10.1093/embo-reports/kvd001](https://doi.org/10.1093/embo-reports/kvd001). eprint: <http://embor.embopress.org/content/1/1/18.full.pdf>. URL: <http://embor.embopress.org/content/1/1/18>.

- [56] WATSON J. D. and CRICK F. H. C. "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid." In: *Nature* 171.4356 (1953). 10.1038/171737a0, pp. 737–738. DOI: 10.1038/171737a0.
- [57] Peter Walter, Ibrahim Ibrahimi, and Gunter Blobel. "Translocation of proteins across the endoplasmic reticulum. I. Signal recognition protein (SRP) binds to in-vitro-assembled polysomes synthesizing secretory protein." In: *The Journal of Cell Biology* 91.2 (1981), pp. 545–550. ISSN: 0021-9525 1540-8140. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2111968>.
- [58] Stefan Washietl, Ivo L. Hofacker, and Peter F. Stadler. "Fast and reliable prediction of noncoding RNAs." In: *Proceedings of the National Academy of Sciences of the United States of America* 102.7 (2005), pp. 2454–2459. DOI: 10.1073/pnas.0409169102. eprint: <http://www.pnas.org/content/102/7/2454.full.pdf>. URL: <http://www.pnas.org/content/102/7/2454.abstract>.
- [59] M.S. Waterman and T.F. Smith. "RNA secondary structure: a complete mathematical analysis." In: *Mathematical Biosciences* 42.3 (1978), pp. 257–266. ISSN: 0025-5564. DOI: 10.1016/0025-5564(78)90099-8. URL: <http://www.sciencedirect.com/science/article/pii/0025556478900998>.
- [60] Michael S Waterman. "Secondary structure of single-stranded nucleic acids." In: *Adv. math. suppl. studies* 1 (1978), pp. 167–212.
- [61] Kevin M Weeks. "Advances in RNA structure analysis by chemical probing." In: *Current opinion in structural biology* 20.3 (2010), pp. 295–304. ISSN: 0959-440X. DOI: 10.1016/j.sbi.2010.04.001. URL: <http://europepmc.org/articles/PMC2916962>.
- [62] Nicholas J. P. Wiebe and Irmtraud M. Meyer. "Transat—A Method for Detecting the Conserved Helices of Functional RNA Structures, Including Transient, Pseudo-Knotted and Alternative Structures." In: *PLOS Computational Biology* 6.6 (June 2010), pp. 1–22. DOI: 10.1371/journal.pcbi.1000823. URL: <https://doi.org/10.1371/journal.pcbi.1000823>.
- [63] Sebastian Will, Kristin Reiche, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. "Inferring Non-Coding RNA Families and Classes by Means of Genome-Scale Structure-Based Clustering." In: *PLoS Comput Biol* 3.4 (2007), e65. ISSN: 1553-7358. DOI: 10.1371/journal.pcbi.0030065.
- [64] Sebastian Will, Tejal Joshi, Ivo L. Hofacker, Peter F. Stadler, and Rolf Backofen. "LocARNA-P: Accurate boundary prediction and improved detection of structural RNAs." In: *RNA* 18.5 (2012), pp. 900–14. ISSN: 1469-9001. DOI: 10.1261/rna.029041.111.
- [65] Winkler Wade C, Nahvi Ali, Sudarsan Narasimhan, Barrick Jeffrey E, and Breaker Ronald R. "An mRNA structure that controls gene expression by binding S-adenosylmethionine." In: *Nat Struct Mol Biol* 10.9 (Sept. 2003). 10.1038/nsb967, pp. 701–

707. ISSN: 1072-8368. DOI: 10.1038/nsb967. URL: http://www.nature.com/nsmb/journal/v10/n9/supinfo/nsb967/_S1.html.
- [66] Terrence N. Wong, Tobin R. Sosnick, and Tao Pan. "Folding of noncoding RNAs during transcription facilitated by pausing-induced nonnative structures." In: *Proceedings of the National Academy of Sciences* 104.46 (2007), pp. 17995–18000. DOI: 10.1073/pnas.0705038104. eprint: <http://www.pnas.org/content/104/46/17995.full.pdf>. URL: <http://www.pnas.org/content/104/46/17995.abstract>.
- [67] A. Xayaphoummine, T. Bucher, F. Thalmann, and H. Isambert. "Prediction and statistics of pseudoknots in RNA structures using exactly clustered stochastic simulations." In: *Proceedings of the National Academy of Sciences* 100.26 (2003), pp. 15310–15315. DOI: 10.1073/pnas.2536430100. eprint: <http://www.pnas.org/content/100/26/15310.full.pdf>. URL: <http://www.pnas.org/content/100/26/15310.abstract>.
- [68] Peter Yakovchuk, Ekaterina Protozanova, and Maxim D. Frank-Kamenetskii. "Base-stacking and base-pairing contributions into thermal stability of the DNA double helix." In: *Nucleic Acids Research* 34.2 (2006), pp. 564–574. DOI: 10.1093/nar/gkj454. eprint: oup/backfile/content_public/journal/nar/34/2/10.1093_nar_gkj454/2/gkj454.pdf. URL: <http://dx.doi.org/10.1093/nar/gkj454>.
- [69] Yanofsky Charles. "Attenuation in the control of expression of bacterial operons." In: *Nature* 289.5800 (1981). 10.1038/289751a0, pp. 751–758. DOI: 10.1038/289751a0.
- [70] Jing Yun A. Zhu, Adi Steif, Jeff R. Proctor, and Irmtraud M. Meyer. "Transient RNA structure features are evolutionarily conserved and can be computationally predicted." In: *Nucleic Acids Research* 41.12 (2013), p. 6273. DOI: 10.1093/nar/gkt319. eprint: oup/backfile/content_public/journal/nar/41/12/10.1093_nar_gkt319/3/gkt319.pdf. URL: <http://dx.doi.org/10.1093/nar/gkt319>.
- [71] Zuker M and Stiegler P. "Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information." In: *Nucleic Acids Research* 9.1 (1981), pp. 133–148. ISSN: 0305-1048 1362-4962. URL: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC326673>.
- [72] Michael Zuker and David Sankoff. "RNA secondary structures and their prediction." In: *Bulletin of Mathematical Biology* 46.4 (1984), pp. 591–621. ISSN: 1522-9602. DOI: 10.1007/BF02459506. URL: <http://dx.doi.org/10.1007/BF02459506>.

