Albert-Ludwigs University of Freiburg

Master Thesis

Constrained RNA-RNA interaction prediction

Author:
Rick Gelhausen

Supervisor:
Dr. Martin Raden

Examiner:
Prof. Dr. Rolf Backofen
Prof. Ivo L. Hofacker (University of Vienna)

A thesis submitted in fulfillment of the requirements
for the degree of Master of Science
in the
Bioinformatics Group,
Department of Computer Science

Submitted on 1st June 2018

DECLARATION

I hereby declare, that I am the sole author and composer of my Thesis and that no other sources or learning aids, other than those listed, have been used. Furthermore, I declare that I have acknowledged the work of others by providing detailed references of said work. I hereby also declare, that my Thesis has not been prepared for another examination or assignment, either wholly or excerpts thereof.

Freiburg,	
Place, Date	Signature

First of all, I want to thank my supervisor Martin Raden for introducing me to IntaRNA and suggesting me this topic. Thanks for all the help and feedback that helped me improve this work.

I want to thank my parents who always support me.

I want to thank my brothers and all my friends who always keep me entertained.

A special thanks goes to André and Frank for poof-reading and to Kristin who designed the awesome cover for my thesis.

Lastly, I want to thank my laptop for not crashing, even after running experiments for 3 weeks straight.

Abstract

Computational prediction of RNA-RNA interactions has become a fast growing topic over the last few years. Many different RNA-RNA interaction prediction tools have been developed, which all use vastly varying methods, ranging from alignment methods over minimum free energy methods to machine learning approaches. None of these tools seem to consider the steric 3D constraints of RNA molecules. These constraints could implicate that short intermolecular helices are more likely due to the restricted unpaired regions in stem loops.

This thesis introduces constraints on the intermolecular helix lengths for the prediction tool IntaRNA to improve the overall prediction quality. To understand the composition of RNA structures, more than 3,000 known RNA secondary structures, of any type and organism, were analysed in this thesis. From this analysis, the distribution of helix lengths were learnt, in order to find good starting parameters for the helix length constraints.

In order to evaluate the performance of the developed constraints, I created the IntaRNA benchmark, which is used to compare to the original IntaRNA predictors.

The experiments showed that the new constraint helped to improve the prediction quality of IntaRNA, while reducing the runtime. Further, the results suggest that restricting helices too much has negative effects on the performance.

Zusammenfassung

Die computer-gestützte Vorhersage von RNA-RNA Interaktionen ist, in den letzten Jahren, zu einer stark wachsenden Thematik geworden. Daher werden immer mehr RNA-RNA Interaktionsvorhersage-Tools entwickelt. Diese Tools verwenden unterschiedliche Methoden um Interaktionen vorherzusagen. Darunter *alignment* Methoden, *minimum free energy* Methoden und neuerdings auch *machine learning* Ansätze.
Keines dieser Tools scheint jedoch die sterischen 3D-Beschränkungen von RNA-Molekülen zu berücksichtigen. Genau diese Beschränkungen könnten allerdings bedeuten, dass kurze intermolekulare Helizes wahrscheinlicher sind, da ungepaarte Regionen in sogenannten *stem loops* aufgrund ihrer dreidimensionalen Verdrehung längenbeschränkt sind.
In dieser Arbeit stelle ich neue Methoden zur Einschränkung der intermolekularen Helixlängen für IntaRNA vor. Um die Zusammensetzung von RNA-Strukturen zu verstehen, habe ich über 3000 bekannte RNA-Sekundärstrukturen unterschiedlicher Typen und Organismus analysiert. Dies gab mir eine Intuition zur Verteilung der Helixlängen. Dadurch konnte ich gute Startparameter für die Helixlängen-Beschränkung ermitteln. Um die Qualität der neuen *prediction modi* zu bewerten, habe ich die IntaRNA-Benchmark erstellt. Diese erlaubt mir Vergleiche zwischen neuen und alten Modi zu ziehen.
Die Experimente haben gezeigt dass die neuen Methoden zur Einschränkung der intermolekularen Helixlängen die Vorhersagequalität von IntaRNA verbessern. Dabei wird gleichzeitig auch die Laufzeit reduziert. Außerdem zeigten die Resultate dass zu kleine Helizes einen negativen Effekt auf die Leistung von IntaRNA haben.

Contents

Chapter 1

Introduction

As a result of newly developed methods, a large amount of new RNA-based regulators have been experimentally discovered over the last few years. Even though it is possible to retrieve regulatory targets for these regulators experimentally in a wet lab, it is a tedious work. Due to the ever increasing amount of discovered RNA-based regulators, computer-driven RNA-RNA interaction prediction algorithms have been developed to support researchers. One such tool is IntaRNA (Busch et al., 2008; Richter, 2012), which was developed by Prof. Backofens bioinformatics group at the University of Freiburg. IntaRNA is an efficient RNA-RNA interaction prediction tool that incorporates both the accessibility of interaction sites and a user-definable seed region.
Currently, the team around Prof. Backofen is working on a reimplementation of IntaRNA. This new implementation, IntaRNAv2 (Mann et al., 2017), is faster and easier to extend. This allows the introduction and development of new constraints and prediction-modi to further enhance the prediction quality. For the rest of this work, IntaRNA refers to the new version.
In this thesis, I will introduce a new constraint on the length of intermolecular helices of RNA-RNA interactions. This might sound counter-intuitive at first, as stackings are the most stable and therefore energetically most favourable structural elements. The idea is that the steric constraints of the tertiary structure of RNA molecules hinder the formation of long intermolecular helices. This leads me to believe that many of such long helices are artefacts of RNA-RNA interaction prediction tools. As a result, I want to determine whether a constraint on the helix lengths helps to improve the results of RNA-RNA prediction tools such as IntaRNA.
To this end, I will first analyse RNAStrand, the RNA Structure and statistical Analysis Database (Andronescu et al., 2008), to get an overview of the distribution of helices in known RNA structures. Therefore, I will determine the distribution of helix lengths, i.e. the number of consecutive base pairs forming the helix, in all structures, including pseudo-knotted structures. Additionally I will analyse unpaired substructures. Built on the insights of this analysis, I will then introduce a new prediction method that incorporates constraints on intermolecular helix lengths.
I differentiate between two different approaches of incorporating the constraints. First, I consider only tightly stacked helices, allowing no bulges or interior loops. Second, I allow limited bulge and -interior loops inside the helices. For both methods, original

IntaRNA-like seed constraints are integrated.
Finally, I introduce the IntaRNA benchmark, which I created to evaluate the newly created predictors and compare them to the original IntaRNA recursions. The new benchmark runs user-definable configurations of IntaRNA on a large set of sRNA queries and mRNA targets, in order to determine how well a series of experimentally proven interactions is predicted.

1.1 Structure of the thesis

In the first chapter, the biological background and the underlying algorithmic details of IntaRNA are introduced. The second chapter details the analysis of RNAStrand and the introduction of the new limited helices constraints. Chapter three contains the results and the comparison against the original IntaRNA recursion. Chapter four lists related work. Chapter five gives an outlook on future work. The conclusion of the thesis is given in the final chapter.

1.2 Biological background

In this thesis, I will focus on Ribonucleic acids (RNA) which are transcribed from Deoxyribonucleic acid (DNA).
The RNA molecules are represented as a sequence $S \in \{A, C, G, U\}^*$, where A, C, G, U are the respective bases of the nucleotide chain, adenine (A), cytosine (C), guanine (G) and uracil (U).
There are two main classes of RNA, the coding RNA (cRNA) and the non-coding RNA (ncRNA). The cRNAs are involved in the translation process, where the RNA encodes for proteins. The composition of an RNA sequence is especially important for cRNAs. Whereas, the ncRNAs perform different functions, like the regulation of gene expressions and are (typically) not involved in the translation into proteins.
An example for ncRNAs are bacterial small RNAs (sRNA). These are highly structured small-chained non-coding RNAs. They can have many different functions, such as the modification of the function of proteins or the regulation of gene creation by binding to messenger RNA (mRNA). IntaRNA was designed to predict the interaction between sRNA queries and mRNA targets, but can also be applied to other RNA types.
RNA sequences fold into structures that determine the function of an RNA molecule, which is especially important for ncRNAs. These structures are created when bases form base pairs via hydrogen bonds. Due to their high binding strength, the Watson-Crick base pairs $G-C$ and $A-U$ as well as the wobble base pair $G-U$ are considered.
Two interacting bases that belong to the same RNA molecule form *intra*molecular structures, as seen in Figure 1.1. On the other hand, if the two paired bases belong to different RNA molecules, they form *inter*molecular structures. RNA-RNA interaction prediction aims at predicting these intermolecular structures between two RNA molecules, which is an extremely important step in understanding the function of ncRNAs. Nevertheless, Intra- and intermolecular structures are not mutually exclusive. A model interaction is shown in Figure 1.2.

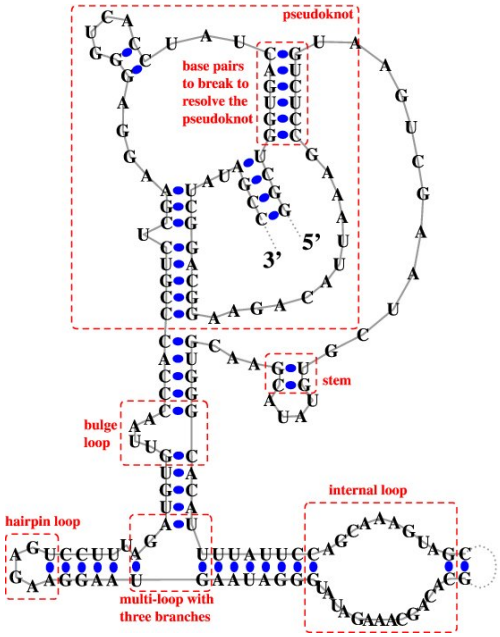


Figure 1.1: *Secondary structure for the RNase P RNA molecule of Methanococcus marapaludis from the RNase P Database (Brown et al., 1994); Red boxes mark structural features, such as stackings (stem), bulges, hairpin-loops, interior-loops, multi-loops and pseudo-knotted structures. This Figure was taken from the RNAStrand webpage. (Andronescu et al., 2008)*

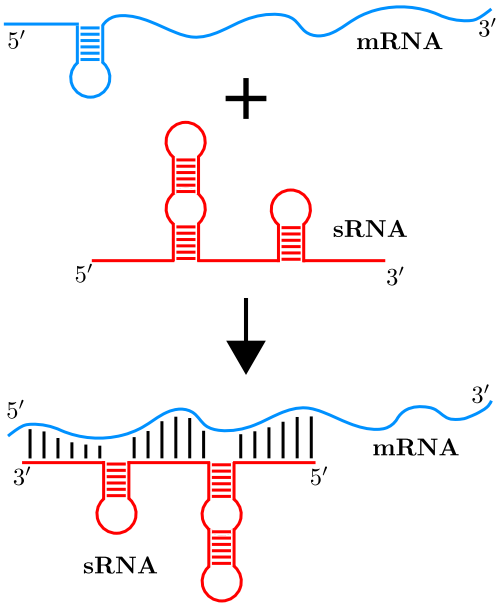


Figure 1.2: *Model interaction between an sRNA query and an mRNA target. Inspired by a figure from Vazquez-Anderson and Contreras (2013).*

Explicit intramolecular structures are not considered by IntaRNA. IntaRNA instead applies an accessibility measure to improve its results, which will be described in more detail in the IntaRNA section.
Formally, an RNA secondary structure P of S is a set of base pairs:

$$P \subseteq \{(i,j) \mid 1 \leq i < j \leq n, S_i \text{ and } S_j \text{ complementary}\},$$

where n = $\mid S \mid$ and for all (i,j), $(i',j') \in P$:

$$(i = i' \Leftrightarrow j = j') \text{ and } i \neq j'.$$

In reality, an RNA molecule has a 3-dimensional structure. Due to the high complexity of predicting these tertiary structures, only secondary structures are considered in this thesis. Further, different types of RNA secondary structures exist, namely nested and crossing structures. Crossing structures contain pseudo-knots, where two structure parts overlap, as shown in Figure 1.3. Pseudo-knots are no real knots when considering tertiary structure, but resemble knot-like shapes when depicting the secondary structure. The algorithmic complexity of the structure prediction increases with the complexity of the allowed pseudo-knot types (Condon et al., 2004). Therefore, pseudo-knots are not considered in IntaRNA.

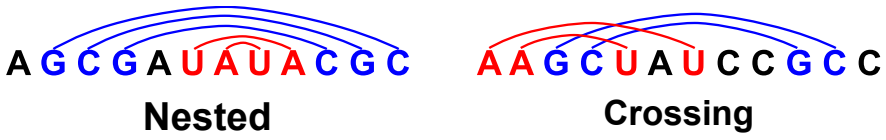


Figure 1.3: *Linear Feynman Diagrams of a nested and a crossing structure. The arcs represent base pairs.*

Typically, there is a distinction between multiple structural elements, based on the relation of base pairs and unpaired bases. These elements are hairpin-loops, stackings, internal/bulge loops and multi-loops. Secondary structures can be decomposed into these structural elements. Examples for intramolecular structure are shown in Figure 1.1. A common way to visualise structures is the so called dot bracket notation, where each base pair is represented by an opening and closing bracket, and unpaired bases are represented with a point.

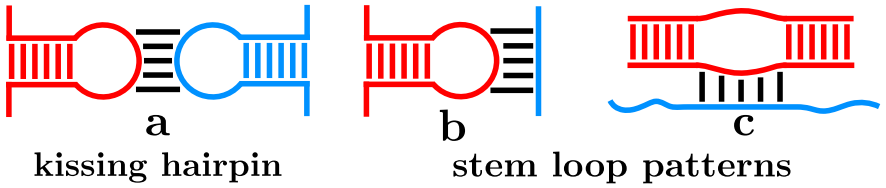


Figure 1.4: *Model of several special interaction cases. (a) kissing hairpin loops, (b) interaction with the unpaired region of a hairpin loop and (c) interaction with the unpaired region of an interior loop.*

It is important to note that even though secondary structures are used to reduce the complexity of the problem, the aim is to find biologically correct interactions.

Therefore, the folding into tertiary structures has to be taken into account. Figure 1.4 shows interactions, especially the kissing complex of hairpin loops, that are attributed to the tertiary structure folding. These interactions are subject to steric 3D constraints (Tinoco and Bustamante, 1999), which limit the size of their according binding regions. This means that the unpaired regions of hairpin loops, and likely internal loops, do not become arbitrarily large. Consequently, I assume that constrained intermolecular helix lengths better reproduce the biological conditions, leading to an increase in prediction quality.
Before further analysing this new constraint, I will first give an introduction to IntaRNA, which I use in the following to implement and test my helix length constraint. As IntaRNA uses energy minimisation to find the optimal RNA-RNA interactions, I begin with the energy model, followed by an introduction to structure probability computation, which are used by IntaRNA.

1.3 Energy computation and Nearest Neighbor Model

The energy minimised by IntaRNA is the free energy. The free energy can be seen as the amount of energy stored in an RNA structure. A positive term provides energy (e.g in the form of heat), whereas a negative energy term describes the amount of energy that is needed to dissolve all base pairs of the RNA structure. This means that the lower the free energy, the more energy has to be applied to disrupt the system. Therefore, a structure is more stable, the lower the free energy. The stablest structure is, consequently, the minimum free energy (mfe) structure. As the free energy is hard to calculate, the usage of energy differences is customary. The free energy is calculated with respect to the unstructured open chain, i.e. the structure containing no base pairs.
For IntaRNA, the underlying energy model to estimate the free energy of a given RNA secondary structure is the Nearest Neighbor Model (Borer et al., 1974). DeVoe and Tinoco (1962) discovered that vertical stackings of bases contribute the most to RNA helix stability. Therefore, directly neighboured bases have to be taken into account when estimating the energy contribution of a base pair. This leads to the introduction of the Nearest Neighbor Model.
The Nearest Neighbor Model uses a loop-based structure decomposition, e.g. the structural elements introduced earlier (Figure 1.1), to create an energy estimate. It focuses on the energy contributions of base pair neighbours. Thereby, only stackings to the enclosed base pairs are considered in order to avoid duplication of energy contributions for stackings. Further, only the unpaired bases adjacent to base pairs are taken into account, as stackings of unpaired bases are less predictable and stable.
The *terminal mismatch pairs* describe the first unpaired bases that follow a stacking, e.g. in a hairpin loop. They contribute to the energy of the system. A similar energy contribution exists for unpaired bases in bulge or internal loops.
Energy contributions for external base pairs, that are not enclosed by any other base pairs, are called *dangling end contributions*.
The energy $E(P)$ of a nested secondary structure P can be estimated by the sum of all loop contributions, for a given loop decomposition (see Figure 1.5).

$$E(P) = \sum_{(i,j)\in P} \begin{cases} e^H(i,j) & : \text{if hairpin loop} \\ e^{SBI}(i,j,k,l) & : \text{if stack, bulge, or internal loop} \\ e^M(i,j,x,x') & : \text{if multi-loop} \end{cases},$$

where e^H, e^{SBI} and e^M provide the context-sensitive energy contributions of hairpin, stack, bulge, internal loops and multi-loops, respectively. (k,l) describes the enclosed base pair for e^{SBI}. x and x' represent the number of unpaired bases and the number of enclosed helices for e^M, respectively.
e^{SBI} is defined as:

$$e^{SBI} = \begin{cases} e^S(i,j,i+1,j-1) & : \text{if stack} \\ e^B(i,j,i+1,j') \text{ or } e^B(i,j,i',j-1) & : \text{if bulge} \\ e^I(i,j,i',j') & : \text{if internal loop} \end{cases},$$

where an internal loop (i,j,i',j') has to fulfil the following conditions:

- $i < i' < j' < j$,
- $(i'-i)+(j-j') > 2$,
- there is no base pair enclosed between (i,j) and (i',j'),

and the bulge case is a special one-sided internal loop.
The exponential number of possible multi-loop combinations demand the usage of an energy estimate using the following formula

$$e^M(i,j,x,x') = e_a^M + e_b^M x + e_c^M x',$$

where e_a^M is a pseudo-energy parameter scoring the multi-loop closing base pair (i,j), e_b^M is the penalty for the enclosed unpaired bases x and e_c^M scores the number of enclosed helices x'. These parameters are derived from experimental data.

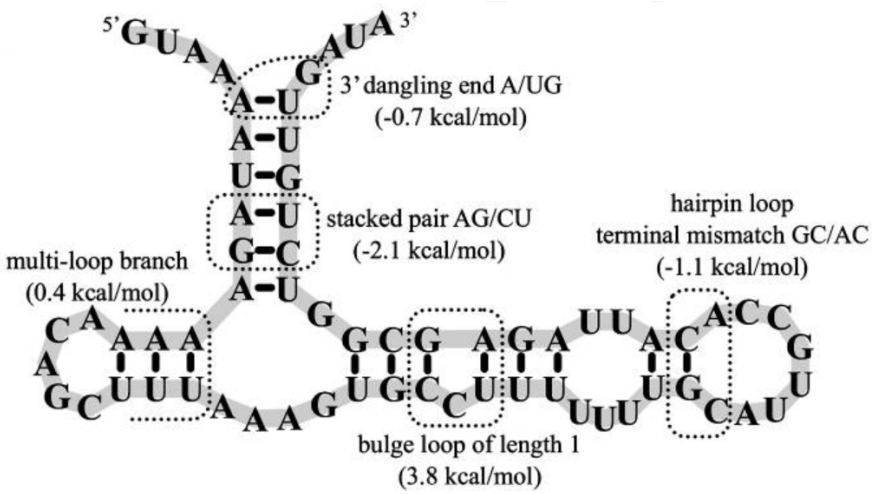


Figure 1.5: *The energy contributions for different structural elements. This Figure was taken from (Andronescu et al., 2010)*

There are multiple sets of these energy contributions available for use. Among the most used energy parameters are the Turner parameters (Mathews et al., 1999), also applied for IntaRNA. They can be found in the Nearest Neighbor Data Base (NNDB) (Turner and Mathews, 2010).
All energy terms in the Nearest Neighbor Model are dependant on temperature and ionic conditions.
As mentioned before, the stablest structure is the structure with the lowest energy, i.e the mfe structure. The stablest structure is, in general, considered to be the functional fold, i.e the structure that fulfils the function of an RNA molecule, which is the anticipated result. Prediction tools for intramolecular structures include among others, UNAfold (Markham and Zuker, 2008) and the Vienna Package (Lorenz et al., 2011).

1.4 Probabilities and McCaskill algorithm

The free energy is used to calculate, among others, structure probabilities, i.e. how likely it is for a certain structure to be formed. In order to determine these probabilities, an according probability distribution is required.
The Boltzmann distribution is, according to the principal of maximum entropy (Jaynes, 1957), the best probability distribution for the calculation of structure or base pair probabilities. It gives us a huge information gain, with a low information content. Therefore, probabilities are calculated according to their Boltzmann weights.

$$w(P) = \exp\left(\frac{-E(P)}{RT}\right),$$

where E is a specific energy of a structure P, R is the gas constant used to calculate the energy for a single molecule and T is the temperature.
Using these Boltzmann weights, the partition function Z can be calculated. Z is the sum over all Boltzmann weights for all structures of a given set $\mathcal{P}$, where the latter is also referred to as *structural ensemble*.

$$Z = \sum_{P \in \mathcal{P}} w(P)$$

Z is required in the calculation of structure and base pair probabilities. These probabilities are calculated for the thermodynamic equilibrium. This means that there are no observable changes on a macroscopic level.
The probability of a structure P, within a given structural ensemble $\mathcal{P}$, can be computed by

$$Pr[P|\mathcal{P}] = \frac{w(P)}{Z},$$

As the underlying energy model is an estimation and simplification of the truth, the structure with the highest probability is not necessarily the functional structure, i.e. the biologically correct one. However, it is safe to assume that the functional structure is among the most probable structures.

