COMPUTATIONAL METHODS FOR RNA-RNA INTERACTION PREDICTION

by

Raheleh Salari
B.Sc., Sharif University of Technology, 2003
M.Sc., Sharif University of Technology, 2005

a Thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Summer 2010

APPROVAL

Name: Raheleh Salari

Degree: Doctor of Philosophy

Title of Thesis: Computational Methods for RNA-RNA Interaction Prediction

Examining Committee: Dr. Funda Ergun
Chair

Dr. S. Cenk Sahinalp, Senior Supervisor
Computer Science, Simon Fraser University

Dr. Anne Condon, Supervisor
Computing Science, University of British Columbia

Dr. Peter Unrau, SFU Examiner,
Molecular Biology and Chemistry, Simon Fraser University

Dr. Peter F. Stadler, External Examiner,
Computer Science and Bioinformatics, Leipzig University

Date Approved: July 29, 2010

Declaration of Partial Copyright Licence

Abstract

Non-Coding RNAs (ncRNAs) such as microRNAs play an important role in the gene regulation. Studies on both prokaryotic and eukaryotic cells show that ncRNAs usually bind to their target mRNA to regulate the translation of corresponding genes. Therapeutic applications of RNA interference and antisense RNA regulation strongly motivate the problem of predicting whether two RNAs interact. In the past few years, high-throughput sequencing technologies have identified a large set of new regulatory ncRNAs, but the number of experimentally verified targets is considerably low. Thus, computational target prediction methods are in high demand. Current methods for predicting ncRNA-target mRNA interactions suffer from low specificity and accuracy. Moreover, their high computational complexity makes them impractical for genome-wide target prediction problems.

In this dissertation, we present fast and accurate computational methods for prediction and analysis of binding thermodynamics between two RNAs, typically oligonucleotides and target RNAs. We develop a partition function algorithm to compute the stability and probability of binding between two RNAs. Partition function is a scalar value from which various thermodynamic quantities can be derived. For example, the equilibrium concentration of each complex nucleic acid species, the heat capacity and the melting temperature of interacting nucleic acids can be calculated based on the partition function of the complex.

In order to reduce the time and space requirements of the computational RNA-RNA interaction prediction problem, we introduce an efficient algorithm that can predict the optimal interaction between two RNAs. Our algorithm applying a technique called *sparsification* has been able to reduce both time and space requirements of the interaction prediction by a linear factor. Finally, we propose a fast heuristic method for multiple binding sites prediction, based on the site accessibility and binding probabilities, that can be used for genome-wide target prediction problems.

To my family with love

“Don’t be afraid, we’re all together!”

— Iranian protesters, June 2009

Acknowledgments

It is a great pleasure that I would like to thank those who have supported me in any respect during my Ph.D. study.

First and foremost, my utmost gratitude to my supervisor, Prof. Cenk Sahinalp, for his encouragement, guidance and support from the initial to the final level of my Ph.D. study. Beside the technical skills in the computational biology, I have learned the necessary requirements of being a good researcher from him. I thank him most sincerely for sharing his knowledge and experience.

I offer my regards and blessings to my supervisor Prof. Anne Condon, my SFU examiner Dr. Peter Unrau, and my external examiner Prof. Peter Stadler who accepted to be in my Ph.D. committee and helped me with valuable discussions and comments.

I was delighted to work with Prof. Rolf Backofen. His nice personality and his energy in doing research have been always an inspiration for me. I would like to thank my coauthors Dr. Mathias Möhl and Dr. Sebastian Will who have made a remarkable contribution to the work. Our collaboration is an enjoyable experience for me.

Great thanks to my wonderful friends for their support and friendship during these years. My friends from Lab for Computational Biology at SFU who made the office a pleasant active environment, Fereydoun, Iman, Faraz, Farhad, Phoung, Alex, and Emre. My dear friends Behrooz, Arina, and Saba who have been with me through these years and did not let me to feel loneliness even for a single moment.

Last, but of course not least, I heartily thank my family for the strong motivation that they gave me to follow my studies. While studying abroad, having their support was the most valuable thing for me.

Raheleh Salari

Contents

List of Tables

List of Figures

Chapter 1

Introduction

RNA (ribonucleic acid) is an important type of molecule inside living cells which exhibits several functions. Until recently RNA was thought to have only two functions: (i) to transmit information between DNA and proteins in the form of a messenger RNA (mRNA) and (ii) to decode information in protein synthesis in the form of a ribosomal RNA (rRNA) or a transfer RNA (tRNA). However, the discovery of microRNAs (miRNAs) and the advent of genome-wide transcriptomics have shown that RNA plays several important roles in living organisms that extend far beyond being a mere intermediate in protein biosynthesis [85].

A large fraction of the genome sequences consists of RNA transcripts that do not code for proteins and are called Non-coding RNAs (ncRNAs). Non-coding RNA genes produce highly abundant and functionally important RNAs. Non-coding RNAs can interact with proteins, small molecules, and other RNAs. An ncRNA, by binding to a protein, controls its activity and accessibility. A Riboswitch, which is part of an mRNA molecule, binds to a small molecule and causes critical changes in the associated gene's activity. But in fact most of the ncRNAs exhibit their functions through interaction with other RNA molecules. Regulatory ncRNAs play a crucial role in gene expression post-transcriptionally through base pairing with a target mRNA as per the eukaryotic miRNAs and small interfering RNAs (siRNAs) [13, 44, 102], antisense RNAs [95, 18], or bacterial small regulatory RNAs (srRNAs) [38]. It has been predicted that only miRNAs regulate at least one-third of all human genes [57].

Regulatory RNAs such as miRNAs and siRNAs are usually very short and have full sequence complementarity to the targets. However some of the regulatory antisense RNAs are relatively long and are not fully complementary to their target sequences. They exhibit their regulatory functions by establishing stable joint structures with target mRNA initiated by one or more loop-loop interactions.

1.1 Motivation

RNA interference (RNAi) is a process in which RNA molecules such as miRNAs and siRNAs bind to their target mRNAs and cause translational repression or activation, mRNA degradation, or changes in mRNA stability. RNAi has three known biological functions: (i) to defend cells against foreign genes such as viruses and transposons, (ii) to trigger the RNA-induced silencing complex (RISC) to cleave specific mRNAs, (iii) to increase gene transcription as part of a promoter. Therefore, RNAi mechanism provides a natural way to control the activity of specific genes.

RNAi technology is an invaluable research tool, allowing much more rapid characterization of the function of known genes. Large-scale functional genomics approaches for target identification in human cells are indebted to RNAi technology. This technology can reveal novel genes involved in disease processes. Applications of RNAi screening for the identification of novel genes implicated in apoptosis, cell division, and drug resistance support the enormous promise of this technology [14].

RNAi has several applications in drug discovery and therapeutics, particularly for viral infections, cancer, and brain diseases. Small interfering RNAs (siRNAs) and short hairpin RNAs (shRNAs) are used to block replication of specific viruses. For cancer therapy several RNAi-based strategies such as the inhibiting of over-expressed oncogenes, promoting apoptosis, regulating cell cycle, antiangiogenesis and enhancing the efficacy of chemotherapy and radiotherapy have been designed. Although RNAi technology has become an excellent therapeutic strategy, yet RNAi-based drug designing is not as straightforward as it was thought to be. An RNAi experiment should deal with difficulties in different steps including designing an siRNA sequence with specific structural characteristics, delivering it into the cells, and finally evaluating its result at both the mRNA and the protein levels with minimum off-targeting effect [67].

These days more and more applications of RNAi in drug discovery and treatment of diseases are discovered. Although still no siRNA is approved for medical use, a number of R&D initiatives and clinical trials are currently underway. During past years, several studies have demonstrated the role of RNAi in treatment of diseases such as hemophilia, hepatitis B, hepatitis C, and HIV. In April 2010, GenomeWeb News released a list of 10 new potential RNAi-based drugs. One example of these drugs is TransDerm, a treatment for the rare skin disorder Pachyonychia congenital caused by a mutation in any one of the dozens of genes encoding keratins. The drug targets a particular mutation in one of these genes. In June 2010, an exciting discovery [70] was published that suggests that miRNA can be a therapeutic target for increasing good cholesterol.

In addition to endogenous regulatory RNAs in RNAi, antisense oligonucleotides perform as exogenous inhibitors of gene expression. In eukayotes, antisense RNAs interactions are involved in several biological processes such as splicing, RNA editing, rRNA modification, and development regulation. Antisense technology has been also used as a research tool for therapeutic purposes. Antisense oligonucleotides can be fed to metabolic networks for specific control of the the metabolism. Blocking the production of disease-causing proteins, artificial regulatory RNA molecules promise to treat human diseases such as cancer, rheumatoid arthritis, brain diseases, and viral infections. However, the drug Fomivirsen [39] is the only antisense-based drug which has reached the market so far. Antisense technology needs further biological studies and analysis to be as effective as expected in disease therapy.

With recent progress in sequencing methods, a huge set of ncRNAs has become known. However there is no high throughput method to detect their associated targets. Consequently, there are increasing interests in computational target prediction methods. The first set of computational methods for predicting ncRNA-target mRNA interactions suffered from over-simplifying the types of interactions allowed. As a result they could not accurately predict many known interactions, particularly those involving long ncRNAs. More precisely, these methods either restricted the interactions to external positions, or they allowed interactions with at most one interaction site. These restrictions were lifted by two independently developed methods, which provided the first set of algorithms for predicting a precise interaction structure of two RNA strands: (i) the algorithm by Pervouchine [74] maximizes the total number of base pairs, and (ii) a more general method by Alkan et al. [4], minimizes the total free energy of the interacting RNA strands using a nearest neighbor energy model. Alkan et al. also provide proof of the NP-hardness of the general problem, together with a precise definition of interaction types that can be handled in polynomial time.

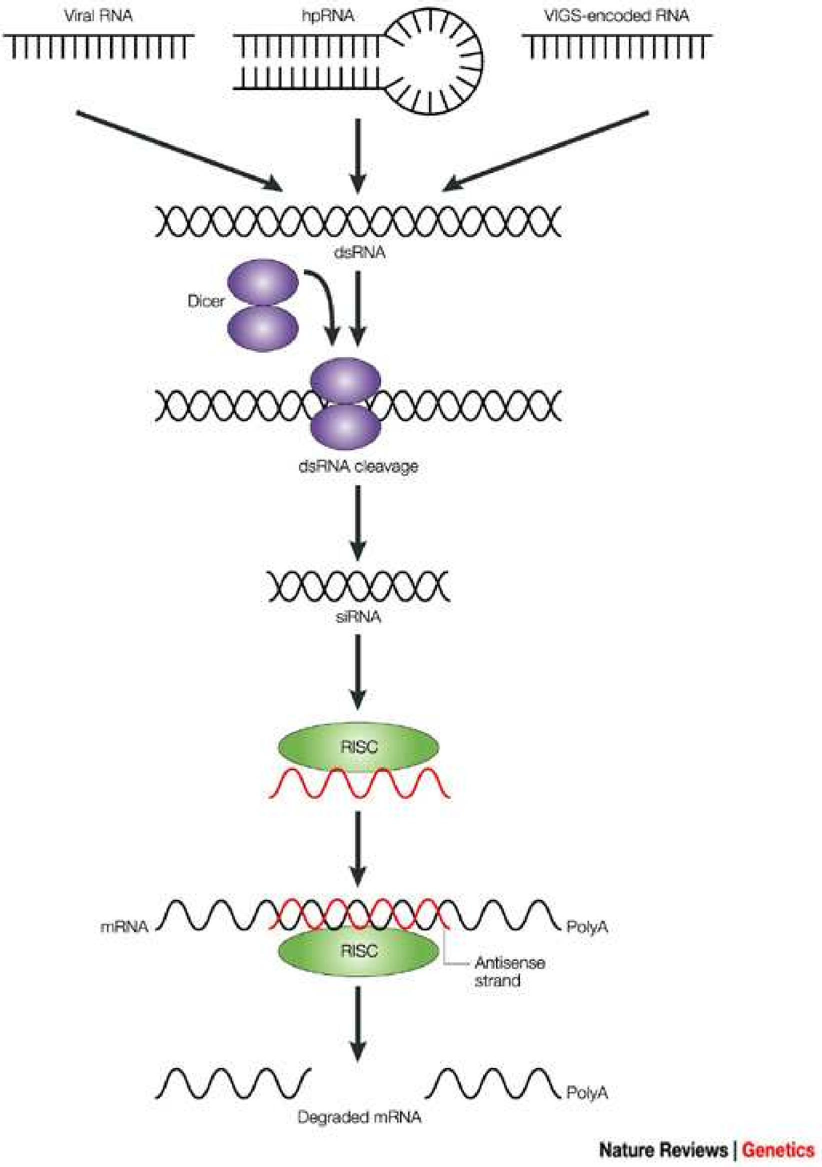


Figure 1.1: RNA interference mechanism [97]. Double-stranded RNA (dsRNA) from replicating viral RNA, viral-vector-derived (VIGS, or virus-induced gene silencing) RNA or hairpin RNA (hpRNA) transcribed from a transgene, is processed by a Dicer-containing complex to generate siRNAs. The antisense strand of siRNA incorporated into the RNA-induced silencing complex (RISC) bind to its target mRNA. Depending on the degree of complementarity between the siRNA and its target mRNA, RISC may either block the translation machinery or cleave the target.

One key problem with the above approaches for predicting a general joint structure [74, 4] is that they all have a worst case running time of $O(n^6)$ and a space complexity of $O(n^4)$. While this complexity might be acceptable when analyzing only a few putative sRNA-target interaction pairs, we are now faced with a situation in which the amount of data to be analyzed is vastly increasing. To give an example, a recent mapping of transcripts using tiling arrays in the budding yeast *S. cerevisiae* [25] with 5,654 annotated open reading frames (ORF) has found 1555 antisense RNAs that overlap at least partially with the ORFs at the opposite strand. Currently, it is unclear what these antisense RNAs are doing - whether they target only their associated sense mRNA or have also other mRNA targets, and whether they always form a complete duplex or more complex joint structures such as multiple kissing hairpins. The same situation appears in many other species. Thus, there is an urgent need for a time and space efficient interaction prediction method that is able to handle complex joint structures.

In order to characterize the effectiveness of the predicted interaction, it needs to be further analyzed by a quantitative analysis of binding thermodynamics between oligonucleotides and target RNA. The specificity of interaction depends on the stability of intermolecular and intramolecular base pairs. Therefore, a method which accurately computes the probability and stability of interactions between two RNAs is greatly in demand.

1.2 Contributions

In this thesis, we (re)define the problem of computationally predicting the interaction between two RNAs and explain the main challenges for a general computational method for the RNA-RNA interaction prediction problem. Our goal is to design and implement an efficient computational method to be reliably used for target prediction purposes while providing quantitative analysis of binding effectiveness. More specifically, we present the following contributions:

- We introduce our energy model for interaction between nucleic acid strands (published in [22]). Our interaction energy model is an extension of the nearest neighbor thermodynamics energy model for RNA secondary structure that contains new components for joint secondary structure between two RNAs. Energy functions for the interaction components are defined in a similar way to the other nearest neighbor thermodynamics rules. The interaction energy model uses validated thermodynamics parameters in the standard energy model and defines only a few new parameters related to the interaction components. These new parameters are trained over a set of known short interactions.

- We develop a dynamic programming algorithm to compute the *interaction partition function* over the whole ensemble of almost all physically possible individuals and joint secondary structures (published in [22]). Our algorithm with $O(n^6)$ time and $O(n^4)$ space complexity considers the most general type of interactions introduced in the literature. Our partition function algorithm can be used to compute various thermodynamic quantities such as the equilibrium concentration of each complex nucleic acid species, heat capacity, and the melting temperature of interacting nucleic acids. We verify our algorithm by computing (i) the equilibrium concentration for the OxyS-fhlA complex and (ii) the melting temperature for RNA pairs available in the literature. In both experiments our algorithm shows high accuracy.

- We show how to reduce both time and space complexity of the minimum total free energy for the joint structure using a technique called *sparsification*. Sparsification technique uses the observation that the resulting DP-matrices are sparse. As in previous applications of sparsification to problems related to RNA folding, our approach exploits a triangle inequation on the dynamic programming matrix. Assuming the *polymer-zeta* property for interacting RNAs, we show an efficiency gain by a linear factor. This *polymer-zeta* property basically states that the probability of a base pair decreases with its size, i.e. there are only few long range base pairs. Our sparsified algorithm (published in [80]) reduces the complexity of the original algorithm from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n) + n^3)$ space for some function $\psi(n)$, which turns out to have small values for the range of n that we encounter in practice. Under the assumption that the polymer-zeta property holds for RNA-structures, we demonstrate that $\psi(n) = O(n)$ on average, resulting in a linear time and space complexity improvement over the original algorithm.

- There are several evidences [19] suggesting that interaction is a multi-step process that involves: (i) unfolding of the two RNA structures to expose the bases needed for hybridization, (ii) the hybridization at the binding site, and (iii) restructuring of the complex to a new minimum free energy conformation. We present a heuristic approach (published in [79]) that can predict interactions involving multiple binding sites by: (i) identifying the collection of accessible regions up to a maximum length w for both input RNA sequences, (ii) using a matching algorithm, computing a set of ”non-conflicting” interactions between the accessible regions which have the highest overall probability of occurrence. Our method computes the most probable non-conflicting matching of accessible regions with $O(n^2w^4 + n^3/w^3)$ time and $O(w^4 + n^2/w^2)$ space complexity., where w is the window size for of accessible region.

The software is implemented in a C++ package and can be accessed through our website for *taveRNA:RNA suite* at ”http://www.compbio.cs.sfu.ca/taverna”.

1.3 Organization of the thesis

The rest of the thesis is organized as follows. In Chapter 2, we first describe the problem of RNA-RNA interaction prediction. Then we present an overview of the existing related computational approaches and summarize the general issues related to the previous works. In Chapter 3, an interaction energy model which is an extension of the RNA secondary standard energy model is presented. This model can handle different interaction components in an RNA-RNA joint secondary structure. Chapter 4 introduces our partition function algorithm for two interacting nucleic acid strands. After explaining the details and recursion cases of the algorithm, several applications of its implementation `piRNA` are discussed. At the end of the chapter, the algorithm is verified through experiments for computing (i) the equilibrium concentration for OxyS-fhlA complex and (ii) the melting temperature for RNA pairs available in the literature. Later, possible solutions for the problem of high complexity requirements of RNA-RNA interaction prediction methods are studied. In Chapter 5, we introduce a technique called *sparsification*. We develop a sparsified algorithm that can predict the optimal interaction between two RNAs. We show our algorithm achieves an efficiency gain by a linear factor in comparison to the original algorithm for RNA-RNA interaction prediction. Chapter 6 explains our fast heuristic algorithm for multiple binding sites prediction, based on the site accessibility and binding probabilities, that can be used for genome-wide target prediction problems. Finally, in Chapter 7 we offer a summary and conclusion of our contributions to the RNA-RNA interaction prediction problem, as well as a discussion of possible directions for future work.

Chapter 2

Definition and Background

In this chapter, we define the general problem of interaction prediction between two RNA molecules. We first present some preliminary definitions related to the RNA sequence, structure and interaction used through text or figures of the thesis. Since the interaction between two RNAs can be considered as their joint secondary structure, we start by a short review on the single RNA secondary structure prediction problem. A short introduction on the structure prediction problem, the energy model of folding and some major strategies are described here. Later we define the RNA-RNA interaction prediction problem and discuss about the current approaches for this problem.

2.1 Preliminaries

The two RNAs are denoted by **R** and **S**. Strand **R** is indexed from 1 to L_R in $5'$ to $3'$ direction and **S** is indexed from 1 to L_S in $3'$ to $5'$ direction. Note that the two strands interact in opposite directions, e.g. **R** in $5' \rightarrow 3'$ with **S** in $3' \leftarrow 5'$ direction. Each nucleotide is paired with at most one nucleotide in the same or the other strand. The subsequence from the i^{th} nucleotide to the j^{th} nucleotide in a strand is denoted by $[i, j]$. We refer to the i^{th} nucleotide in **R** and **S** by i_R and i_S respectively. An intramolecular base pair between the nucleotides i and j in a strand is called an *arc* and denoted by a bullet $i \bullet j$. Two arcs $i \bullet j$ and $i' \bullet j'$ are *pseudoknot* if $i < i' < j < j'$ or $i' < i < j' < j$. An intermolecular base pair between the nucleotides i_R and i_S is called a *bond* and denoted by a circle $i_R \circ i_S$. Two bonds $i_R \circ i_S$ and $j_R \circ j_S$ are called *crossing bonds* if $i_R < j_R$ and $i_S > j_S$ or $i_R > j_R$ and $i_S < j_S$. An interaction arc $i_R \bullet j_R$ in R *subsumes* a subsequence $[i_S, j_S]$ in S if there is at least one bond $k_R \circ k_S$, where $i_R < k_R < j_R$ and $i_S < k_S < j_S$, and for all bonds $k_R \circ k_S$, if $i_S \leq k_S \leq j_S$ then $i_R < k_R < j_R$. Analogously, interaction arcs in S can subsume subsequences in R. Two interaction arcs $i_R \bullet j_R$ and $i_S \bullet j_S$ are part of a *zigzag*, if there is a bond $k_R \circ k_S$, where $i_R < k_R < j_R$ and $i_S < k_S < j_S$, but neither $i_R \bullet j_R$ subsumes $[i_S, j_S]$ nor $i_S \bullet j_S$ subsumes $[i_R, j_R]$.

2.1.1 Recursion Diagrams

In this thesis dynamic programming (DP) algorithms are represented in a graphical notation using the recursion diagrams. Within the recursion diagrams, a horizontal line indicates the phosphate backbone, a solid curved line indicates an arc, and a dashed curved line encloses a region and denotes its two terminal bases which may be paired or unpaired. Letters within a region specify a recursive quantity. White regions are recursed over and blue regions indicate those portions of the secondary structure that are fixed at the current recursion level and contribute to the energy as defined by the energy model. Green and red regions have the same recursion cases as the corresponding white regions, except that for the green regions multiloop energy and for red regions kissing loop energy is applied, i.e. the corresponding penalties for each unpaired base and base pair should be applied. A solid vertical line indicates a bond, a dashed vertical line denotes two terminal bases of a region which may be base paired or unpaired, and a dotted vertical line denotes two terminal bases of a region which are assumed to be unpaired. A terminal determined by $\bullet$ is starting point of either an interaction arc or a bond.

2.2 RNA Secondary Structure

RNA molecule is a linear polymer in which the nucleotides are linked together by means of phosphodiester bridges, or bonds. The base types are Adenine (A), Cytosine (C), Guanine (G), and Uracil (U). In a process called hybridization, some pairs of nucleotides from one or two different RNAs creates hydrogen bonds. Two nucleotides that are connected via hydrogen bonds are called a base pair. The canonical Watson-Crick base pairs, (A-U) and (G-C) are the strongest ones. The thermodynamic stability of a wobble base pair (U-G) is also comparable to that of a Watson-Crick base pair. Base pairing through alternate hydrogen bonding patterns are non-canonical.

A set of base pairs between the nucleotides of an RNA strand forms an RNA secondary

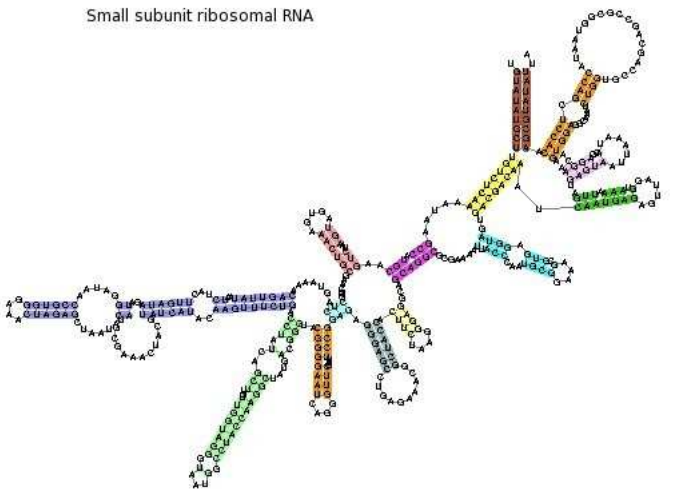


Figure 2.1: Example of RNA secondary structure (from Wikipedia).

structure. Figure 2.1 shows the secondary structure of a small subunit of ribosomal RNA. For many RNA molecules, structure of RNA molecule determines both function and mechanism behind that function. During past decades several experimental and computational techniques have been developed to determine the RNA structures.

Experimental methods. Here we briefly mention some of the experimental approaches to analyze the RNA structures (please see [33] for a survey). One of the earliest method to determine the structure of novel RNAs is structure probing of nucleic acids. This technique can determine individual components of an existing structure such as the existence of a given base pair. Structure probing analysis can be done through many different methods, which include chemical probing, hydroxyl radical probing, nucleotide analog interference mapping (NAIM), and in-line probing.

The other widely used approach is X-ray crystallography. Crystal structures of proteins began to be solved in the late 1950s. Before 2000, only three RNA crystal structures were available. Since 2000, the structure determination of large and complex RNAs by X-ray crystallography has been achieved, initiated by the analysis of the ribosomal subunits, and now including several structures. Improvements in techniques for the synthesis, purification, crystallization and derivatization of large RNAs, as well as the development of advanced software, was essential for these spectacular achievements. Although the number of RNA structure determinations has grown slowly, the average structure size has dramatically increased. The closest competing method is nuclear magnetic resonance (NMR) spectroscopy. Structure determination by NMR spectroscopy usually consists of several following phases, each using a separate set of highly specialized techniques. The sample is prepared, resonances are assigned, restraints are generated and a structure is calculated and validated. Crystallography can solve structures of arbitrarily large molecules, whereas NMR is restricted to relatively small ones (less than 70 kilodaltons).

A structural technique specialized for visualizing dynamic macromolecular complexes of 200 kilodaltons or larger, including RNAs, is single-particle cryo-electron microscopy (cryo-EM). There are no size limitation. Cryo-EM of ribonucleoproteins combined with single-particle reconstruction enables the visualization of each of its transitional states that can be efficiently trapped. Cryo-EM maps are used to fit high-resolution X-ray structures, when available, illustrating the complementarities between the two experimental approaches.

Although traditional methods (e.g. chemical probing, and mass spectrometry) are replaced by recent modern technologies such as X-ray crystallography, Nuclear Magnetic Resonance (NMR) spectroscopy, and Cryo-electron microscopy (Cryo-EM), the experimental methods are still time consuming and costly processes. Consequently, only few RNA secondary structures have been experimentally determined.

Computational methods. The problem of computationally predicting RNA secondary structure was first introduced more than thirty years ago. Although the problem is one of the earliest in computational biology, it has attracted some fresh attention due to the recent discoveries of new classes of non-coding RNAs. Additional interest in the problem comes from the study of complete genomes of RNA viruses.

Several definitions for the "optimal" RNA secondary structure has been presented. Usually it is interpreted as the most stable structure under a specific energy model. Also it can be defined as the most conserved secondary structure. Recently a notion of centroid has been introduced as the most common structure. Based on this notion the most common structure is the one with the minimum distance from the all possible structures (center) in Boltzmann distribution.

Definition (RNA secondary structure prediction problem) Given an RNA sequence $\mathbf{R}$, compute the set of canonical base pairs $\{i \bullet j | 1 \leq i < j \leq n\}$ of the optimal secondary structure.

The following constraints are usually imposed on RNA secondary structure problem:

- Multi pairing is prohibited - each base i is paired with at most one base j.
- Sharp U-turn is prohibited - if $i \bullet j$ is a base pair then $j > i + 3$.
- Pseudoknot is prohibited - if $i \bullet j$ and $i' \bullet j'$ are two base pairs such that $i < i'$, then $i < j < i' < j'$.

2.2.1 RNA secondary structure prediction methods

One of the earliest attempts to predict RNA secondary structure was made by Nussinov and co-workers who used dynamic programming method for maximizing the number of base pairs [72]. Nussinov algorithm simply works as follows. Let $N^{\mathbf{X}}(i,j)$ (for $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$) denotes the maximum number of base pairs of the subsequence $[i,j]$ of the single sequence $\mathbf{X}$. The dynamic programming formulation of $N^{\mathbf{X}}(i,j)$, corresponding to the recursion cases in Figure 5.1, are calculated by the following recursions

$$N^{\mathbf{X}}(i,j) = \max \begin{cases} N^{\mathbf{X}}(i+1,j) & (a) \\ \max\limits_{\substack{i<k\leq j \\ X[i],X[k] \text{ complementary}}} \begin{pmatrix} 1 + N^{\mathbf{X}}(i+1,k-1) \\ + N^{\mathbf{X}}(k+1,j) \end{pmatrix} & (b) \end{cases} \tag{2.1}$$

In case (a) base $X[i]$ is not part of any base pair and in case (b) base $X[i]$ forms a base pair with base $X[j]$. The complexity of the algorithm is $O(n^3)$ with memory requirement of $O(n^2)$. Since the base pair counting model used in Nussinov algorithm is in fact a basic additive energy model, the algorithm can be easily adapted to calculate the optimal structure based on a more accurate additive energy model such as nearest neighbor thermodynamic model (NNTM).

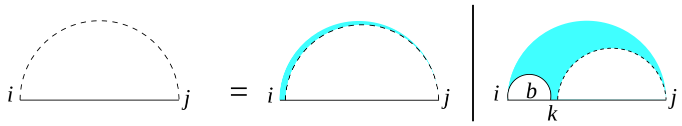


Figure 2.2: Recursion cases for computing the maximum base pairing secondary structure of $[i, j]$.

Nussinov et al. published an adaptation of their approach to use a simple NNTM in 1980 [71]. Later Zuker et al. [105] and Lyngso et al. [61] refined dynamic programming approach for the more accurate versions of NNTM. The popular `mfold` [104] and its more efficient version `RNAfold` [47] (from the `Vienna package`) are implementations of these algorithms.