ABSTRACT

The function of non-coding RNAs often depends on evolutionary conserved structures. Structure formation, however, is a dynamic, co-transcriptional, process. Recent experiments indicate that in addition to the final native state, transient structures can be functional and evolutionary conserved.

We investigated predicted folding trajectories of homologous RNAs to detect novel functionally important structures. Little is known about the conservation of transient structures and only few are experimentally confirmed. Using programs like *Kinwalker* and *Kinefold* we simulated kinetic folding trajectories and studied their conservation.

In the first step we built a dataset of predicted folding trajectories and experimentally confirmed transient structures, starting from the examples published in Zhu et al.. This dataset contains experimental data and simulated folding trajectories for 5 RNA families comprising 437 sequences. A semi-automated pipeline allows to continually update the database.

Our analysis showed that most experimentally known transient structures are indeed conserved. In addition we predicted new conserved folding intermediates. The dataset was used to compare existing kinetic RNA folding programs, to investigate what factors are essential for successful modeling of the folding kinetics.

Further we developed a novel algorithm for aligning folding trajectories. This trajectory alignments show that homologous RNAs fold along highly similar pathways that converge strongly at functional intermediates.

Finally, we developed a background model for the frequency of structure patterns in random sequences. To this end we designed a grammar that generates sequences containing a desired structure pattern and analyzed the asymptotics of the corresponding counting series via their generating functions.

ZUSAMMENFASSUNG

Die Funktion von nicht-codierenden RNA ist stark von ihrer evolutionär konservierten Struktur abhängig. Die Faltung der Struktur ist ein co-transkriptioneller, dynamischer Prozess. Experimente deuten darauf hin, dass nicht nur das Endprodukt der Faltung sondern auch transiente Strukturen funktional und daher konserviert sind.

Um neue funktionell wichtige Strukturen zu entdecken, untersuchten wir Konservierungsmustern in simulierten Faltungstrajektorien homologer RNAs. Über die Konservierung von transienten Strukturen ist bisher wenig bekannt und nur wenige solche Strukturen wurden experimentell nachgewiesen. Etablierte Programme wie Kinwalker und Kinefold ermöglichen uns allerdings, die kinetische Faltungstrajektorien zu simulieren und ihre Konservierung zu erforschen.

Als ersten Schritt erstellten wir einen Datensatz aus vorhergesagten Faltungstrajektorien und experimentell nachgewiesenen transienten Strukturen, basierend auf Beispielen aus Zhu et al.. Aktuell enthält dieser Datensatz Experimentaldaten und simulierte Faltungstrajektorien für fünf RNA-Familien (gesamt 437 Sequenzen). Weiters definierten wir einen teilautomatischen Prozess zur laufenden Aktualisierung der Datenbank.

Unsere Analyse bestätigt, dass die meisten experimentell nachgewiesenen Strukturen konserviert sind. Zusätzlich konnten wir neue konservierte intermediäre Strukturen vorhersagen. Der Datensatz wurde außerdem verwendet um Simulationsprogramme für kinetischer RNA-Faltung zu vergleichen und wichtige Faktoren für die erfolgreiche Modellierung der RNA-Faltung zu identifizieren.

Weiters entwarfen wir einen neuen Algorithmus zum Alignieren von Faltungstrajektorien. Diese Alignments zeigen, dass homologe RNAs ähnlichen Faltungspfaden folgen die an funktionell wichtigen intermediären Strukturen konvergieren.

Schlussendlich entwickelten wir ein Hintergrundmodell für die Frequenz von Strukturmustern in zufälligen Sequenzen. Dafür entwarfen wir eine Grammatik, die gewissen Strukturmotiven folgende Sequenzen generiert, und die Asymptotik der zugehörigen Folge von Zählkoeffizienten anhand ihrer erzeugenden Funktionen analysiert.