It is also possible to calculate the probability that a specific base pair appears.

$$Pr[(i,j)|\mathcal{P}] = \sum_{\substack{P \in \mathcal{P} \\ P \ni (i,j)}} \frac{w(P)}{Z}$$

where $Pr[(i,j)|\mathcal{P}]$ is the probability that base pair (i,j) occurs, given $\mathcal{P}$. These base pair probabilities can be represented in a dot plot and give a good overview what the most probable structure could look like, since the most probable base pairs are likely contained in the most probable structures.
Furthermore, the probability that a given region, from position i to j, of a structure is unpaired can be determined by

$$Pr_u[i,j] = \frac{Z^u_{i,j}}{Z},$$

where $Z^u_{i,j}$ is the partition function of all structures with subsequence $[i,j]$ unpaired, i.e.

$$Z^u_{i,j} = \sum_{P \subset \mathcal{P}^u_{i,j}} w(P) = Z(\mathcal{P}^u_{i,j}),$$

where $\mathcal{P}^u_{i,j}$ is the ensemble of all structures that are unpaired between i and j, i.e.

$$\mathcal{P}^u_{i,j} = \{P \mid \nexists (k,l) \in P : i \leq k \leq j \text{ or } i \leq l \leq j\} \subseteq \mathcal{P}_{\text{all}},$$

where $\mathcal{P}_{\text{all}}$ is the ensemble of all structures that can be formed from a sequence.
The unpaired probability is very important, as it allows the calculation of the accessibility of single stranded regions (Mückstein et al., 2006), which is one of the main features of IntaRNA.
In the following section, I will thoroughly explain how the different probabilities are calculated using the McCaskill algorithm.

1.4.1 McCaskill

Preliminaries

The McCaskill algorithm (McCaskill, 1990) is used to calculate the partition function Z for a given sequence S, which can be used to compute probabilities. The Figures and recursions in this section were inspired by the lecture material of RNA bioinformatics lecture (Raden and Backofen, 2018).
The basic idea is to use an algorithm, similar to the Zuker algorithm (Zuker and Stiegler, 1981), to sum up the Boltzmann weights for all possible structures. The important part is to count every structure only once, which requires the creation of a special multi-loop handling.

There are four matrices required in the algorithm:

$$Q_{i,j} = Z_{\mathcal{P}_{i,j}} \qquad Q^m_{i,j} = Z^m_{\mathcal{P}^{1bd}_{i,j}}$$
$$Q^b_{i,j} = Z^b_{\mathcal{P}_{i,j}} \qquad Q^{m1}_{i,j} = Z^m_{\{P \in P^{1bd}_{i,j} | \text{only one exterior base pair in P}\}}$$

$Q_{i,j}$ contains the summed Boltzmann weights for all structures which only contain bonds in range $[i, j]$. $Q^b_{i,j}$ has the additional property that (i, j) forms a base pair. $Q^m_{i,j}$ requires at least one base pair in range $[i, j]$. $Q^{m1}_{i,j}$ enforces exactly one exterior base pair within (i, j), i.e a base pair that is not enclosed by any other base pair. The final result $Z = Z_{\mathcal{P}_{all}}$ is contained in $Q_{1,|N|}$.

Matrices

The energy contributions used in this section were introduced in section 1.3.
The calculation of the Q_{ij} matrix (see Equation 1.1 below) is sketched in Figure 1.6.

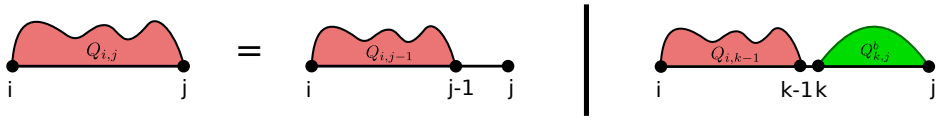


Figure 1.6: *Sketch of the Q_{ij} matrix computation.*

$$Q_{i,j} = Q_{i,j-1} + \sum_{i \le k < j} Q_{i,k-1} \cdot Q^b_{k,j}, \tag{1.1}$$

where Q^b_{ij} is defined in Equation 1.2 and sketched in Figure 1.7.

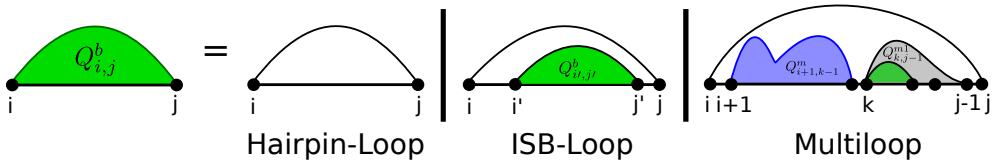


Figure 1.7: *Sketch of the Q^b_{ij} matrix calculation.*

$$Q^b_{ij} = \sum \begin{cases} \exp\left(\frac{-e^H(i,j)}{RT}\right) \\ \sum_{i<i'<j'<j} \left(Q^b_{i',j'} \cdot \exp\left(\frac{-e^{SBI}(i,j,i',j')}{RT}\right)\right) \\ \sum_{i<k<j} \left(Q^m_{i+1,k-1} \cdot Q^{m1}_{k,j-1} \cdot \exp\left(\frac{-e^M_a}{RT}\right)\right) \end{cases}, \tag{1.2}$$

where the energy contributions for the different possible structure elements are added up. If i and j do not form a base pair, then $Q^b_{ij} = 0$. The computation of the Q^m_{ij} (1.3) and Q^{m1}_{ij} (1.4) matrices are sketched in Figure 1.8 and 1.9, respectively.

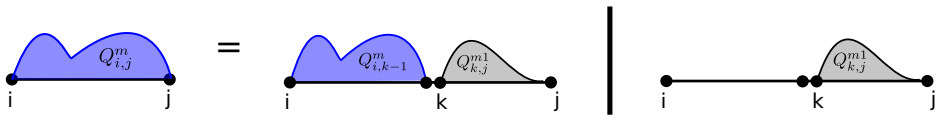


Figure 1.8: *Sketch of the Q^m_{ij} matrix calculation.*

$$Q^m_{i,j} = \sum_{i \leq k < j} \left(Q^m_{i,k-1} + \exp\left(\frac{-(k-i)e^M_c}{RT} \right) \right) \cdot Q^{m1}_{kj} \tag{1.3}$$

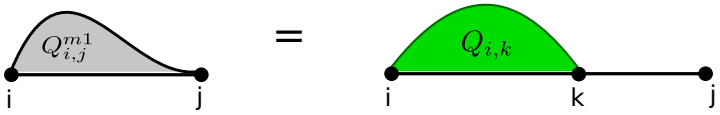


Figure 1.9: *Sketch of the Q^{m1}_{ij} matrix calculation.*

$$Q^{m1}_{ij} = \sum_{i < k \leq j} Q^b_{i,k} \cdot \exp\left(\frac{-e^M_b}{RT} \right) \cdot \exp\left(\frac{-(j-k)e^M_c}{RT} \right) \tag{1.4}$$

The matrices $Q^b_{i,j}$, $Q^m_{i,j}$ and $Q^{m1}_{i,j}$ are initialized with 0. For the single-stranded sequence there is no base, therefore, there will not be an entry in these matrices. $Q_{i,j}$ is initialized with 1 as it covers the single-stranded sequence.

Base pair probabilities:

The probabilities for single base pairs can be calculated using the McCaskill recursions. There are three possible locations for a base pair (k, l). The calculations are shown in Equation 1.5, 1.6 and 1.10 below.

1. (k, l) is an external base pair, as shown in Figure 1.10:

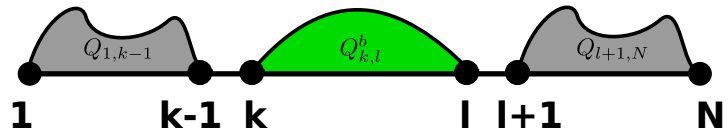


Figure 1.10: *Sketch of the external base pair case, with (k, l) being the external base pair.*

$$p^E_{kl} = \frac{Q_{1,k-1} \cdot Q^b_{k,l} \cdot Q_{l+1,n}}{Q_{1,n}} \tag{1.5}$$

2. (k,l) is the inner base pair of a stacking, bulge- or interior loop closed by base pair (i,j), where $i < k < l < j$, as sketched in Figure 1.11.

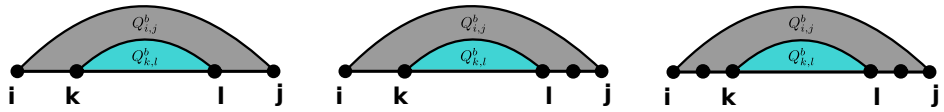


Figure 1.11: *Sketch of a base pair limiting a stacking, bulge or an interior loop.*

$$p_{kl}^{SBI}(i,j) = p_{ij} \frac{\exp\left(\frac{-e^{SBI}(i,j,k,l)}{RT}\right) Q^b_{k,l}}{Q^b_{i,j}} \tag{1.6}$$

The probability $p_{i,j}$ that base pair (i,j) is formed is computed in an outside recursion, before the computation of p_{kl}^{SBI}. $p_{i,j}$ is then corrected by taking the additional constraint that the loop i,j,k,l is formed. The numerator of the fraction is the partition function of all structures containing base pair (k,l). It is corrected by the denominator $Q^b_{i,j}$ which is the partition function of all structures containing base pair (i,j).

3. (k,l) closes an inner helix of a multi-loop closed by base pair (i,j), where $i < k < l < j$.

   $p^M_{kl}(i,j) = p_{ij} \cdot Pr[\text{Multiloop with inner base pair } (k,l) \text{ closed by } (i,j) \mid (i,j)]$

   There are again three locations for (k,l) inside the multi-loop:

   (a) (k,l) is the leftmost base pair, as shown in Figure 1.12:

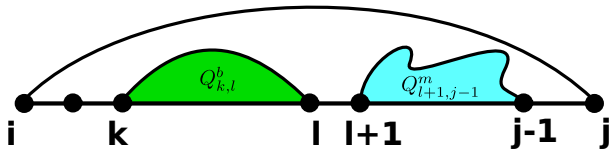


Figure 1.12: *Sketch of the multi-loop case where (k,l) is the leftmost base pair.*

$$\frac{Q^b_{k,l} \cdot Q^m_{l+1,j-1} \cdot \exp\left(\frac{-\left(e_a^M + e_b^M + (k-i-1)e_c^M\right)}{RT}\right)}{Q^b_{i,j}} \tag{1.7}$$

(b) (k, l) is the middle base pair, as shown in Figure 1.13:

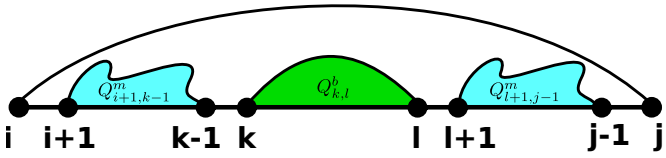


Figure 1.13: *Sketch of the multi-loop case where (k, l) is the middle base pair.*

$$\frac{Q^m_{i+1,k-1} \cdot Q^b_{k,l} \cdot Q^m_{l+1,j-1} \cdot \exp\left(\frac{-(e^M_a + e^M_b)}{RT}\right)}{Q^b_{i,j}} \tag{1.8}$$

(c) (k, l) is the rightmost base pair:

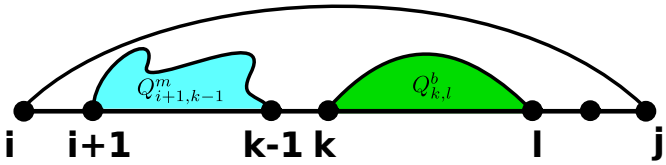


Figure 1.14: *Sketch of the multi-loop case where (k, l) is the rightmost base pair.*

$$\frac{Q^m_{i+1,k-1} \cdot Q^b_{k,l} \cdot \exp\left(\frac{-(e^M_a + e^M_b + (j-l-1)e^M_c)}{RT}\right)}{Q^b_{i,j}} \tag{1.9}$$

The probability for the multi-loop case is then formed by combining the previous three equations (Eq.1.7, Eq. 1.8 and Eq. 1.9)

$$\begin{aligned} p^M_{kl}(i,j) = \frac{p_{ij}}{Q^b_{i,j}} \cdot \Bigg(& Q^b_{k,l} \cdot Q^m_{l+1,j-1} \cdot \exp\left(\frac{-(e^M_a + e^M_b + (k-i-1)e^M_c)}{RT}\right) \\ & + Q^m_{i+1,k-1} \cdot Q^b_{k,l} \cdot Q^m_{l+1,j-1} \cdot \exp\left(\frac{-(e^M_a + e^M_b)}{RT}\right) \\ & + Q^m_{i+1,k-1} \cdot Q^b_{k,l} \cdot \exp\left(\frac{-(e^M_a + e^M_b + (j-l-1)e^M_c)}{RT}\right) \Bigg) \end{aligned} \tag{1.10}$$

The overall probability for a base pair (k, l) is denoted:

$$Pr[(i,j)|\mathcal{P}] = p^E_{kl} + \sum_{i<k,l<j} p^{SBI}_{kl}(i,j) + \sum_{i<k,l<j} p^M_{kl}(i,j)$$

Probabilities of unpaired regions:

A very important concept for RNA-RNA interaction prediction is the calculation of the probability of unpaired regions, which are possible targets for interactions.
These probabilities $Pr_u[i,j]$ can again be calculated using a variant of the McCaskill recursions (McCaskill, 1990), as introduced in (Mückstein et al., 2006).

$$Pr_u[i,j] = \frac{Z_{\mathcal{P}^u_{i,j}}}{Z_{\mathcal{P}_{all}}}$$

The probability that region $[i,j]$ is unpaired, where $\mathcal{P}^u_{i,j}$ is the set of structures where region $[i,j]$ is unpaired and $\mathcal{P}_{all}$ the set of all structures for a given sequence.
There are two different locations for an unpaired region. It is either exterior or enclosed by a base pair. An exterior region is enclosed by no base pairs. The enclosed region is either enclosed by a hairpin, an interior/bulge loop or a multi-loop. Using a disjoint decomposition of $\mathcal{P}^u_{i,j}$, the different cases can be viewed independently.

Case I $[i,j]$ is exterior:

$$Pr_u[i,j \mid exterior] = \frac{Q_{1,i-1} \cdot 1 \cdot Q_{j+1,N}}{Q_{1,N}} \tag{1.11}$$

where N is the length of the sequence and 1 is the Boltzmann weight of the unpaired region. The multiplication is justified as this is an independent decomposition of the sequence.

Case II $[i,j]$ is enclosed by base pair (p,q):

$$Pr_u[i,j \mid enclosed] = \sum_{p<i,j<q} \frac{Pr[(p,q)|\mathcal{P}]}{Q^b_{p,q}} \cdot Q^{pq}_{i,j} \tag{1.12}$$

where $Pr[(p,q)|\mathcal{P}]$ is the probability that (p,q) forms a base pair. $Q^b_{p,q}$ is the partition function of all structures enclosed by (p,q) and $Q^{pq}_{i,j}$ is the partition function of all structures that are unpaired in region $[i,j]$ enclosed by a base pair (p,q).

$Q^{pq}_{i,j}$ is computed by summing over the different cases of structural elements.

1. The Hairpin Loop case, as sketched in Figure 1.15:

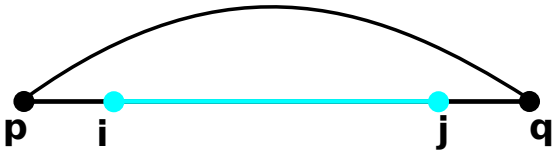


Figure 1.15: *Sketch of the hairpin loop case.*

$$V_H = \exp\left(\frac{-e^H(p,q)}{RT}\right) \tag{1.13}$$

2. The Stacking, Internal- and Bulge Loop cases, as sketched in Figure 1.16:

Figure 1.16: *Sketch of the stacking, bulge and interior loop case.*

$$V_{SBI} = \sum_{\substack{p<i\leq j<k \\ \text{or} \\ l<i\leq j<q}} \exp\left(\frac{-e^{SBI}(p,q,k,l)}{RT}\right) \cdot Q^b_{k,l} \tag{1.14}$$

3. The Multi-loop case, as sketched in Figure 1.17:

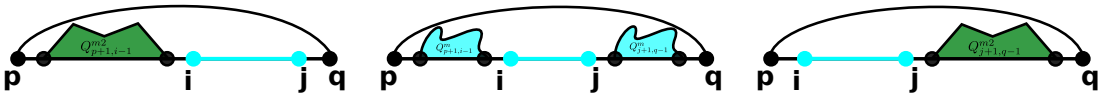


Figure 1.17: *Sketch of the multi loop cases.*

$$\begin{aligned} V_M = \sum_{p<i\leq j<q} \quad & Q^{m2}_{p+1,i-1} \cdot \exp\left(\frac{-(q-i)e^M_c}{RT}\right) \\ & + Q^m_{p+1,i-1} \cdot \exp\left(\frac{-(j-i+1)e^M_c}{RT}\right) \cdot Q^m_{j+1,q-1} \\ & + \exp\left(\frac{-(j-p)e^M_c}{RT}\right) \cdot Q^{m2}_{j+1,q-1} \\ \text{with } Q^{m2}_{i,j} = \sum_{p<k<q} & Q^m_{p,q} \cdot Q^{m1}_{k+1,q} \end{aligned} \tag{1.15}$$

where Q^{m2} ensures at least two helices.

$Q_{i,j}^{pq}$ is then computed by combining Equation 1.13, 1.14 and 1.15:

$$Q_{i,j}^{pq} = V_H + V_{SBI} + V_M$$

The probability $Pr_u[i,j]$, that $[i,j]$ is unpaired, is calculated by combining both cases (see Eq. 1.11 and Eq. 1.12):

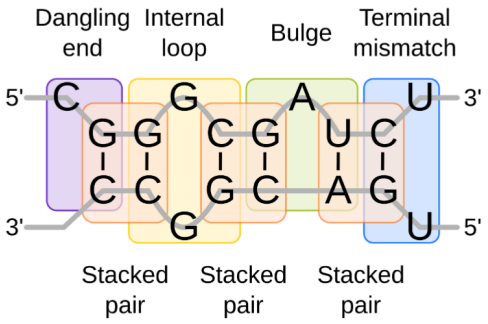
$$\begin{aligned} Pr_u[i,j] &= Pr_u[i,j \mid exterior] + Pr_u[i,j \mid enclosed] \\ &= \frac{Q_{1,i-1} \cdot 1 \cdot Q_{j+1,N}}{Q_{1,N}} + \sum_{p<i,j<q} \frac{Pr[(p,q)|\mathcal{P}]}{Q_{p,q}^{b}} \cdot Q_{i,j}^{pq} \end{aligned}$$

1.5 IntaRNA

IntaRNA is an RNA-RNA interaction prediction tool developed for the prediction of mRNA target sites for bacterial small regulatory RNAs. IntaRNA uses energy minimisation in order to find the optimal interaction. The energy minimised is composed of an accessibility and a hybridisation energy.

The main IntaRNA recursion is limiting the loop sizes in the interaction to 16 nucleotides and it allows no intramolecular base pairs in the interacting sub-sequences.

Figure 1.18 shows an example interaction for IntaRNA. It shows the main structural elements that are considered in IntaRNA, stackings, bulge and interior loops. Multiloops are not considered as they are introducing intramolecular structure into the interaction site. In addition, it gives an overview over the energy contributions of each structural element when using the nearest neighbor model with the parameters provided by Mathews et al. (1999). This includes the contributions for dangling ends, terminal mismatches as well as the intermolecular initiation energy, further called E_{init}.



Structural element	Nucleotides	Free energy [kcal/mol]
Dangling end	(C,GC)	−0.3
Stacked pair	(GC,GC)	−3.3
Internal loop	(GC,G|G,CG)	−2.1
Stacked pair	(CG,GC)	−2.4
Bulge	(GC,A,UA)	+1.6
Stacked pair	(UA,CG)	−2.4
Terminal mismatch	(CG,UU)	−1.2
Intermolecular initiation		+4.1
Overall free energy	$\sum$	−6.0

Figure 1.18: *Example of an interaction formed by two short RNA sequences, taken from (Richter, 2012). The table shows the energy contributions for each of the presented structural elements of the interaction site. The nearest neighbor model using the energy parameters by Mathews et al. (1999) is taken for calculating the energy values. These parameters were taken from the Nearest Neighbor Database (NNDB) (Turner and Mathews, 2010).*

In the following, I will first introduce the RNAup-like exact recursions (Mückstein et al., 2006) and then give an overview over the two main contributions of IntaRNA, the incorporation of a seed region and an heuristic version that greatly improves the runtime and memory consumption.