In addition, some methods have been employed to speed up the dynamic programing algorithm of RNA folding. Using Valiant's approach, Akutsu [2] showed how to reduce the worst case running time of RNA folding problem to $O(n^3(\log\log n)/(\log n)^{1/2})$. Valiant approach [93] is a classical algorithm for context-free recognition in less than cubic time. That approach combining the new paper on the all-pairs shortest path problem [21] can achieve a worst case bound of $O(n^3(\log^3 \log n)/(\log^2 n))$ [91]. Moreover, an approach based on Four-Russian algorithm has been presented to speed up the minimum free energy RNA structure perdition [36] to $O(n^3/\log(n))$. Four-Russion technique is a general strategy to achieve a worst-case speed-up of dynamic programming.

Recently, the dynamic programming sparsification technique has been used to reduce the time and space complexity of calculation of the minimum free energy structure for a single RNA sequence folding [98, 11]. As a result a (roughly) linear reduction in the time and space complexity was achieved on average. To be more precise, the time complexity of RNA-folding was reduced from $O(n^3)$ to $O(nZ)$ and the space complexity was reduced from $O(n^2)$ to $O(Z)$, where Z is a sparsity factor satisfying $n \leq Z \leq n^2$. An estimation [98] of the expected value of a parameter related to Z, based on a probabilistic model for polymer folding and measured by simulations, shows that Z is significantly smaller than $O(n^2)$. The time and space reduction is based on the assumption that RNA structures or consensus structures, in the simultaneous alignment and folding of RNAs, satisfy the *polymer-zeta* behavior. The polymer-zeta property states that in any long polymer chain the probability of having arc between two monomers with distance m converges to $b.m^{-c}$, where $b, c > 0$

are some constants.

In 1990, McCaskill introduced a recursive approach to compute the partition function and base pair probabilities for RNA secondary structures in Boltzmann ensemble [65]. In Boltzmann distribution the probability of each structure is related to its thermodynamics free energy. The minimum free energy structure is the most probable structure in the entire Boltzmann distribution over the set of all possible RNA secondary structures. One alternative approach for RNA secondary structure prediction is to estimate the centroid structure in the Boltzmann distribution that maximizes the expectation of the accuracy of prediction [29, 42, 59]. If the similarity of two structures is defined as the number of their common base pairs, the centorid structure is simply the structure formed of base pairs with probability higher than 1/2.

Stochastic context-free grammars (SCFGs) have been also employed as a probabilistic model for RNA secondary structure [53]. The features and their associated parameters of SCFGs specify probability distributions over possible transformations which they may not have any physical meaning. In this approach statistical learning algorithms are used to drive such parameters. `CONTRAfold` program [32] which is based on conditional log-linear model (CLLM), a probabilistic model which generalizes upon SCFG, attracted a lot of interest due to its high accuracy in RNA folding problem.

An alternative direction in RNA secondary structure prediction aims to improve the predictive power of single RNA folders by simultaneously predicting the structure of two or more functionally similar RNA sequences. This general approach typically aims to optimize a linear combination of (i) the free energy of the *alignment/consensus* of the RNA sequences and (ii) a score derived from covarying mutations or common motifs among the RNA sequences.

There are two basic flavors of this general approach. The first one (e.g. used by `alifold` program [46]), assumes that the precomputed multiple alignment between the input RNA sequences corresponds to the alignment between their substructures. The structure is then derived by folding the multiple alignment of the sequences with the goal of minimizing a linear combination of the total free energy and a score derived from the covarying mutations among aligned bases.

The second flavor suggests to perform the sequence alignment and the structure prediction simultaneously [81, 37, 64, 12, 99, 43]. When formulated as a rigorous dynamic programming procedure, the computational complexity of this technique becomes very high;

it requires $O(n^6)$ time even for two sequences (and $O(n^{3m})$ time for m sequences [26]. In order to decrease the computational complexity, it may be possible to restrict the number of substructures from each RNA sequence to be aligned to the substructures from other sequences. In the RNAscf program [12], for example, this is done through a preprocessing step which detects all statistically significant potential stems of each RNA sequence by performing a local alignment between the sequence and its reverse complement. When computing the *consensus structure*, only those substructures from each RNA sequence which are enclosed by such stems are considered for being aligned to each other. The ultimate goal is again minimizing a linear combination of the free energy of the consensus sequence and the similarity of the aligned sequences and the structures they imply.

Note that this method crucially relies on the correctness of the multiple sequence alignment; thus its prediction quality is usually good for highly similar sequences (60% or more) but can be quite poor for more divergent sequences.

The role of locally significant structural elements in determining the global structure of an RNA molecule is further illustrated by *consensus folding* technique. In this approach, rather than minimizing free energy, the goal is to first extract all potential stems of each input RNA sequence. The consensus structure is then computed through determining the largest set of compatible potential stems that are common to a significant majority of the RNA sequences. A good example that uses the consensus folding technique is the comRNA program [49] which, once all stems of length at least ℓ are extracted from individual sequences, computes the maximum number of compatible stems that are common to at least k of the sequences via a graph theoretic approach. As one can expect, the consensus technique relies on the availability of many sequences that are functionally and structurally related.

Alkan et al. [5] present an alternative approach for capturing locally stable structural elements by delocalizing the thermodynamic cost of forming an RNA substructure. In addition to the total free energy, Alkan et al. incorporate the *energy density* of substructures which is defined as the length normalized free energy of a substructure w.r.t the nearest neighbor thermodynamic model. This approach thus uses the sum of the energy densities of individual substructures as the second component of the linear optimization function, giving the stable but small structural elements the chance to compete with global elements.

The impact of locally stable substructure is also justified by kinetic folding of RNA. The energy landscape of some RNA sequences contains local optima, in which the folding process may become trapped. Kinetic approaches (e.g. `Kinfold` [35], `RNAkinetis` [24]) study the dynamics of RNA folding on such an energy landscape. Using stochastic simulation and Markov chain, these methods aim to model the folding pathway over a couple of time steps. Generally speaking, these methods have a set of rules for transition between structural states which are in the simplest case adding or removing a base pair.

In addition to all the above approaches which focus on unseudoknoted secondary structures, a number of algorithms have been designed to study pseudoknotted structures during recent years. There are two major problems concerning the analysis of pseudoknotted RNAs. First, only few pseudoknoted structures have been known. Second, the prediction of pseudoknots is computationally very expensive. The full problem is known to be NP-hard [60], and efficient algorithms exist only for restricted classes of pseudoknots. The running times of the algorithms which predict the minimum free energy secondary structure for limited classes of pseudoknotted structures [92, 78, 3, 60, 31], range from $O(n^4)$ to $O(n^6)$ while each handles a different class of structures (Condon et al. present a study on the relationships of the various classes of pseudoknotted structures [23]). All these algorithms use the properties of the restricted class in a dynamic programming approach to efficiently solve the prediction problem. Rivas and Eddy algorithm [78] can handle the most general class of structures. Although, the algorithm of Dirks and Pierce [31] can be considered more general than the others, because it can calculate the partition function as well as the minimum free energy secondary structure. Even the most efficient algorithm by Reeder and Giegerich [76] still has a high running time of $O(n^4)$, although it strongly restricts the class of predictable pseudoknots.

2.3 RNA-RNA interaction

Many regulatory RNAs such as microRNAs and small interfering RNAs (miRNAs/siRNAs) are very short (21 to 25 nt) and have full sequence complementarity to the targets. However, some of the regulatory antisense RNAs are relatively long and are not fully complementary to their target sequences. They exhibit their regulatory functions by establishing stable joint structures with target mRNA involving one or more loop-loop interactions. Figure 2.3 shows the OxyS-fhlA complex in *E.coli* that contains two loop-loop interaction sites.

Brunel et al. [19] present a study of the structure and function of RNA loop-loop interactions through some well known examples. Table 2.1 shows the list of the interesting

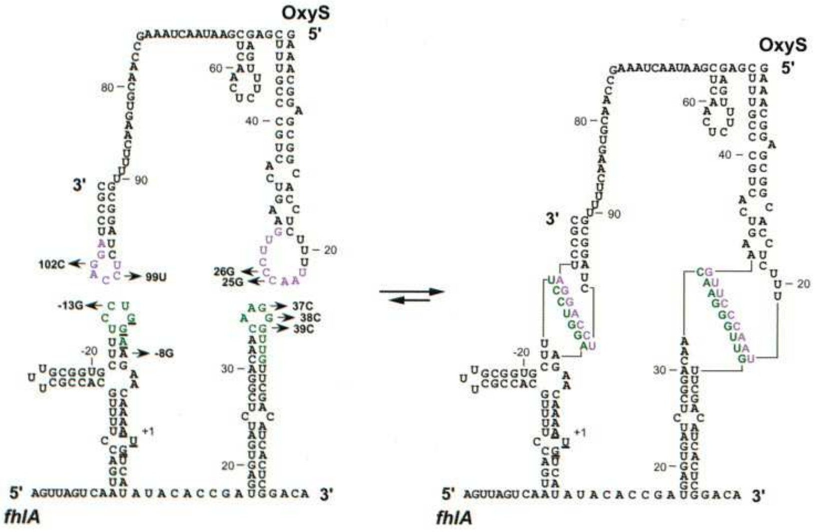


Figure 2.3: Interaction structure of small RNA molecule OxyS (antisense RNA) and fhlA (target) [9].

complexes and their functionality studied in [19]. The first intermolecular loop-loop interactions were observed between complementary anticodons of different tRNA pairs. This observation and the similar ones anticipated the high potential role of hairpin loops to trigger RNA intermolecular recognition. Hairpin loops, due to both functional and structural properties, are perhaps the most adaptable motifs for initiating the interaction. First, hairpin loops are well accessible provided that they are not engaged in the intramolecular architecture. Second, their structural versatility allows them to adopt particular conformations enabling a proper presentation of nucleotides that initiate the recognition process. This initiation step generally involves Watson-Crick pairing of a few nucleotides, preferentially G-C pairs. The initial reversible complex is subsequently converted into a more stable complex through helix propagation or stabilization by a protein. A large diversity of the stabilization mechanisms is observed. It appears that RNA structures have evolved either to freeze the initial complex, or alternatively to convert initial interactions by propagating helices along topologically feasible pathways. Stabilization of the initial complex may also be assisted by proteins and/or formation of additional contacts. RNA loop-loop interactions thus appear to be widely used to facilitate molecular recognition and trigger a variety of dynamic pathways.

Table 2.1: Examples of loop-loop interactions involved in the formation of several prokaryotic antisense/target complexes [19].

Trans-acting RNA	Target	Function	References
Plasmids (IncFII-relatives); CopA	repA mRNA	Replication control; translation inhibition	[58, 73, 54]
(IncB and Inc$I\alpha$-relatives); RNAI	repZ mRNA	Replication control; translation inhibition	[83, 10]
(ColE1-relatives); RNAI	RNAII (preprimer)	Replication control; primer maturation	[90, 89]
(pT181 and pIP501-relatives); RNAI	repC mRNA	Replication control; transcriptional attenuation	
(ColE2-relatives); RNAI	rep mRNA	Replication control; translation inhibition	
pAD1; RNAII	RNAI	Post-segregational killing; translation inhibition	
(IncFI/FII-relatives); FinP RNA	traJ mRNA	Control of conjugation; translation inhibition	[55, 94]
Bacteriophages P22; Sar RNA	ant mRNA	Lysis/lysogeny switch; translation inhibition	[82]

Bacterial E. coli OxyS	fhlA mRNA;
	rpoS mRNA
		Oxidative stress;
		translation inhibition;
		sequestration of Hfq
			[6, 103, 9]

B. subtilis tRNAs	Aminoacyl-tRNA synthetase mRNAs	Aminoacyl-tRNA synthetase regulation; transcription antitermination	[40]

The problem of computationally predicting interaction between RNAs has been attracted substantial interest in recent years. Currently the goal is to predict not only simple interactions between miRNAs and mRNA targets, but even more complex interactions involving loop-loop interactions and multiple binding sites. The RNA-RNA interaction prediction problem in general can be considered as the RNA joint secondary structure prediction one. A joint secondary structure between two RNA sequences is a set of base pairs where each nucleotide is paired with at most one other nucleotide, either internal or external.

Definition (RNA-RNA interaction prediction problem) Given two RNA sequences **R** and **S**, compute the set of canonical base pairs $\{i_R \bullet j_R | 1 \leq i_R < j_R \leq L_R\} \cup \{i_S \bullet j_S | 1 \leq i_S < j_S \leq L_S\} \cup \{i_R \circ i_S | 1 \leq i_R \leq L_R \wedge 1 \leq i_S \leq L_S\}$ of the optimal joint secondary structure.

The RNA-RNA interaction prediction problem in its general case is known to be NP-hard [4], but simplified versions of the problem that consider restricted types of interactions under the specific energy models can be handled in polynomial times. There has been no standard energy model for interaction previously. All the suggested energy models are over simplified and useful only for a specific class of interactions.

2.3.1 RNA-RNA interaction prediction methods

During the last few years, several computational methods emerged to study the interaction between two RNAs. Based on their performance and approach we put them into four different categories.

Concatenating the input sequences into a single sequence. Early attempts to analyze the thermodynamics of multiple interacting strands concatenate input sequences in some order and consider them as a single strand. For example, `pairfold` [8] and `RNAcofold` [17] from Vienna package concatenate the two input sequences into a single strand and predict its minimum free energy structure by treating the boundary between two adjacent RNA sequences as a special loop. For the multiple input sequences, Andronescu et al. [8] suggest to compute minimum free energy structure of concatenation of all different permutation orders of input sequences. Furthermore, `RNAcofold` provides the similar extension of McCaskill's partition function algorithm to compute base pairing probabilities, and equilibrium concentrations of duplex structures. Dirks et al. [30] present a method, as a part of `NUPack`, that concatenates the input sequences in all unique cyclic permutation orders and computes the partition function for the whole ensemble of complex species, carefully considering symmetry and sequence multiplicities. Dirks et al. brought out the fact that for complexes of interacting strands in which some strands are identical (e.g. AA), over counting correction to the partition function recursions is necessary. Dirks et al. proved that the correctness can be easily done for each permutation by dividing the calculated partition function to a value v corresponding to the number of rotations of the cyclic permutation that results in the same permutation. For example, $v = 4$ for AAAA, $v = 3$ for ABABAB, and $v = 2$ for ABAABA.

Despite all the above advances, the methods based on concatenating the sequences are not accurate at all in general, as even if pseudoknots are considered, some useful interactions are excluded (for example loop-loop interaction) while many physically impossible interactions are included (for example physically impossible crossing interactions).

Avoiding internal base pairing. Alternatively, several methods avoid internal base pairing in either strand, and compute the minimum free energy secondary structure for their hybridization under this constraint (`RNAhybrid` [77], `UNAFold` [28, 62], `TargetRNA` [88], `RNAduplex` and `RNAplex` [86] from Vienna package). These approaches naturally work only for simple cases involving typically very short strands. Many regulatory RNAs such as microRNAs and small interfering RNAs (miRNAs/siRNAs) are very short and have almost full sequence complementarity to the targets. These methods are appropriate to find the energetically most favorable hybridization sites of small regulatory RNAs in large target RNAs.

`RNAhybrid` program calculates the MFE hybridizations of all possible start positions in the miRNA and in the target. Bulge loops (i.e., stretches of unpaired nucleotides in either of the sequences) and internal loops (i.e., stretches of unpaired nucleotides in both sequences) are restricted to a constant maximum length in either sequence (which is set to 15 as a default value). If m and n are the lengths of the target and the miRNA, respectively, and c is the maximal length of a loop in either sequence, the space consumption of the algorithm is of the order $O(mn)$, and the time consumption is of the order $O(c^2mn)$. If m is much larger than n and c, which is usually the case for miRNAs and their potential targets, the space and time consumption is linear to the target length m.

`RNAduplex` and `RNAplex` follow the similar approaches as `RNAhybrid`, but `RNAplex` uses a simplified energy model which makes it faster and thus more suitable for longer RNA sequences. In addition a length penalty is considered to focus the target search on short stable interactions.

Given the sequence of an sRNA gene in a particular organism, `TargetRNA` program outputs a ranked list of predicted targets for the sRNA. The program begins by consulting a database of protein coding genes for the related organisms. For each protein coding gene in the organism, the program extracts the mRNA sequence corresponding to the protein coding region along with user-specified regions upstream and downstream of the coding sequence, extending into the 5'-UTR and 3'-UTR, respectively. `TargetRNA` then evaluates and sorts the potential for interaction between every extracted mRNA sequence and the sRNA, and assigns each a hybridization score. In the hybridization score for two RNA sequences, intramolecular base pairings are not considered and pseudoknots are not allowed. To calculate the hybridization score of an sRNA and candidate mRNA target, `TargetRNA` can use either of two different hybridization score models for RNA sequence interactions: an individual base pair model or a stacked base pair model. The individual base pair model of hybridization scoring is based on a straightforward extension of the SmithWaterman dynamic programming algorithm [84], except that instead of assessing homology potential, base pairing potential is assessed. The stacked base pair model of hybridization scoring is based on stacking and destabilizing energies of interacting sequences, where the calculation of the optimal hybridization score for two sequences is comparable with folding RNA sequences [105] without allowing intramolecular base pairings. Note that all the above methods are not able to predict RNA complexes involving loop-loop interactions. However, as mentioned earlier loop-loop interactions play important functionalities.

Considering interaction as a multi step process. There are several evidences [96, 19, 66, 41] that interaction can be considered as a multi step process: 1) unfolding of the two molecules to expose bases needed for hybridization, 2) the hybridization at the binding site, and 3) restructuring of the complex to a new minimum free energy conformation. Based on this idea a third set of methods [69, 52, 20] are designed to predict the (most likely) hybridization between the unpaired regions of the secondary structures of two individual RNAs. The energy score of the interaction is calculated as the sum of the two energy contributions: (i) the energy necessary to open the binding site ΔG_{open} and (ii) the energy gained from hybridization ΔG_{hybrid}. Note that the energy of the open region is assumed to be unchanged by the binding of the oligo.

`RNAup` [69, 68] presents an extension of the standard partition function approach [65] that computes the probabilities that a sequence interval remains unpaired. The corresponding probability is computed as the ratio between the partition functions of the all secondary structures in which the specific interval is unpaired, and the ensamble of all secondary structures. Let $P_u[i, j]$ be the probabilities that a sequence interval $[i, j]$ remains unpaired, then $\Delta G_{open} = (1/\beta) ln P_u[i, j]$, where β is the inverse of the temperature times Boltzmann's constant. ΔG_{open} is calculated for all regions up to a maximum size w for two interacting sequences $\mathbf{R}$ and $\mathbf{S}$. The computation of the hybridization part is performed similar to `RNAhybrid` - the binding region contains a set of stacks, bulge and internal loops. The memory requirement is $O(n^2 + nw^3)$, and the required CPU time scales as $O(n^3 + nw^5)$.

Kertesz et al. [52] developed a quantitative study to examine the effect of site accessibility on miRNA-mRNA interaction. Their method, `PITA`, starts with a genome-wide search for miRNA target *seed regions* and tries to extend these sites in one direction. The seed region is a subsequence of seven or eight bases at the $5'$ end of animal miRNAs. ΔG_{open} is calculated, in a way similar to `RNAup`, for the regions including highly conserved seeds. ΔG_{hybrid} is computed such that miRNA and target are paired according to pairing constraints imposed by seed. The results of the study in several genomes show that site accessibility is as important as sequence match in the seed for determining effectiveness of binding between miRNA and target mRNA. More precisely, the analysis suggest that as seeds are more conserved and regions are more accessible, the target sites are most likely to be preferred evolutionary and thus functionally important.

`IntaRNA` [20] integrated both the accessibility of the target sites and the existence of a user-definable seed in a general approach for arbitrary RNAs. The program is validated to predict targets for bacterial sRNAs, but it can be used to find other RNA-RNA interactions as well.

Although this type of methods show reasonable accuracy in predicting interaction of single binding sites, but unfortunately they are not able to predict the interactions while the complex contains multiple binding sites.

Predicting the joint secondary structure. The last set of approaches compute the minimum total energy joint structure between two interacting strands under different energy models. Pervouchine [74] devised a dynamic programming algorithm to maximize the number of base pairs among interacting strands. A follow up work by Kato et al. [51] proposed a grammar based approach to RNA-RNA interaction prediction. More generally Alkan et al. [4] studied the joint secondary structure prediction problem under three different models: 1) base pair counting, 2) stacked pair energy model, and 3) loop energy model. Alkan et al. proved that the general RNA-RNA interaction prediction under all three energy models is an NP-hard problem. Therefore, they suggested some natural constraints on the topology of possible joint secondary structures which are satisfied by all examples of complex RNA-RNA interactions in the literature. The resulting algorithms compute the minimum free energy secondary structure among all possible joint secondary structures that do not contain (internal) pseudoknots, crossing interactions (i.e. external pseudoknots), and zigzags (zigzag happens when two interacting loops, interact with other region as well). Alkan et al. present the first experimental confirmation of the total free energy minimization approach via correctly predicting the joint structure formed by a number of interacting RNA pairs.

Although these approaches are the most general ones as they can cover almost all possible types of interactions known so far, their significant resource requirements limit their applicability. All the above approaches for predicting a general joint structure have a worst case running time of $O(n^6)$ and a space complexity of $O(n^4)$. While this complexity might be acceptable when analyzing only a few putative sRNA-target interaction pairs, we are now faced with the situation that the amount of data to be analyzed is vastly increasing. Thus, there is an urgent need for a time and space efficient interaction prediction method that is able to handle complex joint structures.

Chapter 3

Interaction Energy Model

In this section we propose an energy model for interaction between RNA strands. Our interaction energy model is an extension of the nearest neighbor thermodynamics energy model for RNA secondary structure. The nearest neighbor thermodynamics energy model describes a framework to calculate the free energy of a given RNA secondary structure.

3.1 Nearest neighbor thermodynamics energy model

Much of the literature on RNA secondary structure prediction is devoted to the *thermodynamic approach* which aims to minimize the sum of the contributions of certain *structural features* to the global free energy of the RNA molecule. Perhaps the most widely used structural features and their associated thermodynamic parameters are provided by the *nearest neighbor thermodynamic model* (NNTM) where the free energy contribution of a given base pair is a function of its nearest base pair [87]. The NNTM has been developed in conjunction with the development of dynamic programming folding algorithm. The independence assumptions in this model is dictated by independence assumptions needed by recursive dynamic programming algorithms.

Since 1999, the NNTM has been accepted as the standard energy model of RNA secondary structure. An unpseudoknotted secondary structure s of a single nucleic acid strand $\mathbf{S}$, in the standard energy model, is decomposed into loops, and a free energy is associated with every loop in s. The total free energy G_s is sum of the loop free energies. The standard energy model consists of the following loop types:

- Empty: $G_{i,j}^{\text{empty}}$ is the free energy of a subsequence $[i, j]$ that contains no base pairs and is external to all loops. Its energy contribution is assumed to be zero.
- Hairpin: $G_{i,j}^{\text{hairpin}}$ is the free energy of a hairpin closed by the arc $i \bullet j$. The energy contribution of this loop depends on the subsequence and the loop size.
- Interior: $G_{i,k_1,k_2,j}^{\text{interior}}$ is the free energy of the interior loop enclosed by the closing arc $i \bullet j$ and the interior arc $k_1 \bullet k_2$. This free energy depends on the closing base pairs and the loop size. An interior loop is called bulge iff one side of the loop has zero length. Stacked pairs are a special case of bulge loops in which case the size of the loop is zero. A stem is a series of stacked pairs.
- Multi: $G_{U,B}^{\text{multi}}$ is the energy of a multiloop with B base pairs and U unpaired bases. It is approximated by

  $$G_{U,B}^{\text{multi}} = \alpha_1 + \alpha_2 U + \alpha_3 B,$$

  in which α_1 is the penalty for the formation of the multiloop, α_2 is the penalty per each unpaired base in the multiloop, and α_3 is the penalty per each base pair facing to the multiloop.

Figure 3.1 shows the elementary substructures in 5S rRNA secondary structure. The parameters associated with the NNTM have been determined from optical melting experiments and have been measured quite precisely over the years. Moreover some (heuristic) scoring models [63] have been used to overcome the difficulty of experimental procedures for some cases. For instance, the energy of a multi-branch loop is approximated by a function of number of branches, number of unpaired bases, dangling bases and the closing base pairs.

Recently some restrictions of laboratory experiments for certain structural features of the NNTM led to the use of statistical learning methods for estimating their associated energy parameters [7]. In this case, the parameters should be trained using large collections of RNA sequences annotated with known secondary structures; clearly the accuracy and the richness of the training set has a direct consequence on the accuracy of the predicted structures.

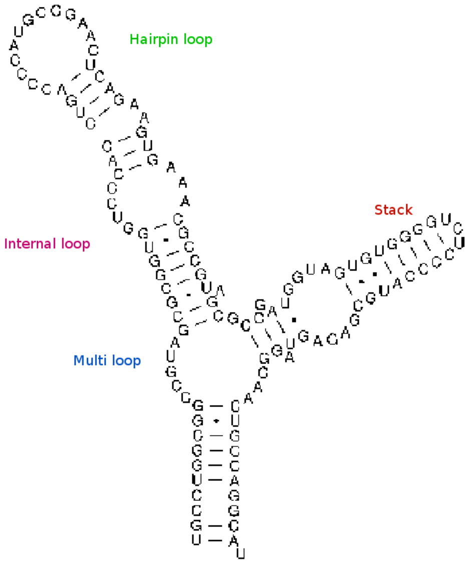


Figure 3.1: An example of RNA secondary structure containing all elementary substructures.

3.2 Interaction Energy Model

Similar to the standard energy model, an interaction structure rs of two nucleic acid strands $\mathbf{R}$ and $\mathbf{S}$, is decomposed into substructure components, and a free energy is associated with every component. The total free energy G_{rs} is sum of the free energies of components. The standard energy model consists of the following loop types: 1) Hairpin, 2) Interior, and 3) Multiloop. However, in an interaction structure of two strands under our assumptions, where pseudoknots, crossing bonds, and zigzags are not allowed, new kinds of components can appear. We extend the standard energy model by defining those new kinds of interaction components. Our extended energy model consists of the following new components:

- Hybrid: $G^{\text{hybrid}}_{\{k_R^i \circ k_S^i\}}$ is the free energy of a joint secondary structure consisting of a series of bonds, $k_R^i \circ k_S^i$, $i = 1, \ldots, m$, with no intramolecular base pairing or branching. We call such a component *hybrid* (Figure 3.2). We define the energy associated with a hybrid component by

$$G^{\text{hybrid}}_{\{k_R^i \circ k_S^i\}} = \beta_1 + \sigma \sum_{i=1}^{m-1} G^{\text{interior}}_{k_R^i, k_R^{i+1}, k_S^{i+1}, k_S^i}, \tag{3.1}$$

in which β_1 is the penalty for the formation of the hybrid, and $\sigma \leq 1$ is the ratio of the free energy of intermolecular to that of intramolecular interior loops (as suggested by [4]). Note that with $\beta_1 = 0, \sigma = 1$, G^{hybrid} is identical to the energy proposed by `RNAhybrid`, first introduced by Rehmsmeier et al. [77] which considers only one hybrid component for mRNA/target duplexes and does not allow any intramolecular structure,

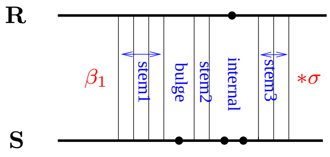


Figure 3.2: A hybrid component between the two strands whose free energy is $G^{\text{hybrid}} = \beta_1 + \sigma(G^{stem_1} + G^{bulge} + G^{stem_2} + G^{internal} + G^{stem_3})$.

- Kissing: $G^{\text{kissing}}_{U^k,B^k}$ is the energy of an intramolecular loop (hairpin, interior, or multiloop) that makes interaction with the other strand and has B^k base pairs and U^k unpaired bases. Such component is called a *kissing loop* (Figure 3.3). The energy associated with a kissing loop is given by

$$G^{\text{kissing}}_{U^k,B^k} = \beta_2 U^k + \beta_3 B^k, \tag{3.2}$$

  in which B^k is the number of base pairs and U^k the number of unpaired bases in the kissing loop, β_2 is the penalty per each unpaired base, and β_3 is the penalty per each base pair. Note that in our model we use different β_1 and σ values for a hybrid component covered by a kissing loop.

- Inter-hybrid: $G^{\text{inter-hybrid}}$ is the energy of an intermolecular loop bounded by two bonds belonging to two consecutive hybrid components. Bases in either sequence facing this kind of loop might be the end points of only arcs and not bonds. We call such a component *inter-hybrid loop* (Figure 3.4). In this work the energy contribution of an inter-hybrid loop is assumed to be zero.

Figure 3.5 shows the interaction energy model components in OxyS-fhlA complex structure. We use a different β_1 penalty and σ for a hybrid component covered by a kissing loop. The parameters for a hybrid component that is not covered by a kissing loop is denoted by

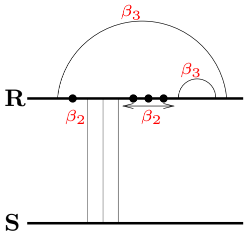


Figure 3.3: A kissing loop in **R** that interacts with the other strand **S**. In this case, the free energy of the kissing loop is $G^{\text{kissing}} = 4\beta_2 + 2\beta_3$.

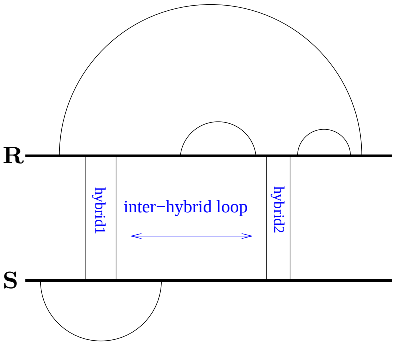


Figure 3.4: An example of inter-hybrid loop in interaction structure of two strands **R** and **S**.

β'_1 and σ'. We add an AU penalty to the energy of a hybrid component per each terminal AU base pair; this penalty is motivated by [101]. Similar to `RNAhybrid`, the interior loops in a hybrid component are restricted to a constant maximum length, in either sequence, which is set to 15 in this work.

Because of lack of data we could not use any learning method to find appropriate parameters for our model. We manually optimized the parameters for the melting temperature experiment reported in [101] and also perturbed them a bit for the equilibrium concentration computation, based on intuition and a few trial and errors. The default values for our

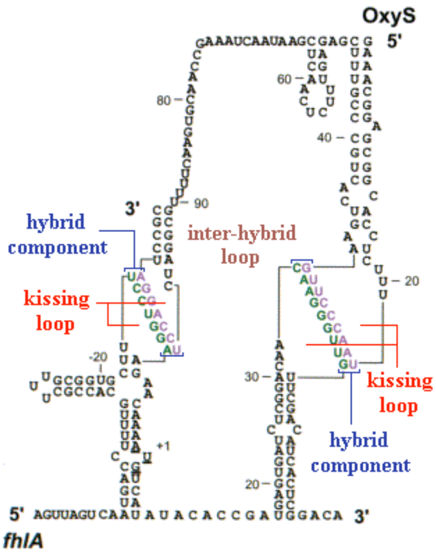


Figure 3.5: Interaction components of OxyS-fhlA pair presented in [9].

energy parameters are

$$\begin{aligned} &\beta_1 = 4.5, \beta_2 = \beta_3 = 0.1, \sigma = 0.7, \\ &\beta_1' = 2.5, \sigma' = 0.8 \qquad (3.3) \\ &AU_{penalty} = 0.45 \end{aligned}$$

In future, provided a large set of interaction data, the parameters can be trained more accurately. Further studies should approve the correctness and accuracy of the interaction energy model and its parameters by extensive laboratory experiments.

Chapter 4

Partition Function for Interacting Nucleic Acid Strands

There is a significant interest in quantitative analysis of binding thermodynamics between oligonucleotides and target RNAs. Here we address the problem of calculating how likely two RNA or DNA strands are to interact. We present a computational method to predict the probability of intermolecular and intramolecular base pairing. Based on the base pair probabilities, one can quantitatively measure the strength, probability, and stability of the complex.

To predict base-pairing probability of any two bases of interacting nucleic acids, it is necessary to compute the *interaction partition function* over the whole ensemble of possible individual and joint secondary structures. Partition function is a scalar value from which various thermodynamic quantities can be derived. For example, the equilibrium concentration of each complex nucleic acid species and also the melting temperature of interacting nucleic acids can be calculated based on the partition function of the complex.

We present a partition function algorithm for analyzing the thermodynamics of two interacting nucleic acid strands considering the most general type of interactions introduced in the literature. We give a dynamic programming algorithm to compute the partition function over almost all physically possible interaction secondary structures in $O(n^6)$ time and $O(n^4)$ space complexity. We verify our algorithm by computing (i) the equilibrium concentration for OxyS-fhlA complex and (ii) the melting temperature for RNA pairs available in the literature. In both experiments our algorithm shows high accuracy.

4.1 Problem Definition

The partition function is a weighted sum over the set of all possible secondary structures S

$$Q(T) = \sum_{s \in S} e^{-G_s/RT} \tag{4.1}$$

where R is the universal gas constant and T is the temperature.

Efficient algorithms for computing the partition function for a single strand have been given. McCaskill gave the first partition function algorithm for a single unpseudoknotted nucleic acid strand [65], and Dirks and Pierce gave a partition function algorithm for a single strand allowing pseudoknots [31]. However, computing the partition function for multiple interacting strands has not been properly addressed. In previous attempts multiple strands are concatenated in some order and partition function for the resulting single strand is computed [28, 17, 30]. That approach is not accurate at all because even if pseudoknots are considered, some useful interactions are excluded while many physically impossible interactions are included (for example physically impossible crossing interactions). On the other hand, considering all possible secondary structures makes the problem NP-hard [4]. Therefore, we only consider all possible secondary structures that do not contain pseudoknots, crossing bonds, and zigzags.

Interaction Partition Function (IPF) Problem

Given a pair of nucleic acid strands **R** and **S**, and a temperature T, compute the partition function, $Q^I(T)$, over S^I the set of all possible single or duplex secondary structures that do not contain pseudoknots, crossing bonds, and zigzags.

Input: nucleic acid strands **R** and **S**.

Output:

$$Q^I(T) = \sum_{s \in S^I} e^{-G_s/RT}.$$

It is important to note that designing an algorithm to compute the partition function is more challenging than giving an algorithm to predict the minimum free energy secondary structure, because for partition function the algorithm should guarantee that every structure is considered exactly once.

4.2 The Algorithm: piRNA

We give a recursive algorithm, called Partition function for InteRacting Nucleic Acids (`piRNA`), for the IPF problem. In all of our recursions, the considered cases are disjoint. This fact shows that every possible secondary structure is reached by exactly one trajectory in the recursion process. Our algorithm guarantees to consider all possible secondary structures exactly once.

We present our algorithm using recursion diagrams [31, 78]. Our algorithm computes two types of recursive quantities: 1) the partition function of a subsequence $[i, j]$ in one strand, and 2) the joint partition function of subsequences $[i_R, j_R]$ and $[i_S, j_S]$. A *region* is the domain over which a partition function is computed. *Terminal bases* are the boundaries of a region. For the first type, region is $[i, j]$ with i and j terminal bases. For the second type, region is $[i_R, j_R] \times [i_S, j_S]$ with i_R, j_R, i_S, and j_S terminal bases. The *length pair* of region $[i_R, j_R] \times [i_S, j_S]$ is $(l_R = j_R - i_R + 1, l_S = j_S - i_S + 1)$. Our algorithm starts with $(l_R = 1, l_S = 1)$ and considers all length pairs incrementally up to $(l_R = L_R, l_S = L_S)$. For a fixed length pair (l_R, l_S), recursive quantities for all the regions $[i_R, i_R{+}l_R{-}1]\times[i_S, i_S{+}l_S{-}1]$ are computed.

4.2.1 Partition Function for Non-Interacting Subsequences

For computing the partition function of a subsequence in one strand we use McCaskill's algorithm [65]. McCaskill's algorithm is shown in Figure 5.2, in which $Q_{i,j}$ is the partition function for the subsequence $[i, j]$.

In Figure 5.2, the first case of $Q_{i,j}$ corresponds to an empty structure (that constitutes no base pairs) whose free energy is assumed to be zero, thus its contribution to the partition function is $e^{-G^{empty}_{i,j}/RT} = 1$. In the other case, there exists at least one arc and the leftmost one is $k_1 \bullet k_2$. It contributes $Q^b_{k_1,k_2} Q_{k_2+1,j}$ to the partition function, therefore,

$$Q_{i,j} = 1 + \sum_{i \le k_1 < k_2 \le j} Q^b_{k_1,k_2} Q_{k_2+1,j}. \tag{4.2}$$

The second line shows the cases of $Q^b_{i,j}$ which is the partition function for the subsequence $[i, j]$ assuming i and j are base paired. The arc $i{\bullet}j$ can close different substructures: hairpin, interior, or multiloop. The energy contribution of each substructure is calculated based on the standard thermodynamics energy model.

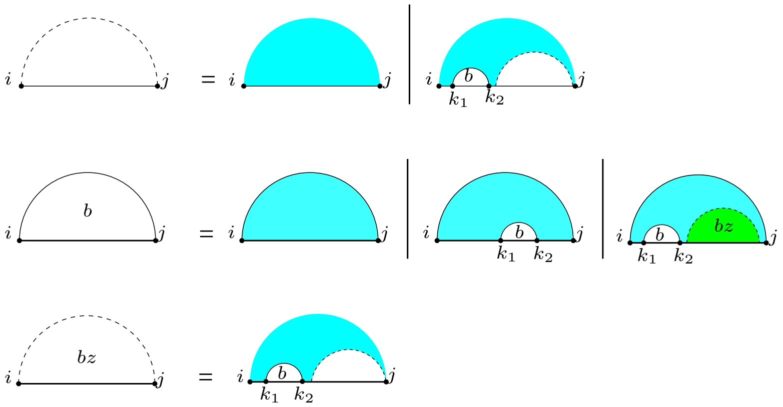


Figure 4.1: McCaskill's algorithm: recursion for $Q_{i,j}$, the partition function for the subsequence $[i,j]$. Above, $Q^{b}_{i,j}$ is the partition function for the subsequence $[i,j]$ assuming i and j are base paired, and $Q^{bz}_{i,j}$ is the partition function for the subsequence $[i,j]$ assuming there is at least one arc in the region.

$$\begin{aligned} Q^{b}_{i,j} =& e^{-G^{\text{hairpin}}_{i,j}/RT} + \sum_{i \le k_1 < k_2 \le j} e^{-G^{\text{interior}}_{i,k_1,k_2,j}/RT} + \\ & \sum_{i \le k_1 < k_2 \le j} Q^{b}_{k_1,k_2} Q^{bz.green}_{k_2+1,j-1} \; e^{-(\alpha_1+\alpha_2(k_1-i-1)+\alpha_3)/RT}. \end{aligned} \tag{4.3}$$

The third line shows cases of $Q^{bz}_{i,j}$ which is the partition function for the subsequence $[i,j]$ assuming the region constitutes at least one arc. Therefore,

$$Q^{bz}_{i,j} = \sum_{i \le k_1 < k_2 \le j} Q^{b}_{k_1,k_2} Q_{k_2+1,j}. \tag{4.4}$$

As mentioned before, a green region is contained in a multiloop. The region has the same recursion as if it was white, however the base pair and unpaired base penalties of multiloop should be applied to it. Explicitly,

$$Q^{bz.green}_{i,j} = \sum_{i \le k_1 < k_2 \le j} Q^{b}_{k_1,k_2} Q^{\cdot green}_{k_2+1,j} \; e^{-(\alpha_2(k_1-i-1)+\alpha_3)/RT}, \tag{4.5}$$

$$Q^{\cdot green}_{i,j} = e^{-\alpha_2(j-i-1)/RT} + \sum_{i \le k_1 < k_2 \le j} Q^{b}_{k_1,k_2} Q^{\cdot green}_{k_2+1,j} \; e^{-(\alpha_2(k_1-i-1)+\alpha_3)/RT}. \tag{4.6}$$

4.2.2 Partition Function for Non-Interacting Gapped Subsequences

The gap partition function Q^g is defined by the recursion in Figure 4.2. This quantity is similar to the g in Dirks-Pierce's algorithm [31]. For $Q^g_{i,d,e,j}$, we assume $i \bullet j$ and $d \bullet e$. There are two groups of cases: 1) there is no more spanning arc in the region, and 2) there is at least another outermost spanning arc $k_1 \bullet k_2$. In both groups, there could be some additional structure in the region. If there is no additional structure in the region, then the spanning region is an interior loop. If there is at least one arc in either side of the region, then the spanning region forms a multiloop and penalty of multiloop should be applied.

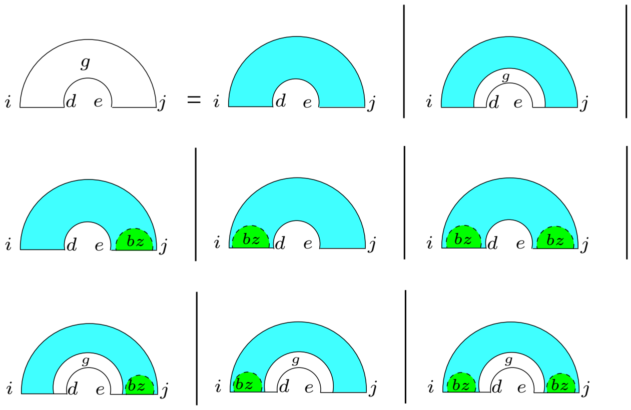


Figure 4.2: Recursion for $Q^g_{i,d,e,j}$ the partition function for the subsequence $[i,j]$ excluding the gap $[d,e]$ assuming $i \bullet j$ and $d \bullet e$.

Later, in the recursion of our interaction partition function, we need an extended version of Q^g where d and e do not necessarily form a base pair and the gap interval $[d+1, e-1]$ contains either some direct bonds loop or multiple interaction arcs. The two extended gapped partition functions are called Q^{gm} and Q^{gk}. $Q^{gm}_{i,d,e,j}$ and $Q^{gk}_{i,d,e,j}$ are the partition functions for $[i,j]$ excluding the gap $[d,e]$, assuming i and j are base paired. For Q^{gm} the gap contains multiple interaction arcs, and for Q^{gk}, the gap contains direct bond (see Figure 4.3). Therefore,

$$Q^{gm}_{i,d,e,j} = Q^{\cdot green}_{i+1,d} Q^{\cdot green}_{e,j-1} + \sum_{\substack{i<k_1\leq d \\ e\leq k_2<j_S}} Q^g_{i,k_1,k_2,j} Q^{\cdot green}_{k_1+1,d} Q^{\cdot green}_{e,k_2-1}, \tag{4.7}$$

and

$$Q^{gk}_{i,d,e,j} = Q^{.red}_{i+1,d} Q^{.red}_{e,j-1} + \sum_{\substack{i<k_1\le d\\ e\le k_2<j_S}} Q^{g}_{i,k_1,k_2,j} Q^{.red}_{k_1+1,d} Q^{.red}_{e,k_2-1}. \tag{4.8}$$

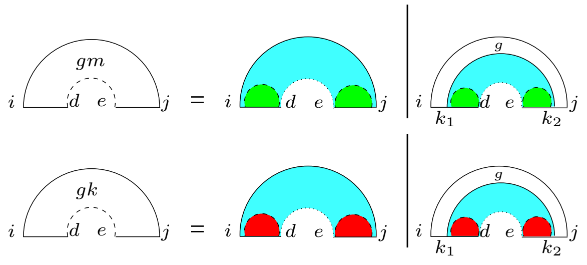


Figure 4.3: Recursion for $Q^{gm}_{i,d,e,j}$ and $Q^{gk}_{i,d,e,j}$ the partition functions for $[i,j]$ excluding the gap $[d,e]$, assuming i and j are base paired. For Q^{gm} the gap contains multiple interaction arc, and for sQ^{gk}, the gap contains direct bond.

4.2.3 Partition Function for Interacting Subsequences

In the following, we present all cases of $Q^{I}_{i_R,j_R,i_S,j_S}$ which is the interaction partition function for the region $[i_R,j_R] \times [i_S,j_S]$. A solid vertical line indicates a bond, a dashed vertical line denotes two terminal bases of a region which may be base paired or unpaired, and a dotted vertical line denotes two terminal bases of a region which are assumed to be unpaired. Figure 5.3 shows the cases of Q^I: 1) there is no bond between the two subsequences, 2) the leftmost bond is a direct bond in both subsequences, and 3) the leftmost bond is covered by an arc in at least one subsequence. Therefore,

$$\begin{aligned} Q^{I}_{i_R,j_R,i_S,j_S} =& Q_{i_R,j_R} Q_{i_S,j_S} + \sum_{\substack{i_R\le k_1<j_R\\ i_S\le k_2<j_S}} Q_{i_R,k_1-1} Q_{i_S,k_2-1} Q^{Ib}_{k_1,j_R,k_2,j_S} + \\ & \sum_{\substack{i_R\le k_1<j_R\\ i_S\le k_2<j_S}} Q_{i_R,k_1-1} Q_{i_S,k_2-1} Q^{Ia}_{k_1,j_R,k_2,j_S}, \end{aligned} \tag{4.9}$$

Figure 5.4 shows the recursion for $Q^{Ib}_{i_R,j_R,i_S,j_S}$, the interaction partition function for the region $[i_R,j_R] \times [i_S,j_S]$ assuming $i_R \circ j_S$ is a bond. Since we have penalties for opening and closing a hybrid component, the recursion for Q^{Ib} has to determine whether the region

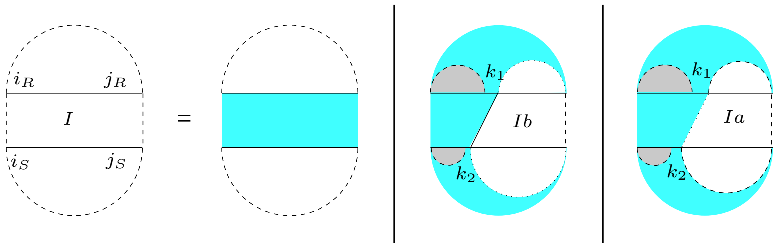


Figure 4.4: Cases of the interaction partition function $Q^{I}_{i_R,j_R,i_S,j_S}$. The first case constitutes no bonds. In the second case, the leftmost bond is a direct bond on both subsequences. In the third case, the leftmost bond is covered by an interaction arc in at least one subsequence.

contains one or several hybrid components. In all cases, Q^{Ih} contains the full hybrid component containing the bond $i_R \circ j_S$ (see Figure 4.7 for Q^{Ih} recursion). The first possibility reflects the case where we have only one hybrid component. In the other cases, we have always at least two hybrid components. The subsequent intermolecular bond starts a new hybrid component iff 1) it is not direct in at least one subsequence, i.e. it is covered by an arc in the associated regions (case 2 of the Q^{Ib} recursion), or 2) there is at least one arc between the two successive intermolecular bonds (case 3 and 4 of the Q^{Ib} recursion). Using the additional matrices Q^{Ihh} and Q^{Ihb}, we get

$$\begin{aligned} Q^{Ib}_{i_R,j_R,i_S,j_S} =& Q^{Ihh}_{i_R,j_R,i_S,j_S} + \sum_{\substack{i_R<k_1<j_R \\ i_S<k_2<j_S}} Q^{Ihb}_{i_R,k_1,i_S,k_2} Q^{Ib}_{k_1,j_R,k_2,j_S} + \\ & \sum_{\substack{i_R<k_1<j_R \\ i_S<k_2<j_S}} Q^{Ihh}_{i_R,k_1,i_S,k_2} Q^{Ia}_{k_1,j_R,k_2,j_S}. \end{aligned} \tag{4.10}$$

The quantities Q^{Ihh} and Q^{Ihb} are defined by the recursion diagrams in Figure 4.6 and equivalently by the following equations:

$$Q^{Ihb}_{i_R,j_R,i_S,j_S} = \sum_{\substack{i_R\le k_1\le j_R \\ i_S\le k_2\le j_S}} e^{-\beta_1/RT} Q^{Ih}_{i_R,k_1,i_S,k_2} (Q^{bz}_{k_1+1,j_R} Q_{k_2+1,j_S} + Q^{bz}_{k_2+1,j_S}) \tag{4.11}$$

and

$$Q^{Ihh}_{i_R,j_R,i_S,j_S} = \sum_{\substack{i_R\le k_1\le j_R \\ i_S\le k_2\le j_S}} e^{-\beta_1/RT} Q^{Ih}_{i_R,k_1,i_S,k_2} Q_{k_1+1,j_R} Q_{k_2+1,j_S}, \tag{4.12}$$

in which Q^{Ih} is the interaction partition function for a hybridization region (Figure 4.7).

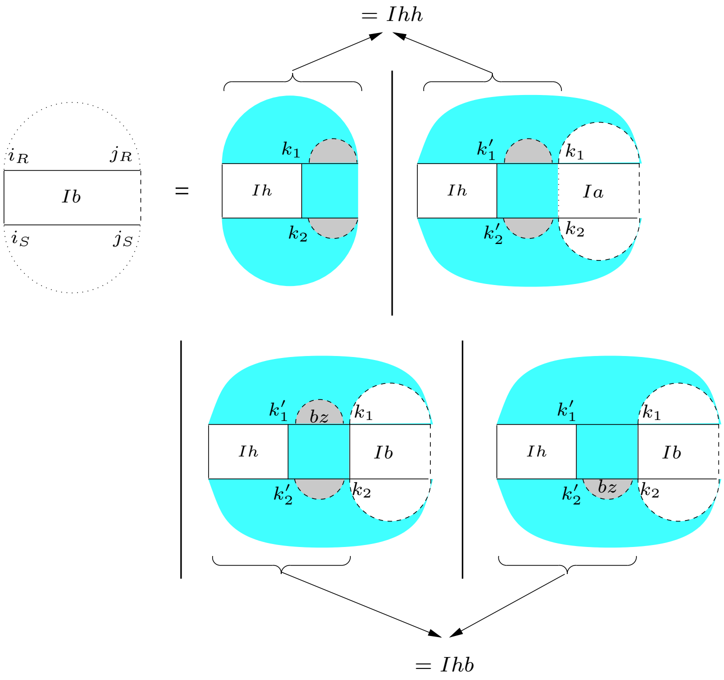


Figure 4.5: Recursion for $Q^{Ib}_{i_R,j_R,i_S,j_S}$ assuming $i_R \circ j_S$ is a bond. We show a version of the recursion that contains two split points in each sequence for simplicity reasons. However, this would increase the complexity and can easily be resolved by introducing two additional matrices Q^{Ihh} and Q^{Ihb} for the region $[i_R, k_1] \times [i_S, k_2]$ as indicated by the arrows.

Figure 4.7 shows the recursion for Q^{Ih}. Since we do not allow isolated bond the base case of Q^{Ih} is an interior loop, otherwise it can be an isolated bond. Two cases is possible: 1) there is no bond other than $i_R \circ j_S$ and $i_S \circ j_R$ in the region, and 2) there exist more bonds between $i_R \circ j_S$ and $i_S \circ j_R$, the leftmost of which is $k_1 \circ k_2$. Precisely,

$$
\begin{aligned}
Q^{Ih}_{i_R,j_R,i_S,j_S} =& e^{-\sigma G^{\text{interior}}_{i_R,j_R,i_S,j_S}/RT} + \\
& \sum_{\substack{i_R \le k_1 \le j_R \\ i_S \le k_2 \le j_S}} e^{-\sigma G^{\text{interior}}_{i_R,k_1,i_S,k_2}/RT} Q^{Ih}_{k_1,j_R,k_2,j_S}.
\end{aligned}
\tag{4.13}
$$

Figure 4.8 shows the cases of $Q^{Ia}_{i_R,j_R,i_S,j_S}$ for which at least one of i_R and j_S is the end point of interaction arc: 1) $i_R \bullet k_1$ subsumes $[i_S, k_2]$ and k_2 is not base paired with i_S, 2) $i_S \bullet k_2$ subsumes $[i_R, k_1]$ and i_R is not base paired with k_1, and 3) $i_R \bullet k_1$ and $i_S \bullet k_2$ are

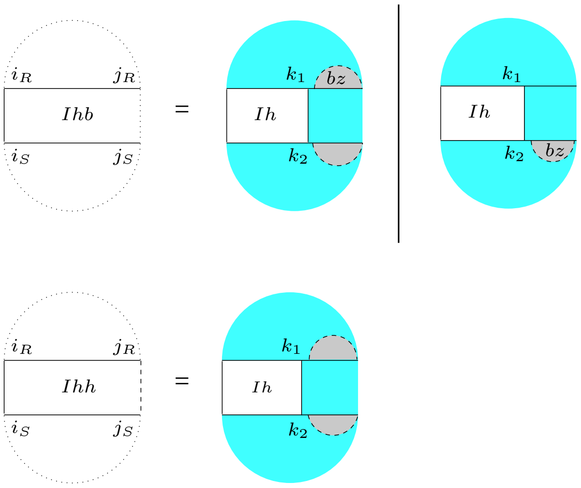


Figure 4.6: Cases of $Q^{Ihb}_{i_R,j_R,i_S,j_S}$ and $Q^{Ihh}_{i_R,j_R,i_S,j_S}$ whose region contains one hybrid component on the left. Here, region $[i_R, k_1] \times [i_S, k_2]$ represents a hybrid component. Figure 4.7 shows the recursion for Q^{Ih}.

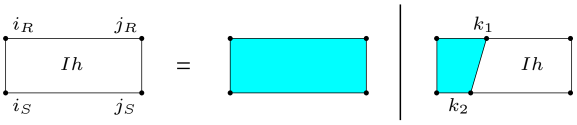


Figure 4.7: Cases of $Q^{Ih}_{i_R,j_R,i_S,j_S}$ the interaction partition function for a single hybrid component.

equivalent. If only one of i_R and i_S is the end point of an interaction arc while the other one is the end point of a bond, then the interaction arc subsumes the other subsequence. If both i_R and i_S are end points of interaction arcs, then one of the arcs subsumes the other one or they are equivalent. Therefore,

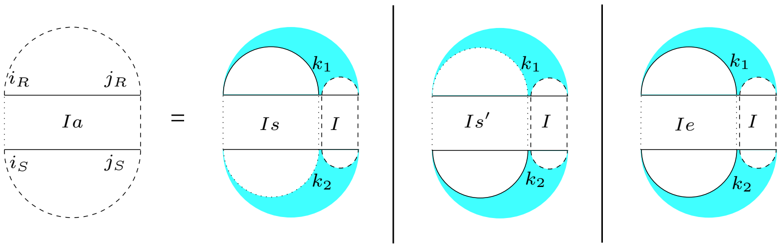


Figure 4.8: Cases of $Q^{Ia}_{i_R,j_R,i_S,j_S}$, for which we assume at least one of i_R and j_S is the end point of an interaction arc.

$$\begin{aligned} Q^{Ia}_{i_R,j_R,i_S,j_S} = & \sum_{\substack{i_R<k_1\leq j_R\\ i_S\leq k_2\leq j_S}} Q^{Is}_{i_R,k_1,i_S,k_2} Q^{I}_{k_1+1,j_R,k_2+1,j_S} + \\ & \sum_{\substack{i_R\leq k_1\leq j_R\\ i_S<k_2\leq j_S}} Q^{Is'}_{i_R,k_1,i_S,k_2} Q^{I}_{k_1+1,j_R,k_2+1,j_S} + \\ & \sum_{\substack{i_R<k_1\leq j_R\\ i_S<k_2\leq j_S}} Q^{Ie}_{i_R,k_1,i_S,k_2} Q^{I}_{k_1+1,j_R,k_2+1,j_S}, \end{aligned} \tag{4.14}$$

in which $Q^{Is}_{i_R,k_1,i_S,k_2}$ is the interaction partition function of $[i_R,k_1]\times[i_S,k_2]$ assuming $i_R\bullet k_1$ is an interaction arc that subsumes $[i_S,k_2]$, $Q^{Is'}_{i_R,k_1,i_S,k_2}$ is the symmetric counterpart of Q^{Is}, and $Q^{Ie}_{i_R,k_1,i_S,k_2}$ is the interaction partition function of $[i_R,k_1]\times[i_S,k_2]$ assuming $i_R\bullet k_1$ and $i_S\bullet k_2$ are equivalent interaction arcs.

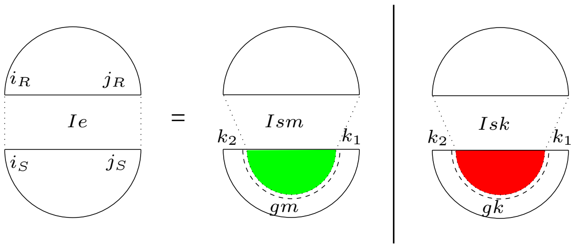


Figure 4.9: Cases of $Q^{Ie}_{i_R,j_R,i_S,j_S}$, for which $i_R\bullet j_R$ and $i_S\bullet j_S$ are equivalent interaction arcs.

For Q^{Ie}, it does not make any difference which one of the covering arcs $i_R\bullet j_R$ and $i_S\bullet j_S$ is extracted first. We first extract the covering arc from **S** (see Figure 4.9). Extracting the covering arc, the remaining subsequence of **S** contains either at least one direct bond, in which case kissing loop penalty should be applied, or multiple interaction arcs, in which case multiloop penalty should be applied. Hence, Figure 4.9 is appropriately colored by green and red to remind the type of penalty. So, we have

$$Q^{Ie}_{i_R,j_R,i_S,j_S} = \sum_{i_S<k_1<k_2<j_S} Q^{Ism.green}_{i_R,j_R,k_1,k_2} Q^{gm}_{i_S,k_1-1,k_2+1,j_S} + Q^{Isk.red}_{i_R,i_S,k_1,k_2} Q^{gk}_{i_S,k_1-1,k_2+1,j_S}. \quad (4.15)$$

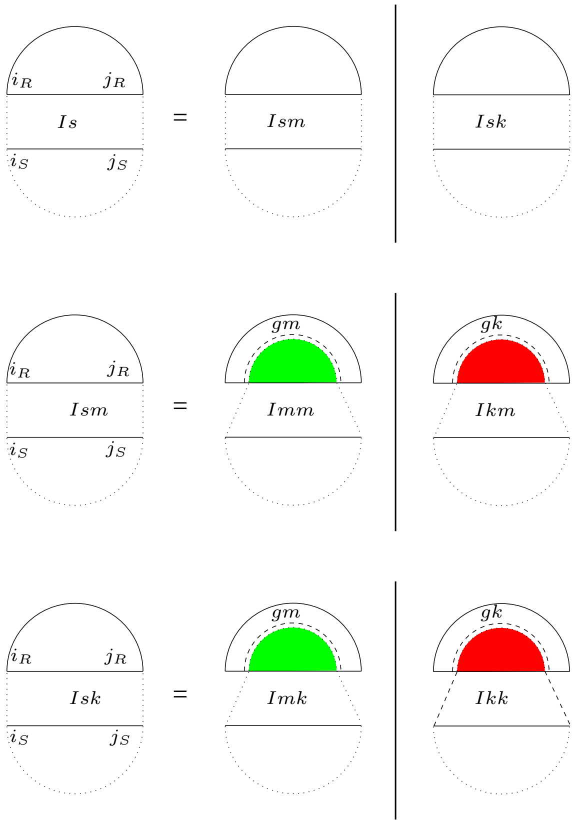


Figure 4.10: Recursion for $Q^{Is}_{i_R,j_R,i_S,j_S}$, interaction partition function assuming $i_R \bullet j_R$ is an interaction arc subsuming $[i_S, j_S]$. In Q^{Ism}, $[i_S, j_S]$ contains multiple interaction arcs and in Q^{Isk}, $[i_S, j_S]$ contains at least one direct bond.