1.5.1 Exact Recursions

There are two major components in IntaRNA that determine the quality of RNA-RNA interactions between two sub-sequences of sequences S^1 and S^2, the hybridisation energy $H(i, j, k, l)$ and the accessibility of the interaction sites.
The hybridisation energy is calculated using the Nearest Neighbor Energy Model. It represents the hybridisation minimum free energy of two sub-sequences, where the leftmost positions of both sub-sequences form a base pair.
For simplification purposes, $E_{3'}^{dangle}$, $E_{5'}^{dangle}$ and E_{mm}^{term} will not be considered in the following recursions. Further, I will refer to the following recursions as the original IntaRNA recursions for the rest of my thesis.
For sub-sequences $S_i^1...S_k^1$ and $S_j^2...S_l^2$, where S^1 is ordered from $5'$ to $3'$ and S^2 in the reverse order:

$$H(i, j, k, l) = \min\{E(P) \mid (i, j) \in P \wedge (k, l) \in P\}$$

The hybridization energy is calculated with a Zuker-like recursion.

$$H(i,j,k,l) = \min \begin{cases} E_{init} \\ \quad : \text{if } (S_i^1, S_j^2) \text{ can pair, } i = k \text{ and } j = l, \\ \min\limits_{r,s}\{e^{SBI}(i,j,r,s) + H(r,s,k,l)\} \\ \quad : \text{if } (S_i^1, S_j^2) \text{ and } (S_k^1, S_l^2) \text{ can pair, } i \neq k \text{ and } j \neq l, \\ \infty \\ \quad : \text{otherwise.} \end{cases}$$

where e^{SBI} is the energy contribution for a stack, bulge or internal loop introduced in section 1.3.

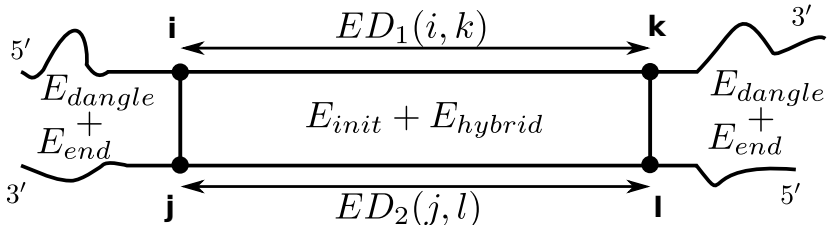


Figure 1.19: *The energy contributions needed in the IntaRNA recursions.*

The accessibility represents the energy required to make the interaction site single-stranded. It is calculated as the energy difference between the energy of the ensemble of all structures $\mathcal{P}$ that can be formed by S and the energy of the ensemble of all structures $\mathcal{P}^u$, where the interaction site is single-stranded.

This energy difference $ED(i,j)$ is computed using a partition function approach as introduced by (McCaskill, 1990).
The free energy of the ensemble $\mathcal{P}$ is:

$$E^{ens}(\mathcal{P}) = -RT \cdot ln(Z_{\mathcal{P}}),$$

It follows that $ED(i,k)$ is:

$$ED(i,k) = E^{ens}(\mathcal{P}^u_{i,k}) - E^{ens}(\mathcal{P}), \tag{1.16}$$

Both the accessibility and the hybridisation energy are combined to form the extended hybridisation energy. All needed energy contributions are sketched in Figure 1.19. The extended hybridisation energy of a specific hybridisation between $S^1_i...S^1_k$ and $S^2_j...S^2_l$ is defined by:

$$C(i,j,k,l) = \begin{cases} H(i,j,k,l) + ED_1(i,k) + ED_2(j,l) \\ \quad : \text{if } (S^1_i, S^2_j) \text{ and } (S^1_k, S^2_l) \text{ can pair, } i \neq k \text{ and } j \neq l, \\ \infty \\ \quad : \text{otherwise.} \end{cases} \tag{1.17}$$

1.5.2 seed interactions

A feature observed in certain RNA-RNA interactions are seed regions. A seed region is an interaction region with near perfect complementarity. Seed regions were discovered first for animal microRNAs (Isaac, 2005; Brennecke et al., 2005; Doench and Sharp, 2004). Later, it was discovered that this also applies to many bacterial sRNAs (Tjaden et al., 2006; Bouvier et al., 2008).
The incorporation of seed regions is the first improvement of IntaRNA over the exact RNAup recursions. A special constraint is added for the predicted interactions. Thus, at least one seed is required in an interaction site.
In order to implement seed regions into IntaRNA, Busch et al. (2008) introduced the following seed features that can be controlled by the user:

- B: the number of perfectly paired bases in the seed region,
- $b^{\max}, b^{\max}_m, b^{\max}_s$: the maximal number of unpaired bases in the seed region in both sequences, in the mRNA

An additional *seed* matrix is introduced to incorporate seed regions into IntaRNA. The minimal free energy of a hybridisation between sub-sequences $S^1_i...S^1_k$ and $S^2_j...S^2_l$ that include B' base pairs is expressed by $seed(i,j,k,l;B')$. Thus, the number of unpaired bases for the mRNA and the sRNA are determined by $k-i+1-B'$ and $l-j+1-B'$, respectively, and have to follow the according maximal values from above, i.e. $b^{\max}, b^{\max}_m$ and $b^{\max}_s$.

The *seed* matrix is defined as:

$$E_{seed}(i,j,k,l;B') = \begin{cases} \min\limits_{\substack{p,q \text{ with} \\ k-p+1 \geq B'-1 \\ l-q+1 \geq B'-1}} \begin{pmatrix} e^{SBI}(i,j,p,q) \\ +seed(p,q,k,l;B'-1) \end{pmatrix} \\ \quad : \text{if } (S_i^1, S_j^2) \text{ and } (S_k^1, S_l^1) \text{ can pair and } 2 < B' \leq B, \\ e^{SBI}(i,j,k,l) \\ \quad : \text{if } (S_i^1, S_j^2) \text{ and } (S_k^1, S_l^1) \text{ can pair and } B' = 2, \\ \infty \\ \quad : \text{otherwise.} \end{cases} \tag{1.18}$$

with $k-p+1 \geq B'-1$ and $l-q+1 \geq B'-1$ ensuring that $B'-1$ base pairs can be formed between both sub-sequences $S_p^1 ... S_k^1$ and $S_q^2 ... S_l^1$, respectively. Let $l_m = k-i+1$ and $l_s = l-j+1$ be the lengths of intervals $[i,k]$ and $[j,l]$. Then, $seed(i,j,k,l;B)$ is only valid if $l_m - B \leq b_m^{max}, l_s - B \leq b_s^{max}$ and $l_m + l_s - 2B \leq b^{max}$, ensuring the seed features listed above.

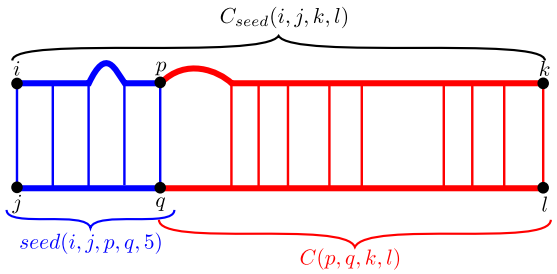


Figure 1.20: *The relation between matrix $C_{seed}(i,j,k,l)$, $seed(i,j,p,q,5)$ and $C(p,q,k,l)$. $seed(i,j,p,q,5)$ contains the minimal hybridisation energy of a seed region with five base pairs that is enclosed by base pairs (i,j) and (p,q). $C(p,q,k,l)$ contains the minimal extended hybridisation energy of sub-sequences $S_p^1 ... S_k^1$ and $S_q^2 ... S_l^2$. Inspired by a figure from (Richter, 2012).*

In order to find the optimal mfe interaction containing at least one seed, an additional matrix $C_{seed}(i,j,k,l)$, containing the energy scores of interactions with a seed region is introduced. It is sketched in Figure 1.20. The matrix is defined as:

$$C_{seed}(i,j,k,l) = \begin{cases} \min \begin{cases} \min\limits_{p,q} \begin{pmatrix} e^{SBI}(i,j,p,q) + C_{seed}(p,q,k,l) \\ -ED(p,k) - ED(q,l) \\ +ED(i,k) + ED(j,l) \end{pmatrix} \\ \min\limits_{\substack{p,q \text{ with} \\ l_m \leq b_m^{max}+B \\ l_s \leq b_s^{max}+B \\ l_m+l_s \leq b^{max}+2B}} \begin{pmatrix} seed(i,j,p,q;B) + C(p,q,k,l) \\ -ED(p,k) - ED(q,l) \\ +ED(i,k) + ED(j,l) \end{pmatrix} \end{cases} \\ \quad : \text{if } (S_i^1, S_j^2) \text{ and } (S_k^1, S_l^2) \text{ can pair}, i \neq k \text{ and } j \neq l \\ \infty \\ \quad : \text{otherwise.} \end{cases}$$

where the first part covers the case in which a seed region was already found right of (p,q). The second part represents the case, where no seed region was found right of (p,q). Instead a seed region is found between (i,j) and (p,q). When considering the example in Figure 1.20, the $seed(i,j,k,l;5)$ contains no accessibility contributions, whereas the $C(i,j,k,l)$ matrix contains the accessibility contribution. In order to ensure that the contributions are correct for the C_{seed} matrix, the ED values for the intervals $[p,k]$ and $[q,l]$ have to be replaced by the ED values for the entire region formed by the intervals $[i,k]$ and $[j,l]$. The accessibility values are not additive, therefore they have to be replaced.

The underlying implementation of IntaRNA uses a dynamic programming (DP) approach in order to realise the presented recursions. First, DP matrices are filled with the energy values resulting from the recursions. Then, a traceback technique is used in order to create the optimal interaction output.

1.5.3 Heuristic recursion

The exact recursions lead to a time and space complexity of $O(n^2m^2)$, when restricting the considered intermolecular interior loop length, with n and m being the lengths of the query and target sequence, respectively. Therefore, they are not applicable for genome-wide screens. IntaRNA introduces a heuristic to reduce both time and space complexity. This heuristic is based on the sparsification technique, which means that for matrix $C(i,j,k,l)$ many entries contain the same values. These values are often not used in following recursion steps. Therefore, the idea is to consider, for each interaction start i,j, only the optimal right interaction with boundaries k,l instead of all possible interaction ranges. This reduces the space complexity to $O(nm)$ as only the best value is stored for each (i,j) start. This reduction also applies to the time complexity, as not all possible ranges have to be considered any more.

The heuristic version of $C(i,j,k,l)$, $C(i,j)$ is defined as:

$$C(i,j) = \begin{cases} \min\limits_{p,q} \left(e^{SBI}(i,j,p,q) + C(p,q)\right) \\ \quad : \text{if } (S_i^1, S_j^2) \text{ can pair,} \\ \infty \\ \quad : \text{otherwise.} \end{cases}$$

Similar simplifications can be applied to the recursions that incorporate a seed region, effectively reducing their overall time and space complexity to $O(nm)$.
In the next chapter, I will introduce my own recursions. As the exact recursions are easier to understand and visualise, I formulated the exact RNAup-like recursions of my constraints. The same heuristic can be applied to my recursions. I implemented the heuristic versions in order to allow benchmarking on a large dataset. This will be discussed in more detail in the next chapter.

Chapter 2

Limited Stacking

In this thesis, I aim at extending IntaRNA by new prediction modi, which enforce a limit on the helix lengths, for intermolecular helices in the interaction region, to further improve the prediction quality. In order to find decent values for the helix length limit, I analysed known RNA structures.
To this end, the RNA secondary STRucture and statistical ANalysis Database (RNAStrand) (Andronescu et al., 2008) was used. At the time of this thesis, RNAStrand contained 4666 RNA secondary structures of various types and organisms. The data, used in this thesis, was downloaded in November 2017. The data in the RNAStrand database was assembled from other databases (Westbrook et al. (2003); Cannone et al. (2002); Andersen et al. (2006); Sprinzl and Vassilenko (2005); Brown (1999); Griffiths-Jones et al. (2005); Berman et al. (1992)). It is well suited to get an overview of the distribution of helix lengths and other useful statistics, such as the distribution of unpaired regions. RNAStrand offers dot bracket notations for the structure of every molecule in the database. These are single-molecule structures, i.e. intramolecular structures. The analysis of the structures was split into two parts, the analysis of helices and that of unpaired regions.
In the first part, the analysis of helices, two cases are distinguished, helices in all structures and helices in pseudo-knotted structures, called pseudo-knot helices in the following. The reasoning for the analysis of pseudo-knot helices, is related to an observation described by Thirumalai (1998). RNA molecules fold in two separate phases. Simply put, the nested RNA secondary structure forms first, followed by the folding of crossing secondary structure elements into the tertiary structure. The second step basically represents the formation of pseudo-knotted structures, called pseudo-knots. As pseudo-knots form after the creation of the nested secondary structure, they are subject to similar constraints as intermolecular stems, as the existing secondary structure impairs the size of possible interaction regions. Therefore, the pseudo-knot helices might give valuable information about intermolecular helices. The analysis of all structures provides the opportunity to compare them to the pseudo-knotted structures.
The second part, the analysis of unpaired regions, reveals regions that are free and can potentially interact with other RNA molecules. This analysis could provide information about possible seed regions, as explained in the IntaRNA section of the introduction.
Unfortunately, there are also several downsides to RNAStrand. The database is very

diverse, but several RNA types are overrepresented, therefore the statistics are biased, e.g. there are 726 transfer messenger RNAs compared to only 41 cis regulatory elements. This problem can be solved by normalising the data, as done in RNAStrands analysis tool. As the main purpose of my analysis is to get an overview of the distributions of helices in known RNA, I am not interested in specific values. My hope is that the general observations will remain roughly the same even when not normalising the data. Further, there are many sequences that contain special characters, other than the *AGCU* bases. To avoid any problems with faulty structures, I chose to omit these sequences and their respective structure in a preprocessing step. One exception is the "~" character. It only appears on unpaired positions, due to which the affected bases were removed rather than the whole structure. Moreover, two sequences in the database did not have the same length as their according dot-bracket notations and were removed as well. After both preprocessing steps, 3,313 of the initial 4,666 molecules were used for further analysis.

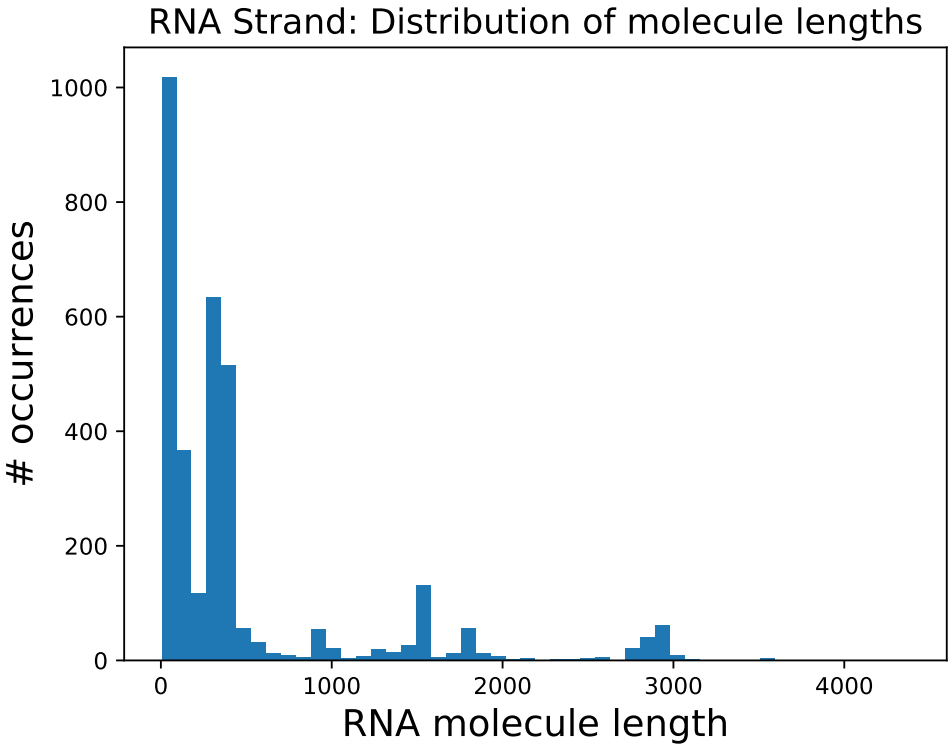


Figure 2.1: *The distribution of lengths in RNAStrand e.g. the number of occurrences of each molecule length within the database. The x-axis represents the different molecule lengths. The y-axis shows the number of occurrences.*

Figure 2.1 provides an overview of the distribution of molecule/structure lengths for all molecules in RNAStrand. Most structures are of short length, but there are also several long structures. This allows for a more general analysis of the helix lengths. For example, it allows to analyse whether helix lengths are dependent on sequence lengths.

2.1 Distribution of helices in known RNA structures

As already mentioned, the distribution of helices in known RNA structures will provide the necessary constraint parameters. Further, it might give an intuition whether the idea of limiting helix lengths is justified, i.e. there is a clear trend towards a certain helix length. First, the distribution of helix lengths in RNAStrand will be analysed, which should give an overview of the range of possible helix lengths. Then, both the number of helices and the maximum helix length are investigated for a more detailed understanding of the problem. During every step of this analysis, the according distributions for pseudo-knot helices will be considered in order to determine differences and similarities.
To start, I evaluated the distribution of helix lengths within the database. In order to do this, I created a script that reads all structures inside RNAStrand and returns for every molecule the different helix lengths it contains. Here, I defined a helix as follows. Let S be a fixed sequence. Let P be an RNA structure for S.
A base pair, $(i, j) \in P$ is part of a helix if there exists a base pair $(i', j') \in P$ such that:

- $i < i' < j' < j$
- $i' - i - 1 \leq b$
- $j - j' - 1 \leq b$

where b represents a user-defined, maximally allowed bulge-size. This is a generalisation, as it also allows short interior loop-like structures inside of a helix.
The pseudo-knot helices were collected using the same method. The only difference is that helices formed by "()" base pairs in the dot bracket notation were ignored. All other types of bracket notations describe different pseudo-knots.
Figure 2.2, shows the distribution of helix lengths in RNAStrand for different bulge-sizes. It was created by counting the occurrences of all helix lengths for all 3,313 molecules. It uses a logarithmic scale for the y-axis to give a more detailed overview. The results shown in the plots are as expected. Short helices are dominant, whereas long ones are very infrequent in comparison. The idea of limiting stack sizes looks promising, when regarding the large difference between the occurrence of long and short helices. Allowing different maximal bulge-sizes increases the diversity of admitted helices. Therefore, the distribution shifts slightly towards the longer helices, as anticipated.
The nature of my helix definition causes an even larger shift for increasing bulge-sizes, as interior-loops are also acceptable. Nevertheless, the overall trend remains the same. There is an increasing number of long helices, but the very short helices still dominate by far.
Figure 2.3 shows the occurrences of different helix lengths for pseudo-knot helices. When comparing them to overall helix lengths, one can see that pseudo-knot helices are very limited in size. As a single base pair is not regarded a helix, they range from 2 to 9 base pairs, where short helices are again more common than longer ones.
Observing both Figure 2.2 and Figure 2.3, the idea of limiting helix lengths seems viable, as the small helices dominate substantially. However, there are two properties that are not observable in these plots. First of all, structures have variable amounts of

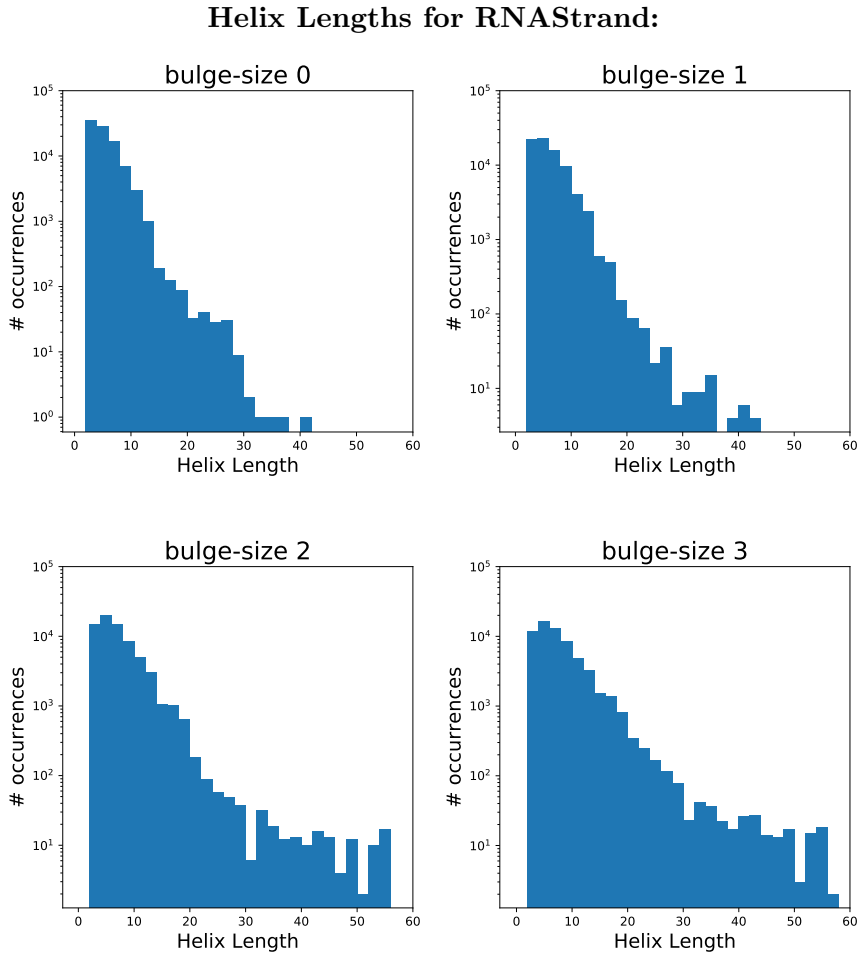


Figure 2.2: *The distribution of helix lengths in RNAStrand. The x-axis represents the different occurring helix lengths. The y-axis represents the number of occurrences for each helix length on a log scale.*

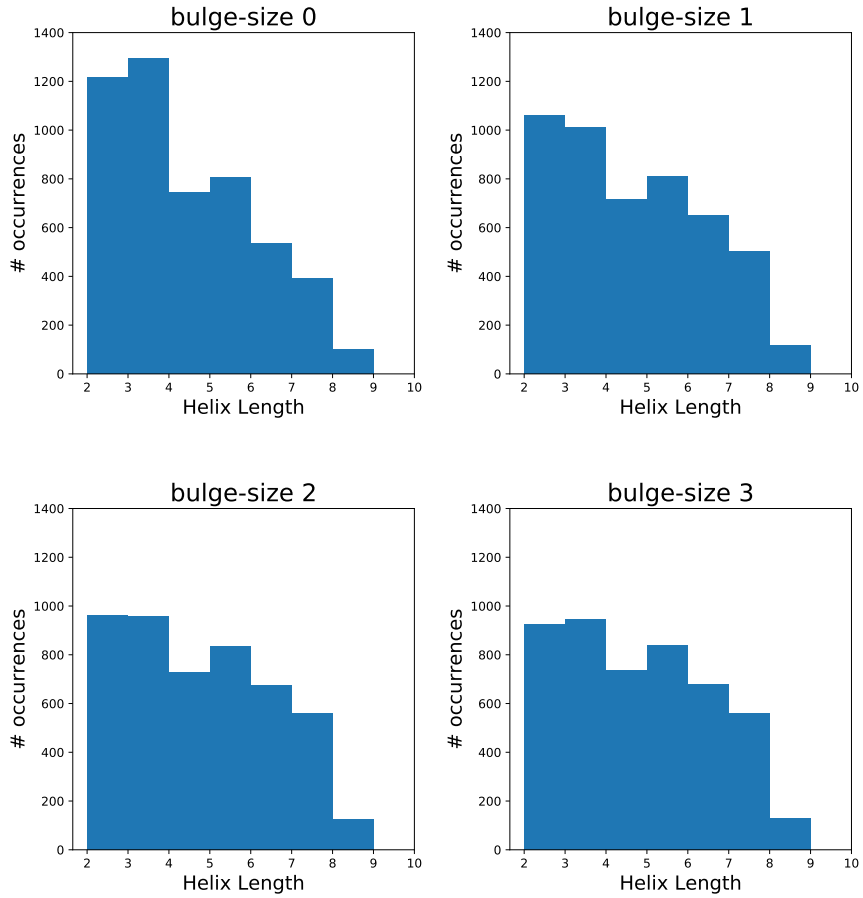


Figure 2.3: *The distribution of helix lengths belonging to pseudo-knots in RNAStrand for different bulge-sizes. The x-axis represents the different occurring helix lengths. The y-axis represents the number of occurrences for each helix length.*

helices, longer sequences likely allow more helices than short ones. Secondly, structures are not composed of a single helix length, they consist of many short helices and a few longer ones. Figure 2.4 gives an overview of the number of helices that occur in the database and how often each of them appears. Most structures contain only a small amount of helices and increasing bulge-sizes further reinforce this. The more bases in a bulge are allowed, the longer the helix lengths become and the fewer helices will fit into the structure.

Figure 2.5 shows the number of occurring helices for each molecule length in the database. As multiple lengths appear more than once, the arithmetic mean was taken. This Figure confirms what might seem obvious, the longer a structure, the more individual helices occur. This just represents the general trend, there are obviously longer

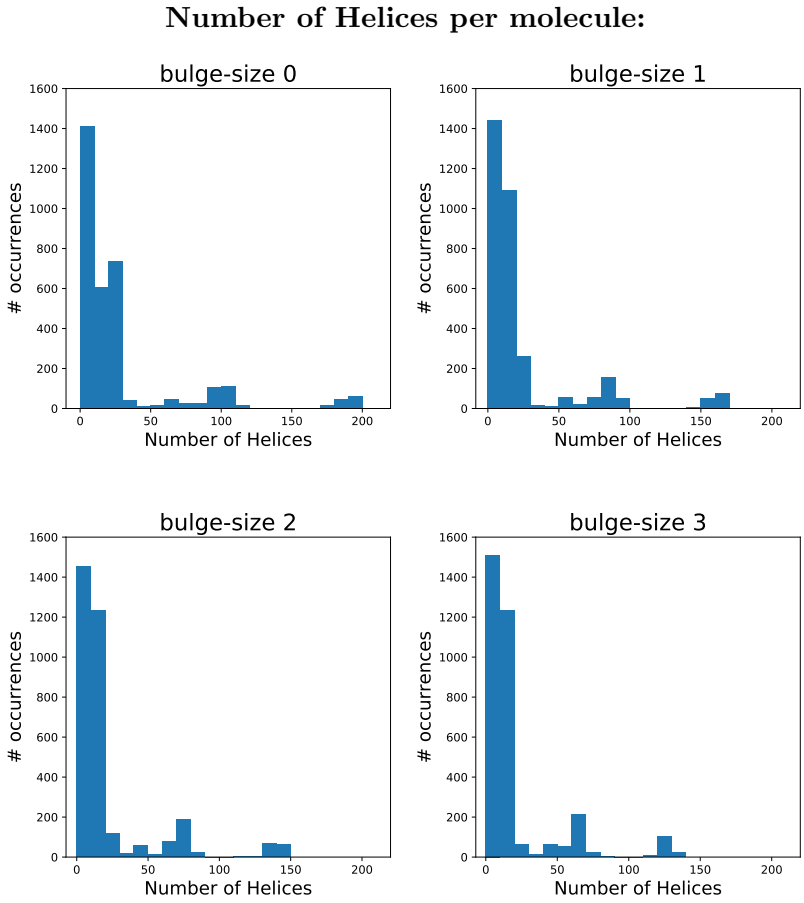


Figure 2.4: *The distribution of the number of helices for each molecule in RNAStrand for different bulge-sizes. The x-axis represents the number of helices. The y-axis represents the number of occurrences for each number of helices.*

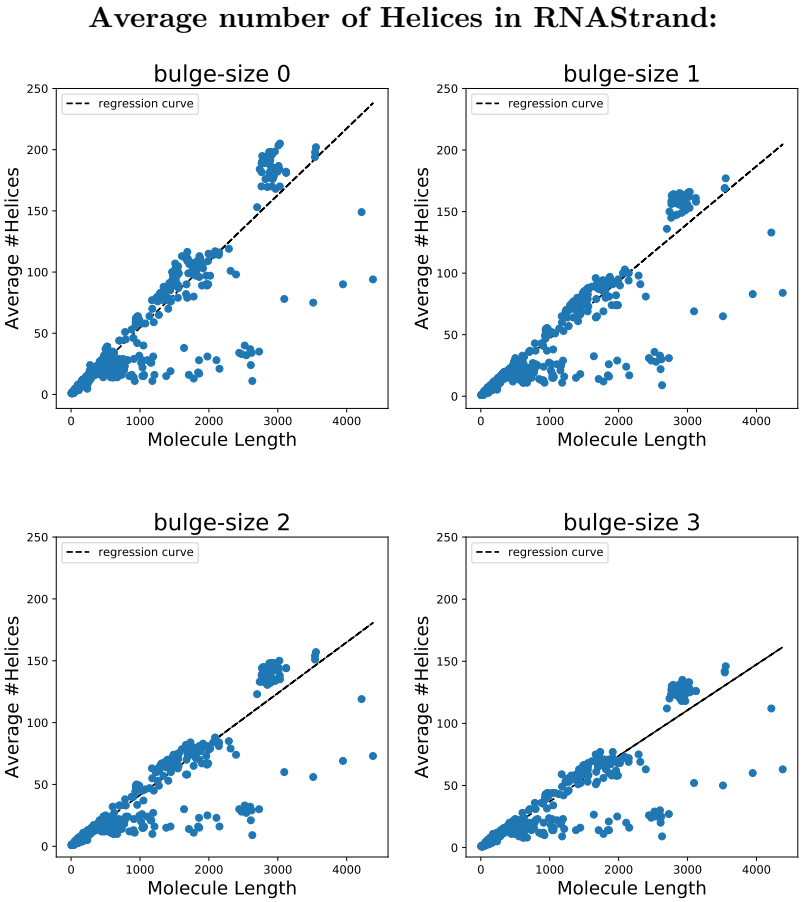


Figure 2.5: *The average number of helices per molecule length in RNAStrand for different bulge-sizes. The x-axis represents the different molecule lengths contained in the database. The y-axis represents the average number of helices contained in the structure of each molecule.*

structures with longer and fewer helices, containing long unpaired regions. Considering RNAStrand, molecules with a small amount of up to about 20 helices are dominant. Figure 2.6 and 2.7 show that most structures that contain pseudo-knots only contain a small number of them. Further, longer structures are more likely to have more pseudo-knotted structures.

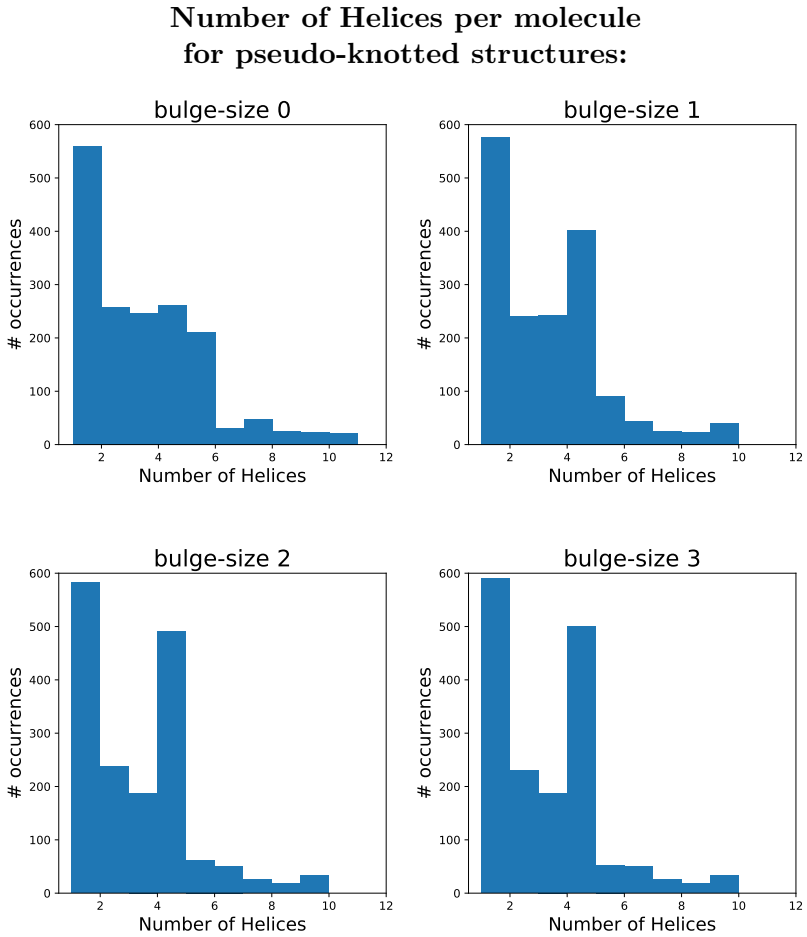


Figure 2.6: *The distribution of the number of helices for each molecule in RNAStrand for different bulge-sizes and for pseudo-knot helices. The x-axis represents the number of helices. The y-axis represents the number of occurrences for each number of helices.*

The more valuable information, regarding helix length constraints, is the maximum helix length. If the average helix length is low, there could still be single large helices in each structure. Limiting helix lengths too much would lead to mistakes in those cases.

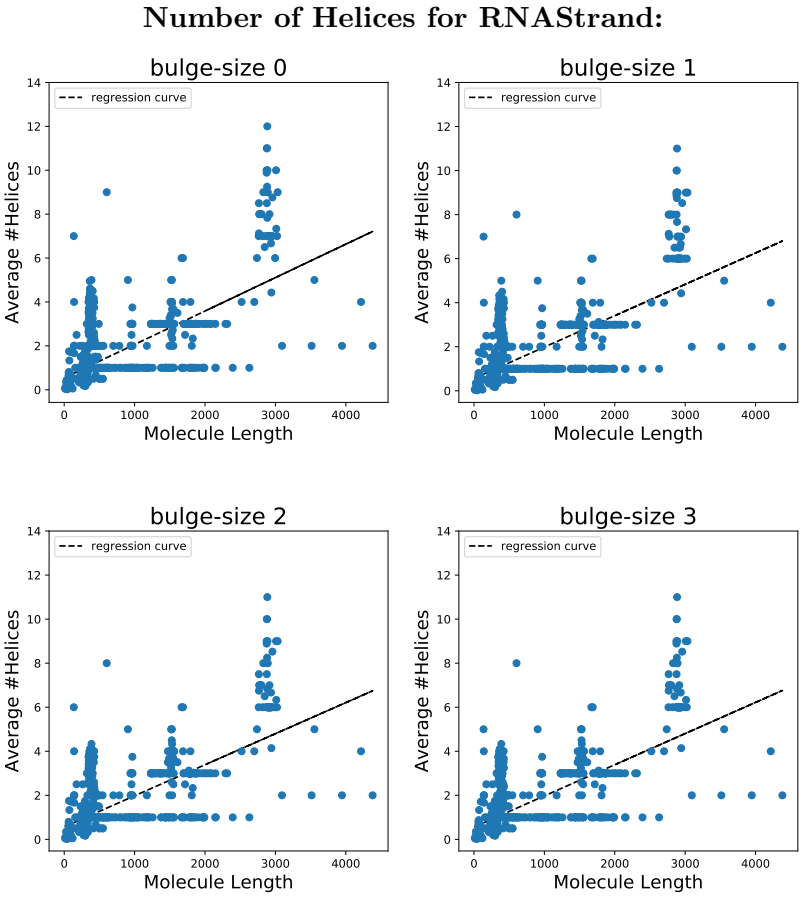


Figure 2.7: *The distribution of the number of helices for each molecule in RNAStrand for different bulge-sizes, where the helices belong to pseudo-knots. The x-axis represents the number of helices. The y-axis represents the number of occurrences for each number of helices.*

Figure 2.8 shows how often certain maximum helix lengths occur in RNAStrand. In order to create this plot, the maximum helix length was determined for each molecule in the database. Most structures seem to contain a helix length of around 6 to 12 base pairs. Raising bulge-sizes cause a notable increase in helix lengths. The number of maximum helix lengths in the range of 6 to 12 have already roughly halved for a bulge-size of 2. This has to be taken into consideration when allowing large bulge-sizes, but it could also be caused by the fact that I allow interior loops as well.

To see how the length of a molecule influences the maximum helix length, Figure 2.9 shows the maximum helix lengths for each molecule length. When multiple molecules have the same length, the overall maximum was considered. Despite many outliers, longer structures tend to have slightly larger maximum helix lengths. The red line represents the average helix length, which was calculated by adding up all helix lengths

of structures with the same length and dividing by the number of all helices in said structures.
It shows that on average the helix lengths lie between 5 and 6. This gives a notion of how far the maximum stack length strays from the average.

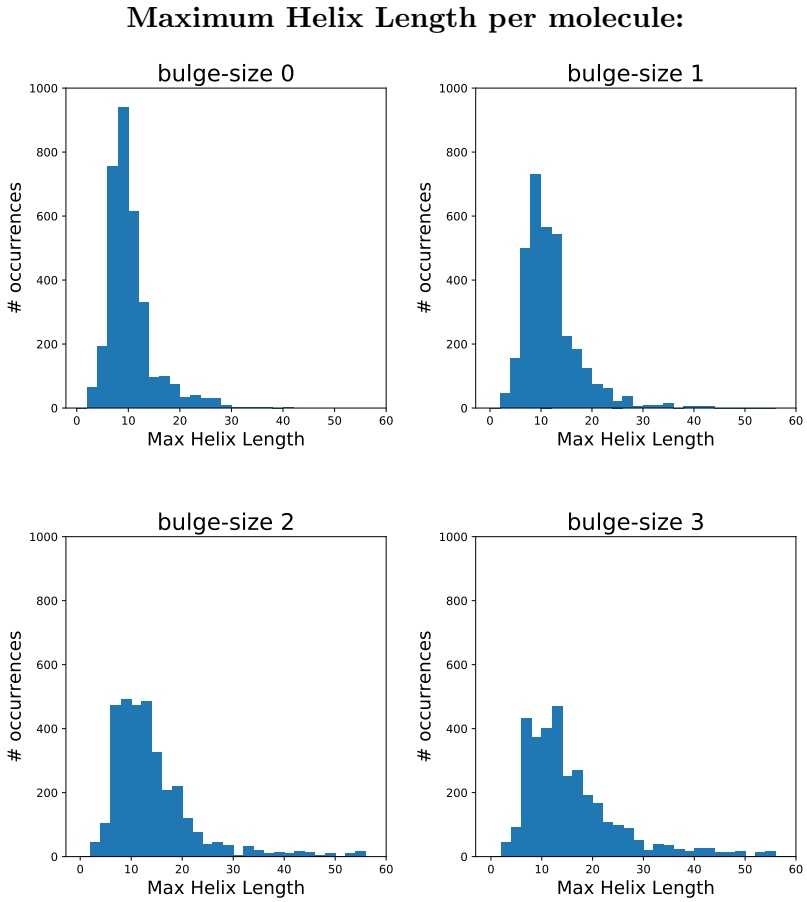


Figure 2.8: *The maximum helix length per molecule allowing different bulge-sizes. The x-axis represents the maximum helix lengths appearing in RNAStrand. The y-axis represents the number of occurrence of each of the maximum helix lengths.*

Due to the short pseudo-knot helix lengths, the maximum helix lengths are also short. As can be seen in Figure 2.10, the maximum helix lengths are distributed over the entire range of helix lengths, with a slight tendency towards the longer helices when increasing the bulge-size.
By analysing the general helix distribution, the overall helix sizes in combination with the maximum helix lengths suggest that the idea of limiting helix sizes is not unreasonable. The maximum helix lengths seem to lie around 10 base pairs for low bulge-sizes. Nevertheless, there are many structures with substantially higher helix lengths.
In contrast, the pseudo-knot helices consist of maximum 9 base pairs, independent of

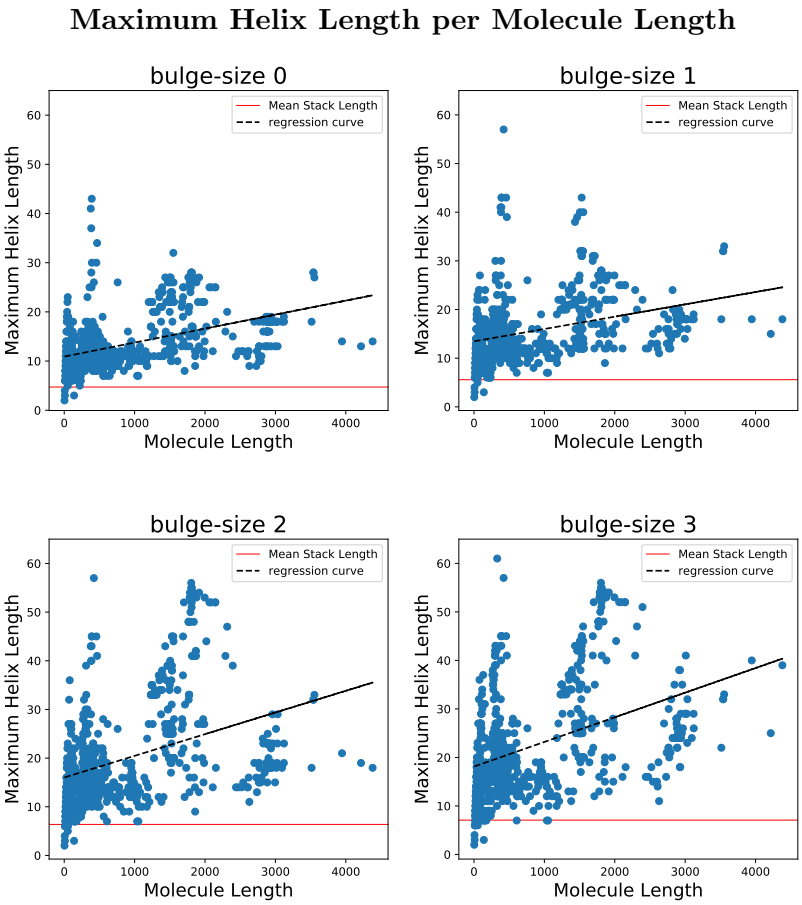


Figure 2.9: *The maximum helix length for each molecule length in RNAStrand allowing different bulge-sizes. The x-axis represents the molecule lengths occurring in RNAStrand. The y-axis represents the maximum helix length.*

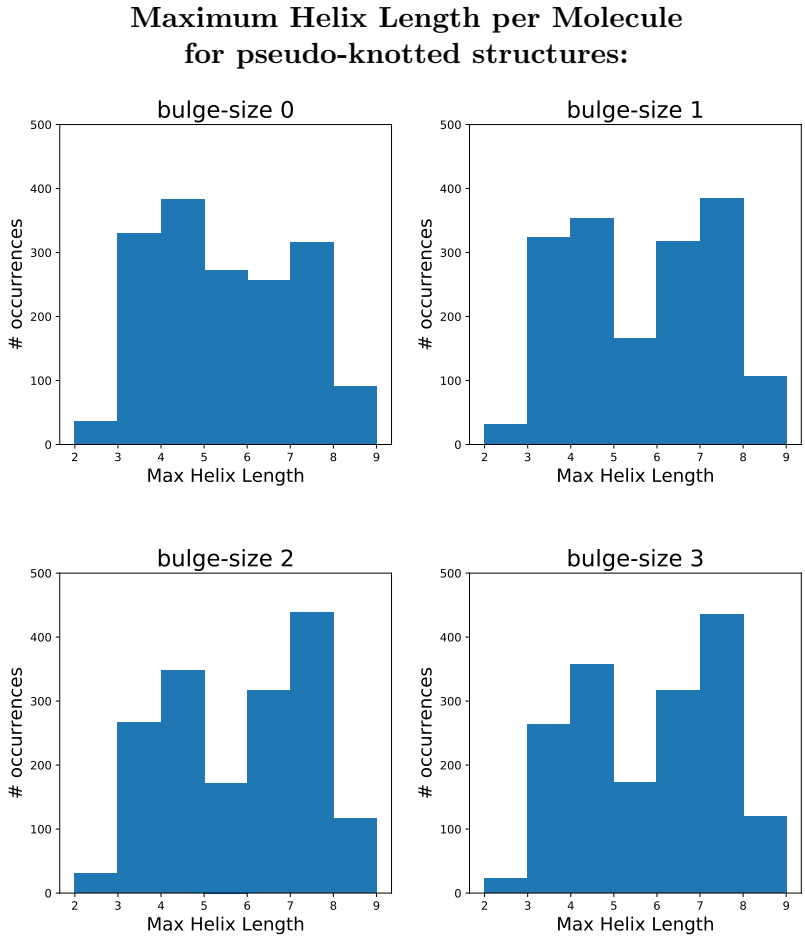


Figure 2.10: *The maximum helix length per molecule for different bulge-sizes, where the helices belong to pseudo-knots. The x-axis represents the maximum helix lengths appearing in RNAStrand. The y-axis represents the number of occurrence of each of the maximum helix lengths.*

the sequence length. Assuming that pseudo-knot helices underlie similar constraints than those imposed on intermolecular interactions, this would confirm that limiting helix sizes could actually be biologically correct. However, as mentioned before, the data is biased, and during the preprocessing more than 1,500 structures were filtered out. Therefore, the representations are not exact. Nonetheless, they provide sufficient information suggesting that a maximum helix length of 9-10 base pairs is a good starting choice for testing the new prediction modi.