Assuming $i_R \bullet j_R$ is an interaction arc that subsumes $[i_S, j_S]$, $Q^{Is}_{i_R,j_R,i_S,j_S}$ is the partition function for $[i_R, j_R] \times [i_S, j_S]$. Since the union of the cases of Q^{Isk} and Q^{Ism} comprise the cases of Q^{Is},

$$Q^{Is}_{i_R,j_R,i_S,j_S} = Q^{Isk}_{i_R,j_R,i_S,j_S} + Q^{Ism}_{i_R,j_R,i_S,j_S}. \tag{4.16}$$

In particular, Q^{Isk} contains all cases of Q^{Is} in which $[i_S, j_S]$ has at least one direct bond, and Q^{Ism} contains all cases of Q^{Is} in which $[i_S, j_S]$ includes multiple interaction arcs. Similarly, we extract the covering arc from Q^{Isk} and Q^{Ism} to obtain Q^{Imm}, Q^{Imk}, Q^{Ikm}, and Q^{Ikk}, where k stands for kissing (or equivalently containing a direct bond) and m for multiple interaction arcs. The quantities $Q^{Imm}_{i_R,j_R,i_S,j_S}$, $Q^{Ikm}_{i_R,j_R,i_S,j_S}$, $Q^{Imk}_{i_R,j_R,i_S,j_S}$, and $Q^{Ikk}_{i_R,j_R,i_S,j_S}$ are defined by recursions in Figs. 4.11, 4.12, 4.13, and 4.14. Note that all four terminal bases of their region can only be the end points of a bond or of an interaction arc. In summary:

- Q^{Imm} includes all cases that have multiple interaction arcs in both $[i_R, j_R]$ and $[i_S, j_S]$.
- Q^{Imk} includes all cases where $[i_R, j_R]$ has multiple interaction arcs and $[i_S, j_S]$ has at least one direct bond.
- Q^{Ikm} is symmetric to Q^{Imk} with respect to $\mathbf{R}$ and $\mathbf{S}$.
- Q^{Ikk} includes all cases where both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one direct bond.

In Q^{Imm}, both subsequences $[i_R, j_R]$ and $[i_S, j_S]$ include multiple interaction arcs and have no direct bond (Figure 4.11). All four terminal bases are endpoints of interaction arcs. Since i_R and j_S are endpoints of interaction arc, there must exist an Q^{Ia} on the left side of the region. This Q^{Ia} has no direct bond on both subsequences from $\mathbf{R}$ and $\mathbf{S}$, which we call $Q^{Ia_{nn}}$. The bases j_R and i_S are also end points of interaction arc, so there are interaction arcs on the right side of the Q^{Imm} in both subsequences. These arcs can have three types: 1) arc in subsequence $[i_R, j_R]$ subsumes the arc in subsequence $[i_S, j_S]$, 2) arc in subsequence $[i_S, j_S]$ subsumes the arc in subsequence $[i_R, j_R]$, or 3) two arcs are equivalent. Note that for multiple interaction arcs there are an Q^{Ie}, Q^{Is} or $Q^{Is'}$ on both left and right side of the region. The left one is contained in an extracted Q^{Ia}, and the right one is extracted separately. This scheme will continue for the other cases with multiple interaction arcs.

In Q^{Imk}, subsequence $[i_R, j_R]$ has multiple interaction arcs and subsequence $[i_S, j_S]$ has at least one direct bond (Figure 4.12). Here, i_R and j_R are the end points of an interaction arc and i_S and j_S are the end points of a bond or interaction arc. Since i_R is the end point of an interaction arc, there must exist an Q^{Ia} on the left side of the region. The Q^{Ia}

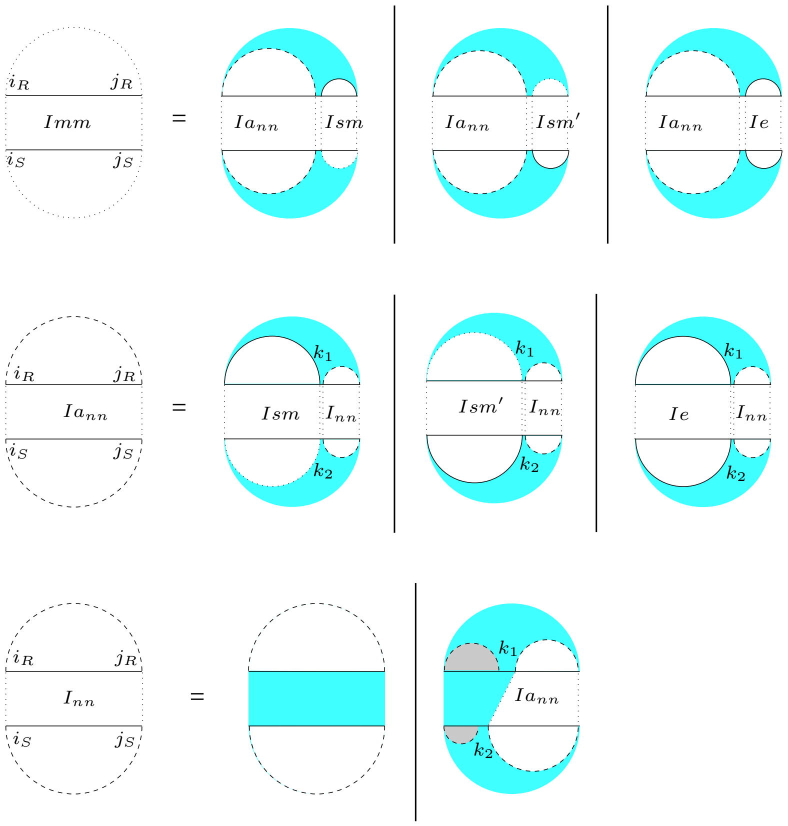


Figure 4.11: Recursions for $Q^{Imm}_{i_R,j_R,i_S,j_S}$ assuming both $[i_R, j_R]$ and $[i_S, j_S]$ have multiple interaction arcs.

has no direct bond in the subsequence of **R**, but it can have two cases with direct bond in subsequence of **S**. We denote the special Q^{Ia} that has at least one direct bond in the subsequence of **S** by $Q^{Ia_{nd}}$. In this case, the arc on the right side of the subsequence of **R** can have three types: 1) it subsumes an interacting region in $[i_S, j_S]$, 2) it is subsumed by the interaction arc on the right side of $[i_S, j_S]$, 3) it is equivalent to the interaction arc on the right side of $[i_S, j_S]$. Note that the arc on $[i_S, j_S]$ can only subsume subsequences with multiple interaction arcs. If Q^{Ia} has no direct bond in **S** subsequence (denoted by $Q^{Ia_{nn}}$), the arc on the right side of $[i_R, j_R]$ should subsume a subsequence on the right side of $[i_S, j_S]$ that has at least direct bond. The quantity Q^{Ikm} is symmetric to Q^{Imk} with respect to **R** and **S** (Figure 4.13).

In Q^{Ikk}, both subsequences of **R** and **S** have at least one direct bond, and all four terminal bases of the region can be end points of bond or interaction arc (Figure 4.14). We go through the cases based on different possibilities of terminal bases. If two terminal bases at the same side of the region are end points of a bond, then obviously they are base paired, otherwise at least one of them is the end point of an interaction arc.

In the first case of Figure 4.14, all four terminal bases are end points of bond, i.e. $i_R \circ j_S$ and $j_R \circ i_S$. This case is similar to Q^{Ib} with a bond on its right. We denote this special Q^{Ib} by Q^{Ib_r} which is shown in Figure 4.15. If just $i_R \circ j_S$, then there is an Q^{Ib} on left side of the region. In that case, the right side has three cases: 1) the right side of $[i_R, j_R]$ contains an interaction arc that subsumes a subsequence on the right side of $[i_S, j_S]$, 2) the right side of $[i_S, j_S]$ contains an interaction arc that subsumes a subsequence on the right side of $[i_R, j_R]$, and 3) there are equivalent interaction arcs on the right sides of $[i_R, j_R]$ and $[i_S, j_S]$. If just $j_R \circ i_S$, then the case is similar to an Q^{Ia} with a bond on its right. We denote this special Q^{Ia} by Q^{Ia_r} (Figure 4.15).

Now consider cases in which terminal bases neither on the left nor on the right make bond with one another. In this type of cases, there must exist an Q^{Ia} on the left side of the region. This Q^{Ia} may contain direct bonds on either subsequence. Denote the special Q^{Ia} that has at least one direct bond in both subsequences by $Q^{Ia_{dd}}$. The right side of the region has three cases: 1) there is an interaction arc on the right side of the remaining subsequence of $\mathbf{R}$ that subsumes a subsequence on the right side of $\mathbf{S}$, 2) there is an interaction arc on the right side of the subsequence of $\mathbf{S}$, that subsumes a subsequence on the right side of $\mathbf{R}$, and 3) there are equivalent interaction arcs on the right sides of the subsequences of $\mathbf{R}$ and $\mathbf{S}$. Denote the special Q^{Ia} that has at least one direct bond in the subsequence of $\mathbf{R}$ by $Q^{Ia_{dn}}$. There must exist an interaction arc on the right side of the subsequence of $\mathbf{R}$ that subsumes a subsequence on the right side of $\mathbf{S}$. Note that the subsequence on the right side of $\mathbf{S}$ should have at least one direct bond. We denote the special Q^{Ia} that has at least one direct bond in the subsequence of $\mathbf{S}$ by $Q^{Ia_{nd}}$. In that case, there must exist an interaction arc on the right side of the subsequence of $\mathbf{S}$ that subsumes a subsequence on the right side of $\mathbf{R}$. Note that the subsequence on the right side of $\mathbf{R}$ should have at least one direct bond.

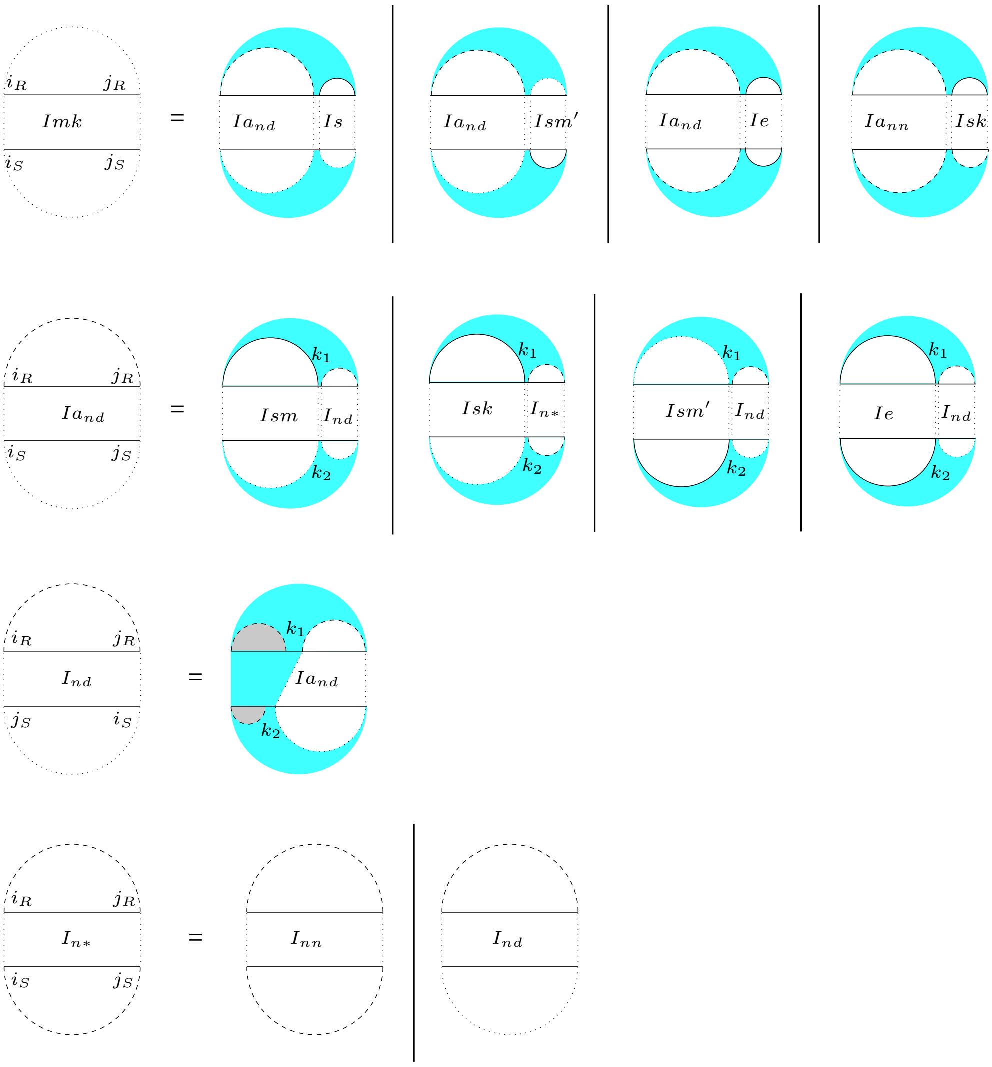


Figure 4.12: Recursions for $Q^{Imk}_{i_R,j_R,i_S,j_S}$ assuming $[i_R, j_R]$ has multiple interaction arcs and $[i_S, j_S]$ has at least one direct bond.

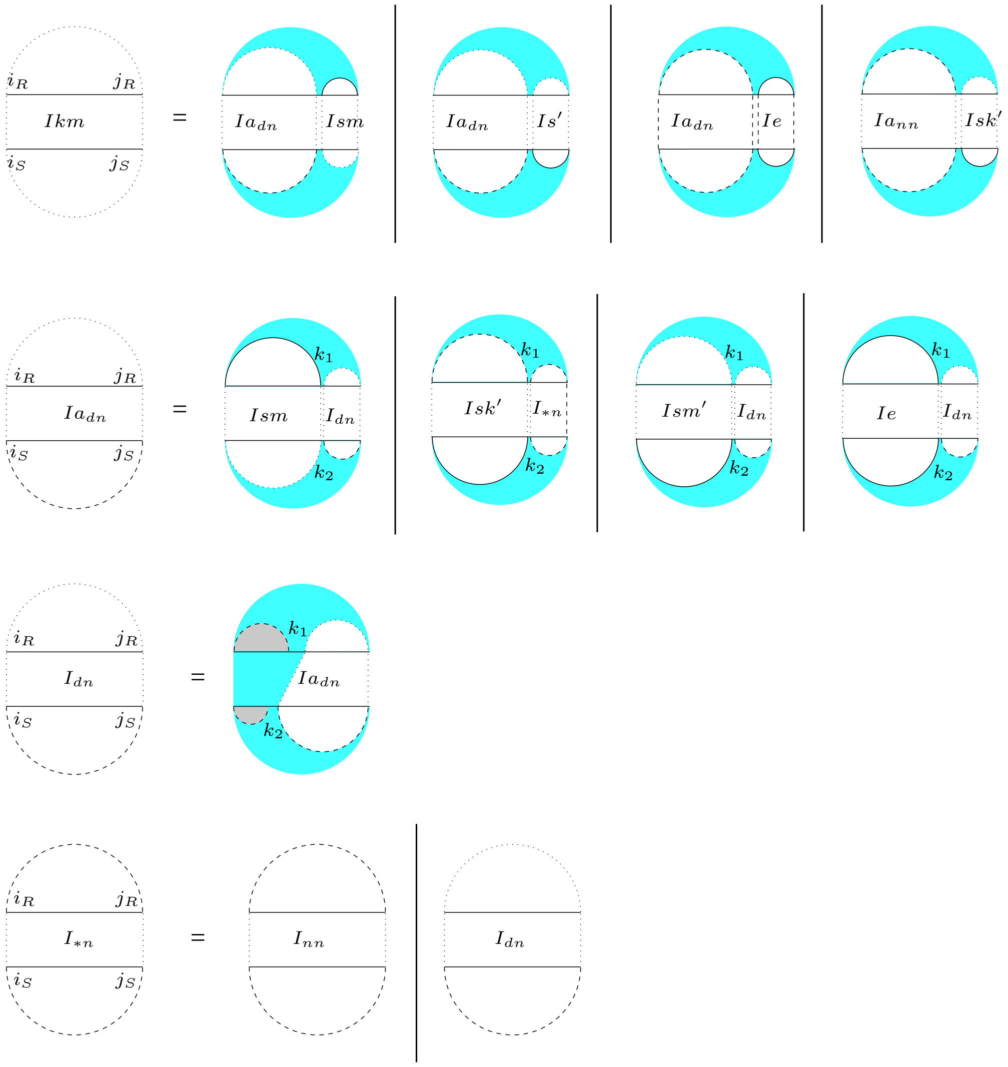


Figure 4.13: Recursions for $Q^{Ikm}_{i_R,j_R,i_S,j_S}$ assuming $[i_R, j_R]$ has at least one direct bond and $[i_S, j_S]$ has multiple interaction arcs.

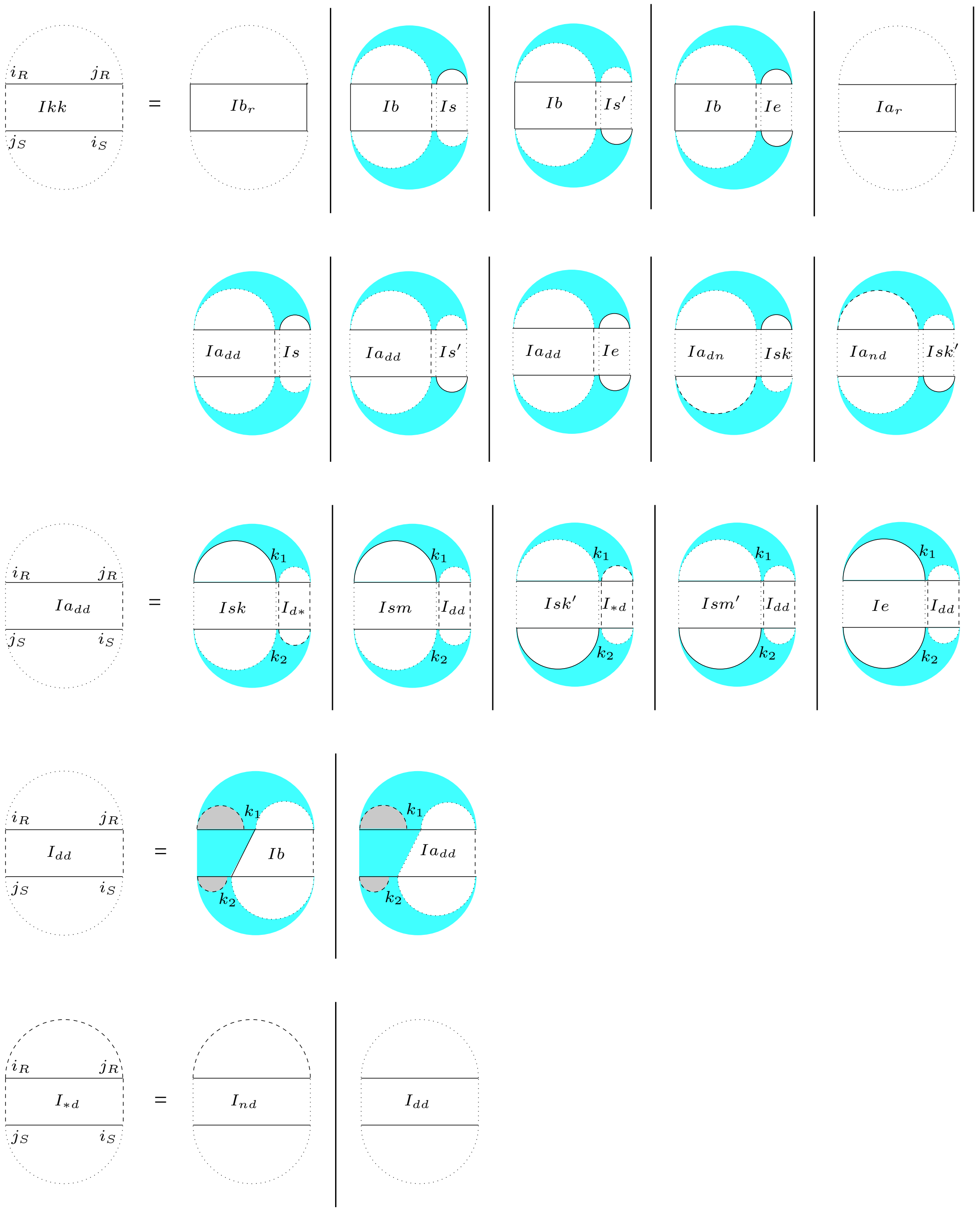


Figure 4.14: Recursions for $Q^{Ikk}_{i_R,j_R,i_S,j_S}$ assuming both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one direct bond.

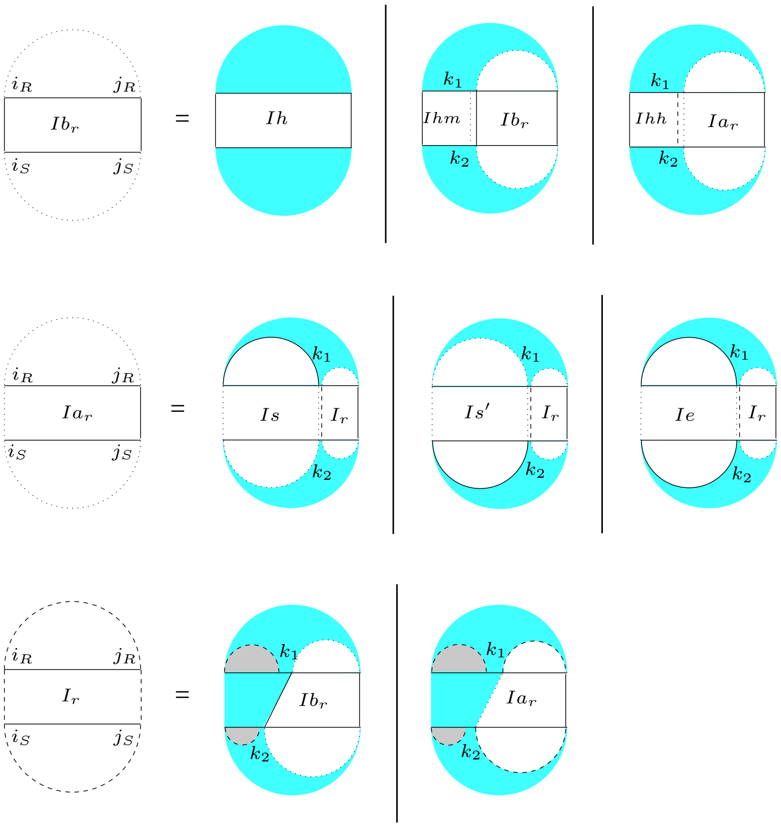


Figure 4.15: The quantities Q^{I_r}, Q^{Ib_r} and Q^{Ia_r} are some auxiliary quantities similar to Q^I, Q^{Ib} and Q^{Ia} except that there is a bond on their right side.

4.3 Applications

We describe how to compute the equilibrium concentrations, heat capacity and melting temperature from partition functions based on the method in [28].

Given two nucleic acid strands **R** and **S**, the concentrations of **R**, **S**, **RR**, **SS**, **RS** species, denoted by $N_{\mathbf{R}}$, $N_{\mathbf{S}}$, $N_{\mathbf{RR}}$, $N_{\mathbf{SS}}$, $N_{\mathbf{RS}}$ respectively. The partition function Q for a system at a given temperature T, volume V and all possible distributions of the initial material, N^0, between the species is computed as:

$$Q = \sum \frac{N^0_{\mathbf{R}}!N^0_{\mathbf{S}}!}{N_{\mathbf{R}}!N_{\mathbf{S}}N_{\mathbf{RR}}!N_{\mathbf{SS}}!N_{\mathbf{RS}}!}(Q_{\mathbf{R}})^{\mathbf{R}}(Q_{\mathbf{S}})^{\mathbf{S}}(Q^I_{\mathbf{RR}})^{\mathbf{RR}}(Q^I_{\mathbf{SS}})^{\mathbf{SS}}(Q^I_{\mathbf{RS}})^{\mathbf{RS}} \tag{4.17}$$

In the equilibrium, the free energy of a closed system at constant temperature, volume, and pressure tends toward a minimum [56]. The equilibrium distributions of $N_{\mathbf{R}}$, $N_{\mathbf{S}}$, $N_{\mathbf{RR}}$, $N_{\mathbf{SS}}$, and $N_{\mathbf{RS}}$ are determined by the minimization of the free energy.

Equilibrium concentrations are calculated from the chemical equilibrium constants

$$\begin{aligned} K_{\mathbf{R}} &= \frac{Q^I_{\mathbf{RR}}}{Q^2_{\mathbf{R}}} = \frac{N_{\mathbf{RR}}}{N^2_{\mathbf{R}}}, \\ K_{\mathbf{S}} &= \frac{Q^I_{\mathbf{SS}}}{Q^2_{\mathbf{S}}} = \frac{N_{\mathbf{SS}}}{N^2_{\mathbf{S}}}, \\ K_{\mathbf{RS}} &= \frac{Q^I_{\mathbf{RS}}}{Q_{\mathbf{R}}Q_{\mathbf{S}}} = \frac{N_{\mathbf{RS}}}{N_{\mathbf{R}}N_{\mathbf{S}}}, \end{aligned} \tag{4.18}$$

under the constraint

$$N_{\mathbf{RS}} = N^0_{\mathbf{R}} - 2N_{\mathbf{RR}} - N_{\mathbf{R}} = N^0_{\mathbf{S}} - 2N_{\mathbf{SS}} - N_{\mathbf{S}}. \tag{4.19}$$

The chemical potentials of the species can be obtained by differentiating the free energy $-RTln(Q)$ with respect to the concentrations of their corresponding molecules. Thus we have $\mu_{\mathbf{R}} = -RT\frac{\partial ln(Q)}{\partial N_{\mathbf{R}}}$ for **R** (as well as $\mu_{\mathbf{S}}$, $\mu_{\mathbf{RR}}$, $\mu_{\mathbf{SS}}$, and $\mu_{\mathbf{RS}}$ for the other species).

The free energy of the whole ensemble of species can be represented as

$$F = \mu_{\mathbf{R}}N_{\mathbf{R}} + \mu_{\mathbf{S}}N_{\mathbf{S}} + \mu_{\mathbf{RR}}N_{\mathbf{RR}} + \mu_{\mathbf{SS}}N_{\mathbf{SS}} + \mu_{\mathbf{RS}}N_{\mathbf{RS}}. \tag{4.20}$$

The heat capacity is a quantity for the amount of heat that is required to raise the temperature of the solution. Heat capacity is expressed in units of joules per kelvin. From statistical thermodynamics it is well known that the heat capacity, C_p, is derived from the second derivative of the free energy, F, with respect to the temperature T.

$$C_p = -T\left(\frac{\partial^2 F}{\partial T^2}\right)_p \tag{4.21}$$

One has to make the above calculations for varying temperatures over the desired range. To compute C_p for a particular temperature T_k an approach based on the derivation proposed by Vienna Group [47] is used. Melting temperature T_m is the temperature at witch $\frac{\partial C_p}{\partial T} = 0$, where 50% of the strands are unfolded.

4.4 Experimental Results

Here, we report our implementation of the algorithm and two types of experiments we performed to test the predictive power of our algorithm:
(1) A novel experiment (which, to our knowledge has not been performed successfully by any other program to date), uses our algorithm to predict the equilibrium concentration of an RNA-RNA complex, in particular the OxyS-fhlA interaction [9][1]. We successfully predicted the equilibrium concentrations for OxyS with wild-type fhlA and 4 other fhlA mutants.
(2) Predicting the melting temperature of RNA duplexes is an important application of the partition function for interacting nucleic acid pairs [28]; our first experiment thus test how accurately our algorithm predicts the melting temperature of RNA pairs collected from the several sources in the literature with respect to the accuracy of available alternatives, `RNAcofold` from Vienna package v1.7.2 [17] and `UNAFold v3.6` which is a new version of former `mfold` [62]. We remind the reader that `RNAcofold` concatenates the two RNA strands and computes the partition function for the resulting single strand. Therefore, it does not consider many cases that our algorithm considers. `UNAFold v3.6`, on the other hand, simplifies the problem by forbidding intramolecular base pairing. It computes the partition function of the two strands over just hybridization structures. As can be expected, our algorithm consistently outperforms the alternatives in all three data sets.

Note that the parameters used by our program in the above experiments have been manually optimized as computational learning methods for fine tuning the parameters require prohibitive computational resources. It may be possible to improve the accuracy of our program through a better selection of parameters.

[1] Equilibrium concentrations of another complex formed by CopA/I-CopT is also available in the literature [45], however the interaction has tertiary structural components, i.e. a very long pair of kissing hairpins forming a helix, anti-helix pair with a long gap in between. Alkan et al [4] were able to establish the most likely joint structure between this RNA pair only through post processing. This complex requires some additional constraints on the lengths of interacting loops which are not incorporated into our model due to additional computational complexity they would impose.

4.4.1 Implementation

We remind the reader that the time and space complexity of our algorithm are $O(n^6)$ and $O(n^4)$ respectively; here $n = \max(L_R, L_S)$ is the maximum length of the two input strands. We implemented our algorithm in C++, and used the energy functions and energy parameters of `UNAFold v3.6` for a single strand [62]. For our own interaction energy model, the parameters used by our program are given in the next section. We use a different β_1 penalty and σ for a hybrid component that is covered by a kissing loop. The parameters for a hybrid component that is not covered by a kissing loop is denoted by β'_1 and σ'. Similar to `RNAhybrid`, the interior loops in a hybrid component are restricted to a constant maximum length, in either sequence, which is set to 15 in this work.

Since our algorithm considers many more possible secondary structures in comparison to alternative methods, our program has a higher running time. Fortunately, our algorithm can be easily parallelized as the dynamic programming tables computed by our program on subsequence pairs depend only on their (proper) subregions. We parallelized our program using OpenMP 3.0. Our experiments were performed on a large scale shared memory parallel platform with 64 PPC 1.9 GHz processors with 256 GB RAM. We ran our program for strands of length between 5 nt to 120 nt. The running time of our program for short strands (∼20 nt) was less than 1 minute - for longer strands (∼120 nt) it was about 10 hours.

4.4.2 Data Sets

The first data set that we used for predicting melting temperature contains all 9 different RNA pairs reported in Table 3 of [101]. It contains almost complementary 5-7nt RNA pairs that were designed to optimize the thermodynamic parameters for terminal base pairs. Their melting temperatures vary from 29.8°C to 53.7°C.