2.2 Distribution of unpaired regions in known RNA structures

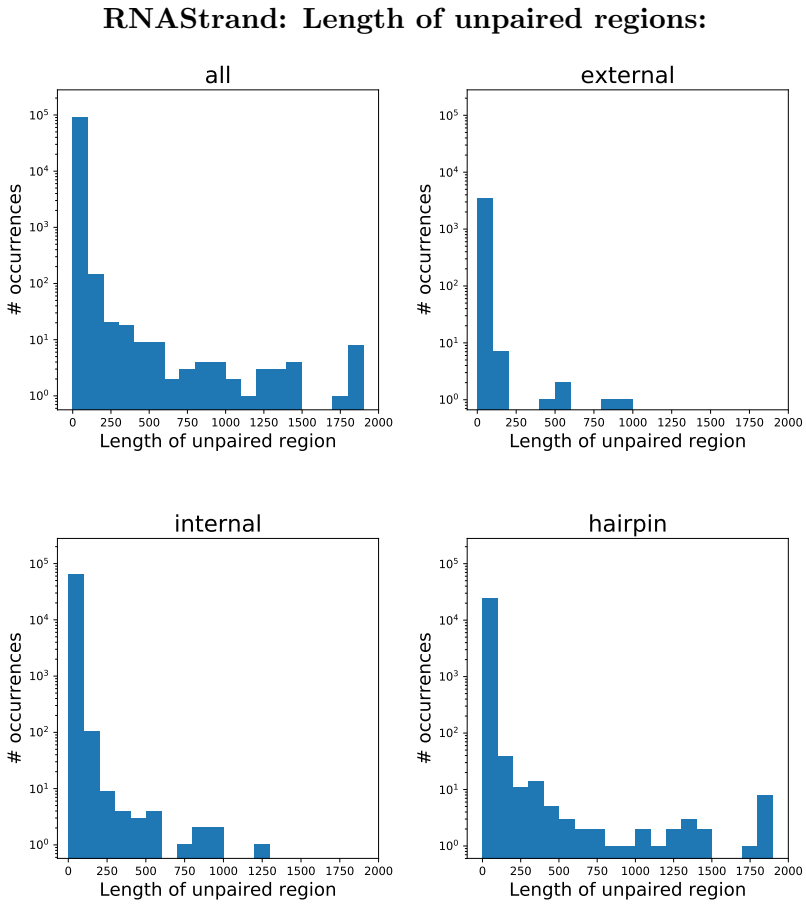


Figure 2.11: *Distribution of unpaired region lengths, classified into 3 categories, external, internal and hairpin. The x-axis represents the different lengths of unpaired regions. The y-axis represents the occurrences of each unpaired region length.*

After having analysed the distribution of helices in known RNA structures, another

approach is to analyse unpaired regions. In order to do this, I created another method, which reads the 3,313 valid RNA structures from RNAStrand and outputs the lengths of unpaired regions, classified into three different categories. First, the external unpaired bases, which belong to no structural elements. Secondly, the hairpin category, which contains all unpaired regions that are part of a hairpin loop. And lastly, the internal category, which comprises all other unpaired regions, i.e those belonging to interior loops, bulges and multi-loops.
Furthermore, single base pairs are regarded as part of an unpaired region, as they are quite unstable. Only unpaired regions with more than 2 bases are considered. Figure 2.11 shows the distribution of unpaired region lengths for the different categories. The majority of unpaired regions range from 3 to 75 bases. The external unpaired regions are the least represented, followed by internal regions. Most of the longer regions seem to be part of hairpin loops.
The analysis of the unpaired regions suggests that there is a large range of different unpaired region lengths. This shows the potential for many seed regions, as most of the unpaired regions are short.

2.3 Prediction of interactions with limited helix length

After the analysis of RNAStrand, the next step is to apply what I have learned to IntaRNA by creating new prediction methods.
In the following, I will differentiate between tightly stacked helices that allow no bulge/interior-loops and those that allow them. In both cases, a seed-based variant will be introduced that incorporates user-definable seed regions into the recursion, as seen before with the original IntaRNA recursions. In order to implement the different methods without having to alter the main recursion every time, I introduce a *helix* function, similar to the *seed* function (Equation 1.18) used by IntaRNA.
The basic idea of the *helix* function is to pre-compute all possible combinations of forming a helix beforehand and store the corresponding energies in a matrix. These helix energies are then used by the predictor to find the optimal interaction with limited helix lengths.
The *helix* method $E_{helix}(i, j, k, l; n_I, n_B)$ returns for fixed intervals $[i, k]$ and $[j, l]$ the best interaction energy, using the following user-definable constraints:

- n_I: the maximal number of unpaired bases allowed for each interior loop between each base pair of a helix.
- n_B: the maximal number of base pairs allowed in a helix. By definition, a helix has at least two base pairs.

The energy contributions for dangling ends as well as the ED-values introduced in Equation 1.16 are usually added in the following recursions. They are omitted in order to keep the recursions readable.

2.3.1 no bulge/interior loop

I will start with the simple version, allowing no unpaired bases in the helix. In other words, n_I is set to 0.
Figure 2.12 shows a sketch of the *helix* function E_{helix}.

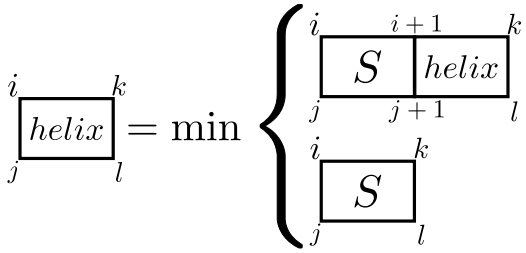


Figure 2.12: *Sketch of the E_{helix} function, denoted as helix and S representing a stacking of two base pairs.*

As there are no bulges or interior loops allowed in the *helix* function E_{helix}, it can be expressed as a sum of stacks:

$$E_{helix}(i,j,k,l;0,n_B) = \begin{cases} \sum\limits_{i<s\leq(k-i)} \left(e^S(i+s-1,j+s-1,i+s,j+s)\right) \\ \quad : \text{if } k-i=l-j<n_B, \\ \infty \\ \quad : \text{otherwise.} \end{cases}$$

In order to find the optimal interaction, all valid intervals have to be analysed, this is done in the predictor function, sketched in Figure 2.13.

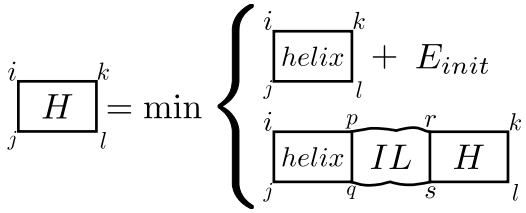


Figure 2.13: *Sketch of the predictor method. helix being the helix function, H the predictor method and IL an interior loop*

The predictor is defined as:

$$H(i,j,k,l) = \min \begin{cases} E_{helix}(i,j,k,l,0,n_B) + E_{init} \\ \min\limits_{\substack{p,q \\ r,s}} \begin{pmatrix} E_{helix}(i,j,p,q,0,n_B) \\ +e^{SBI}(p,q,r,s) + H(r,s,k,l) \end{pmatrix} \end{cases}, \tag{2.1}$$

where the first part describes the case of having a single helix and adds the according initial energy. The second part captures the expansion case when having multiple

helices separated by interior loops. To avoid connecting multiple helices into one long helix, the interior loop case does not allow stacks in the predictor. The predictor returns the mfe interaction with limited helix length.

seed variant

The seed variant incorporates seed regions into the helix computation. This is done by combining the E_{helix} and E_{seed} functions, sketched in Figure 2.14.

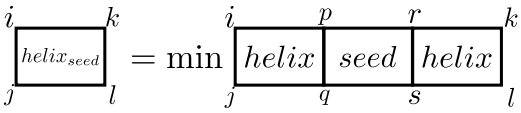


Figure 2.14: *Depiction of the decomposition strategy for the E^{seed}_{helix} computation.*

The *helix* function with seeds E^{seed}_{helix} is defined as:

$$E^{seed}_{helix}(i,j,k,l;0,n_B) = \min_{\substack{p,q \\ r,s}} \begin{pmatrix} E_{helix}(i,j,p,q) \\ +E_{seed}(p,q,r,s) \\ +E_{helix}(r,s,k,l) \end{pmatrix} \tag{2.2}$$

where $k-i+1 \leq n_B$ and $l-j+1 \leq n_B$ in order to ensure that the maximum base pair constraint is fulfilled. The E^{seed}_{helix} function is just a screen over all possible combinations of embedding a seed into a helix. For simplicity, the parameters of the E functions were omitted. They are dependent on one-another and when combined cannot exceed n_B.
Due to the independence of the seed and helix constraints, it is possible to allow unpaired bases in the seed, even when not allowing unpaired bases in the helix constraints. Therefore, it is not possible, without further constraints, to avoid unpaired bases for the E^{seed}_{helix} function.
This returns the minimum free energy of a helix with seed for two fixed intervals $[i,k]$ and $[j,l]$. Consequently, all valid intervals have to be analysed in a predictor method, in order to find the optimal interaction, the predictor is sketched in Figure 2.15.
The predictor is defined as:

$$H_S(i,j,k,l) = \min \begin{cases} E^{seed}_{helix}(i,j,k,l) + E_{init} \\ \min\limits_{\substack{p,q \\ r,s}} \begin{pmatrix} E^{seed}_{helix}(i,j,p,q) \\ +e^{SBI}(p,q,r,s) + H(r,s,k,l) \end{pmatrix} \\ \min\limits_{\substack{p,q \\ r,s}} \begin{pmatrix} E_{helix}(i,j,p,q) \\ +e^{SBI}(p,q,r,s) + H_S(r,s,k,l) \end{pmatrix} \end{cases}$$

The first part describes the initial case of having the helix with seed and the according initial energy. The second part handles the case of having a helix with seed followed by helices without seed, separated by an interior loop. The last part describes the case of having helices without seed before a helix with seed region.

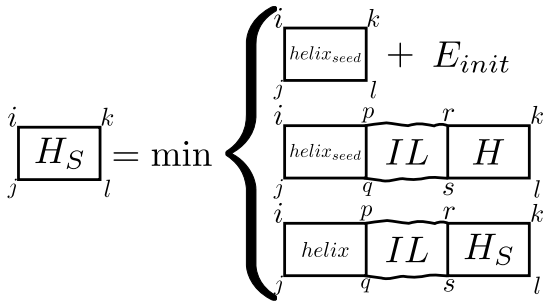


Figure 2.15: *Sketch of the predictor method when allowing a helix with seed region, where H_S denotes the predictor with seed, H the predictor without seed, helix the E_{helix} function, $helix_{seed}$ the E^{seed}_{helix} function and IL the interior loop.*

When using the new H_S matrix within the C computation in Equation 1.17, one can identify the mfe interaction with limited helix length and incorporated seed region, including accessibility penalties.

2.3.2 limited bulge/interior loop

A different approach is to allow (very) small bulges and internal loops within a helix. This is achieved by ensuring that $n_I > 0$. n_I encodes the number of unpaired bases that are allowed between each pair of base pairs in a helix, i.e. the individual interior loop or bulge size. Therefore, the maximum number of possible unpaired bases is:

$$(n_B - 1) * n_I$$

The *helix* function is defined as:

$$E_{helix}(i,j,k,l;n_I,n_B) =$$

$$\min_{2\le B\le n_B} \begin{cases} \min\limits_{\substack{p,q \text{ with} \\ k-p+1\ge B'-1 \\ l-q+1\ge B'-1}} \begin{pmatrix} e^{SBI}(i,j,p,q) \\ +E_{helix}(p,q,k,l;n_I,B'-1) \end{pmatrix} & 2 < B' \le B, \\ e^{SBI}(i,j,k,l) & B = 2, \\ \infty & \text{otherwise.} \end{cases}$$

with B being the current number of base pairs in the helix. The *helix* method returns the minimum free energy of a helix for two fixed intervals $[i,k]$ and $[j,l]$. As unpaired bases are allowed in this case, $p - i + 1 = 2 + n'_I$ and $q - j + 1 = 2 + n''_I$ need to hold, as well as $k - i + 1 = 2 + n'_I$ and $l - j + 1 = 2 + n''_I$ when $B = 2$, where $n'_I + n''_I \le n_I$. The conditions $k - p + 1 \ge B' - 1$ and $l - q + 1 \ge B' - 1$ ensure that $B' - 1$ base pairs are possible in $[p,k]$ and $[q,l]$, as well as unpaired bases.

To find the optimal helix, all allowed small interior/bulge loop combinations have to be considered. This leads to an exponential growth of the complexity. This is treated further in the following complexity section.

Like before all intervals have to be considered to find the optimal interaction with limited helix length. This is done by using the same predictor as before (Equation 2.1). As unpaired bases are now allowed inside a helix, a minimum interior loop size is introduced, which is $n_I + 1$. This ensures that the interior/bulge loop that connects two helices is larger than any loop allowed within a single helix, otherwise the two helices would be merged into a (too large) single one. All *helix* function variants are implemented to be easily exchangeable.

seed variant

The *helix* function that allows both seed regions and small interior/bulge loops is the same as that used when unpaired bases are prohibited. Like with the predictor method, only the *helix* function has to be changed to the newly introduced one.
To find the optimal interaction, the predictor, introduced in Equation 2.3.1, is used by simply replacing the according *helix* functions with the new one.

2.3.3 Complexity analysis

The complexity of the newly introduced exact RNAup-like (Mückstein et al., 2006) recursions does not allow genome-wide searches in reasonable time. Let the length of

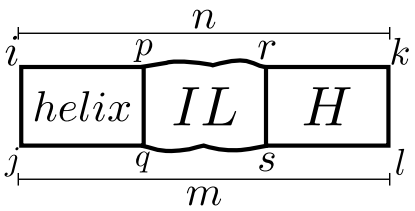


Figure 2.16: *Representation of the recursion part that adds the most to the complexity of the method.*

the input sequences S1 and S2 be n and m, respectively, and I assume that we have to screen over the entire lengths for interval $[i, k]$ and $[j, l]$ as shown in Figure 2.16. As the energies are always calculated for given intervals $[i, k]$ and $[j, l]$, the space complexity is $O(n^2m^2)$. Furthermore, the starting and endpoints of the interior loop, (p, q) and (r, s) respectively, have to be determined. This would lead to an overall time complexity of $O(n^4m^4)$. However, I limit the maximum helix length n_B to $[2, 15]$. This ensures that only a constant part of the sequence has to be screened. Consequently the overall time complexity is $O(n^3m^3)$.
In order to get a feasible runtime, I applied several simplification techniques and heuristics, that were introduced by (Busch et al., 2008) to improve the original IntaRNA recursions. Figure 2.17 provides an overview over the space and time complexity, given these improvements.
First of all, a maximum interior loop size of 16 nucleotides is introduced. Therefore, $r - p + 1 \leq 16$ and $s - q + 1 \leq 16$ hold and the time complexity is reduced to $O(n^2m^2)$, as r and s are not dependent on the size of the input any more.

Then, a heuristic is applied. To this end, instead of considering all interaction ranges, for each interaction start (i, j), only the optimal right boundary (k, l) is considered. This reduces the space and time complexity to $O(nm)$. The heuristic is based on the idea that the matrix $H(i, j, k, l)$ is sparse and only a small part is used in subsequent recursion steps, which ensures that the found interactions will still be the mfe or near-mfe interactions.
In the *helix* functions the best energy for each interaction start (i, j) are stored together with the according length of the helix for both sequences. Knowing the length for a helix in both sequences exchanges the screen over the limited (p, q) end with a constant lookup, which enables a slight runtime improvement. The asymptotic complexity remains at $O(n^2m^2)$.
The presented techniques do also apply to the seed variants of the new predictors, as the seed size is also restricted. The variants that allow unpaired bases in the helix need further restrictions on the number of allowed unpaired bases between each stacked base pair n_I. For my purposes n_I is limited to a maximum of 2, as the number of possible combinations grows exponentially with the size of n_I and the number of allowed base pairs n_B. Further, I only store the best unpaired combination for each interaction start (i, j) to ensure that the time and space complexity does only increase by a constant factor.

bounded lengths	heuristic	space	time
		$O(n^2m^2)$	$O(n^3m^3)$
✓		$O(n^2m^2)$	$O(n^2m^2)$
	✓	$O(nm)$	$O(n^2m^2)$
✓	✓	$O(nm)$	$O(nm)$

Figure 2.17: *Overview of the space and time complexity given certain improvement techniques, under the assumption that the unpaired bases are restricted.*

Chapter 3

Results

In order to evaluate the performance of the newly created predictors for IntaRNA, I will run them on a large dataset. In multiple experiments, I will compare with the original IntaRNA recursions, which will be run on the same dataset, for varying parametrisations of my predictors. For that reason, I created the IntaRNA-benchmark, which I will describe in the following section.

3.1 Benchmark

The IntaRNA-benchmark is equivalent to the CopraRNA-benchmark (Wright, 2016) from a theoretical point of view, but I have changed much on a technical level. The aim was to automatise as many steps of the benchmarking process as possible, but at the same time make it fully customisable.

3.1.1 Theoretical background

The IntaRNA benchmark contains bacterial sRNA queries and mRNA targets. We focus on these RNA types, as IntaRNA is especially successful on the prediction of bacterial regulatory sRNA. IntaRNA outperformed competing programs by determining the exact target sites with higher accuracy.
The genome sequences were originally taken from the GenBank database of the National Centre for Biotechnology Information (NCBI) (Benson et al., 2008). This dataset comprises 4,319 target regions from the *E.coli* genome (GenBank accession number NC_000913) and 4,552 target regions from the *Salmonella typhimurium* genome (Genbank accession number NC_003197).
The targets are genomic subregions around the start codon of the respective mRNA including 200 nucleotides upstream and 100 nucleotides downstream. For one, because most sRNAs bind their target gene in a region around the start codon. Further, genomic subregions are easily extractable from the GenBank.
Further, I have 15 sRNAs for *E.coli* and 15 sRNAs for *Salmonella*. Most of these sRNA act as post-transcriptional regulators by base-pairing to a target messenger RNA (mRNA). Further, IntaRNA incorporates the accessibility of target sites and the

existence of seed regions into the prediction, which is especially interesting for bacterial sRNA.
Therefore, these sRNA-mRNA interactions are ideal for evaluating the performance of new IntaRNA constraints.
To evaluate the performance, I use experimentally verified sRNA targets. The set comprises 149 verified sRNA target. For these verified sRNA targets, only the information that they are regulated is known. The according informations about the location and structure of the interaction is unknown.
The evaluation method is similar to a method used by (Tjaden et al., 2006). In this method the 149 verified sRNA targets are considered as true interactions. Each of the 15 sRNAs in *E.coli* may interact with any of the 4,319 target regions. The same is applicable to the 4,552 target regions for each sRNA in *Salmonella*. As a result, there are 133,065 potential interactions, of which only 149 are considered true interactions, leaving 132,916 unsupported interactions. What is tested is whether the verified targets have, in relation to the other possible mRNAs, better overall energy values, i.e lower energy compared to the unsupported interactions. In other words, I check how well the verified target performs, when taking the predictions of all potential interactions as my background distribution.
In this benchmark, a result file is computed for each sRNA query per call. These result files contain a list of target candidates and are sorted according to their computed energy scores. Like this, the results are ordered from the most favourable, the one with the lowest energy, to the most unfavourable interaction. Then, a rank is calculated for each entry among the verified interactions (true interactions). The rank describes how favourable the interaction is, e.g. in what row of the result file of the IntaRNA call it appears. With this in mind, it is preferable to have as many low-ranking interactions as possible. In other words, the more low ranks are obtained, the more often IntaRNA predicted the experimentally verified interactions among its best results.
To visualise the performance of an IntaRNA prediction mode, I use receiver operating characteristic (ROC) curves. The X-axis describes the number of target predictions per query RNA, while the Y-axis represents the number of true positives. In other words, for each X, the number of ranks that are smaller or equal to X are counted and represented on the Y-axis. Like this, multiple prediction modes can be plotted into the same graph to compare their performance. As ROC curves can be hard to interpret, especially when overloaded with many curves, I also provide the option to use violin plots. The user can provide a reference prediction, the difference between the reference curve and each prediction mode is visualised in a violin plot. This makes it easier to view the increase or decrease of performance compared to a given curve.