The second data set that we used for computing melting temperature contains all 12 different RNA pairs reported in Table 1 of [27]. These RNA pairs are designed to optimize the thermodynamic parameters for three-way multi loops. In each pair of this data set, the first RNA has ∼20nt and the second one has ∼10nt. The experimental melting temperatures were determined from heat absorption measurements by two different methods which are explained as ”Method 3” and ”Method 4” in [75]. Although these pairs are very similar, the average difference of the two methods for this data set is 2.49°C. This suggests that there may exist RNA pairs with exceptional features in this set.

The third data set that we used for computing melting temperature contains all 62 different RNA pairs reported in Tables 3 and 4 of [64]. These pairs are designed to optimize the thermodynamic parameters for three- and four-way multi loops. In each pair of this data set, the first RNA has 22-40nt and the second one has 10-14nt. Again, the experimental melting temperatures were determined by two different methods. This data set is large enough with longer sequences, and the average difference of the two methods for this data set is 0.7°C, smaller than that of the second data set. Moreover, the variance and maximum of the difference is smaller than those of the second data set. Overall, this data set is more reliable than the previous one. These three data sets are all we were able to collect from the literature.

4.4.3 Equilibrium Concentration

Our first set of experiments, to the best of our knowledge, have not been successfully performed by the use of any available program to date. Here we predict the equilibrium concentrations for OxyS with wild-type fhlA and 4 other fhlA mutants. OxyS is a small untranslated RNA (109 nt) that is induced in response to oxidative stress in E. coli. It acts as a regulator affecting the expression of multiple genes. In particular, OxyS represses the translation of fhlA, a transcriptional activator for formate metabolism, by binding to it. Argaman and Altuvia carried out a series of experiments to measure equilibrium dissociation constants for OxyS with wild-type fhlA and its mutants [9]. To measure the equilibrium dissociation constants, they measured the concentration of OxyS-fhlA complex for a fixed initial OxyS concentration (2nM) and various initial concentrations of fhlA. Their plots are reported in Figure 8 and Table 2 of [9]. Those plots can be predicted from the partition functions for OxyS, fhlA, OxyS-OxyS, fhlA-fhlA, and OxyS-fhlA. To validate our algorithm, we computed these partition functions using our program, and predicted the equilibrium concentrations of OxyS-fhlA complex. Our results are compatible with experimental measurements, as we had expected.

Figure 4.16 shows the experimental measurements and our results. Interestingly, our algorithm predicted the equilibrium concentration of OxyS-fhlA complex quite accurately for the wild-type fhlA and all of its mutants. Note that although we also experimented with RNAcofold and UNAFold in this case, we do not report on their results as they significantly differed from the experimental measurements. This is probably not very surprising as correctly predicting the equilibrium concentrations is a very difficult task and is highly sensitive to the accuracy of the partition functions. We noticed that $Q_{\mathbf{R}}$ and $Q_{\mathbf{S}}$ computed by the three programs are very close because they use the same algorithm for a single strand (i.e. McCaskill's). Therefore based on (4.18), a method can compute equilibrium concentrations correctly only if it computes each individual Q^I accurately. As one can observe in Figure 4.16, our program has been able to predict OxyS-fhlA complex concentrations accurately, thus we can conclude that our program computes Q^I for OxyS-OxyS, fhlA-fhlA, and OxyS-fhlA accurately.

As mentioned above, the parameters used by our program on this data set have been manually optimized. Our energy parameters in this experiment are

$$\beta_1 = 6.6, \beta_2 = \beta_2 = 0.1, \sigma = 0.9, \tag{4.22}$$

$$\beta_1' = 4.5, \sigma' = 0.9. \tag{4.23}$$

4.4.4 Melting Temperature

As mentioned before, predicting the melting temperature of RNA duplexes is one of the most important applications of the partition function for interacting nucleic acid pairs [28]. Table 4.1 shows the melting temperatures computed by our program, `RNAcofold`, and `UNAFold v3.6` for the first data set. In this set, the strands are short, and as we expected, our algorithm is highly accurate with only 1.48°C difference from experimental values on average. It can be seen that `RNAcofold` and `UNAFold` perform relatively poorly, and their predicted melting temperatures differ from the experimental values by about 9°C on average.

Table A.1 shows the melting temperatures predicted by the three programs for the second data set. Each pair is referred to by an identifier $(A, B, \ldots, L)$. Please refer to the Appendix A or [27] to see the exact sequences of each pair. As mentioned before, the experimental melting temperatures were determined from heat absorbance measurements by two different methods which are explained as ”Method 3” and ”Method 4” in [75]. We refer to the melting temperature values computed by ”Method 3” and ”Method 4”, by T_c and T_l respectively. `RNAcofold` accuracy obviously dropped in this case, whereas `UNAFold` accuracy did not change much in comparison to the results for the first data set. The accuracy of our method has also dropped a bit, which may be because of some RNA pairs with exceptional features.

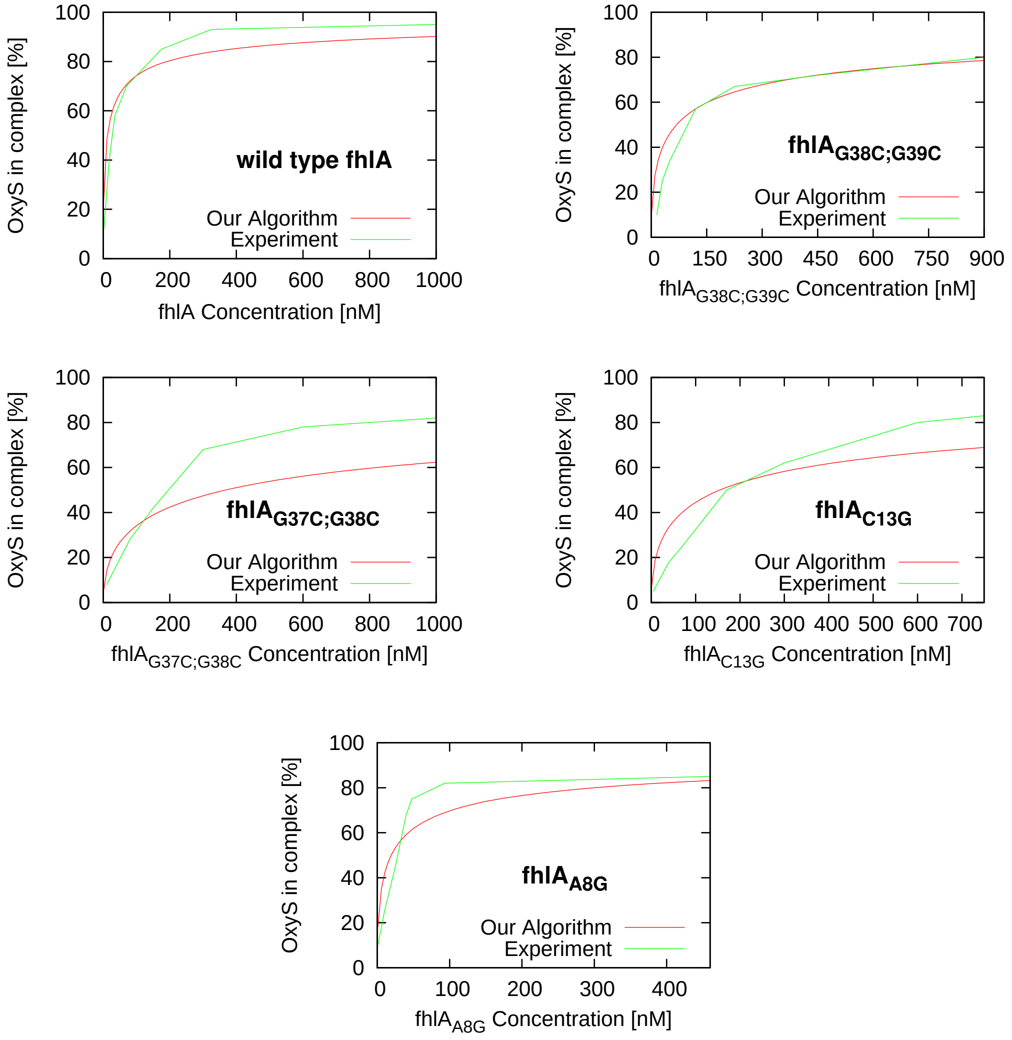


Figure 4.16: Experimental and computational determination of equilibrium constants for pairs of OxyS with wild-type and mutated fhlA. Horizontal axis denotes the initial concentration of fhlA, and the vertical axis denotes the percentage of OxyS in OxyS-fhlA complex. Initial concentration of OxyS was $2 \times 10^{-9} M$ [9].

Table 4.1: Experimental and predicted melting temperatures for the first data set (see Section 4.4.2 and [101]).

Pairs	Experiment	piRNA	RNAcofold	UNAFold
ACGCA/UGCGU	29.8	29.41	42.64	46.14
GCACG/CGUGC	37.5	36.07	46.61	43.91
AGCGA/UCGCU	30.2	30.38	42.68	45.15
GCUCG/CGAGC	37.2	36.88	47.75	44.71
ACUGUCA/UGACAGU	48.2	44.91	56.8	57.59
GUCACUG/CAGUGAC	51.1	49.4	58.44	55.91
AGUCUGA/UCAGACU	45.7	45.47	56.4	56.68
GACUCAG/CUGAGUC	52	49.96	59.11	56.25
GAGUGAG/CUCACUC	53.7	49.97	59.07	56.00
Avg. error		1.48	9.35	8.55

Experimental melting temperatures are calculated using the linear plots of T_M^{-1} vs $\ln(C_T/4)$. Buffer was 1.0 M NaCl, 20 mM sodium cacodylate, and 0.5 mM Na2EDTA, pH 7.0 at 0.2 mM dulexes. All values are in °C.

Table 4.3 presents the melting temperatures predicted by the three programs for the third data set. As you can see, our program has high accuracy and performs significantly better than `RNAcofold` and `UNAFold` for this data set. As we argued before, the third data set is the largest and the most reliable of the three data sets. It is important to note that `RNAcofold` and `UNAFold` both perform poorly either in this case or the two previous cases. Therefore, neither `RNAcofold` nor `UNAFold` are as reliable as our program for melting temperature prediction.

Table 4.3: Predicted melting temperatures for the set RNA pairs from [64].

Pairs	Experiment		piRNA	RNAcofold	UNAFold
	T_l	T_c			
G-GC-G/C-C	45.4	46	56.81	37	21.4
G-GC-G/CaC	51.8	52.2	56.84	37	27.15
G-GC-G/Ca$_2$C	55.9	56	56.86	37	27.12
G-GC-G/Ca$_3$C	58.4	57.3	56.85	37	25.73
G-GC-G/CauaC	57.3	56.9	56.84	37	24.35

Continued on next page

Table 4.3 – continued from previous page

Pairs	Experiment		piRNA	RNAcofold	UNAFold
	T_l	T_c			
G-GC-G/Ca$_4$C	56.7	57.1	56.85	37	25.06
GaGC-G/C-C	51.2	49.3	56.94	37.1	21.25
GaGC-G/CaC	55.2	54.7	56.96	41	21.44
GaGC-G/Ca$_2$C	56.1	55.6	56.98	41	21.46
GaGC-G/Ca$_3$C	56.1	54.9	56.97	41	21.44
GaGC-G/CauaC	54.8	54.3	56.98	37.06	21.47
GaGC-G/Ca$_4$C	55.3	54.8	56.98	37	21.39
Ga$_2$GC-G/C-C	55.1	52.9	57	50.17	36.04
Ga$_2$GC-G/CaC	57	56.4	57.03	48.01	36.8
Ga$_2$GC-G/Ca$_2$C	55.6	55.4	57.05	41.84	36.91
Ga$_2$GC-G/Ca$_3$C	55	54.7	57.03	41.84	36.19
Ga$_2$GC-G/CauaC	55.3	54.5	57.3	52.17	36.74
Ga$_2$GC-G/Ca$_4$C	54.1	53.9	57.05	48.01	34.22
Ga$_2$GCaG/C-C	56.6	56.6	57.18	47.01	36.01
Ga$_2$GCaG/CaC	58.7	58.9	57.18	44.1	36.81
Ga$_2$GCaG/Ca$_2$C	58	58.8	57.2	44.1	36.13
Ga$_2$GCaG/Ca$_3$C	56.5	57.5	57.15	44.1	36.96
Ga$_2$GCaG/CauaC	57.2	56.9	57.48	43	35.92
Ga$_2$GCaG/Ca$_4$C	57.9	57.9	57.17	44.1	34.66
Ga$_2$GCa$_2$G/C-C	56	56.9	57.19	37.17	36.14
Ga$_2$GCa$_2$G/CaC	58.7	59.1	57.2	44.1	36.94
Ga$_2$GCa$_2$G/Ca$_2$C	59.7	59.6	57.19	44.1	36.22
Ga$_2$GCa$_2$G/Ca$_3$C	58.6	58.7	57.16	44.1	35.89
Ga$_2$GCa$_2$G/CauaC	57	57.3	57.74	37	35.03
Ga$_2$GCa$_2$G/Ca$_4$C	57.5	58.1	57.18	44.1	34.93
G-UA-G/C-C	50.4	50.8	56.82	46.26	21.53
G-UA-G/CaC	54.3	55.8	57.88	61.42	34.47
G-UA-G/Ca$_2$C	56.6	57.8	57.89	61.42	41.68
G-UA-G/Ca$_3$C	57.6	58.5	57.88	61.42	40.84

Continued on next page

Table 4.3 – continued from previous page

Pairs	Experiment		piRNA	RNAcofold	UNAFold
	T_l	T_c			
G-UA-G/CauaC	57.9	58.7	57.87	61.41	40.96
G-UA-G/Ca$_4$C	58.6	58.5	57.88	61.43	40.64
GaUA-G/C-C	51.6	51.8	56.96	49.18	21.42
GaUA-G/CaC	55.6	55.7	57.01	37.07	30.98
GaUA-G/Ca$_2$C	56.7	57.4	57.04	50.31	31.46
GaUA-G/Ca$_3$C	56.8	56.9	57	44.17	29.91
GaUA-G/CauaC	57	57.1	56.99	37.07	29.98
GaUA-G/Ca$_4$C	56.8	56.8	57.01	50.31	29.29
G-CG-GC-G/C-C	64.8	65.2	57.24	37	21.38
G-CG-GC-G/CaC	58.8	60.4	57.22	37	21.44
G-CG-GC-G/Ca$_2$C	55.6	56.4	57.35	37	21.38
G-CG-GC-G/Ca$_3$C	55.4	55.3	57.32	37	21.56
G-CG-GC-G/Ca$_4$C	53.9	53	57.19	37	21.38
GaCG-GC-G/C-C	57.3	58.7	57.2	37	21.71
GaCG-GC-G/CaC	59.7	61.2	57.21	37	21.76
GaCG-GC-G/Ca$_2$C	55.4	57.2	57.19	37	21.45
GaCG-GC-G/Ca$_3$C	55.2	56.5	57.11	37	21.42
GaCG-GC-G/CauaC	55.2	55.8	57.09	37	21.38
GaCG-GC-G/Ca$_4$C	55	55.3	57.14	37	21.47
GaCG-GCaG/C-C	58.1	58.8	56.9	37	21.54
GaCG-GCaG/CaC	59.3	59.7	56.99	37	21.76
GaCG-GCaG/Ca$_2$C	57.5	59.4	56.89	37	63.08
GaCG-GCaG/Ca$_3$C	57.9	58.2	56.95	37	21.44
GaCG-GCaG/CauaC	58.9	58.3	56.93	37	21.53
GaCG-GCaG/Ca$_4$C	57.3	58.1	56.84	37	21.46
Ga$_2$CGa$_2$GCa$_2$G/C-C	54.4	55.5	57.12	47.17	67.28
Ga$_2$CGa$_2$GCa$_2$G/CaC	55	56.6	57.04	44.01	67.23
Ga$_2$CGa$_2$GCa$_2$G/Ca$_2$C	55.3	57.2	57.12	51.31	66.09
Avg. difference		T_l	T_l T_c	T_l T_c	T_l T_c

Continued on next page

Table 4.3 – continued from previous page

Pairs	Experiment		piRNA		RNAcofold		UNAFold	
	T_l	T_c						
		0.7	1.87	1.95	14.27	14.41	26.5	26.56

Here each pair is referred to by an identifier. Please refer to the Appendix A or [64] to see the exact sequences of each pair. T_l is calculated using the linear plots of T_M^{-1} vs $\ln(C_T/4)$, and T_c is calculated by the average of melt curve fits. Buffer was 1.0 M NaCl, 20 mM sodium cacodylate, and 0.5 mM Na2EDTA, pH 7.0 at 0.1 mM total strand concentration. All values are in °C.

Please note that we did not use any learning methods for tuning our 6 interaction energy parameters because of the running time of our algorithm. Our interaction energy parameters in melting temperature experiments are

$$\beta_1 = 5.1, \beta_2 = \beta_2 = 0.1, \sigma = 0.92, \tag{4.24}$$

$$\beta_1' = 4.1, \sigma' = 0.95, \tag{4.25}$$

which were manually optimized using only the first data set. The second and third data sets were used as test sets.

Table 4.2: Experimental and predicted melting temperatures for the set of RNA pairs reported in [27].

<table>
<tr><th>Pairs</th><th colspan="2">Experiment</th><th colspan="2">piRNA</th><th colspan="2">RNAcofold</th><th colspan="2">UNAFold</th></tr>
<tr><th></th><th>T_l</th><th>T_c</th><th colspan="2"></th><th colspan="2"></th><th colspan="2"></th></tr>
<tr><td>A</td><td>28.7</td><td>30.3</td><td colspan="2">32.44</td><td colspan="2">50.99</td><td colspan="2">21.52</td></tr>
<tr><td>B</td><td>19</td><td>20.5</td><td colspan="2">31.55</td><td colspan="2">52.55</td><td colspan="2">33.22</td></tr>
<tr><td>C</td><td>33.6</td><td>33.6</td><td colspan="2">32.94</td><td colspan="2">53.11</td><td colspan="2">39.77</td></tr>
<tr><td>D</td><td>33.9</td><td>36</td><td colspan="2">32.43</td><td colspan="2">51.02</td><td colspan="2">26.85</td></tr>
<tr><td>E</td><td>23</td><td>24.4</td><td colspan="2">31.66</td><td colspan="2">52.48</td><td colspan="2">32.22</td></tr>
<tr><td>F</td><td>34.9</td><td>36.9</td><td colspan="2">33.28</td><td colspan="2">54.7</td><td colspan="2">39.91</td></tr>
<tr><td>G</td><td>32.4</td><td>33.6</td><td colspan="2">32.76</td><td colspan="2">49.76</td><td colspan="2">64.27</td></tr>
<tr><td>H</td><td>16.1</td><td>18.9</td><td colspan="2">36.41</td><td colspan="2">57.92</td><td colspan="2">29.76</td></tr>
<tr><td>I</td><td>29</td><td>32.3</td><td colspan="2">32.32</td><td colspan="2">50.99</td><td colspan="2">29.18</td></tr>
<tr><td>J</td><td>32.3</td><td>37.1</td><td colspan="2">37.01</td><td colspan="2">56.92</td><td colspan="2">28.8</td></tr>
<tr><td>K</td><td>23.4</td><td>30.7</td><td colspan="2">31.45</td><td colspan="2">49.36</td><td colspan="2">26.18</td></tr>
<tr><td>L</td><td>33.5</td><td>35.4</td><td colspan="2">32.61</td><td colspan="2">50.51</td><td colspan="2">28.01</td></tr>
<tr><td>Avg. difference</td><td></td><td>T_l</td><td>T_l</td><td>T_c</td><td>T_l</td><td>T_c</td><td>T_l</td><td>T_c</td></tr>
<tr><td></td><td></td><td>2.49</td><td>5.53</td><td>4.19</td><td>24.21</td><td>21.72</td><td>8.86</td><td>9.38</td></tr>
</table>

Here each pair is referred to by an identifier $(A, B, \ldots, L)$. Please refer to the Appendix A or [27] to see the exact sequences of each pair. T_l is calculated using the linear plots of T_M^{-1} vs $\ln(C_T/4)$, and T_c is calculated by the average of melt curve fits. Buffer was 1.0 M NaCl, 20 mM sodium cacodylate, and 0.5 mM Na2EDTA, pH 7.0 at 0.1 mM total strand concentration. All values are in °C.

Chapter 5

Efficient RNA-RNA Interaction Prediction via Sparsification

As mentioned earlier, the key problem with the previous approaches for predicting a general joint structure (please see chapter 2) is that they all have a worst case running time of $O(n^6)$ and a space complexity of $O(n^4)$. While this complexity might be acceptable when analyzing only a few putative sRNA-target interaction pairs, we are now faced with the situation that the amount of data to be analyzed is vastly increasing. To give an example, a recent mapping of transcripts using tiling arrays in the budding yeast *S. cerevisiae* [25] with 5,654 annotated open reading frames (ORF) has found 1555 antisense RNAs that overlap at least partially with the ORFs at the opposite strand. Currently, it is completely unclear what these antisense RNAs are doing - whether they target only their associated sense mRNA or have also other mRNA targets, and whether they always form a complete duplex or more complex joint structures such as multiple kissing hairpins if they overlap only partially is not known. The same situation appears in many other species. Thus, there is urgent need for a more time and space efficient interaction prediction method that is able to handle complex joint structures.

In this chapter we present a new method for calculating the joint structure of interacting RNAs by minimizing their total free energy, which improves time and space efficiency over previous approaches. As first in its class, the method is sufficiently fast to be applied in large scale screening approaches.

We show how to reduce both time and space complexity using an approach called *sparsification*, which uses the observation that the resulting DP-matrices are sparse. As previous applications of sparsification to problems related to RNA folding, our approach exploits a triangle inequation on the dynamic programming matrix. Assuming the *polymer-zeta* property for interacting RNAs, we show an efficiency gain by a linear factor. This *polymer-zeta* property basically states that the probability of a base pair decreases with its size, i.e. there are only few long range base pairs.

Here, we consider a version of the polymer-zeta property for interacting RNAs and develop novel algorithmic approaches as (1) we cannot assume the standard polymer-zeta property for all base pairs as for intermolecular base pairs there is no clear notion of a distance between the bases; (2) the joint interaction prediction problem does not allow to split only at arcs in the recursion, which was crucial in the demonstration of a linear (asymptotic) speed up for problems involving the folding of a single RNA.

We sparsify the dynamic programming tables involved in total free energy minimization first described in Alkan *et al.* [4] on our more general Interaction energy model resulting in a significant reduction in time and space complexity. There are four different cases that need to be sped up, which results in a total of four different candidate lists; for each sequence and each region, we have to consider folding with interaction or without interaction, which gives rise to two types of candidate lists. We emphasize that beyond reducing time complexity, we obtain a similar space reduction even in the intricate setting of the independent candidate lists.

5.1 The Algorithm: Agile inteRNA

In this section we discuss an algorithm for RNA-RNA interaction prediction via total free energy minimization, under the assumption that there are no (internal) pseudoknot, crossing bond (i.e. external pseudoknots), or zigzag in the joint structure. We use sparsification techniques to reduce the complexity of the original algorithm from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n) + n^3)$ space for some function $\psi(n) = O(n)$ on average. To simplify the presentation, we discuss the sparsification for the joint structure prediction via total base pair maximization. Note that RNA-RNA interaction based on base pair maximization is the generalized version of the Nussinov model [72] for single RNA folding and was employed by Pervouchine [74] as well as Alkan et al. [4] for RNA-RNA interaction prediction. Later in this chapter we also provide all concepts for generalizing the algorithm to capture our more realistic interaction energy model in Chapter 3.

5.1.1 Sparsification for Maximizing Base Pairs

Given two RNA sequences **R** and **S**, $N(i_R, j_R, i_S, j_S)$ denotes the maximum number of base pairs in the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$, and $N^{\mathbf{X}}(i, j)$ (for $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$) denotes the maximum number of base pairs of the subsequence $[i, j]$ of the single sequence **X**. The recursion cases for computing the maximum number of base pairs for RNA-RNA interaction are illustrated in Figure 5.1. $N(i_R, j_R, i_S, j_S)$ and $N^{\mathbf{X}}(i, j)$ for $\mathbf{X} \in \{\mathbf{R}, \mathbf{S}\}$ are calculated by the following recursions

$$N(i_R, j_R, i_S, j_S) = \max \begin{cases} N(i_R + 1, j_R, i_S, j_S) & (a) \\ N(i_R, j_R, i_S + 1, j_S) & (b) \\ N(i_R + 1, j_R, i_S + 1, j_S) + 1 & (c) \\ \max\limits_{\substack{i_R < k \leq j_R \\ R[i_R], R[k] \text{ compl.}}} \begin{pmatrix} 1 + N^{\mathbf{R}}(i_R + 1, k - 1) \\ + N(k + 1, j_R, i_S, j_S) \end{pmatrix} & (d) \\ \max\limits_{\substack{i_S \leq k < j_S \\ S[i_S], S[k] \text{ compl.}}} \begin{pmatrix} 1 + N^{\mathbf{S}}(i_S + 1, k - 1) \\ + N(i_R, j_R, k + 1, j_S) \end{pmatrix} & (e) \\ \max\limits_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ R[i_R], R[k_R] \text{ compl.}}} \begin{pmatrix} 1 + N(i_R + 1, k_R - 1, i_S, k_S) \\ + N(k_R + 1, j_R, k_S + 1, j_S) \end{pmatrix} & (f) \\ \max\limits_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ S[i_S], S[k_S] \text{ compl.}}} \begin{pmatrix} 1 + N(i_R, k_R, i_S + 1, k_S - 1) \\ + N(k_R + 1, j_R, k_S + 1, j_S) \end{pmatrix} & (g) \end{cases} \tag{5.1}$$

$$N^{\mathbf{X}}(i, j) = \max \begin{cases} N^{\mathbf{X}}(i + 1, j) & (a) \\ \max\limits_{\substack{i < k \leq j \\ X[i], X[k] \text{ compl.}}} \begin{pmatrix} 1 + N^{\mathbf{X}}(i + 1, k - 1) \\ + N^{\mathbf{X}}(k + 1, j) \end{pmatrix} & (b) \end{cases} \tag{5.2}$$

In Eq. 5.1, the cases (a) and (b) introduce an unpaired base at positions i_R and i_S respectively, and case (c) introduces a bond $i_R \circ i_S$. Cases (d) and (f) introduce an arc at $i_R \bullet k$ and cases (e) and (g) at $i_S \bullet k$, where cases (f) and (g) assume that the arc is an interaction arc and cases (d) and (e) assume that this is not the case.

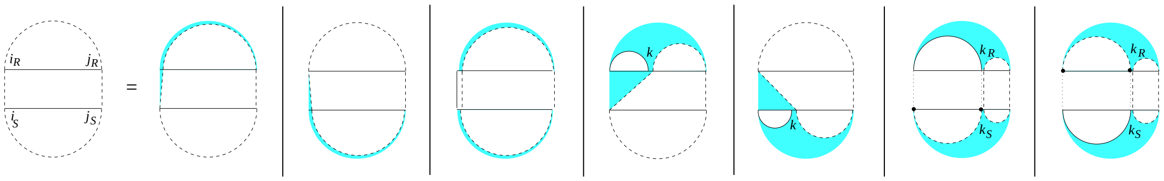


Figure 5.1: Recursion cases for computing the maximum base pairing joint structure of $[i_R, j_R]$ and $[i_S, j_S]$.

Time reduction by sparsification

We will apply a sparsification technique to reduce the number of cases necessary to be considered for Eq 5.1(d)-(g), as well as Eq 5.2(b).

Concerning sparsification, the simple cases are Eq 5.1(d),(e), and Eq 5.2(b), which correspond to the folding of a single sequence. The sparsification of these cases works in close analogy to the sparsification of RNA structure prediction as described by Wexler et al. [98]. We will briefly review their approach adapted to case Eq 5.2(b). Thereafter, we describe sparsification of the complex cases.

Sparsifying recursion cases for single structure folding The key to sparsification is a triangle inequality property of the DP matrix. In the case of $N^{\mathbf{X}}$, for every subsequence $[i, j]$ and $i < k \leq j$ the following inequality holds:

$$N^{\mathbf{X}}(i, j) \geq N^{\mathbf{X}}(i, k) + N^{\mathbf{X}}(k + 1, j).$$

Due to this property, it is sufficient to maximize in Eq. 5.2(b) for each i only over certain candidates k instead of all k with $i < k \leq j$. In this case, k is a candidate for i, iff $N^{\mathbf{X}}(i+1, k) < N^{\mathbf{X}}(i, k)$ and for all $i < k' < k$, $1+N^{\mathbf{X}}(i+1, k'-1)+N^{\mathbf{X}}(k'+1, k) < N^{\mathbf{X}}(i, k)$. Operationally, during the computation of $N^{\mathbf{X}}(i, k)$ we detect that k is a candidate for i by checking that the instance $1 + N^{\mathbf{X}}(i + 1, k - 1) + N^{\mathbf{X}}(k + 1, k)$ of recursion case Eq. 5.2(b) is the only maximal case.

For non-candidates k there exists some k', $i \leq k' < k$, where $N^{\mathbf{X}}(i, k) = N^{\mathbf{X}}(i, k') + N^{\mathbf{X}}(k' + 1, k)$. Then for all $j > k$, $N^{\mathbf{X}}(i, k) + N^{\mathbf{X}}(k + 1, j) = N^{\mathbf{X}}(i, k') + N^{\mathbf{X}}(k' + 1, k) + N^{\mathbf{X}}(k+1, j)$, and by triangle inequality $N^{\mathbf{X}}(i, k)+N^{\mathbf{X}}(k+1, j) \leq N^{\mathbf{X}}(i, k')+N^{\mathbf{X}}(k'+1, j)$. This means that, whenever a non-candidate k yields a maximal value, then there is already a $k' < k$ that yields the same value. Therefore k does not need to be considered, because the smallest such k' is taken into account.

Wexler et al. showed that sparsification reduces the expected time complexity of RNA folding by a linear factor, since the expected number of candidates for each i is constant. The transfer of sparsification to cases Eq 5.1(d) and (e) is straightforward, because only one subsequence is decomposed and the indices of the other subsequence remain fixed.

Sparsifying recursion cases for joint structure folding We extend the sparsification idea to the recursion cases Eq 5.1(f) and (g), which split both sequences and therefore minimize over a pair of split points (k_R, k_S). For the four dimensional matrix $N(i_R, j_R, i_S, j_S)$, the following generalization of the triangle inequality holds.

Observation 5.1.1 (Triangle inequality for $N(i_R, j_R, i_S, j_S)$) *For every subsequence $[i_R, j_R]$ and $[i_S, j_S]$ and for every $i_R < k_R \leq j_R$ and $i_S \leq k_S < j_S$, $N(i_R, j_R, i_S, j_S) \geq N(i_R, k_R, i_S, k_S)$ $+ N(k_R + 1, j_R, k_S + 1, j_S)$.*

Note that in principle both cases Eq 5.1(f) and (g) split the two subsequences at k_R and k_S, respectively, into the pairs $[i_R, k_R]$, $[i_S, k_S]$ and $[k_R + 1, j_R]$, $[k_S + 1, j_S]$. The only difference is that within the first pair of subsequences, $[i_R, k_R]$, $[i_S, k_S]$, case (f) assumes an arc $i_R \bullet k_R$ and case (g) assumes an arc $i_S \bullet k_S$. We consider only the case Eq 5.1(f), the case (g) is analogous.

Definition (Candidate for case Eq. 5.1(f)) For case Eq. 5.1(f), a pair (k_R, k_S) is a *candidate for* (i_R, i_S), iff i_R and k_R are complementary and for all $(k'_R, k'_S) \neq (k_R, k_S)$ with $i_R < k'_R \leq k_R$, $i_S < k'_S \leq k_S$,

$$\begin{aligned} & 1 + N(i_R + 1, k_R - 1, i_S, k_S) + N(k_R + 1, k_R, k_S + 1, k_S) \\ > & 1 + N(i_R + 1, k'_R - 1, k'_S, k_S) + N(k'_R + 1, k_R, k'_S + 1, k_S), \end{aligned}$$

With respect to the recursion case (f) a candidate (k_R, k_S) implies that the instance with $k_R = j_R$ and $k_S = j_S$ (i.e. $1 + N(i_R + 1, k_R - 1, i_S, k_S) + N(k_R + 1, k_R, k_S + 1, k_S)$) is the only maximal instance in the maximization of (f). Furthermore, it implies that none of the cases (a)-(e) in the computation of $N(i_R, k_R, i_S, k_S)$ yields a larger value than case (f).

Lemma 5.1.2 *For correctness of the recursion of Eq. 5.1, in the maximization of Eq. 5.1(f) it suffices to consider only the set of candidates given above.*

Proof For any non-candidate (k_R, k_S), there exists some (k'_R, k'_S) with $i_R - 1 \leq k'_R \leq k_R$, $i_S - 1 \leq k'_S \leq k_S$, $(k'_R, k'_S) \neq (k_R, k_S)$, $(k'_R, k'_S) \neq (i_R - 1, i_S - 1)$, and

$$1 + N(i_R + 1, k_R - 1, i_S, k_S) \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, k_R, k'_S + 1, k_S). \tag{5.3}$$

Note that $k'_R = i_R - 1$ or $k'_S = i_S - 1$ in Eq. 5.3 occurs when (k_R, k_S) is not a candidate due to one of the recursion cases (a)-(e).

Eq. 5.3 and the triangle inequality imply that for all $j_R > k_R$ and $j_S > k_S$

$$\begin{aligned} & 1 + N(i_R + 1, k_R - 1, k_S, j_S) + N(k_R + 1, j_R, k_S + 1, j_S) \\ & \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, k_R, k'_S + 1, k_S) + N(k_R + 1, j_R, k_S + 1, j_S) \\ & \leq N(i_R, k'_R, i_S, k'_S) + N(k'_R + 1, j_R, k'_S + 1, j_S). \end{aligned} \tag{5.4}$$

Non-candidates (k_R, k_S) for (i_R, i_S) do not need to be considered in the recursions of all $N(i_R, j_R, i_S, j_S)$, because there exists a recursion case splitting at (k'_R, k'_S) that yields the same or better score for $N(i_R, k_R, i_S, k_S)$. The equivalent case is considered in the recursion of $N(i_R, j_R, i_S, j_S)$ and, due to Eq. 5.4, yields a greater or equal score. □

Therefore the recursion case Eq. 5.1(f) can be updated such that the maximization runs only over the candidates for this case.

$$\max_{\substack{i_R < k_R \leq j_R \\ i_S < k_S \leq j_S \\ (k_R, k_S) \text{ candidate for } (i_R, i_S)}} \begin{pmatrix} 1 + N(i_R + 1, k_R - 1, i_S, k_S) \\ + N(k_R + 1, j_R, k_S + 1, j_S) \end{pmatrix} \tag{5.5}$$

Analogously, we define candidates for case Eq. 5.1(g). The candidate criterion for Eq. 5.1(g) is stricter than for Eq. 5.1(f), since we require that a candidate for Eq. 5.1(g) is better than all cases Eq. 5.1(a)-(e) and (f).

Expected number of candidates $\psi_1(n)$ denotes the expected number of candidates $k \leq n + i$ for some i in cases Eq. 5.1(d),(e), and Eq. 5.2(b). $\psi_2(n)$ is the expected number of candidates (k_R, k_S), $k_R \leq i_R + n$, $k_S \leq i_S + n$, for some (i_R, i_S) in cases Eq. 5.1(f) and (g).

Applying the described sparsification to all non-constant cases in recursions Eq. 5.1 and Eq. 5.2, yields the following.

Theorem 5.1.3 *$N(1, L_R, 1, L_S)$ can be computed in $O((\psi_1(n) + \psi_2(n))n^4)$ expected time, where $n = max(L_R, L_S)$.*

For a theoretical bound on $\psi_1(n)$ and $\psi_2(n)$, we assume the polymer-zeta property holds for each one of the RNA sequences that are involved in the interaction (with the other RNA sequence). The polymer-zeta property states that in any long polymer chain the probability of having arc between two monomers with distance m converges to $b.m^{-c}$, where $b, c > 0$ are some constants. For a polymer as a self-avoiding random walk on a square lattice, it has been known that $c > 1$ [34]. The exponent c for the denaturation transition of DNA in both 2D and 3D models is found to be larger than 2 [50]. Since RNA folds similar to other polymers, one can assume that RNA folding obeys the polymer-zeta property; i.e. the probability that a structure is formed over the subsequence of length m converges to $b.m^{-c}$, where $c > 1$. Although the property is not proven for RNA molecules, there is empirical evidence, as shown by Wexler et al. [98], that a version of polymer-zeta property holds for RNA molecules as well.

Lemma 5.1.4 *Assume that the two interacting RNAs independently satisfy the* polymer-zeta *property with* $c > 1$*, i.e. there exist constants* $b > 0$ *and* $c > 1$ *such that the probability for any internal base pair* $i \bullet (i+m)$ *is bounded by* $b \cdot m^{-c}$ *- even when two RNAs interact. Then* $\psi_1(n) = O(1)$ *and* $\psi_2(n) = O(n)$*.*

Proof $\psi_1(n) = O(1)$ follows from Wexler et al. [98]. For $\psi_2(n) = O(n)$, consider all candidates (k_R, k_S) for (i_R, i_S) and case Eq. 5.1(f). (Case Eq. 5.1(g) is symmetric.) Note that in Eq. 5.1(f), $i_R \bullet k_R$. For a fixed k_S analogously to Wexler et al. [98], the expected number of k_R with $i_R \bullet k_R$ is $b\sum_{i=1}^{n} i^{-c} < b\sum_{i=1}^{\infty} i^{-c}$ which converges to a constant for $c > 1$. Hence for each of the $O(n)$ possible values of k_S, k_R takes only a constant number of different values and hence on average we have $O(n)$ such candidates. □

Space efficient strategy

The space complexity of the algorithm can be reduced from $O(n^4)$ to $O(n^3 + \psi(n)n^2)$ as follows. The matrices $N^{\mathbf{R}}$ and $N^{\mathbf{S}}$ only require $O(n^2)$ space. All cases for the computation of an entry $N(i_R, j_R, i_S, j_S)$ only rely on entries $N(i'_R, j'_R, i'_S, j'_S)$ that satisfy one of the following two properties. (i) $j'_R \in \{j_R - 1, j_R\}$ and $j'_S \in \{j_S - 1, j_S\}$ or (ii) $N(i'_R, j'_R, i'_S, j'_S)$ corresponds to some candidate of the respective case, i.e. in case Eq. 5.1(d) $j'_R + 1$ is a candidate for $i'_R - 1 = i_R$, in case (e) $j'_S + 1$ is a candidate for $i'_S - 1 = i_S$, in case (f) (j'_R+1, j'_S) is a candidate for $(i'_R-1, i'_S) = (i_R, j_R)$, and in case (g) (j'_R, j'_S+1) is a candidate for $(i'_R, i'_S - 1) = (i_R, j_R)$. As shown in the following algorithm, all values that satisfy (i)

```
Algorithm: Space efficient evaluation of Eq. 5.1
precompute matrices $N^{\mathbf{R}}$ and $N^{\mathbf{S}}$ ;
initialize empty lists for candidates ;
for $j_R = 1..L_R$ do
    allocate and init matrix slice $N(\cdot, j_R, \cdot, \cdot)$ ;
    for $j_S = 1..L_S$, $i_R = j_R..1$, $i_S = j_S..1$ do
        compute $N(i_R, j_R, i_S, j_S)$ ;
        if $j_R$ is candidate for $i_R$ and Eq. 5.1(d) then
            store $N^{\mathbf{R}}(i_R + 1, j_R - 1, i_S, j_S)$ in list for $i_R$ and Eq. 5.1(d)
        else if $j_S$ is candidate for $i_S$ and Eq. 5.1(e) then
            store $N^{\mathbf{S}}(i_S + 1, j_S - 1)$ in list for $i_S$ and Eq. 5.1(e)
        else if candidate for Eq. 5.1(f) then
            store $N(i_R + 1, j_R - 1, i_S, j_S)$ in list for $(i_R, i_S)$ and Eq. 5.1(f)
        else if candidate for Eq. 5.1(g) then
            store $N(i_R, j_R, i_S + 1, j_S + 1)$ in list for $(i_R, i_S)$ and Eq. 5.1(g)
        end
    end
    free matrix slice $N(\cdot, j_R - 1, \cdot, \cdot)$ ;
end
```

can be stored in a three dimensional matrix and all values that satisfy (ii) can be stored in candidate lists of length $\psi(n)$ for each of the $O(n^2)$ instances of (i_R, i_S).

Note that, in the pseudocode, we maintain two three dimensional matrices, namely $N(\cdot, j_R, \cdot, \cdot)$ and $N(\cdot, j_R - 1, \cdot, \cdot)$ during the computation of the values for j_R. In practice, we save half of this memory, because any entry $N(\cdot, j_R - 1, \cdot, j_s)$ can be freed as soon as all $N(\cdot, j_R, \cdot, j_S)$ are computed.

Trace-Back We describe the recursive trace-back starting from a matrix entry (i_R, j_R, i_S, j_S). Computing the Trace-back involves some recomputation. First, the entire matrix slice $N(\cdot, j_R, \cdot, j_S)$ is recomputed unless it is already in memory. This requires access to only entries in the same matrix slice and candidates. Then, the best case in the recursion for $N(i_R, j_R, i_S, j_S)$ is identified. In cases (a)-(c), we recurse to the respective entry. In cases (d)-(g), which split in a first and second entry, we first recurse to the second one, which is in the same matrix slice. Then, we free the memory for the current matrix slice and recurse to the first entry, which will cause recomputation. Since each entry is recomputed at most once, the trace-back does not affect the asymptotic complexity.

5.1.2 Sparsification for Minimizing Free Energy

Alkan et al. [4] describe minimization of the free energy of RNA-RNA-interaction based on a simple stacked-pair energy model assuming there are no pseudoknot, crossing bond, and zigzag in the joint structure. Here we discuss an algorithm for RNA-RNA interaction free energy minimization on the same type of interactions based on the interaction energy model. We call our algorithm `Agile inteRNA`, as it is in fact the efficient version of `inteRNA` by Alkan et al. [4, 1]. Since the general recursive structure of this algorithm is identical to base pair maximization, our sparsification technique can be applied to reduce their time and space complexity in the same way. Compared to base pair maximization, these recursions distinguish several matrices representing differently scored substructures. Notably, they are formulated such that all cases that split an entry (i_R, j_R, i_S, j_S) at (k_R, k_S) are of the same form as cases Eq. 5.1(f) and (g) or k_R and k_S are bounded due to the loop length restriction of the energy model. Achieving the same space complexity requires one additional consideration. For assigning correct energy to internal loops formed by interaction arcs, an entry (i_R, j_R, i_S, j_S) can depend on (i'_R, j'_R, i_S, j_S), where j'_R is neither j_R nor $j_R - 1$. However, $j_R - j'_R$ is still bounded by the maximal loop length ℓ of the energy model, i.e. $j_R - j'_R < \ell$. Hence, it suffices to store ℓ matrix slices $(\cdot, j'_R, \cdot, \cdot)$ for $j_R - \ell < j'_R \leq j_R$.

Theorem 5.1.5 *The MFE interaction of two RNAs of maximal length n can be computed in expected time $O((\psi_1(n) + \psi_2(n))n^4)$ and expected space $O((\psi_1(n) + \psi_2(n))n^2 + n^3)$.*

5.1.3 Minimum free energy RNA-RNA interaction prediction

In this section we present our recursive algorithm for RNA-RNA interaction free energy minimization which is compatible to sparsification technique and is based on the interaction energy model. The minimum free energy joint structure $M(i_R, j_R, i_S, j_S)$ derived from one of the seven possible cases shown in Figure 5.3. The first two cases are when i_R or i_S is an unpaired base. In third case i_R interacts with i_S, this bond starts a special type of joint structure denoted by Ib and it is explained in Figure 5.4. The forth and fifth cases are when i_R or i_S is forming intramolecular base pairs. In other possible cases either $i_R \bullet k_R$ is an interaction arc subsuming $[i_S, k_S]$ or $i_S \bullet k_S$ is an interaction arc subsuming $[i_R, k_R]$. The DP algorithm for free energy minimization based on sparsification strategy, $M(i_R, j_R, i_S, j_S)$, is defined as follows:

$$M(i_R, j_R, i_S, j_S) = \max \begin{cases} M(i_R+1, j_R, i_S, j_S) & (a) \\ M(i_R, j_R, i_S+1, j_S) & (b) \\ M^{Ib}(i_R, j_R, i_S, j) & (c) \\ \max\limits_{\substack{i_R<k\le j_R \\ k \text{ candidate for } (i_R)}} \begin{pmatrix} M^{\mathbf{R}.b}(i_R, k) \\ \quad + M(k+1, j_R, i_S, j_S) \end{pmatrix} & (d) \\ \max\limits_{\substack{i_S\le k<j_S \\ k \text{ candidate for } (i_S)}} \begin{pmatrix} M^{\mathbf{S}.b}(i_S, k) \\ \quad + M(i_R, j_R, k+1, j_S) \end{pmatrix} & (e) \\ \max\limits_{\substack{i_R<k_R\le j_R \\ i_S<k_S\le j_S \\ (k_R,k_S) \text{ candidate for } (i_R, i_S)}} \begin{pmatrix} M^{Is}(i_R, k_R, i_S, k_S) \\ \quad + M(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (f) \\ \max\limits_{\substack{i_R<k_R\le j_R \\ i_S<k_S\le j_S \\ (k_R,k_S) \text{ candidate for } (i_R, i_S)}} \begin{pmatrix} M^{Is'}(i_R, k_R, i_S, k_S) \\ \quad + M(k_R+1, j_R, k_S+1, j_S) \end{pmatrix} & (g) \end{cases} \tag{5.6}$$

$$M^{\mathbf{X}}(i,j) = \max \begin{cases} M^{\mathbf{X}}(i+1, j) & (a) \\ \max\limits_{\substack{i<k\le j \\ (k) \text{ candidate for } (i)}} \begin{pmatrix} M^{\mathbf{X}.b}(i, k) \\ \quad + M^{\mathbf{X}}(k+1, j) \end{pmatrix} & (b) \end{cases} \tag{5.7}$$

Here $M^{Ib}(i_R, j_R, i_S, j_S)$ (Figure 5.4) is the minimum free energy for the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$ assuming $i_R \cdot j_S$ is an interaction bond, and $M^{Is}(i_R, j_R, i_S, j_S)$ (Figure 5.5) is the minimum free energy for the joint structure of $[i_R, j_R]$ and $[i_S, j_S]$ assuming $i_R \circ j_R$ is an interaction arc subsuming $[i_S, j_S]$. $M^{Is'}$ is symmetric to M^{Is} where $i_S \circ j_S$ is an interaction arc subsuming $[i_R, j_R]$. In Q^{Isl}, $[i_S, j_S]$ contains at least one interaction arc and in Q^{Isk}, $[i_S, j_S]$ contains at least one direct bond. The other auxiliary matrices are Q^{Ill}, Q^{Ilk}, Q^{Ikl}, and Q^{Ikk} (Figure 5.8).

- Q^{Ill} includes all cases where both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one interaction arc.
- Q^{Ilk} (symmetric to Ikl) includes all cases where $[i_R, j_R]$ has at least one interaction arc and $[i_S, j_S]$ has at least one direct bond.
- Q^{Ikk} includes all cases where both $[i_R, j_R]$ and $[i_S, j_S]$ have at least one direct bond.

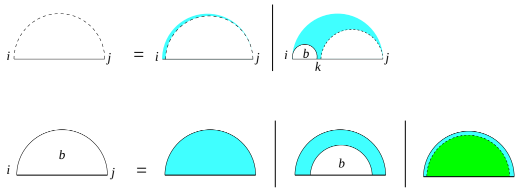


Figure 5.2: Recursion cases for MFE single structure.

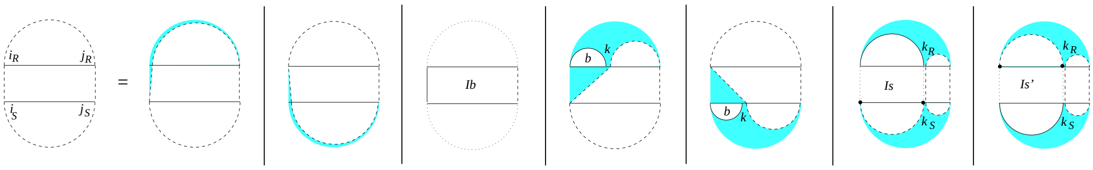


Figure 5.3: Recursion cases for MFE joint structure.

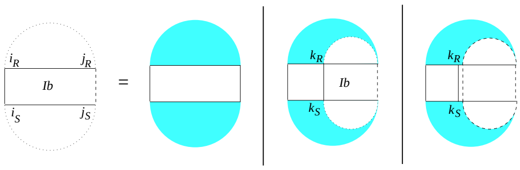


Figure 5.4: Recursion cases for MFE joint structure while $i_R \circ j_S$ is a bond. Here $i_R < k_R \leq \min i_R + \ell, j_R$ and $i_S < k_S \leq \min i_S + \ell, j_S$ w. ℓ is the maximal loop length.

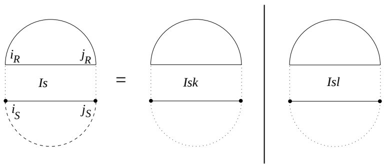


Figure 5.5: In recursive quantity Is, $i_R \bullet j_R$ is an interaction arc which subsumes interval $[i_S, j_S]$. The subsumed area contains at least one direct bond or at least one interaction arcs.

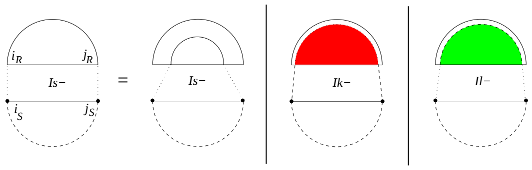


Figure 5.6: Recursion cases for Isl or Isk which extract the interaction arc $i_R \bullet j_R$.

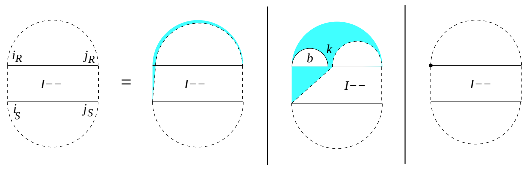


Figure 5.7: In Ikk, Ikl, Ilk, or Ill, if the terminal point i_R (or j_S) is not an end point of interaction bond or arc, some recursions should be applied to extract the internal structure.

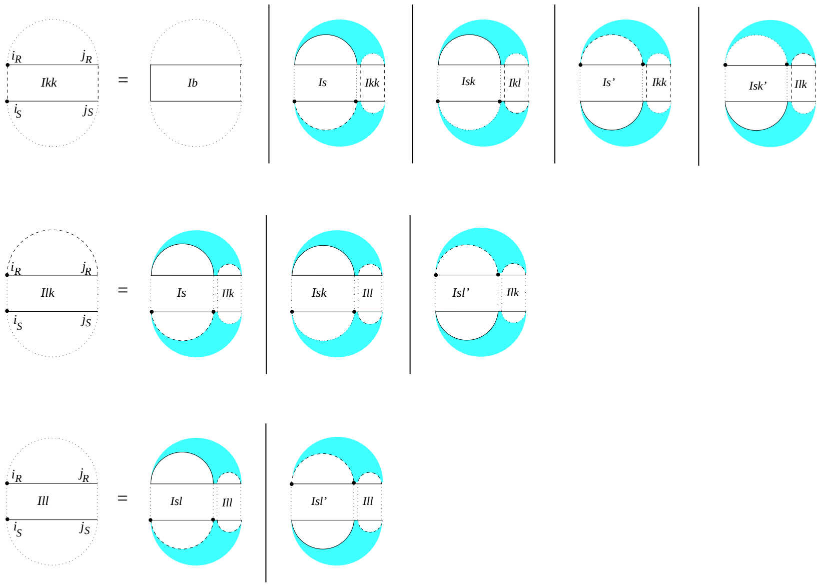


Figure 5.8: Recursion for joint structures that has direct interactions on both subsequences (Ikk), direct interaction on one subsequence and interaction arc on the other (Ikl and Ilk which are symmetric), and interaction arcs on both subsequences (Ill).

5.2 Experimental Results

For evaluating the effect of sparsification on RNA-RNA-interaction, we implemented three variants of the total free energy minimization algorithm for RNA-RNA-interaction prediction: the first variant does not perform any sparsification, the second employs sparsification for improving the time complexity, and the third improves both time and space complexity.

Below, we first evaluate the accuracy of the total free energy minimization algorithm for RNA-RNA interaction prediction. We present an assessment of sensitivity, positive prediction value, and F-measure of our method on the data set of Kato et al. [51] which involves five distinct RNA-RNA interactions. Note that sparsification does not affect the calculated free energy values (i.e. optimality of the calculated joint free energy of the interaction), the accuracy of the predicted interactions is identical to the original approaches for general RNA-RNA-interactions based on the same scoring scheme.

Later we demonstrate that sparsification leads to a significant reduction of the time and space requirements in practice. Then we study the relationship between the sequence length and the number of candidates per each base on a large set of confirmed RNA-RNA interactions and study the average time/space behavior of the algorithms.

5.2.1 Accuracy of total free energy minimization

In this section, we assess the performance of our total minimum free energy algorithm for RNA-RNA interaction prediction. For this purpose we used the 5 RNA-RNA complexes from Kato et al. [51] test set. We compared our results with two state-of-the-art methods for joint structure prediction: (1) the grammatical approach by Kato et al. [51] (denoted by EBM as energy-based model), and (2) the DP methods for two models presented by Alkan et al. [4] (denoted by SPM as stacked-pair model and LM as loop mode and implemented in [1]).

In order to estimate the accuracy of prediction, we measured the sensitivity and PPV defined as follows:

$$sensitivity = \frac{number\ of\ correctly\ predicted\ base\ pairs}{number\ of\ true\ base\ pairs}, \tag{5.8}$$

$$PPV = \frac{number\ of\ correctly\ predicted\ base\ pairs}{number\ of\ predicted\ base\ pairs}. \tag{5.9}$$

As another measure of accuracy we calculated F-measure which considers both sensitivity and PPV. F-measure is the harmonic mean of sensitivity and PPV, and its formula is as follows:

$$F = \frac{2 \times sensitivity \times PPV}{sensitivity + PPV}. \tag{5.10}$$

Table 5.1: Prediction accuracy of competitive RNA-RNA joint structure prediction methods.

RNA-RNA	Sensitivity				PPV			
interaction pairs	Agile inteRNA	EBM	SPM	LM	Agile inteRNA	EBM	SPM	LM
CopA-CopT	1.000	0.909	0.955	0.864	0.846	0.800	0.778	0.760
DIS-DIS	1.000	0.786	0.786	0.786	1.000	0.786	0.786	0.786
IncRNA$_{54}$-RepZ	0.875	0.917	0.875	0.875	0.792	0.830	0.778	0.778
R1inv-R2inv	1.000	0.900	1.000	1.000	1.000	0.947	1.000	1.000
Tar-Tar*	1.000	1.000	1.000	1.000	0.875	0.933	0.875	0.875
Average	0.975	0.902	0.923	0.905	0.902	0.859	0.843	0.840

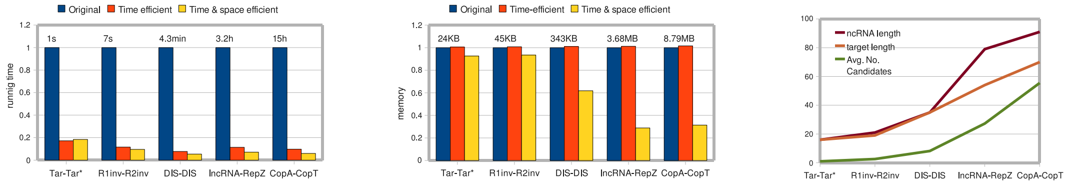
Table 5.2: F-measure values of competitive RNA-RNA joint structure prediction methods.

RNA-RNA	F-measure			
interaction pairs	Agile inteRNA	EBM	SPM	LM
CopA-CopT	0.917	0.851	0.857	0.809
DIS-DIS	1.000	0.786	0.786	0.786
IncRNA$_{54}$-RepZ	0.831	0.871	0.824	0.824
R1inv-R2inv	0.900	0.923	1.000	1.000
Tar-Tar*	0.933	0.965	0.933	0.933
Average	0.916	0.879	0.880	0.870

Tables 5.1 and 5.2 show comparison between the accuracy of our method and other competitors. As it can be seen, our method based on the three accuracy measures outperformed the competitors. For Tar-Tar* and R1inv-R2inv pairs that both RNAs are relatively short ($\sim$ 20nt), all methods were accurate enough. However, for DIS-DIS which is not still long (35nt), only our method was able to predict the interaction while the other approaches returned no interaction. CopA-CopT and IncRNA$_{54}$-RepZ are a bit longer ($\sim$ 60nt); CopA-CopT has two disjoint binding sites and IncRNA$_{54}$-RepZ has a continuous binding site. Our method outperformed the others in predicting the joint structure of CopA-CopT, while IncRNA$_{54}$-RepZ was predicted more accurately by EBM. We did not compare the running time between these methods due to the fact that each one uses different platform and hardware.

5.2.2 Time and space requirements of total free energy minimization

We applied the three variants of the MFE algorithm to five distinct RNA-RNA interactions reported by Kato et al. [51], which were used to assess the accuracy of available RNA-RNA interaction methods.



(a) Run-time improvement (b) Space improvement (c) Average No. candidates

Figure 5.9: Performance of three variants of the RNA-RNA interaction prediction algorithm via total free energy minimization, on a set of interactions compiled by Kato et al. [51]. All values for time and space usage are normalized by the usage of the non-sparsified algorithm, for which absolute time/space usage figures are also given.

Figure 5.9 shows (in absolute terms) time and space usage of the algorithms (with or without sparsification) on a Sun Fire X4600 server with 2.6 GHz processor speed. The results show that sparsification significantly improves the performance of the algorithms. In fact, Figure 5.9 demonstrates that as the RNA sequences in question get longer, the relative performance of the sparsified algorithms (with respect to the non-sparsified ones) improve. Although the pure time optimization causes a small space overhead due to maintaining the candidate lists, the time and space optimization not only improve the space utilization, as expected, but also results in further reduction in running time.

5.2.3 Number of Candidates

The time and space complexity of the (time and space) sparsified RNA-RNA-interaction prediction algorithm is linearly proportional to the (average) number of interaction partner candidates per base. Figure 5.9(c) shows how the average number of candidates (k_R, k_S) change as the lengths of the two RNA sequences increase. While the non-sparsified algorithms need to consider a quadratic number of split points (k_R, k_S), the number of candidates (and hence the number of split points) is much lower for the sparsified algorithms.

In order to observe the effects of sparsification on a much larger data set involving longer RNA sequences, we employ the algorithm for RNA-RNA interaction prediction which maximizes the number of (internal and external) base pairs. The data set we use for this purpose includes 43 pairs of ncRNAs and their known target mRNAs. This set not only includes (i) the data set of Kato et al. [51], but also (ii) a recently compiled test set of Busch et al. [20] consisting of 18 sRNA-target pairs, as well as (iii) all ncRNA-target interactions of E.coli from NPinter [100]. Among these interactions 32 are from E.coli, 8 are from Salmonella typhimurium and 3 are from HIV. Since the majority of the known ncRNAs bind to their target mRNAs in close proximity of the start codon, we extracted - as the target region - a subsequence comprising 300nt upstream and 50nt downstream of the first base of the start codon of each mRNA from GenBank [15]. As a result, the maximum sequence length is 227nt for ncRNAs and 350nt for target mRNAs.