3.1.2 Technical background

In contrast to the CopraRNA-benchmark, the IntaRNA-benchmark can also record time and maximum memory consumption of each IntaRNA call. It can be downloaded from `https://github.com/BackofenLab/IntaRNA-benchmark`.
The time and maximum memory consumption is collected using the *os.wait4* command by tapping the *resource.getrusage*. The time is extracted from the *ru_utime* field,

while the maximum consumption is read from *ru_maxrss* field. The *ru_maxrss* field contains the maximum resident set size, i.e. the maximum amount of memory that was attributed to a given process currently handled by *os.wait4*.
Further, I introduced a system using callIDs to allow running different parametrisations of IntaRNA simultaneously, in order to compare the performance of each individual call. The IntaRNA-benchmarking scripts are written in python3 and allow multiple command-line arguments for customisation.
The benchmarking begins with the *calls* script. The script calls IntaRNA, with user-defined parameters, for each sRNA query, mRNA target combination and thus creates a result file for each sRNA. For newer versions of IntaRNA and ViennaRNA (Lorenz et al., 2011), it also allows the pre-computation of target ED-values and subsequent re-usage of these values in order to fasten up the benchmarking process substantially. Then, the *benchmarking* script is called, which uses the *verified interactions* file to generate the rank for each interaction. Lastly, the *plot_performance* script can be used to create an ROC or violin plot. Further, it allows the visualisation of memory and time consumption.
The *mergeBenchmark* and the *clearAll* scripts are not vital to the benchmarking process but can be used to easily merge the results of multiple calls of the benchmark script to visualise them in one plot or remove certain callIDs respectively.

3.2 Hardware specifications

All experiments were run on a machine equipped with an *Intel Core i5-62000U 2.3GHz* processor and 8GB available RAM. This highlights that IntaRNA has a low memory requirement which makes it applicable on normal work stations. In order to fasten up the benchmarking process, all experiments were run using multi-processing on 3 cores. This reduces the runtime while roughly increasing the maximum memory consumption by a factor of 3, which also influences the times and maximum memory data mentioned in the experiments section. Further, the accessibility energies (ED-values) for the targets were calculated in advance and read from file, to avoid re-computation.

3.3 Experiments

In order to test whether limited helix lengths improve the prediction quality of IntaRNA, I experimented with different parametrisations of the two new prediction methods. The heuristic version of all predictors was used to run the experiments, as the runtime complexity of the exact versions makes them impractical for genome-wide screens. Nevertheless, the runtime of the heuristic version did not allow me to test all possible parametrisations I intended, as there is a large amount of different parametrisation combinations that could potentially yield interesting results.
The original predictors were run with their default values and are used as a reference for comparison.
In the following, I will refer to the predictor that allows no unpaired bases as *stackingOnly*. When unpaired bases are allowed, it is called *unpaired*. The suffix *seed* refers to the seed variant of a predictor.

3.3.1 Overview

To get an overview over the performance of the different new prediction modi, I compare them to the original IntaRNA recursions. Therefore, the default values for all predictors were used. The maximum size of the helices n_B is thereby limited to 10. The number of unpaired bases n_I is set to 2, for variants of the predictors. The number of seed base pairs is 7 by default and no unpaired bases are allowed in the seed region.

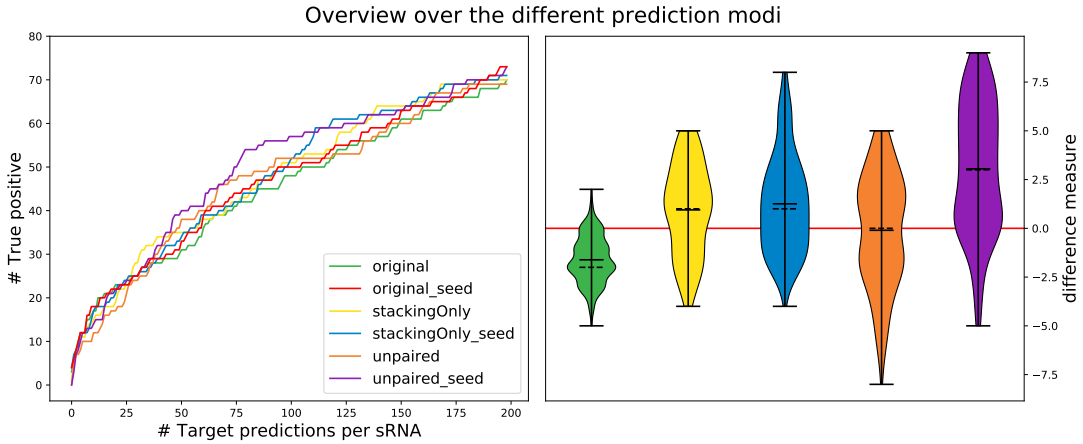


Figure 3.1: *Overview over the different predictors using their default values up to 200 target predictions. n_B is set to 10, n_I is set to 2 and the number of seed base pairs is set to 7.* ***stackingOnly*** *references the E_{helix} method without unpaired bases, whereas* ***unpaired*** *represents the ones allowing unpaired bases. The suffix* ***_seed*** *denotes the seed variants of the functions. The left figure shows the performance of the different predictors using ROC curves. The right figure shows the difference between the different predictors and the original predictor with seed (red). In the violin plots, the dashed line represents the median and the straight line the mean.*

The left plot in Figure 3.1 shows the performance of all prediction modi, using ROC curves. The *original_seed* curve shows the results for the original predictor with incorporated seed regions. It is the best performing predictor in IntaRNA so far, therefore I use it as a reference to evaluate the performance of the newly developed methods. It is coloured in red in every plot.
In order to give a clearer overview over the performance of each predictor, I calculated for each method their difference to the reference predictor. This means that for each number of target predictions plotted, the difference of the true positive values between the reference and the new predictor is taken. When the difference is positive the new predictor performed better than the original one, for the according number of target predictions. The difference measure is visualised in the right figure using violin plots.
It is important to note that while this measure allows for an easier comparison and gives a better intuition than the ROC curves, it can be misleading when not considering both plots. Generally speaking, the measure only quantifies how often the curve of the new predictor is above or below the reference curve. The problem is that the more

target predictions are allowed, the less important the results. With this in mind, both curves have to be taken into account when rating the performance of a predictor.
The ROC plot shows that the original predictor has, for the current default values, still the best starting values, i.e. the best values for a very low number of target predictions. The predictor that allows no unpaired bases and no seed, here *stackingOnly*, has worse starting values but overtakes the original predictor at around 25 target predictions. The seed variant of the *stackingOnly* predictor slightly improves the starting values. It only has a performance boost at around 125 allowed target predictions, which is not very impactful. The same behaviour is observable for the predictors allowing unpaired bases. The seed-less variant of the *unpaired* predictor has the worst performance among all predictors, given the default parameters. Whereas, the *unpaired_seed* predictor shows very promising results. It does not perform as well as the *stackingOnly_seed* predictor up to around 40 target predictions. On the other hand, it outperforms the other predictors by a large margin on the range $[50, 115]$. Given the starting values of the *unpaired_seed* predictor, it is hard to say which one is better. What is clear is that the seed helps increasing the prediction quality in both methods.

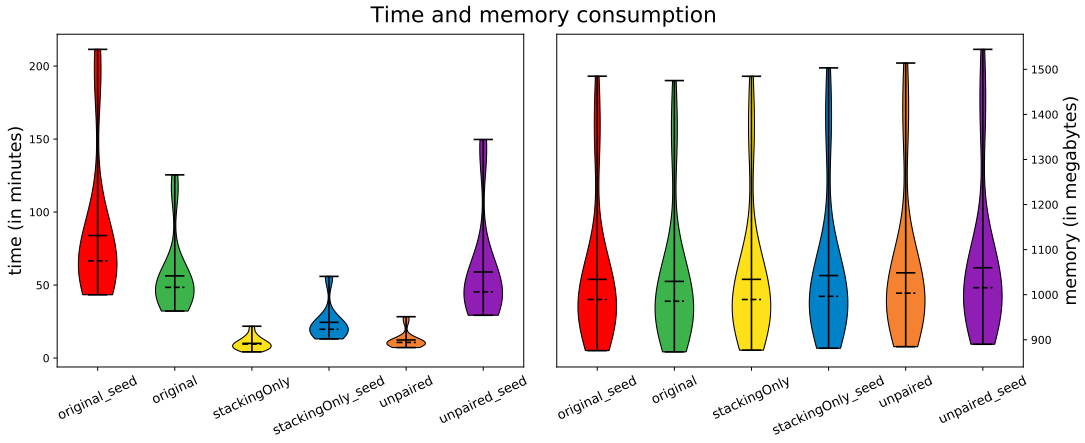


Figure 3.2: *Overview over the time (left) and maximum memory consumption (right) for the predictors in the overview shown in Figure 3.1.*

Figure 3.2 shows the runtime and maximum memory consumption for the different predictors from the overview shown in Figure 3.1. These are the runtime and maximum memory values for the heuristic version using multiprocessing on 3 cores for each predictor. It is clearly visible, that the introduction of a maximum helix length does not only show potential for improving the performance of IntaRNA. The runtime is greatly decreased, while the maximum memory consumption stays roughly the same. My focus lies in creating the wanted functionality of a maximum helix length, as a proof of concept. Therefore, not everything is optimised to the point it could be. Nevertheless, the maximum helix length reduces the range that has to be considered in every step, leading to a greatly reduced runtime. The upper bounds of the violin plots are caused by the sRNA queries *GcvB* and *SgrS*, as they are more than double the length of the other sRNA used in the benchmark.

The reason for the overhead of the seed variants is easy to see when looking at Equation 2.2 or the corresponding sketch in Figure 2.14. The seed variants need three matrices in total. The E_{helix} and the E_{seed} matrices are required to calculate the E_{helix}^{seed} matrix. The much higher gap for the *unpaired* predictor is explained by its more complex structure, e.g. when a helix of 5 base pairs is not possible for the predictor allowing no unpaired bases it makes no sense to test 6 or more base pairs, for a given starting position. This is not the case for the *unpaired_seed* predictor as a small bulge or internal loop could potentially allow longer helices. Therefore, all combinations have to be taken into account for the *unpaired_seed* predictor, while the *stackingOnly_seed* predictor has more efficient break conditions.

The overview showed that, given the default values, all new predictors outperform the currently best predictor at some point. Apart from the *unpaired* predictor, who is especially bad in the beginning. Nevertheless, even though the *stackingOnly* predictors come very close to the starting values of the original predictors, none manages to outperform them.

In order to see whether different parametrisation might lead to even better performance, I will analyse each method in more detail. I want to see how different numbers for maximum helix length and minimum helix length, as well as different maximum energy values influence the performance of each predictor.

3.3.2 no bulge/internal loop

In order to get all the results in time, I used the setup described in the hardware specification, but used an *AMD Ryzen 1700X Eight-Core* processor instead. That allowed me to queue 4 IntaRNA calls using 3-threads each via hyper-threading. Therefore, the relative difference in runtime remains the same, but the overall runtime reduces.

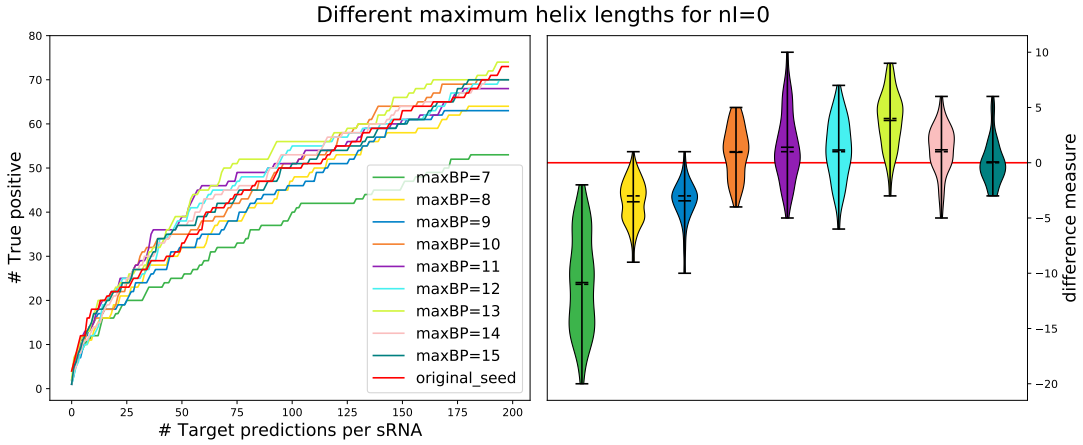


Figure 3.3: *The influence of different maximum helix lengths for the* stackingOnly *predictor without seed. For the description of the plot order, see Figure 3.1.*

simple variant

The default value for the maximally allowed helix length is just an assumption based on the results from the analysis of RNAStrand (Andronescu et al., 2008). Therefore, I tested different n_B ranging from 7, the default seed size, to 15, in order to determine whether the overall quality of the prediction improves or not.
I started with the *stackingOnly* predictor that allows no seed. Figure 3.3 describes how different maximum helix lengths influence the result of the predictor. It clearly shows that very small n_B have a negative effect on the prediction quality. In contrast, $n_B = 11$ and $n_B = 13$ show really good performance increases when comparing to the default value $n_B = 10$. Especially, $n_B = 13$ shows very promising results, as it starts to overtake the original predictor at around 15 target predictions.
In the next step, I tried to improve the results of the predictor using both the minimum helix length and the maximum energy parameter. To do this, I used the overall best performing maximum helix length $n_B = 13$.

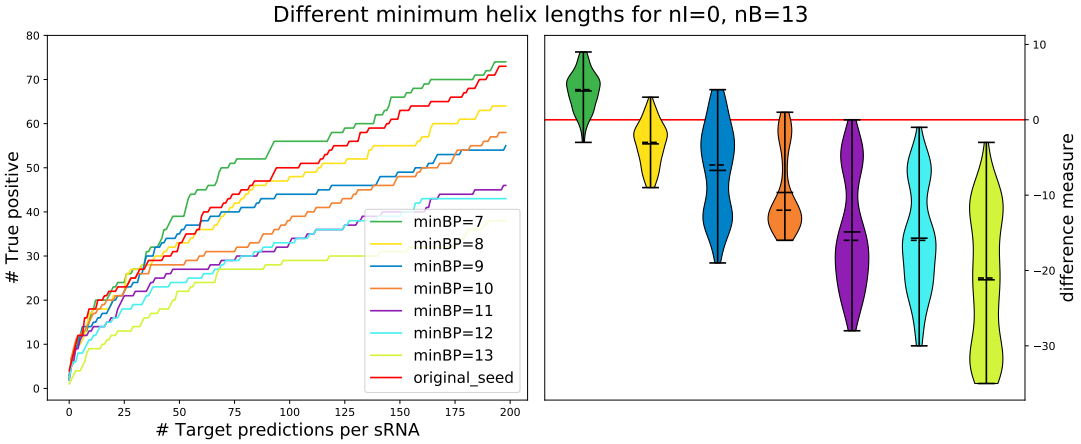


Figure 3.4: *The influence of different minimum helix lengths for the* stackingOnly *predictor without seed. For the description of the plot order, see Figure 3.1.*

In Figure 3.4 the impact of different minimum helix lengths on the overall prediction quality is shown. *minBP*=7 represents the default case that is also represented in Figure 3.3 as *maxBP*=13. The minimum helix length has in fact a very negative effect on the prediction quality of the *stackingOnly* predictor. In some cases, it slightly improves the starting values but it causes a noticeable performance drop for the rest of the plot.

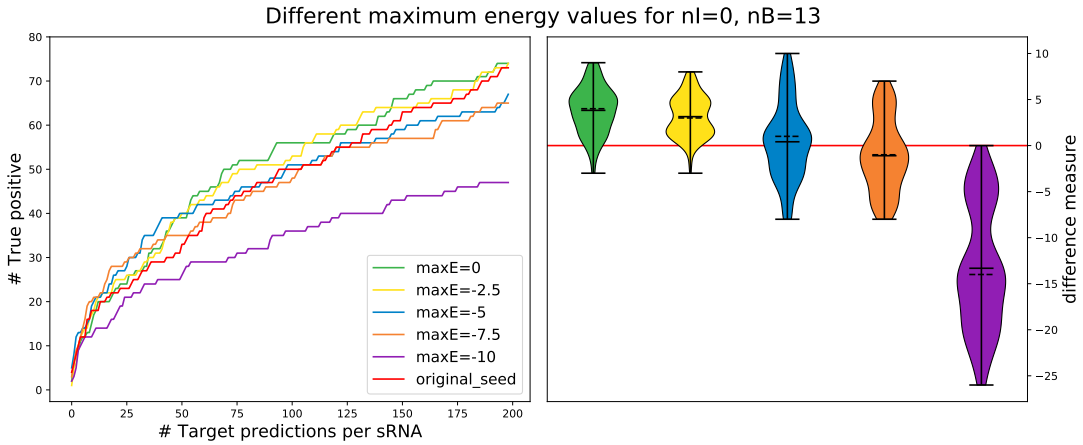


Figure 3.5: *The influence of different maximum energy values for the* stackingOnly *predictor without seed. For the description of the plot order, see Figure 3.1.*

Figure 3.5 shows different maximum energy values for the *stackingOnly* predictor. It becomes clear that, even though the performance decrease is not as strong as for the minimum helix length parameter, the prediction quality drops when using lower energy values.
In order to see whether the seed variant behaves differently, I continued by performing the same experiments for the *stackingOnly_seed* predictor.

seed variant

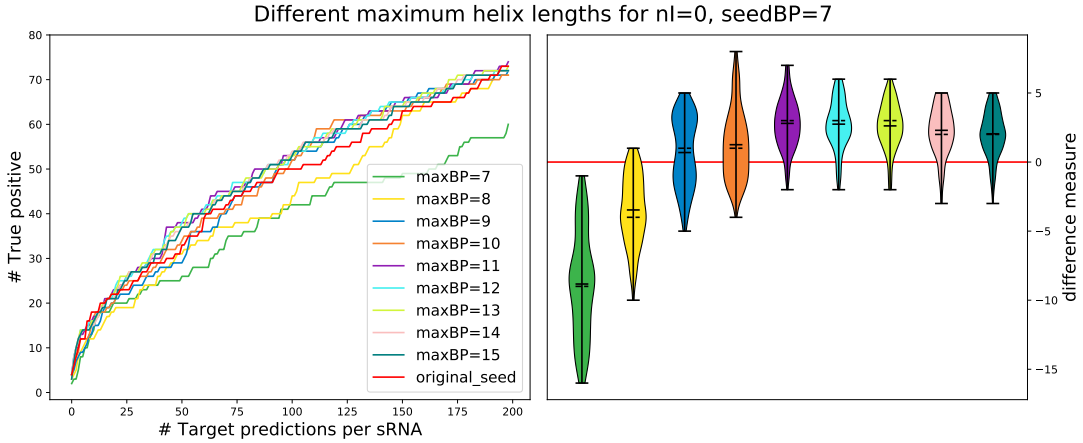


Figure 3.6: *The influence of different maximum helix lengths for the* stackingOnly_seed *predictor. For the description of the plot order, see Figure 3.1.*

I started again by evaluating different maximum helix lengths, as shown in Figure

3.6. In contrast to the predictor without seed, $n_B = 11$ performs better than $n_B = 13$. Nevertheless, all values $n_B \in [11, 15]$ are relatively good for *stackingOnly_seed*. Consequently, I will continue my analysis using $n_B = 11$ as the value for comparison. As before, I continue testing different minimum helix lengths and maximum energy values in order to see whether there is a difference to the *stackingOnly* predictor.

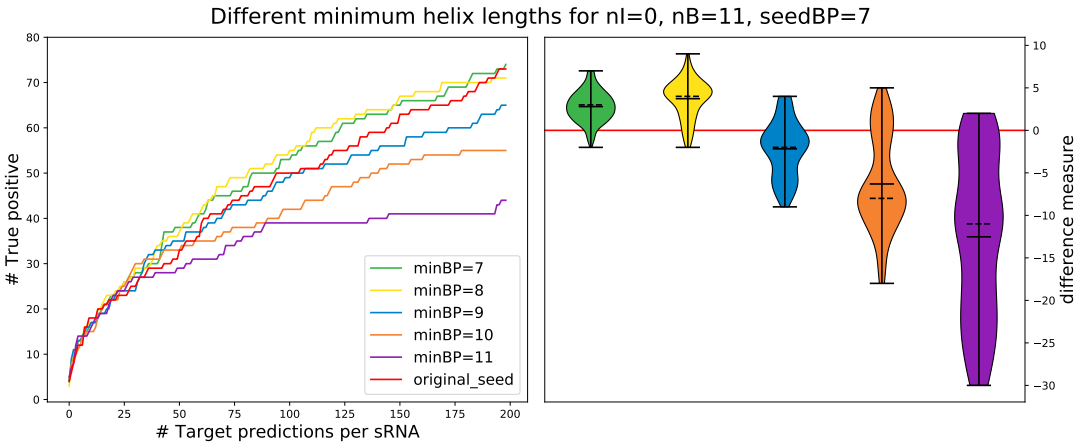


Figure 3.7: *The influence of different minimum helix lengths for the* stackingOnly_seed *predictor. For the description of the plot order, see Figure 3.1.*

In Figure 3.7, the influence of different minimum helix lengths is highlighted. For the *stackingOnly_seed* predictor, the minimum helix lengths show no real improvement. But the prediction quality in not reduced as badly as for the predictor without seed. All in all, the minimum helix length does not help increase the prediction quality for this predictor.
Figure 3.8 describes the influence of different maximum energy values on the prediction quality. This means that only helices that have an energy lower than the given *maxE* are accepted. As a consequence, for values of *maxE* that are too low, near to no interactions are valid any more. This leads to a bad performance. A *maxE* of 0 is the default value. The values -2.5, -5 and -7.5 are better than the reference predictor. Especially, for a *maxE* of -7.5, the predictor has better results from around 2 target predictions to the end of the plot, with improvements of up to around 18 true positive values.
As mentioned before, the idea behind maximum energy values is similar to that of a minimum helix length. This is true in general, but the maximum energy allows more flexibility. Where the minimum helix length is fixed, different length combinations are still possible for the given maximum energy, e.g. a helix formed solely by G-C base pairs is allowed to be shorter as it has a lower energy than a helix formed only by A-U base pairs. When testing different *maxE* values close to -7.5 like -7 and -8, they showed similar improvements, but -7.5 remained the overall best value.