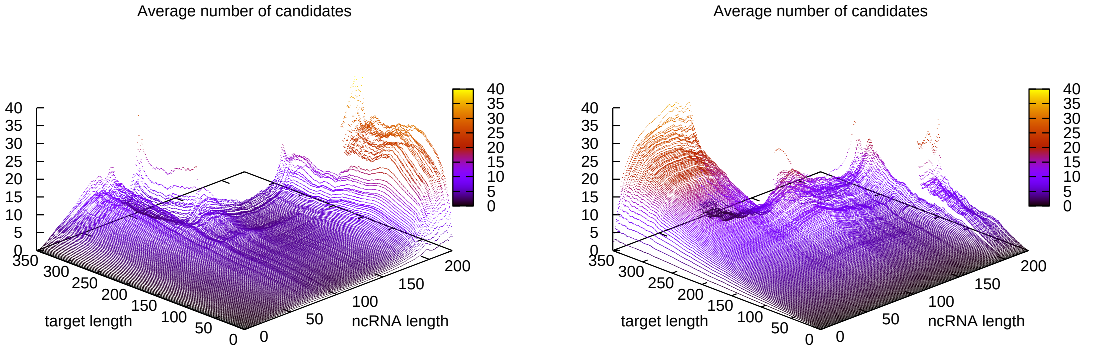
The experimental results on this larger data set confirm that the sparsification technique works for a single RNA folding via base pair maximization: the average number of candidates for those cases is low (roughly 5) as previously reported by Wexler et al. [98].

The recursion cases Eq. 5.1(f) and (g) split both RNAs simultaneously at points (k_R, k_S). Therefore they dominate the running time of the algorithm. For these cases, we counted the candidates that were considered during the computation of (the maximum number of base pairs of) each subsequence pair. The average number of candidates for different subsequence lengths, both for ncRNAs and mRNAs are depicted in Figure 5.10 - specific cases that correspond to Eq. 5.1(f) as well as Eq. 5.1(g) are provided separately. Note that the average number of candidates are generally low regardless of the sequence lengths: among all possible combinations of split points (k_R, k_S) (respectively in ncRNA and mRNA), even for the longest subsequences (e.g. ncRNA length $l_S = 252$ and mRNA length $l_R = 202$), no more than 40 pairs (of the possible 252 x 202 $= 50,904$ combinations for this example) are actual candidates on the average.[1]

5.2.4 Total number of fragments for different ncRNA and target subsequence lengths

The following graph shows the total number of fragments for different ncRNA and target subsequence lengths. The white region on top right of the plot in Figure 5.11 ($l_R > 111 \wedge l_S > 252$ and $l_R > 202 \wedge l_S > 153$) denotes the area that there are no fragments in our data set.

[1] Note that certain combinations of l_R and l_S there is no value for the number of candidates due to the fact that there is no data for $l_R > 111$ and $l_S > 252$ as well as $l_R > 202$ and $l_S > 153$.



(a) for case Eq. 5.1(f)

(b) for case Eq. 5.1(g)

Figure 5.10: Average number of candidates as a function of subsequence lengths.

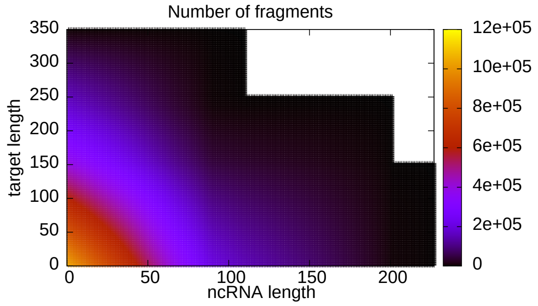


Figure 5.11: Total number of fragments for different ncRNA and target lengths.

Chapter 6

Fast Binding Sites Prediction

There are several evidences suggesting that RNA-RNA interaction is a multi step process [19, 66, 41] that involves: 1) unfolding of the two RNA structures to expose the bases needed for hybridization, 2) the hybridization at the binding site, and 3) restructuring of the complex to a new minimum free energy conformation. In this chapter we present a fast heuristic algorithm to predict interaction involving multiple binding sites based on the observation that the independent secondary structure of an RNA molecule is mostly preserved even after it forms a joint structure with another RNA. The above observation has been used by different methods for target prediction (see chapter 2 for an overview). However, most of these methods focus on predicting interactions involving only a single binding site, and are not able to predict interactions involving multiple binding sites. In contrast, our heuristic approach can predict interactions involving multiple binding sites by: (1) identifying the collection of accessible regions for both input RNA sequences, (2) using a matching algorithm, computing a set of ”non-conflicting” interactions between the accessible regions which have the highest overall probability of occurrence.

Note that an accessible region is a subsequence in an RNA sequence which, with ”high” probability, remain unpaired in its secondary structure. Our method considers the possibility of interactions being formed between one such accessible region from an RNA sequence with more than one such region from the other RNA sequence. Thus, in step (1), it extends the algorithm by Mückstein et al. for computing the probability of a specific region being unpaired [68] to compute the joint probability of two (or more) regions remaining unpaired. Because an accessible region from an RNA typically interacts with no more than two accessible regions from the other RNA, we focus on calculating the probability of at most two regions remaining unpaired: within a given RNA sequence of length n, our method can calculate the probability of any pair of regions of length $\leq w$ each, in $O(n^4w)$ time and $O(n^2)$ space. In step (2), on two input RNA sequences of length n and m ($n \leq m$), our method computes the most probable non-conflicting matching of accessible regions in $O(n^2w^4 + n^3/w^3)$ time and $O(w^4 + n^2/w^2)$ space.

6.1 The Heuristic Algorithm: inRNAs

Our heuristic algorithm for RNA-RNA interaction prediction problem is based on the idea that the external interactions mostly occur between pairs of unpaired regions of single structures. We aim to predict interactions of multiple binding sites as long as they have no crossing. The heuristic algorithm contains the following steps:

- Predict the highly accessible regions in each strands. These regions include the loop regions in native structure of RNA strand. In order to predict accessible regions we chose all the regions which remain unpaired with high probability.
- Predict the optimal non-conflicting interactions between the accessible regions. For every pair of accessible regions of two interacting RNAs a cost of interaction is calculated. Then a matching algorithm runs to find the minimum cost non-conflicting subset of interactions.

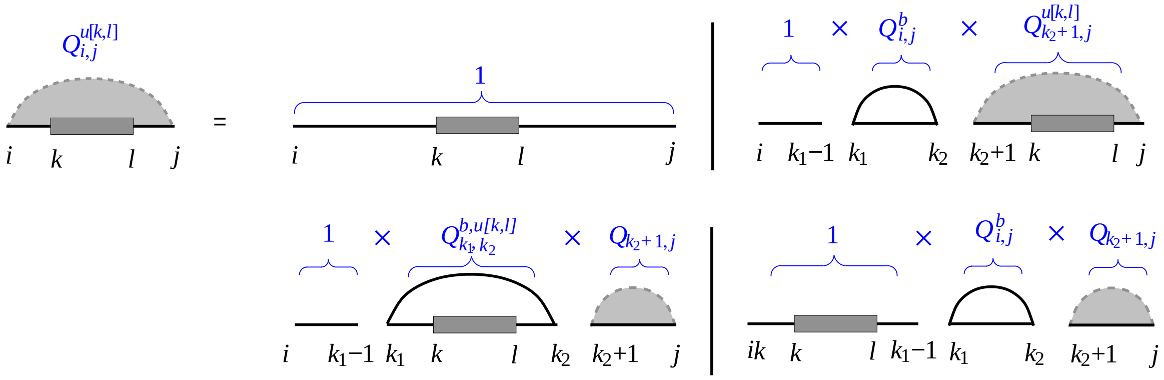
6.1.1 Accessible Regions

For a single RNA sequence an accessible region is a subsequence that remains unpaired in equilibrium state with high probability. The probability of an unpaired region can be calculated based on the algorithm presented in [68]. Here, we are interested in multiple unpaired regions. For this purpose one should compute the joint probabilities for any subset of possible intervals. Since the computation of all joint probabilities needs substantial time and space, here we only consider the joint probability of two unpaired subsequences.

Denoting the set of secondary structures in which the sequence interval $[k, l]$ remains unpaired by $S^{u[k,l]}$, the corresponding partition function is

$$Q^{u[k,l]}(T) = \sum_{s \in S^{u[k,l]}} e^{-G_s/RT}, \tag{6.1}$$

where R is the universal gas constant and T is the temperature. In order to compute the $Q^{u[k,l]}$, the standard recursion for the partition function folding algorithm [65] can be extended as:


$Q_{i,j}^{u[k,l]}$
i k l j
=
1
i k l j
1 × $Q_{i,j}^{b}$ × $Q_{k_2+1,j}^{u[k,l]}$
i k_1−1 k_1 k_2 k_2+1 k l j
1 × $Q_{k_1,k_2}^{b,u[k,l]}$ × $Q_{k_2+1,j}$
i k_1−1 k_1 k l k_2 k_2+1 j
1 × $Q_{i,j}^{b}$ × $Q_{k_2+1,j}$
ik k l k_1−1 k_1 k_2 k_2+1 j

where $i \leq k \leq l \leq j$ and $k_1 \cdot k_2$ is the leftmost base pair. Partition functions $Q_{i,j}^{b,u[k,l]}$ (where $i \cdot j$) and $Q_{i,j}^{m,u[k,l]}$ (where $[i,j]$ is inside a multiloop and constitutes at least one base pair) while the interval $[k,l]$ remains unpaired are derived from the standard algorithm in a similar way. Furthermore, probability of a base pair $p \cdot q$ while $[k,l]$ remains unpaired, $\mathbb{P}(p \cdot q | u[k,l])$, can be calculated by applying the McCaskill algorithm [65] for computing the base pair probability on $Q^{u[k,l]}$. It is easy to see that the desired partition function $Q^{u[k,l]}$ and base pair probability $\mathbb{P}(p \cdot q | u[k,l])$ are computed in same time and space complexity as the standard algorithm by McCaskill ($O(n^3)$ and $O(n^2)$ respectively).

Mückstein et al. [68] introduce an algorithm to compute the probability of unpaired region $\mathbb{P}(u[i,j])$ for a given sequence interval $[i,j]$. Here, we extend the specified algorithm to compute $\mathbb{P}(u[i,j]|u[k,l])$ which is the probability of unpaired region $[i,j]$ while $[k,l]$ remains unpaired. Clearly if some part of $[i,j]$ is within the interval $[k,l]$, the corresponding probability for that part is equal to one. Hence, for computing the probability only parts of $[i,j]$ which are exterior to $[k,l]$ should be considered. Here, without loss of generality we assume $k \leq l < i \leq j$.

For unpaired interval $[i,j]$ there are two general cases: either it is not closed by any base pair, or it is part of a loop. Fiq. 6.1 summarizes the cases of unpaired interval $[i,j]$ as a part of the loop enclosed by base pair $p \cdot q$ while interval $[k,l]$ remains unpaired. In case x interval $[p,q]$ does not contain interval $[k,l]$, and in the other cases $(a-e)$ interval $[k,l]$ lies

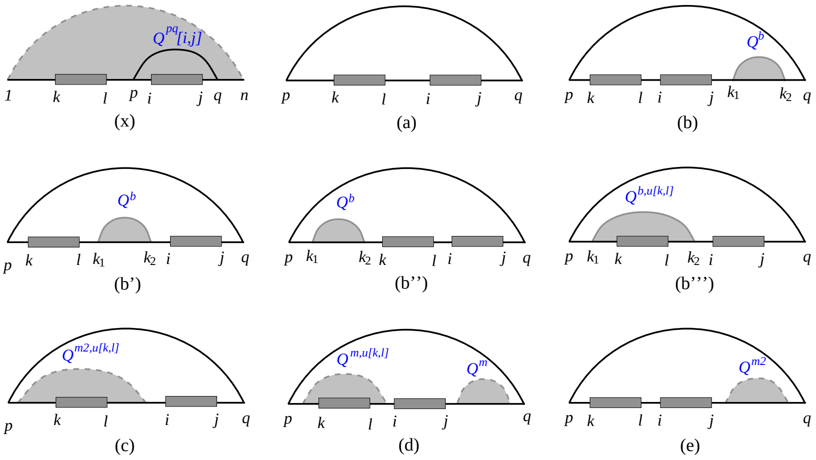


Figure 6.1: Cases of unpaired interval $[i, j]$ within a loop enclosed by $p \cdot q$ while $[k, l]$ remains unpaired.

in interval $[p, q]$. Probability $\mathbb{P}(u[i, j]|u[k, l])$ can be calculated as follows:

$$\begin{aligned}
\mathbb{P}(u[i,j]|u[k,l]) =& \frac{Q^{u[k,l]}_{1,i-1} \times 1 \times Q_{j+1,n}}{Q^{u[k,l]}} \\
&+ \sum_{l<p<i\leq j<q} \mathbb{P}(p \cdot q|u[k,l]) \times \frac{Q^{pq}_{i,j}}{Q^b_{p,q}} \qquad (x) \\
&+ \sum_{p<k\leq l<i\leq j<q} \mathbb{P}(p \cdot q|u[k,l]) \times \frac{Q^{pq,u[k,l]}[i,j]}{Q^{b,u[k,l]}_{p,q}} \qquad (a-e)
\end{aligned} \tag{6.2}$$

$Q^{pq}[i, j]$ which is introduced by Mückstein et al., counts all structures on $[p, q]$ that $[i, j]$ is part of the loop closed by base pair $p \cdot q$. The quantity $Q^{pq,u[k,l]}[i, j]$ is a variant of $Q^{pq}[i, j]$ while $[k, l]$ lies in $[p, q]$. Recursion of $Q^{pq,u[k,l]}[i, j]$ on cases $(a - e)$ displayed in Figure 6.1, is based on different types of loop and position of $[k, l]$. Therefore, we have

$$\begin{aligned} Q^{pq,u[k,l]}[i,j] =& e^{-G^{\text{hairpin}}_{p,q}/RT} && (a) \\ &+ \sum_{\substack{j<k_1<k_2<q| \\ l<k_1<k_2<i|p<k_1<k_2<k}} e^{-G^{\text{interior}}_{i,k_1,k_2,j}/RT} Q^{b}_{k_1,k_2} && (b, b', b'') \\ &+ \sum_{i<k_1<k\leq l<k_2<i} e^{-G^{\text{interior}}_{i,k_1,k_2,j}/RT} Q^{b,u[k,l]}_{k_1,k_2} && (b''') \\ &+ Q^{m2,u[k,l]}_{p+1,i-1} \; e^{-(a+b+c(q-i))/RT} && (c) \\ &+ Q^{m,u[k,l]}_{p+1,i-1} Q^{m}_{j+1,q-1} \; e^{-(a+b+c(j-i-1))/RT} && (d) \\ &+ Q^{m2}_{j+1,q-1} \; e^{-(a+b+c(j-p))/RT} && (e) \end{aligned} \tag{6.3}$$

where Q^{m2} is the partition function of a subsequence inside a multiloop that constitutes at least two base pairs. Q^{m2} which is introduced in Mückstein et al. algorithm can be extended to calculate $Q^{m2,u[k,l]}$. Therefore, the joint probability of two unpaired regions is obtained using

$$\mathbb{P}(u[i,j], u[k,l]) = \mathbb{P}(u[i,j] \mid u[k,l]) \times \mathbb{P}(u[k,l]). \tag{6.4}$$

Mückstein et al. algorithm requires $O(n^3)$ running time and $O(n^2)$ space complexity to compute the probability of unpaired region $\mathbb{P}(u[i,j])$ for every possible interval $[i,j]$ assuming the interval length is limited to size w. Using the the extended algorithm, given sequence interval $[k,l]$ computing $\mathbb{P}(u[i,j], u[k,l])$ for every possible interval $[i,j]$ requires the same time and space complexity. Note that for each interval $[k,l]$, $Q^{u[k,l]}$ should be computed separately. Since there are $O(nw)$ different intervals for a limited interval length w, with $O(n^4w)$ running time and $O(n^2)$ space complexity we are able to compute the joint probabilities for all pairs of unpaired regions. The same idea can be used to compute the joint probability of multiple unpaired regions. However, considering each extra interval increases the running time by a factor of $O(nw)$.

Note that a simplified version of our algorithm which ignores the joint probability of accessibility can be run in $O(n^3)$ time complexity. Moreover, for genome scale studies one can apply the algorithm by Bernhart er al. [16] for computing local base pairing probabilities in $O(nw^3)$ time complexity.

6.1.2 Interaction Matching Algorithm

We are given two lists of non-overlapping accessible regions $T_{\mathbf{R}} = \{r_1, r_2, ..., r_{n'}\}$ and $T_{\mathbf{S}} = \{s_1, s_2, ..., s_{m'}\}$ sorted according to their orders in interacting sequences **R** and **S**. We aim to calculate the optimal set of interaction bonds between the accessible regions under the following constraints: (1) Each accessible region can interact with at most two accessible regions from the other sequence. (2) There is no crossing interaction.

Let $Q^I_{r_i,s_j}$ be the partition function of all possible joint structures of two interacting sequence r_i and s_j, which can be calculated based on our interaction partition function algorithm. Define $I[r_i, s_j] = Q^I_{r_i,s_j} - Q_{r_i}Q_{s_j}$ as the partition function for the set of joint structures that contain some interactions between r_i and s_j . Two accessible regions r_i and s_j are considered to be able to interact if and only if $\mathbb{P}(I[r_i, s_j]) > 1/2$. The cost of interaction between two accessible regions r_i and s_j, $C(r_i, s_j)$, is the sum of the following terms:

- $E_u(r_i)$ and $E_u(s_j)$: the energy difference between the complete ensemble and the ensemble in which the interacting subsequences are left unpaired for both accessible regions. We have $E_u(r_i) = (-RT)(\ln(Q^{u[r_i]}_{\mathbf{R}}) - \ln(Q_{\mathbf{R}})) = (-RT)\ln(\mathbb{P}(u[r_i]))$. Similar equation can be used to calculate $E_u(s_j)$.
- $E_I(r_i, s_j)$: the ensemble energy of interacting joint structure for the two accessible regions where $E_I(r_i, s_j) = (-RT)\ln(\mathbb{P}(I[r_i, s_j]))$.

Cost of interaction between an accessible region r_i and two other accessible regions s_k and s_j is defined as $C(r_i, s_ks_j) = E_u(r_i) + E_u(s_k, s_j) + E_I(r_i, s_ks_j)$, where s_ks_j is the concatenation of two subsequences, and $E_u(s_k, s_j) = (-RT)\ln(\mathbb{P}(u[s_k], u[s_j]))$. Similarly the cost of interaction between two accessible regions from **R** and one accessible region from **S** is defined.

As an option, one can use minimum free energy (MFE) instead of ensemble energy (E_I) to define the cost of interaction. Accessible regions r_i and s_j are considered to be able to interact if and only if $MFE(r_i, s_j) < MFE(r_i) + MFE(s_j)$, i.e. there are some interaction bonds in the minimum free energy joint structure. Therefore, we have $C(r_i, s_j) = E_u(r_i) + E_u(s_j) + MFE(r_i, s_j)$. The cost of interaction of an accessible region r_i with two other accessible regions s_k and s_j is defined as $C(r_i, s_ks_j) = E_u(r_i) + E_u(s_k, s_j) + MFE(r_i, s_ks_j)$.

With $H(i, j)$, we denote the minimum cost non-conflicting set of interactions between the accessible regions $\{r_1, ..., r_i\}$ and $\{s_j, ..., s_{m'}\}$. The following dynamic programming computes $H(i,j)$:

$$H(i,j) = min \left\{ \begin{array}{ll} H(i-1, j+1) + C(r_i, s_j) & (i) \\ \min_{j<k\leq m'}\{H(i-1,k+1) + C(r_i, s_k s_j)\} & (ii) \\ \min_{1\leq k<i}\{H(k-1,j+1) + C(r_k r_i, s_j)\} & (iii) \\ H(i-1,j) & (iv) \\ H(i,j+1) & (v) \\ \infty & (vi) \end{array} \right\} \quad (6.5)$$

where $1 \leq i \leq n'$ and $1 \leq j \leq m'$. The algorithm starts by calculating $H(1, m')$ and explores all $H(i,j)$ by increasing i and decreasing j until $i = n'$ and $j = 1$. The DP algorithm has $O(n'^2 m' + n' m'^2)$ time and $O(n'm')$ space requirements. Also we need $O(n'm'w^6)$ time and $O(w^4)$ space to compute the cost of interaction for every pair of accessible regions. Assuming $n' \geq m'$ and $n' \leq n/w$, we can conclude that this step of the algorithm requires $O(n^2w^4 + n^3/w^3)$ time and $O(w^4 + n^2/w^2)$ space.

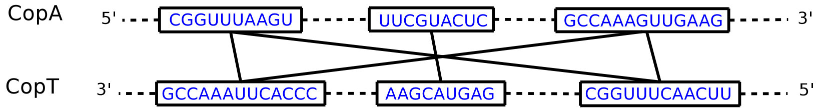


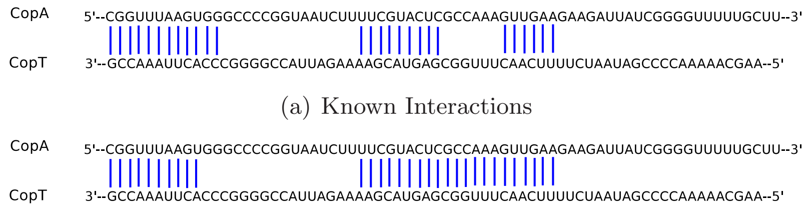
Figure 6.2: Interaction between accessible regions of CopA-CopT: a simple example for interaction matching algorithm.

CopA-CopT is a well known antisense RNA-target complex observed in E.coli [95]. The joint structure of CopA-CopT contains two disjoint binding sites. Figure 6.2 shows the identified accessible regions in CopA and CopT. Two regions connected by an edge are able to interact. Figure 6.3 shows the known and predicted interaction bonds between CopA and CopT. Note that internal bonds of both RNAs are not displayed in this figure.

6.2 Experimental Results

6.2.1 Dataset

In our experiments we used a dataset of 23 known RNA-RNA interactions which includes two recently used test sets. Table 6.1 contains the list of these RNA pairs. The first 18 sRNA-target pairs are compiled and used as test set by `IntaRNA` [20]. Next 5 pairs of RNAs


CopA
5'--CGGUUUAAGUGGGCCCCGGUAAUCUUUUCGUACUCGCCAAAGUUGAAGAAGAUUAUCGGGGUUUUUGCUU--3'
CopT
3'--GCCAAAUUCACCCGGGGCCAUUAGAAAAGCAUGAGCGGUUUCAACUUUUCUAAUAGCCCCAAAAACGAA--5'
CopA
5'--CGGUUUAAGUGGGCCCCGGUAAUCUUUUCGUACUCGCCAAAGUUGAAGAAGAUUAUCGGGGUUUUUGCUU--3'
CopT
3'--GCCAAAUUCACCCGGGGCCAUUAGAAAAGCAUGAGCGGUUUCAACUUUUCUAAUAGCCCCAAAAACGAA--5'

(a) Known Interactions

(b) Predicted Interactions

Figure 6.3: Interaction between CopA and CopT. (a) Natural interactions. (b) Predicted interactions.

which are known to have loop-loop interactions have been used by Kato et al. [51] to evaluate the proposed grammatical parsing approach for RNA-RNA joint structure prediction.

6.2.2 Binding Sites Prediction

For assessing the predictive power of our algorithm, we compared our algorithm with `IntaRNA` [20] and `RNAup` [69]. Based on the experimental results presented by `IntaRNA`, both `IntaRNA` and `RNAup` which incorporate accessibility of target regions, performed better than the other (TargetRNA, RNAHybrid, and RNAplex) competitive programs.

The results of these two programs for the first 18 RNA pairs are as presented in Table 1. in [20]. For the next 5 RNA pairs, we run `IntaRNA` v1.2 with its default settings and `RNAup` (from Vienna package v1.8.4) with the same setting that has been used in the experiments of Table 1. in [20]. `RNAup` has been run using parameter -b which considers the probability of unpaired regions in both RNAs and the maximal length of interaction to 80. In order to estimate the accuracy of programs, we measured the sensitivity, PPV and F-measure for intermolecular base pairs.

Table 6.1 shows the performance results of our program `inRNAs`, `IntaRNA` and `RNAup`. On average our method achieved 85% accuracy while `IntaRNA` and `RNAup` showed respectively 79% and 76% accuracy. This results demonstrate that our method predict RNA-RNA binding sites more accurately in compare to the competitive methods. In this dataset OxyS-fhlA and CopA-CopT are the only ones that have two disjoint binding sites where our method outperformed `IntaRNA` and `RNAup` by up to 30% improvement in F-measure. Both `RNAup` and `IntaRNA` could not predict any correct bond for GcvB-gltI pair, because they missed the binding site. However, `IntaRNA` could get 80% accuracy by considering the suboptimal prediction which is close to the accuracy that we have achieved for this case.

Table 6.1: Prediction accuracy of competitive RNA-RNA binding site prediction methods.

RNA-RNA	Sensitivity			PPV			F-measure		
interaction pairs	inRNAs	IntaRNA	RNAup	inRNAs	IntaRNA	RNAup	inRNAs	IntaRNA	RNAup
DsrA-RpoS	0.808	0.808	0.808	0.778	0.778	0.778	0.793	0.793	0.793
GcvB-argT	0.950	0.950	0.900	0.864	0.950	0.947	0.905	0.950	0.923
GcvB-dppA	1.000	1.000	1.000	0.850	0.586	0.459	0.919	0.739	0.629
GcvB-gltI	0.750	0.000	0.000	0.500	0.000	0.000	0.600	0.000	0.000
GcvB-livJ	0.634	0.955	0.955	0.824	0.955	0.955	0.717	0.955	0.955
GcvB-livK	0.540	0.542	0.542	0.570	0.565	0.565	0.555	0.553	0.553
GcvB-oppA	1.000	1.000	1.000	0.733	0.957	0.957	0.846	0.978	0.978
GcvB-STM4351	0.760	0.760	0.880	1.000	0.905	0.957	0.864	0.826	0.917
IstR-tisAB	0.722	0.879	0.667	1.000	0.960	1.000	0.839	0.918	0.800
MicA-ompA	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
MicA-lamB	1.000	1.000	0.826	1.000	0.821	0.704	1.000	0.902	0.760
MicC-ompC	1.000	1.000	0.727	1.000	0.537	0.410	1.000	0.699	0.524
MicF-ompF	0.960	0.960	0.800	0.960	0.960	0.952	0.960	0.960	0.869
OxyS-fhlA	0.813	0.500	0.375	1.000	1.000	1.000	0.897	0.667	0.545
RyhB-sdhD	0.618	0.588	0.794	0.955	1.000	0.794	0.750	0.741	0.794
RyhB-sodB	1.000	1.000	1.000	1.000	0.818	0.900	1.000	0.900	0.947
SgrS-ptsG	0.566	0.739	0.739	0.765	1.000	1.000	0.651	0.850	0.850
Spot42-galK	0.432	0.409	0.523	0.760	0.643	0.523	0.551	0.500	0.523
CopA-CopT	0.889	1.000	0.556	0.828	0.391	0.652	0.857	0.562	0.600
DIS-DIS	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
IncRNA$_{54}$-RepZ	1.000	0.738	0.750	0.889	0.850	0.857	0.941	0.790	0.800
R1inv-R2inv	1.000	1.000	1.000	0.778	1.000	0.778	0.875	1.000	0.875
Tar-Tar*	1.000	1.000	1.000	0.833	0.833	0.833	0.909	0.909	0.909
Average	0.845	0.819	0.776	0.865	0.805	0.784	0.845	0.791	0.763

Chapter 7

Conclusion and Discussion

The gene therapy applications of RNA interference have been approved by several studies on drug discovery and biomedical research. Many pharmaceutical companies invest in RNA interference technology to provide a new class of drugs. RNA interference technology is expected to bring a revolution in medical treatment. Currently there is no high throughput approach for experimental target prediction. Consequently, computational methods are in high demand for target prediction problems.

This dissertation presents a computational study of the RNA-RNA interaction prediction problem. Our proposed approaches satisfy the main requirements for a method to be useful for genome wide screening problems.

We first present an interaction energy model which is an extension of the standard thermodynamic energy model for an RNA secondary structure. Our model introduces three new components: (i) the hybrid component, (ii) the kissing loop, and (iii) the inter-hybrid loop. The corresponding energy functions of these components are extended versions of hybridization, multi loop, and pseudoknot energy functions. Our model considers almost all physically possible secondary structures that do not contain pseudoknots, crossing bonds, and zigzags. Therefore the interaction energy model can calculate the energy contribution of complex types of interactions including loop-loop interaction and multiple binding sites

Then, we develop an efficient algorithm to compute the partition function of two interacting nucleic acid strands. Using our partition function algorithm, we can compute the probability of interaction as well as several thermodynamic values such as equilibrium concentration, melting temperature, heat capacity, and UV absorption. We verified our algorithm by computing the melting temperature for RNA pairs available in the literature and the equilibrium concentration for the OxyS-fhlA complex. In both experiments our algorithm provides high accuracy. A parallel work by Huang et al. [48], published just a few months after our paper, solves the interaction partition function problem by a dynamic programming algorithm based on a different set of recursion cases. This approach has the same complexity and performance as our method.

We also consider the problem of predicting the joint structure of two interacting RNAs via minimizing their total free energy as a tool for detecting/verifying mRNA targets of regulatory ncRNAs. Earlier approaches to the problem either use a restricted interaction model, not covering many known joint structures, or require significant computational resources for many practical instances. We show that sparsification, a technique that has been applied to single RNA folding prediction, can be applied to the problem of RNA-RNA interaction prediction to improve significantly both the running time and the space utilization of the DP algorithm. In fact, by employing a version of the polymer-zeta property for interacting RNA-structures (a property generally assumed to be held by many polymers and has been empirically shown for single RNAs), we show how to reduce the running time and space of the RNA-RNA interaction prediction problem, from $O(n^6)$ time and $O(n^4)$ space to $O(n^4\psi(n))$ time and $O(n^2\psi(n)+n^3)$ space, for a function $\psi(n) = O(n)$ on average. These theoretical improvements are verified by our experiments; as a result it is now possible to employ computational prediction of RNA-RNA interactions to a much wider range of potential regulatory ncRNAs and their targets.