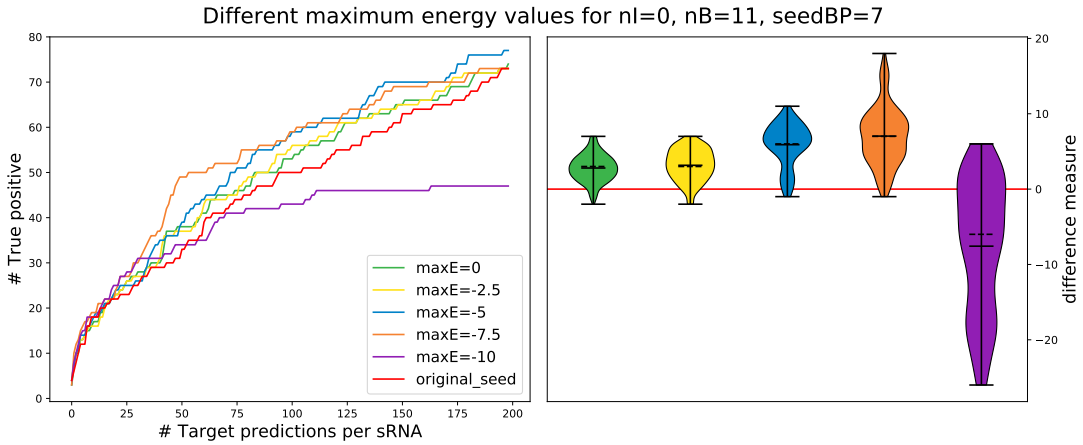
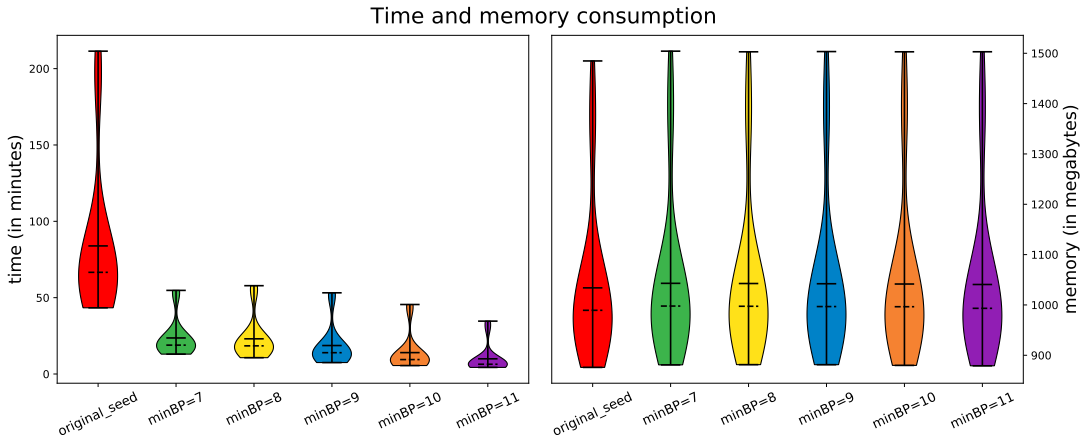
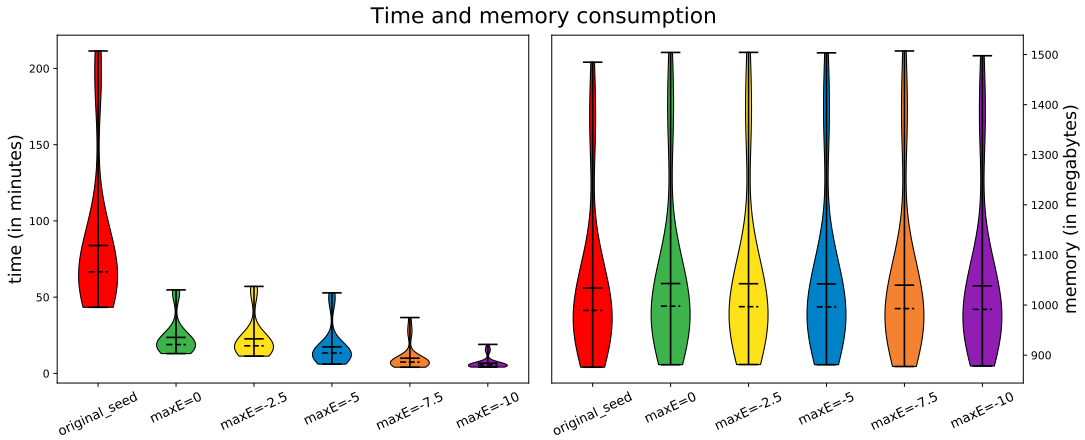


Figure 3.8: *The influence of different maximum energy values for the* stackingOnly_seed *predictor. For the description of the plot order, see Figure 3.1.*

Figure 3.9 shows that both the minimum helix length and the maximum energy values can help to reduce the runtime, while the maximum memory remains the same.
When comparing the influence of the maximum energy parameter for both prediction variants, it becomes clear that the seed seems to help further increase the prediction quality. The reason for this is most likely the implementation of the seed variants. As the E_{helix} matrix is required to create the seed, I set the maximum energy value for E_{helix}, i.e. *stackingOnly*, to 999. This means that all energy values are accepted for the E_{helix} matrix. The reason for this is easily explained when considering the recursion for E^{seed}_{helix} (see Equation 2.2). When not allowing all values for E_{helix} the overall best combination of introducing a seed might not be found. Therefore, the seed variant still allows smaller helices in the final interaction, while enforcing one strong seed. This seems to have a very positive effect on the overall prediction quality. Nevertheless, it might be favourable to make the seed variant independent from the seed-less variant in the future. This allows customising both variants, without affecting each other directly.



(a)



(b)

Figure 3.9: *Overview over the time (left) and maximum memory consumption (right) for different minimum helix lengths (a) and maximum energy values (b) for* $n_I = 0$, $n_B = 11$ *and* $seedBP = 7$.

3.3.3 limited bulge/internal loop

I will now apply the same parametrisation techniques to the *unpaired* predictors. Due to complications during the experimentation phase, I was only able to generate the results for $n_I = 2$. Previous tests showed that the predictor behaves in the same way for both $n_I = 1$ and $n_I = 2$, while $n_I = 1$ leads to slightly better results and a reduced runtime. Nevertheless, the overall behaviour of different parametrisation remains the same.

simple variant

Figure 3.10 shows the behaviour of the *unpaired* predictor, given different numbers of maximum helix lengths. The general idea remains the same as for the *stackingOnly* predictors. Small maximum helix lengths, like 7 and 8, return very bad results. The best performance is achieved for 13 allowed base pairs, while it is hard to say for this predictor, as it performs badly for all values.

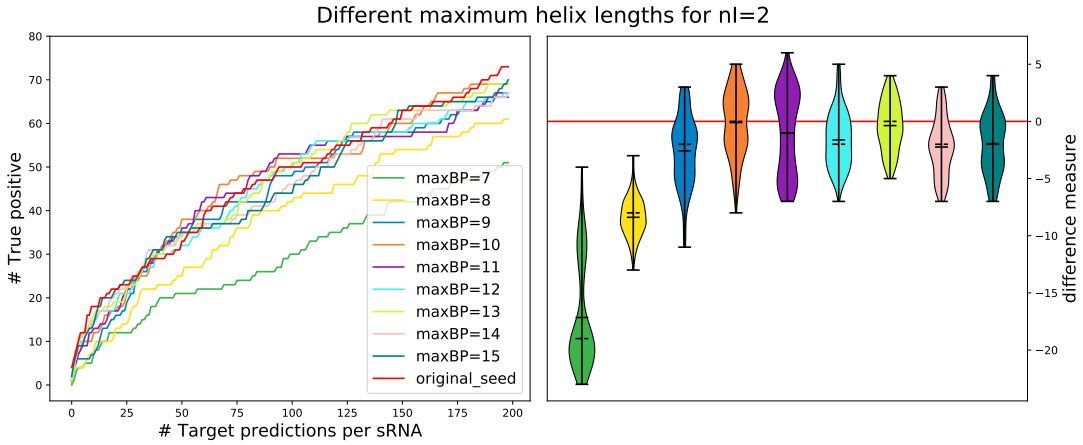


Figure 3.10: *The influence of different maximum helix lengths for the* unpaired *predictor without seed.* n_I *is set to* 2*. For the description of the plot order, see Figure 3.1.*

Due to the overall bad results of the *unpaired* predictor and the lessons learnt from the *stackingOnly* predictor without seed, I did not conduct further experiments with the minimum helix length and maximum energy value parameters. Instead, I continued with a detailed analysis of the *unpaired_seed* predictor.

seed variant

As the *unpaired_seed* predictor already showed promising results in the overview, I also tested different minimum and maximum helix lengths as well as different maximum energy values with this predictor, in an attempt to improve the results.

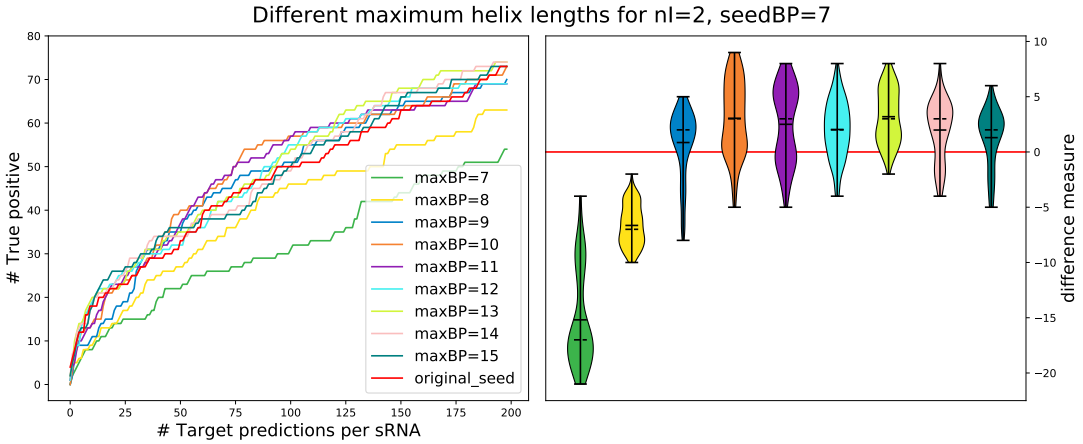


Figure 3.11: *The influence of different maximum helix lengths for the* unpaired_seed *predictor.* n_I *is set to* 2 *and* $seedBP = 7$. *For the description of the plot order, see Figure 3.1.*

Figure 3.11 highlights the influence of different maximum helix lengths for the *unpaired_seed* predictor, with $n_I = 2$ and $seedBP = 7$.
It is hard to assess which maximum helix length value returns the best results for the *unpaired_seed* predictor. Though it seems as if the best value is among the values from 10 to 15. Among these values, $n_B = 13$ and $n_B = 15$ perform best for low numbers of target predictions. In the range between 25 and 50 target predictions, $n_B = 15$ performs better. Further, $n_B = 10$ and $n_B = 11$ have better values in the range $[50, 125]$. Nevertheless, I would say that 13 performs best given the whole range of target predictions, due to the good starting values and the high mean and median value. Therefore, I will use $n_B = 13$ to continue testing the minimum helix length and maximum energy parameters.

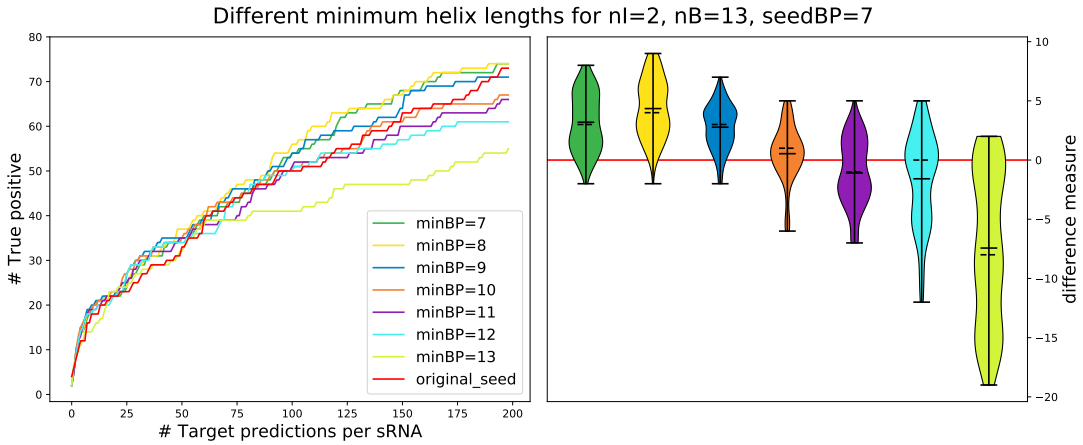


Figure 3.12: *The influence of different minimum helix lengths for the* unpaired_seed *predictor.* n_I *is set to* 2, n_B *to* 13 *and* $seedBP = 7$. *For the description of the plot order, see Figure 3.1.*

Figure 3.12 shows how different minimum helix lengths affect the performance of the *unpaired_seed* predictor. *minBP*=7 represents the performance of the *unpaired_seed* for $n_B = 13$, shown in Figure 3.11. For this predictor, the minimum helix length has no great effect on the prediction quality. It seems to slightly improve for *minBP*=8 but apart from that it reduces.

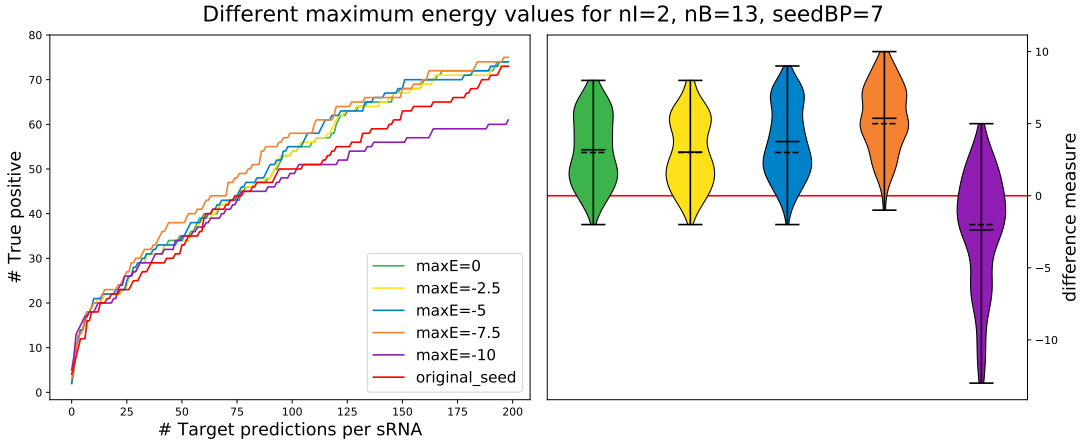
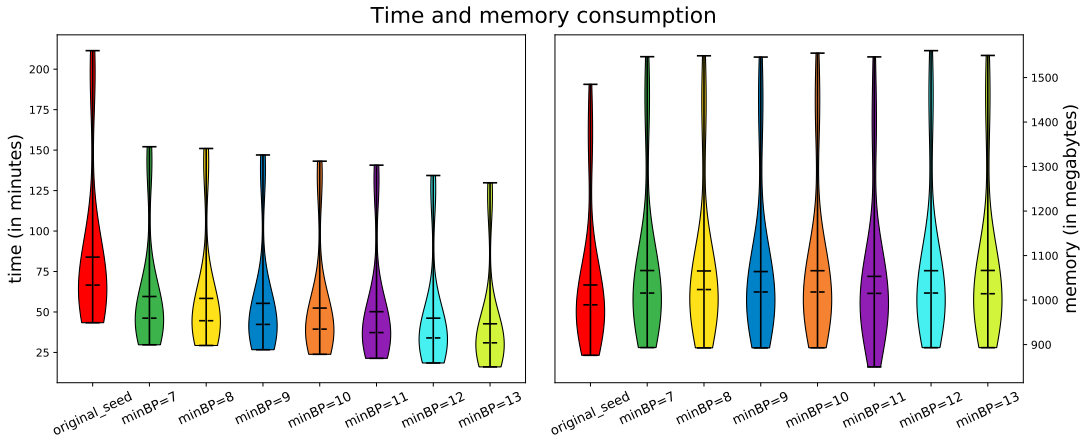


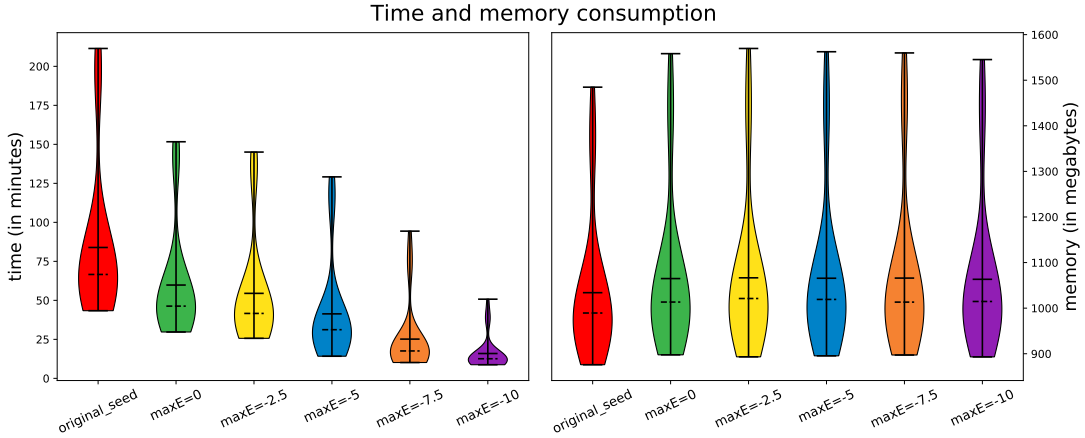
Figure 3.13: *The influence of different maximum energy values for the* unpaired_seed *predictor.* n_I *is set to* 2, n_B *to* 13 *and* $seedBP = 7$. *For the description of the plot order, see Figure 3.1.*

Figure 3.13 shows the influence of different maximum energy values given $n_I = 2$, $n_B = 13$ and *seedBP*=7. *maxE*=0 represents the performance of the *unpaired_seed* predictor, shown in Figure 3.11. Similar to the *stackingOnly_seed* predictor, the different maximum energy values have a positive effect on the prediction quality. This

suggests, that favouring longer helices while at the same time introducing a maximum helix length improves the results. But when comparing the results to those of the minimum helix length in Figure 3.12, the results seem to confirm the suspicion that the flexibility of a maximum energy is better than a static minimum helix length. As before, a value of $maxE{=}-7.5$ returns the overall best results. $maxE{=}-10$ has good starting values, but due to the very low energy requirement many interactions are not viable any more, explaining the large performance drop. Test with $maxE{=}-15$ showed that of the potential 133,065 interactions, only around 60 passed the energy threshold, among which 3 were verified interactions. The maximum energy parameter, like the minimum helix length parameter, can further reduce the runtime of the predictor, as shown in Figure 3.14.



(a)



(b)

Figure 3.14: *Overview over the time (left) and maximum memory consumption (right) for different minimum helix lengths (a) and maximum energy values (b) for* $n_I = 2$, $n_B = 13$ *and* $seedBP = 7$.

3.3.4 Summary

In this chapter, I showed the results of each method for different parametrisations. This was done using the newly created IntaRNA-benchmark.
The results revealed that the maximum helix length constraint helps improving the results of IntaRNA, while at the same time reducing the runtime. For all predictors, low values of n_B showed bad results, while high helix lengths showed good results in general. Nonetheless, the best results were determined for $n_B \in [10, 13]$. Further, I analysed the behaviour of the minimum helix length and maximum energy parameters. It showed that, while both tend to improve the overall prediction quality and runtime, the maximum helix energy achieves the overall best results. As mentioned before, this is most likely due to the fact that the maximum energy value is more flexible. This means that it allows different helix lengths, depending on the composition of the bases involved in the interaction. Whereas, the minimum helix length only allows fix lengths. This makes me think, that a minimum energy value could potentially lead to even better predictions than a maximum helix length.
Moreover, the large gap between the performance of the *unpaired* and *unpaired_seed* predictors could suggest that, either the heuristic of storing only the best combination of unpaired bases in each step or the whole unpaired bases handling is too unrestricted. With the latter, I mean that the current method allows structures that contain internal loops between each pair of base pairs, leading to unreasonable structures. The introduction of a seed that contains no unpaired bases consequently increases the prediction quality, as only a small part of the helix remains that can contain unpaired bases. A solution to both of these problems would be the introduction of additional restrictions on the *unpaired* predictor. This is discussed in more detail in the Future Work chapter of this thesis.
Finally, I summarised the best performing parametrisation of each prediction method in Figure 3.15.