Finally, we introduce a fast heuristic algorithm for RNA-RNA interaction prediction. Our heuristic algorithm for the RNA-RNA interaction prediction problem incorporates the accessibility of unpaired regions and a matching algorithm to compute the optimal set of interactions between the multiple accessible regions. The algorithm has an $O(n^4w + w^6)$ running time and $O(n^2 + w^4)$ space complexity where w is the size of the sequence window in ncRNA and target mRNA. Note that a simplified version of our algorithm which ignores the joint probability of accessibility can be run in $O(n^3w + w^6)$ time complexity. Moreover, for genome scale studies where only the local base pairing probabilities are considered, time complexity is $O(nw^3 + w^6)$. The main advantage of our method is its ability to predict multiple binding sites which have so far only been predictable by expensive algorithms. On a set of known RNA-RNA complexes, our proposed algorithm shows great accuracy, particularly for interaction complexes with multiple binding sites.

7.1 Future Work

The RNA-RNA interaction problem is a relatively new one with many unresolved biological aspects. Our information about the gene regulation mechanism and possible involving components is still very limited. With progress in biological studies as more details of the interaction process are revealed, computational approaches for target prediction can be improved, resulting in more accurate predictions.

Currently, our method considers several folding factors including temperature, PH level, salt and magnesium concentrations, as well as the initial concentrations of two interacting strands. For simplicity we ignore the effect of RNA-induced silencing complex (RISC) and some RNA chaperones such as Hfq known to be involved in the interaction process. One direction for further improvement of interaction prediction problem is to study the effect of more involving components.

The proposed interaction energy model needs experimental validation. The parameters are learned over a limited set of short interaction samples. Perhaps a rich set of training data increases the accuracy of the model. The correctness and accuracy of the model should be approved by extensive laboratory experiments.

Although applying sparsification technique to the minimum free energy RNA-RNA interaction prediction problem results in a significant improvement of time and space complexity, we cannot arrive at a similar result for the interaction partition function problem. The interaction partition function algorithm presented in this thesis has $O(n^6)$ time and $O(n^4)$ space complexity. Recently, Tsur et al. [91] proved that by applying the Valiant algorithm, we can improve the running time of the interaction partition function problem into $O(n^6.\frac{\log^3 \log n}{\log^2 n})$. This time improvement, independent of its theoretical value, is not practically significant, and since the algorithm is complicated it is not worth it to be implemented. In addition, this method does not reduce the space complexity of the interaction prediction problem. Therefore, improving the time and space complexity of our algorithm is a high priority.

Our heuristic algorithm in Chapter 6, as well as its competitors [20, 69], is based on the assumption that interaction usually happens between the accessible regions of two RNA structures. In fact, this method models the interaction as a double-step process. However, as mentioned in Chapter 2, there are several well known RNA-RNA complexes [19] involved in multi-step pathways of interaction. Based on the studied cases, interactions are initiated by one or two loop-loop interactions. Then the structure is reformed into a more stable one by interacting along a topologically feasible pathway. Perhaps the multi-step pathway of interaction can be modeled by a kinetic study.

The current interaction prediction methods consider the two-dimensional model of interaction. Although modeling the interaction problem in three-dimensional space is too complex and impractical at the present time, incorporating a few of topological constraints into the two-dimensional models could be useful.

Appendix A

Data Sets Used in Chapter 4

For verification of our algorithm in predicting the melting temperature, we used three data sets available in the literature. The RNA pairs from the first data set which were originally reported in Table 3 of [101] has been presented in the paper. Here we present the RNA sequence pairs from the second (originally reported in Table 1 of [27]) and third (originally reported in Tables 3 and 4 of [64]) data sets that have been used in Tables 2 and 3 of the paper.

Table A.1: Sequences of the set of RNA pairs reported in Table 2 of the paper.

Pairs	Sequences
A	GGAGCGGCUUCGGCCGGACG/CGUCaaCUCC
B	GGAGaCGGCUUCGGCCGGACG/CGUCauaCUCC
C	GGAGaCGGCUUCGGCCGGCAG/CUGCauaCUCC
D	GGAGgCGGCUUCGGCCGuGACG/CGUCcauaCUCC
E	GGAGaCGGCUUCGGCCGcGACG/CGUCauaCUCC
F	GGAGgCGGCUUCGGCCGuGACG/CGUCauaCUCC
G	GGAGCGGCUUCGGCCGGACG/CGUCCUCC
H	GGAGaCGGCUUCGGCCGGACG/CGUCcauaCUCC
I	GGAGCGGCUUCGGCCGGACG/CGUCauaCUCC
J	GGAGCGGCUUCGGCCGGACG/CGUCcauaCUCC
K	GGAGaCGGCUUCGGCCGcGACG/CGUCcauaCUCC
L	GGAGaCGGCUUCGGCCGaGACG/CGUCcauaCUCC

Table A.2: Sequences of the set of RNA pairs reported in Table 3 of the paper.

Pairs	Sequences
G-GC-G/C-C	GGCAGGCGCUUCGGCGCGGAGG/CCUCCCUGCC
G-GC-G/CaC	GGCAGGCGCUUCGGCGCGGAGG/CCUCCaCUGCC
G-GC-G/Ca$_2$C	GGCAGGCGCUUCGGCGCGGAGG/CCUCCaaCUGCC
G-GC-G/Ca$_3$C	GGCAGGCGCUUCGGCGCGGAGG/CCUCCaaaCUGCC

Continued on next page

Table A.2 – continued from previous page

Pairs	Sequences
G-GC-G/CauaC	GGCAGGCGCUUCGGCGCGGAGG/CCUCCauaCUGCC
G-GC-G/Ca$_4$C	GGCAGGCGCUUCGGCGCGGAGG/CCUCCaaaaCUGCC
GaGC-G/C-C	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCCUGCC
GaGC-G/CaC	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCaCUGCC
GaGC-G/Ca$_2$C	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCaaCUGCC
GaGC-G/Ca$_3$C	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCaaaCUGCC
GaGC-G/CauaC	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCauaCUGCC
GaGC-G/Ca$_4$C	GGCAGaGCGCUUCGGCGCGGAGG/CCUCCaaaaCUGCC
Ga$_2$GC-G/C-C	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCCUGCC
Ga$_2$GC-G/CaC	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCaCUGCC
Ga$_2$GC-G/Ca$_2$C	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCaaCUGCC
Ga$_2$GC-G/Ca$_3$C	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCaaaCUGCC
Ga$_2$GC-G/CauaC	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCauaCUGCC
Ga$_2$GC-G/Ca$_4$C	GGCAGaaGCGCUUCGGCGCGGAGG/CCUCCaaaaCUGCC
Ga$_2$GCaG/C-C	GGCAGaaGCGCUUCGGCGCaGGAGG/CCUCCCUGCC
Ga$_2$GCaG/CaC	GGCAGaaGCGCUUCGGCGCaGGAGG/CCUCCaCUGCC
Ga$_2$GCaG/Ca$_2$C	GGCAGaaGCGCUUCGGCGCaGGAGG/CCUCCaaCUGCC

Continued on next page

Table A.2 – continued from previous page

Pairs	Sequences
Ga$_2$GCaG/Ca$_3$C	GGCAGaaGCGCUUCGGCGCaGGAGG /CCUCCaaaCUGCC
Ga$_2$GCaG/CauaC	GGCAGaaGCGCUUCGGCGCaGGAGG /CCUCCauaCUGCC
Ga$_2$GCaG/Ca$_4$C	GGCAGaaGCGCUUCGGCGCaGGAGG /CCUCCaaaaCUGCC
Ga$_2$GCa$_2$G/C-C	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCCUGCC
Ga$_2$GCa$_2$G/CaC	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCaCUGCC
Ga$_2$GCa$_2$G/Ca$_2$C	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCaaCUGCC
Ga$_2$GCa$_2$G/Ca$_3$C	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCaaaCUGCC
Ga$_2$GCa$_2$G/CauaC	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCauaCUGCC
Ga$_2$GCa$_2$G/Ca$_4$C	GGCAGaaGCGCUUCGGCGCaaGGAGG /CCUCCaaaaCUGCC
G-UA-G/C-C	GGCAGUCGCUUCGGCGAGGAGG /CCUCCCUGCC
G-UA-G/CaC	GGCAGUCGCUUCGGCGAGGAGG /CCUCCaCUGCC
G-UA-G/Ca$_2$C	GGCAGUCGCUUCGGCGAGGAGG /CCUCCaaCUGCC
G-UA-G/Ca$_3$C	GGCAGUCGCUUCGGCGAGGAGG /CCUCCaaaCUGCC
G-UA-G/CauaC	GGCAGUCGCUUCGGCGAGGAGG /CCUCCauaCUGCC
G-UA-G/Ca$_4$C	GGCAGUCGCUUCGGCGAGGAGG /CCUCCaaaaCUGCC
GaUA-G/C-C	GGCAGaUCGCUUCGGCGAGGAGG /CCUCCCUGCC
GaUA-G/CaC	GGCAGaUCGCUUCGGCGAGGAGG /CCUCCaCUGCC

Continued on next page

Table A.2 – continued from previous page

Pairs	Sequences
GaUA-G/Ca$_2$C	GGCAGaUCGCUUCGGCGAGGAGG/CCUCCaaCUGCC
GaUA-G/Ca$_3$C	GGCAGaUCGCUUCGGCGAGGAGG/CCUCCaaaCUGCC
GaUA-G/CauaC	GGCAGaUCGCUUCGGCGAGGAGG/CCUCCauaCUGCC
GaUA-G/Ca$_4$C	GGCAGaUCGCUUCGGCGAGGAGG/CCUCCaaaaCUGCC
G-CG-GC-G/C-C	GGCAGCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCCUGCC
G-CG-GC-G/CaC	GGCAGCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaCUGCC
G-CG-GC-G/Ca$_2$C	GGCAGCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaCUGCC
G-CG-GC-G/Ca$_3$C	GGCAGCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaaCUGCC
G-CG-GC-G/Ca$_4$C	GGCAGCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaaaCUGCC
GaCG-GC-G/C-C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCCUGCC
GaCG-GC-G/CaC	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaCUGCC
GaCG-GC-G/Ca$_2$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaCUGCC
GaCG-GC-G/Ca$_3$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaaCUGCC
GaCG-GC-G/CauaC	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCauaCUGCC
GaCG-GC-G/Ca$_4$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCGGAGG/CCUCCaaaaCUGCC
GaCG-GCaG/C-C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG/CCUCCCUGCC
GaCG-GCaG/CaC	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG/CCUCCaCUGCC

Continued on next page

Table A.2 – continued from previous page

Pairs	Sequences
GaCG-GCaG/Ca$_2$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG /CCUCCaaCUGCC
GaCG-GCaG/Ca$_3$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG /CCUCCaaaCUGCC
GaCG-GCaG/CauaC	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG /CCUCCauaCUGCC
GaCG-GCaG/Ca$_4$C	GGCAGaCGGCUUCGGCCGGCGCGCAAGCGCaGGAGG /CCUCCaaaaCUGCC
Ga$_2$CGa$_2$GCa$_2$G/C-C	GGCAGaaCGGCUUCGGCCGaaGCGCGCAAGCGCaaGGAGG /CCUCCCUGCC
Ga$_2$CGa$_2$GCa$_2$G/CaC	GGCAGaaCGGCUUCGGCCGaaGCGCGCAAGCGCaaGGAGG /CCUCCaCUGCC
Ga$_2$CGa$_2$GCa$_2$G/Ca$_2$C	GGCAGaaCGGCUUCGGCCGaaGCGCGCAAGCGCaaGGAGG /CCUCCaaCUGCC

Bibliography

[1] C. Aksay, R. Salari, E. Karakoc, C. Alkan, and S. C. Sahinalp. taveRNA: a web suite for RNA algorithms and applications. *Nucleic Acids Res.*, 35:W325–329, Jul 2007.

[2] T. Akutsu. Approximation and exact algorithms for RNA secondary structure prediction and recognition of stochastic context-free languages. *J. Comb. Optim.*, 3(2-3):321–336, 1999.

[3] T. Akutsu. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discrete Appl. Math.*, 104(1-3):45–62, 2000.

[4] C. Alkan, E. Karakoc, J. H. Nadeau, S. C. Sahinalp, and K. Zhang. RNA-RNA interaction prediction and antisense RNA target search. *Journal of Computational Biology*, 13(2):267–282, 2006.

[5] C. Alkan, E. Karakoc, S. C. Sahinalp, P. Unrau, A. Ebhardt, K. Zhang, and J. Buhler. RNA secondary structure prediction via energy density minimization. In *Proc. RECOMB, LNBI 3909*, pages 130–142, May 2006.

[6] S. Altuvia, A. Zhang, L. Argaman, A. Tiwari, and G. Storz. The Escherichia coli OxyS regulatory RNA represses fhlA translation by blocking ribosome binding. *EMBO J.*, 17:6069–6075, Oct 1998.

[7] M. Andronescu, A. Condon, H. H. Hoos, D. H. Mathews, and K. P. Murphy. Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, 23(13), 2007.

[8] M. Andronescu, Z. C. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *J. Mol. Biol.*, 345:987–1001, Feb 2005.

[9] L. Argaman and S. Altuvia. fhlA repression by OxyS RNA: kissing complex formation at two sites results in a stable antisense-target RNA complex. *J. Mol. Biol.*, 300:1101–1112, Jul 2000.

[10] K. Asano and K. Mizobuchi. Structural analysis of late intermediate complex formed between plasmid ColIb-P9 Inc RNA and its target RNA. How does a single antisense RNA repress translation of two genes at different rates? *J. Biol. Chem.*, 275:1269–1274, Jan 2000.

[11] R. Backofen, D. Tsur, S. Zakov, and M. Ziv-Ukelson. Sparse RNA folding: Time and space efficient algorithms. In Gregory Kucherov and Esko Ukkonen, editors, *Proc. 20th Symp. Combinatorial Pattern Matching*, volume 5577 of *LNCS*, pages 249–262. Springer, 2009.

[12] V. Bafna, H. Tang, and S. Zhang. Consensus folding of unaligned RNA sequences revisited. *J. Comput. Biol.*, 13:283–295, Mar 2006.

[13] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116(2):281–97, 2004.

[14] S. Bartz and A. L. Jackson. How will RNAi facilitate drug development? *Sci. STKE*, 2005:pe39, Aug 2005.

[15] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. GenBank. *Nucleic Acids Research*, 36(Database issue):D25–30, 2008.

[16] S. H. Bernhart, I. L. Hofacker, and P. F. Stadler. Local RNA base pairing probabilities in large sequences. *Bioinformatics*, 22:614–615, Mar 2006.

[17] S. H. Bernhart, H. Tafer, U. Mückstein, C. Flamm, P. F. Stadler, and I. L. Hofacker. Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol Biol*, 1:3, 2006.

[18] S. Brantl. Antisense-RNA regulation and RNA interference. *Bioch. Biophys. Acta*, 1575(1-3):15–25, 2002.

[19] C. Brunel, R. Marquet, P. Romby, and C. Ehresmann. RNA loop-loop interactions as dynamic functional motifs. *Biochimie*, 84:925–944, Sep 2002.

[20] A. Busch, A. S. Richter, and R. Backofen. IntaRNA: Efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. *Bioinformatics*, page btn544, 2008.

[21] T. M. Chan. More algorithms for all-pairs shortest paths in weighted graphs. In *STOC*, pages 590–598, 2007.

[22] H. Chitsaz, R. Salari, S. C. Sahinalp, and R. Backofen. A partition function algorithm for interacting nucleic acid strands. *Bioinformatics*, 25:i365–373, Jun 2009.

[23] A. Condon, B. Davy, B. Rastegari, S. Zhao, and F. Tarrant. Classifying RNA pseudoknotted structures. *Theor. Comput. Sci.*, 320(1):35–50, 2004.

[24] L. V. Danilova, D. D. Pervouchine, A. V. Favorov, and A. A. Mironov. RNAKinetics: a web server that models secondary structure kinetics of an elongating RNA. *J Bioinform Comput Biol*, 4:589–596, Apr 2006.

[25] L. David, W. Huber, M. Granovskaia, J. Toedling, C. J. Palm, L. Bofkin, T. Jones, R. W. Davis, and L. M. Steinmetz. A high-resolution map of transcription in the yeast genome. *Proc. Natl. Acad. Sci. USA*, 103(14):5320–5, 2006.

[26] E. Davydov and S. Batzoglou. A computational model for RNA multiple structural alignment. *Theor. Comput. Sci.*, 368(3):205–216, 2006.

[27] J. M. Diamond, D. H. Turner, and D. H. Mathews. Thermodynamics of three-way multibranch loops in RNA. *Biochemistry*, 40:6971–6981, Jun 2001.

[28] R. A. Dimitrov and M. Zuker. Prediction of hybridization and melting for double-stranded nucleic acids. *Biophysical Journal*, 87:215–226, 2004.

[29] Y. Ding, C. Y. Chan, and C. E. Lawrence. RNA secondary structure prediction by centroids in a Boltzmann weighted ensemble. *RNA*, 11:1157–1166, Aug 2005.

[30] R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, and N. A. Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Review*, 49(1):65–88, 2007.

[31] R. M. Dirks and N. A. Pierce. A partition function algorithm for nucleic acid secondary structure including pseudoknots. *Journal of Computational Chemistry*, 24(13):1664–1677, 2003.

[32] C. B. Do, D. A. Woods, and S. Batzoglou. CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, 22:e90–98, Jul 2006.

[33] B. Felden. RNA structure: experimental analysis. *Curr. Opin. Microbiol.*, 10:286–291, Jun 2007.

[34] M. E. Fisher. Shape of a self-avoiding walk or polymer chain. *Journal of Chemical Physics*, 44:616–622, 1966.

[35] C. Flamm, W. Fontana, I. L. Hofacker, and P. Schuster. RNA folding at elementary step resolution. *RNA*, 6:325–338, Mar 2000.

[36] Y. Frid and D. Gusfield. A simple, practical and complete $O(n^3/log(n))$-time Algorithm for RNA folding using the Four-Russians Speedup. *Algorithms Mol Biol*, 5:13, Jan 2010.

[37] J. Gorodkin, L. J. Heyer, and G. D. Stormo. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucleic Acids Res.*, 25:3724–3732, Sep 1997.

[38] S. Gottesman. Micros for microbes: non-coding regulatory RNAs in bacteria. *Trends in Genetics*, 21(7):399–404, 2005.

[39] L. R. Grillone and R. Lanz. Fomivirsen. *Drugs Today*, 37:245–255, Apr 2001.

[40] M. Grunberg-Manago H. Putzer and M. Springer. Bacterial aminoacyl-tRNA synthetase: Genes and regulation of expression. *tRNA: Structure, Biosynthesis, and Function*, page 293, 1995.

[41] J. Hackermller, N. C. Meisner, M. Auer, M. Jaritz, and P. F. Stadler. The effect of RNA secondary structures on RNA-ligand binding and the modifier RNA mechanism: a quantitative model. *Gene*, 345:3–12, 2005.

[42] M. Hamada, H. Kiryu, K. Sato, T. Mituyama, and K. Asai. Prediction of RNA secondary structure using generalized centroid estimators. *Bioinformatics*, 25:465–473, Feb 2009.

[43] M. Hamada, K. Sato, H. Kiryu, T. Mituyama, and K. Asai. Predictions of RNA secondary structure by combining homologous sequence information. *Bioinformatics*, 25:i330–338, Jun 2009.

[44] G. J. Hannon. RNA interference. *Nature*, 418(6894):244–51, 2002.

[45] T. A. Hjalt and E. G. Wagner. Bulged-out nucleotides in an antisense RNA are required for rapid target RNA binding in vitro and inhibition in vivo. *Nucleic Acids Res.*, 23:580–587, Feb 1995.

[46] I. L. Hofacker, M. Fekete, and P. F. Stadler. Secondary structure prediction for aligned RNA sequences. *J. Mol. Biol.*, 319:1059–1066, Jun 2002.

[47] I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures (the Vienna RNA Package), 1994.

[48] F. W. Huang, J. Qin, C. M. Reidys, and P. F. Stadler. Partition function and base pairing probabilities for RNA-RNA interaction prediction. *Bioinformatics*, 25:2646–2654, Oct 2009.

[49] Y. Ji, X. Xu, and G. D. Stormo. A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. *Bioinformatics*, 20:1591–1602, Jul 2004.

[50] Y. Kafri, D. Mukamel, and L. Peliti. Why is the DNA denaturation transition first order? *Phys. Rev. Lett.*, 85:4988–4991, Dec 2000.

[51] Y. Kato, T. Akutsu, and H. Seki. A grammatical approach to RNA-RNA interaction prediction. *Pattern Recogn.*, 42(4):531–538, 2009.

[52] M. Kertesz, N. Iovino, U. Unnerstall, U. Gaul, and E. Segal. The role of site accessibility in microRNA target recognition. *Nat. Genet.*, 39:1278–1284, Oct 2007.

[53] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Res.*, 31:3423–3428, Jul 2003.

[54] F. A. Kolb, C. Malmgren, E. Westhof, C. Ehresmann, B. Ehresmann, E. G. Wagner, and P. Romby. An unusual structure formed by antisense-target RNA binding involves an extended kissing complex with a four-way junction and a side-by-side helical alignment. *RNA*, 6:311–324, Mar 2000.

[55] G. Koraimann, C. Koraimann, V. Koronakis, S. Schlager, and G. Hogenauer. Repression and derepression of conjugation of plasmid R1 by wild-type and mutated finP antisense RNA. *Mol. Microbiol.*, 5:77–87, Jan 1991.

[56] L. D. Landau and E. M. Lifshitz. *Statistical Physics.* Pergamon, Oxford, UK, 1969.

[57] B. P. Lewis, C. B. Burge, and D. P. Bartel. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. *Cell*, 120:15–20, Jan 2005.

[58] J. Light and S. Molin. The sites of action of the two copy number control functions of plasmid R1. *Mol. Gen. Genet.*, 187:486–493, 1982.

[59] Z. J. Lu, J. W. Gloor, and D. H. Mathews. Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, 15:1805–1813, Oct 2009.

[60] R. B. Lyngsøand C. N. Pedersen. RNA pseudoknot prediction in energy-based models. *J. Comput. Biol.*, 7:409–427, 2000.

[61] R. B. Lyngsø, M. Zuker, and C. N. Pedersen. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics*, 15:440–445, Jun 1999.

[62] N. R. Markham and M. Zuker. UNAFold: software for nucleic acid folding and hybridization. *Methods Mol. Biol.*, 453:3–31, 2008.

[63] D. H. Mathews, J. Sabina, M. Zuker, and D. H. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, 288:911–940, May 1999.

[64] D. H. Mathews and D. H. Turner. Experimentally derived nearest-neighbor parameters for the stability of RNA three- and four-way multibranch loops. *Biochemistry*, 41:869–880, Jan 2002.

[65] J. S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29:1105–1119, 1990.

[66] N. C. Meisner, J. Hackermller, V. Uhl, A. Aszdi, M. Jaritz, and M. Auer. mRNA openers and closers: modulating AU-rich element-controlled mRNA stability by a molecular switch in mRNA secondary structure. *Chembiochem*, 5:1432–1447, 2004.

[67] S. Mocellin and M. Provenzano. RNA interference: learning gene knock-down from cell physiology. *J Transl Med*, 2:39, 11 2004.

[68] U. Mückstein, H. Tafer, J. Hackermüller, S. H. Bernhart, M. Hernandez-Rosales, J. Vogel, P. F. Stadler, and I. L. Hofacker. Translational control by RNA-RNA interaction: Improved computation of RNA-RNA binding thermodynamics. *Bioinformatics Research and Development*, 13:114–127, 2008.

[69] U. Mückstein, H. Tafer, J. Hackermüller, S. H. Bernhart, P. F. Stadler, and I. L. Hofacker. Thermodynamics of RNA-RNA binding. *Bioinformatics*, 22:1177–1182, May 2006.

[70] S. H. Najafi-Shoushtari, F. Kristo, Y. Li, T. Shioda, D. E. Cohen, R. E. Gerszten, and A. M. Naar. MicroRNA-33 and the SREBP host genes cooperate to control cholesterol homeostasis. *Science*, 328:1566–1569, Jun 2010.

[71] R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 77:6309–6313, Nov 1980.

[72] R. Nussinov, G. Piecznik, J. R. Grigg, and D. J. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35:68–82, 1978.

[73] C. Persson, E. G. Wagner, and K. Nordstrom. Control of replication of plasmid R1: structures and sequences of the antisense RNA, CopA, required for its binding to the target RNA, CopT. *EMBO J.*, 9:3767–3775, Nov 1990.

[74] D. D. Pervouchine. IRIS: intermolecular RNA interaction search. *Genome Inform*, 15:92–101, December 2004.

[75] J. D. Puglisi and I. Tinoco. Absorbance melting curves of RNA. *Meth. Enzymol.*, 180:304–325, 1989.

[76] J. Reeder and R. Giegerich. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, 5:104, 2004.

[77] M. Rehmsmeier, P. Steffen, M. Hochsmann, and R. Giegerich. Fast and effective prediction of microRNA/target duplexes. *RNA*, 10:1507–1517, Oct 2004.

[78] E. Rivas and S. R. Eddy. A dynamic programming algorithm for RNA structure prediction including pseudoknots. *Journal of Molecular Biology*, 285(5):2053–68, 1999.

[79] R. Salari, R. Backofen, and S. C. Sahinalp. Fast prediction of RNA-RNA interaction. *Algorithms Mol Biol*, 5:5, 2010.

[80] R. Salari, M. Möhl, S. Will, S. C. Sahinalp, and R. Backofen. Time and space efficient RNA-RNA interaction prediction via sparse folding. In *RECOMB*, pages 473–490, 2010.

[81] D. Sankoff. Simultaneous solution of the RNA folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, 45(5):810–825, 1985.

[82] K. L. Schaefer and W. R. McClure. Antisense RNA control of gene expression in bacteriophage P22. II. Kinetic mechanism and cation specificity of the pairing reaction. *RNA*, 3:157–174, Feb 1997.

[83] K. R. Siemering, J. Praszkier, and A. J. Pittard. Mechanism of binding of the antisense and target RNAs involved in the regulation of IncB plasmid replication. *J. Bacteriol.*, 176:2677–2688, May 1994.

[84] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, Mar 1981.

[85] G. Storz. An expanding universe of noncoding RNAs. *Science*, 296(5571):1260–3, 2002.

[86] H. Tafer and I. L. Hofacker. RNAplex: a fast tool for RNA-RNA interaction search. *Bioinformatics*, 24:2657–2663, Nov 2008.

[87] I. Tinoco, O. C. Uhlenbeck, and M. D. Levine. Estimation of secondary structure in ribonucleic acids. *Nature*, 230:362–367, Apr 1971.

[88] B. Tjaden, S. S. Goodwin, J. A. Opdyke, M. Guillier, D. X. Fu, S. Gottesman, and G. Storz. Target prediction for small, noncoding RNAs in bacteria. *Nucleic Acids Research*, 34(9):2791–802, 2006.

[89] J. Tomizawa. Control of ColE1 plasmid replication: binding of RNA I to RNA II and inhibition of primer formation. *Cell*, 47:89–97, Oct 1986.

[90] J. Tomizawa, T. Itoh, G. Selzer, and T. Som. Inhibition of ColE1 RNA primer formation by a plasmid-specified small RNA. *Proc. Natl. Acad. Sci. U.S.A.*, 78:1421–1425, Mar 1981.

[91] D. Tsur, S. Zakov, and M. Ziv-ukelson. Reducing the worst case running time of a family of RNA and CFG problems, using Valiant's approach. In *Proc. WABI*, 2010.

[92] Y. Uemura, A. Hasegawa, S. Kobayashi, and T. Yokomori. Tree adjoining grammars for RNA structure prediction. *Theoretical Computer Science*, 210:277 – 303, 1999.

[93] L. G. Valiant. General context-free recognition in less than cubic time. *J. Comput. Syst. Sci.*, 10(2):308–315, 1975.

[94] T. van Biesen and L. S. Frost. The FinO protein of IncF plasmids binds FinP antisense RNA and its target, traJ mRNA, and promotes duplex formation. *Mol. Microbiol.*, 14:427–436, Nov 1994.

[95] E. G. Wagner and K. Flärdh. Antisense RNAs everywhere? *Trends Genet.*, 18:223–226, 2002.

[96] S. P. Walton, G. N. Stephanopoulos, M. L. Yarmush, and C. M. Roth. Thermodynamic and kinetic characterization of antisense oligodeoxynucleotide binding to a structured mRNA. *Biophys. J.*, 82:366–377, Jan 2002.

[97] P. M. Waterhouse and C. A. Helliwell. Exploring plant genomes by RNA-induced gene silencing. *Nat. Rev. Genet.*, 4:29–38, Jan 2003.

[98] Y. Wexler, C. Zilberstein, and M. Ziv-Ukelson. A study of accessible motifs and RNA folding complexity. *Journal of Computational Biology (Special RECOMB 2006 Issue)*, 14(6):856–72, 2007.

[99] S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen. Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, 3:e65, Apr 2007.

[100] T. Wu, J. Wang, C. Liu, Y. Zhang, B. Shi, X. Zhu, Z. Zhang, G. Skogerb, L. Chen, H. Lu, Y. Zhao, and R. Chen. NPInter: the noncoding RNAs and protein related biomacromolecules interaction database. *Nucleic Acids Res.*, 34:D150–152, Jan 2006.

[101] T. Xia, J. SantaLucia, M. E. Burkard, R. Kierzek, S. J. Schroeder, X. Jiao, C. Cox, and D. H. Turner. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, 37:14719–14735, Oct 1998.

[102] P. D. Zamore and B. Haley. Ribo-gnome: the big world of small RNAs. *Science*, 309(5740):1519–24, 2005.

[103] Z. Zhang, S. M. Kang, Y. Li, and C. D. Morrow. Genetic analysis of the U5-PBS of a novel HIV-1 reveals multiple interactions between the tRNA and RNA genome required for initiation of reverse transcription. *RNA*, 4:394–406, Apr 1998.

[104] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, 31:3406–3415, Jul 2003.

[105] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9(1):133–148, 1981.