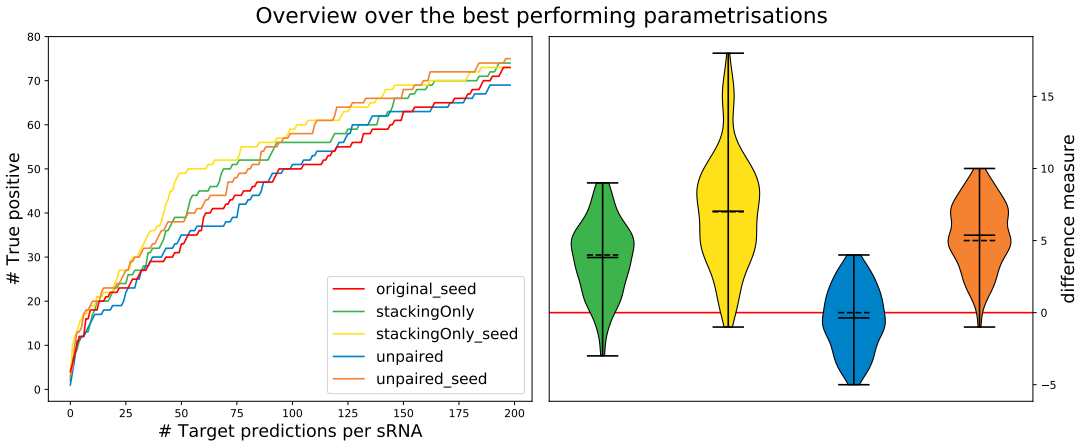


Figure 3.15: *An overview over the best performing parametrisations of each predictor. The Figure shows the* stackingOnly *predictor for* $n_B = 13$*, the* stackingOnly_seed *predictor for* $n_B = 11$*,* $seedBP = 7$ *and* $maxE = -7.5$*. The* unpaired *predictor is represented using* $n_B = 13$ *and the* unpaired_seed *predictor for* $n_B = 13$*,* $seedBP = 7$ *and* $maxE = -7.5$*.*

Chapter 4

Related Work

When discussing related work, there are two aspects that can be covered. For one, I present a constraint for IntaRNA, which is therefore used in mfe-based RNA-RNA interaction prediction. On the other hand, the general concept of reducing the intermolecular helix lengths could be applicable in other methods and fields. For the latter, I was unable to find anything related to the idea of introducing a maximum helix length in order to better incorporate the steric 3D constraints of RNA molecules into the RNA-RNA interaction prediction. To the best of my knowledge no such constraint was tested before.

As this technique aims at improving RNA-RNA interaction prediction, there are many related approaches. Umu and Gardner (2017) created a benchmark of RNA-RNA interaction prediction tools, that compares different approaches of accomplishing this task as well as the performance of each of the most prominent tools using these methods. They describe three main groups of RNA-RNA interaction prediction methods, alignment-like methods, mfe methods and comparative (homology) methods. There is a large amount of tools that use these different methods. Umu and Gardner (2017) give a detailed description of these tools and indicate special methods only used for certain RNA types. They also describe some under-represented methods like machine learning approaches and probabilistic methods.

IntaRNA belongs to the mfe methods. These are further split into three sub-methods, those that consider intra-molecular structure, those that ignore intra-molecular structure and those that incorporate the accessibility of the target regions. IntaRNA belongs to the last group.

It is important to note that each of these tools specialises in certain RNA types, therefore they do not perform as well on large datasets that contain many different RNA types. Some approaches showed to perform quiet well on a large variety of RNA types, but it is unlikely that a method exists, that performs well on every RNA type.

I want to give an overview over three of the more recently developed tools. Accessfold (DiChiacchio et al., 2016), RIsearch2 (Alkan et al., 2017) and RIblast (Fukunaga and Hamada, 2017) are RNA-RNA interaction prediction tools, but they use different approaches than IntaRNA.

The idea behind Accessfold is to better integrate the competition between uni- and bimolecular structure, i.e. the intra- and intermolecular structure, into the prediction

process. Therefore, DiChiacchio et al. (2016) introduced two new algorithms, DensityMin and Accessfold, that used two new approaches for accessibility evaluation, free energy density minimisation and pseudo-energy minimisation. DensityMin was based on the hypothesis that density minimisation leads to shorter, more stable helices between strands, which were thought to outperform potential intramolecular binding partners. Accessfold showed better results than DensityMin, it uses energy minimisation. The energy minimised is composed of a folding free energy change and a pseudo-energy that accounts for the accessibility. This pseudo-energy penalty is added to the folding free energy for each nucleotide forming a base pair. It accounts for accessibility but treats each nucleotide independently. A similar approach of position-wise accessibility terms was already employed in RNAplex (Tafer and Hofacker, 2008). Biophysically, it would be more accurate to add the accessibility energy once for each interaction site. Nonetheless, they showed, for their tested data set, that this pseudo-energy minimisation was able to outperform the mean sensitivity of RNAup (Mückstein et al., 2006), while the difference in average PPV was statistically non- significant.

RIsearch2 introduces a new concept to predict RNA-RNA interactions. Contrary to its predecessor RIsearch (Wenzel et al., 2012), which used a dynamic programming approach based on the Waterman-Gotoh algorithm (Gotoh, 1982), RIsearch2 uses a seed-and-extend approach. This newly developed method is divided into two main steps. In a first step, indices of query and target sequences are built using suffix arrays in order to locate seed regions. This is done by going through the query and target suffix arrays in parallel and finding suffixes with perfect complementarity. In a second step, these seeds are then extended and the hybridisation energy is computed using a simplified energy model introduced in the original RIsearch. It uses a dynamic programming (DP) approach, calculating DP matrices that can be tracebacked to find the resulting interactions. RIsearch2 includes several user-definable parameters, like an hybridisation energy threshold and seed constraints. RIsearch2 is very good at predicting small interfering RNA (siRNA) but can also be applied to general RNA-RNA interactions. Alkan et al. (2017) suggest that RIsearch2 has great potential as a filter in general RNA-RNA interaction screens.

RIblast was developed for computational prediction of lncRNA-RNA interactions. Existing prediction tools are computationally too expensive for extensive screens of large scale lncRNA datasets. Fukunaga and Hamada (2017) introduced two major steps, a database construction and an RNA interaction search, while the latter uses the seed-and-extension approach first introduced in RIsearch2. During the database construction, RIblast calculates approximate accessibility energies of each segment in the target RNA dataset. Then the target RNA sequences are reversed and concatenated. Suffix arrays are created from these concatenated sequences. Finally, search results of short strings are pre-calculated. The approximate accessibility energies, the concatenated sequences, suffix arrays and pre-calculated search results are then stored in the database. During the RNA interaction search step, the approximate accessibility energies and a suffix array for a query sequence are computed. Then, RIblast locates seed regions with hybridisation energies that fulfil a certain threshold. This is done based on two suffix arrays of the query and the database. Then, the interaction energies are calculated by summing up the hybridisation and accessibility energies. Following this,

interactions from gap-less seed regions are extended and interactions that fully overlap others are removed. Lastly, the same is done for interactions from seed regions with gap. Fukunaga and Hamada (2017) show that RIblast performs extremely well when using the Andronescu energy parameters (Andronescu et al., 2010), both on an sRNA and a fungal snoRNA dataset as well as for the lncRNA for which it was developed. The main accomplishment is the huge runtime improvement of this method, where other methods need more than half of a month, RIblast only needs several hours.
RIsearch2 and RIblast are too recent to be featured in the benchmark of Umu and Gardner (2017), therefore it is hard to say how they compare to the other tools benchmarked by them. In their benchmark, Umu and Gardner (2017) concluded that the mfe methods were still the best performing methods, among which RNAup and IntaRNA are featured as the best methods, closely followed by RNAplex (Tafer and Hofacker, 2008). These three tools belong to the same subgroup of mfe methods that use accessibility energies. On their benchmark, the original RIsearch was by far the fastest tool, but did not perform well. Accessfold was the slowest algorithm and showed average results. They also determined that long target RNAs reduced the overall prediction quality, except for RNAplex which was able to effectively detect correct interaction sites for long RNA targets. Comparative methods are a controversial topic, as some research indicates that they could increase prediction accuracy, whereas other results indicate the opposite.
All in all, most authors agree upon the point that a lack of experimentally proven RNA-RNA interactions slows down the creation of general RNA-RNA interaction prediction tools.

Chapter 5

Future Work

The analysis of RNAStrand was conducted to give a fast and general overview over the different aspects of the problem. Therefore, not every aspect was analysed in detail. During the creation of this thesis, I learned many things trough the evaluation of results and doing additional research. Consequently, it would be interesting to further investigate some of these points.

During the analysis of RNAStrand, I investigated pseudo-knotted structures in order to draw conclusions about intermolecular structure. I did this in a very general way, treating all pseudo-knotted structures the same way. This ignores the fact that some pseudo-knots are formed by pairing with hairpin loops, others by pairing with unpaired regions. Due to the steric constraints of tertiary structures presented in the introduction, the hairpin loops allow for less flexibility than the unpaired regions. Therefore, it would be favourable to do a more concrete analysis of these types of pseudo-knotted structures.

In the analysis section, I only investigated the general distribution of unpaired region lengths in the entire dataset. I never analysed how the unpaired regions were distributed in stackings. This lead to the unpaired predictor that allows unpaired bases between every pair of base pairs. Unfortunately, this predictor did not perform as good as the predictor allowing no unpaired bases. The seed implementation in IntaRNA also allows unpaired bases, but it features a parameter restricting the overall number of unpaired bases in a seed. Currently, a heuristic is used for the unpaired case where only the best combination is saved in each step. When introducing an overall maximum, this heuristic can be removed without having a drastic increase of runtime and memory consumption, when restricting the range of the maximum.

Following this idea, I tested a helix function with a maximum unpaired bases restriction on the entire helix. Figure 5.1 shows the performance of this restricted method when setting the overall maximum to 2. It also shows the currently implemented unpaired predictor for $n_B = 10$ and $n_I = 2$ as well as the predictor allowing no unpaired bases and the original recursion. This new method outperforms the current unpaired predictor and it shows great potential. Unfortunately, there was no time to thoroughly test and analyse this new method. Therefore, it would be nice to further analyse this method in the future by testing different parametrisations as for the other predictors.

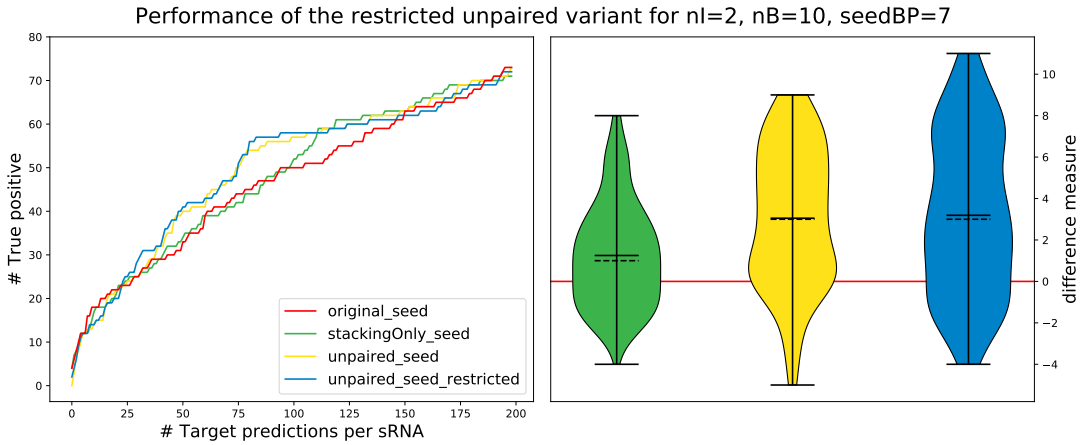


Figure 5.1: *A comparison between the current unpaired variant and potential new variant that restricts the maximum number of unpaired bases in the whole helix. All predictors, beside the original, were run for* $n_B = 10$ *and* $seedBP = 7$*. unpaired_seed was calculated with* $n_I = 2$ *and the restricted case, limited the maximum allowed number of unpaired bases to* 2*.*

Further, there are still many parameters that can be explored in order to see how they affect the helix constraints. For example, it would be interesting to see how different *seedBP* will affect the overall prediction quality.
The difference between a minimum helix length and a maximum energy value, explained in the results chapter, could imply that a minimum energy value is better suited than a fixed maximum helix length. It would be interesting to explore this in the future.
Due to a lack of experimentally proven intermolecular structures, I evaluated a database containing intramolecular structures. It would be favourable to have a large database of intermolecular structures to analyse. This would provide more valuable information and allow improvements on RNA-RNA interaction prediction tools. At the same time, the existing benchmarks could be improved.

Chapter 6

Conclusion

In this thesis, I introduced a new maximum helix length constraint to IntaRNA. It is based on the idea that intermolecular helices cannot become arbitrarily long due to steric constraints of the tertiary structure. I showed empirically that my claims are not unjustified by analysing the helix length distributions of the RNA structures from the RNAStrand database. I introduced new prediction modi to IntaRNAv2 and thoroughly tested them on a newly created benchmark. Nevertheless, I was not able to try all parametrisations I intended as the computation times were still too high for the hardware I used.

I have shown that the predictor that allows no unpaired bases in the helices had the overall best performance for all parametrisations tested. The unpaired predictor was likely too generalised and I assume that the restricted version of this predictor, as suggested in the future work chapter, will greatly improve the prediction quality.

Moreover, the length limitation improved the overall runtime of IntaRNA, while keeping the memory consumption roughly the same.

Furthermore, I have demonstrated that limiting the maximum helix length too much, reduces the overall prediction quality. On the other hand, a higher minimum helix length, simulated by a maximum energy parameter clearly increased the prediction quality, while further reducing the runtime.

The main limitation of this work, is the lack of a database containing intermolecular structures to analyse. The RNAStrand database allowed a general overview due to a certain likeness between intra- and intermolecular structures, but it does not allow concrete observations.

All in all, I can say that the introduction of a maximum helix length shows great promise and needs to be further investigated.

Bibliography

Alkan, F., Wenzel, A., Palasca, O., Kerpedjiev, P., Rudebeck, A. F., Stadler, P. F., Hofacker, I. L., and Gorodkin, J. (2017). RIsearch2: suffix array-based large-scale prediction of RNA-RNA interactions and siRNA off-targets. *Nucleic Acids Res.*, 45(8):e60.

Andersen, E., Rosenblad, M., Larsen, N., Westergaard, J., Burks, J., Wower, I., Wower, J., Gorodkin, J., Samuelsson, T., and Zwieb, C. (2006). The tmRDB and SRPDB resources. *Nucleic Acids Research*, 34:163–168.

Andronescu, M., Bereg, V., Hoos, H. H., and Condon, A. (2008). RNA STRAND: The RNA Secondary Structure and Statistical Analysis Database. *BMC Bioinformatics*, 9(1):340.

Andronescu, M., Condon, A., Hoos, H. H., Mathews, D. H., and Murphy, K. P. (2010). Computational approaches for RNA energy parameter estimation. *RNA*, 16(12):2304–2318.

Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Wheeler, D. L. (2008). GenBank. *Nucleic Acids Res.*, 36(Database issue):25–30.

Berman, H., Olson, W., Beveridge, D., Westbrook, J., Gelbin, A., Demeny, T., Hsieh, S., Srinivasan, A., and Schneider, B. (1992). The nucleic acid database. a comprehensive relational database of three-dimensional structures of nucleic acids. *Biophys J*, 63:751–759.

Borer, P. N., Dengler, B., Tinoco, I., and Uhlenbeck, O. C. (1974). Stability of ribonucleic acid double-stranded helices. *Journal of Molecular Biology*, 86(4):843 – 853.

Bouvier, M., Sharma, C. M., Mika, F., Nierhaus, K. H., and Vogel, J. (2008). Small RNA binding to 5' mRNA coding region inhibits translational initiation. *Mol. Cell*, 32(6):827–837.

Brennecke, J., Stark, A., Russell, R. B., and Cohen, S. M. (2005). Principles of microRNA-target recognition. *PLoS Biol.*, 3(3):e85.

Brown, J. (1999). The Ribonuclease P Database. *Nucleic Acids Research*, 27:314–314.

Brown, J., Haas, E., Gilbert, D., and Pace, N. (1994). The Ribonuclease P Database. *Nucleic Acids Research*, 22:3660–3662.

Busch, A., Richter, A. S., and Backofen, R. (2008). IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics*, 24(24):2849–56.

Cannone, J., Subramanian, S., Schnare, M., Collett, J., D'Souza, L., Du, Y., Feng, B., Lin, N., Madabusi, L., Muller, K., Pande, N., Shang, Z., Yu, N., and Gutell, R. (2002). The comparative RNA web (crw) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, pages 3–15.

Condon, A., Davy, B., Rastegari, B., Zhao, S., and Tarrant, F. (2004). Classifying RNA Pseudoknotted Structures.

DeVoe, H. and Tinoco, I. (1962). The stability of helical polynucleotides: Base contributions. *Journal of Molecular Biology*, 4(6):500 – 517.

DiChiacchio, L., Sloma, M. F., and Mathews, D. H. (2016). AccessFold: predicting RNA-RNA interactions with consideration for competing self-structure. *Bioinformatics*, 32(7):1033–1039.

Doench, J. G. and Sharp, P. A. (2004). Specificity of microRNA target selection in translational repression. *Genes Dev.*, 18(5):504–511.

Fukunaga, T. and Hamada, M. (2017). RIblast: an ultrafast RNA-RNA interaction prediction system based on a seed-and-extension approach. *Bioinformatics*, 33(17):2666–2674.

Gotoh, O. (1982). An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162(3):705–708.

Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S., and Bateman, A. (2005). Rfam: annotating non-coding rnas in complete genomes. *Nucleic Acids Research*, 33:121–124.

Isaac, B. (2005). Prediction and validation of microRNAs and their targets. *FEBS Letters*, 579(26):5904–5910.

Jaynes, E. T. (1957). Information Theory and Statistical Mechanics. *Physical Review*, 106:620–630.

Lorenz, R., Bernhart, S. H., Höner zu Siederdissen, C., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, 6(1):26.

Mann, M., Wright, P. R., and Backofen, R. (2017). IntaRNA 2.0: enhanced and customizable prediction of RNA-RNA interactions. *NAR*.

Markham, N. R. and Zuker, M. (2008). *UNAFold*, pages 3–31. Humana Press, Totowa, NJ.

Mathews, D. H., Sabina, J., Zuker, M., and Turner, D. H. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288(5):911 – 940.

McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119.

Mückstein, U., Tafer, H., Hackermüller, J., Bernhart, S. H., Stadler, P. F., and Hofacker, I. L. (2006). Thermodynamics of RNA–RNA binding. *Bioinformatics*, 22(10):1177–1182.

Raden, M. and Backofen, R. (2018). Lecture Material: RNA Bioinformatics.

Richter, A. S. (2012). *Computational analysis and prediction of RNA-RNA interactions*. PhD thesis, University of Freiburg.

Sprinzl, M. and Vassilenko, K. (2005). Compilation of trna sequences and sequences of trna genes. *Nucleic Acids Research*, 33:139–140.

Tafer, H. and Hofacker, I. L. (2008). RNAplex: a fast tool for RNA-RNA interaction search. *Bioinformatics*, 24(22):2657–2663.

Thirumalai, D. (1998). Native secondary structure formation in RNA may be a slave to tertiary folding. *Proceedings of the National Academy of Sciences*, 95(20):11506–11508.

Tinoco, I. and Bustamante, C. (1999). How RNA folds. *Journal of Molecular Biology*, 293(2):271 – 281.

Tjaden, B., Goodwin, S. S., Opdyke, J. A., Guillier, M., Fu, D. X., Gottesman, S., and Storz, G. (2006). Target prediction for small, noncoding RNAs in bacteria. *Nucleic Acids Res.*, 34(9):2791–2802.

Turner, D. H. and Mathews, D. H. (2010). NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research*, 38:D280–D282.

Umu, S. U. and Gardner, P. P. (2017). A comprehensive benchmark of RNA-RNA interaction prediction tools for all domains of life. *Bioinformatics*, 33(7):988–996.

Vazquez-Anderson, J. and Contreras, L. M. (2013). Regulatory RNAs: charming gene management styles for synthetic biology applications. *RNA Biol*, 10(12):1778–1797.

Wenzel, A., Akbasli, E., and Gorodkin, J. (2012). RIsearch: fast RNA-RNA interaction search using a simplified nearest-neighbor energy model. *Bioinformatics*, 28(21):2738–2746.

Westbrook, J., Feng, Z., Chen, L., Yang, H., and Berman, H. (2003). The Protein Data Bank and structural genomics. *Nucleic Acids Research*, 31:489–491.

Wright, P. R. (2016). *Predicting small RNA targets in prokaryotes - a challenge beyond the barriers of thermodynamic models.* PhD thesis, Albert-Ludwigs-University Freiburg.

Zuker, M. and Stiegler, P. (1981). Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research 9.1*, pages 133–48.