Study of Two Biochemical Models:

Chemical Reaction Networks, and Nucleic Acid Systems

by

Monir Hajiaghayi

B.Sc., Sharif University of Technology, 2008
M.Sc., The University of British Columbia, 2010

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

October 2017

Abstract

The contributions of this thesis are motivated by an exciting challenge at the intersection of computer science and biochemistry: Can we program molecules to do interesting or useful computations? There has been significant progress in programming nucleic acids - particularly DNA molecules - thanks in part to availability of models and algorithms for predicting nucleic acid structure and folding kinetics. At a higher level of abstraction, Chemical Reaction Networks (CRNs) have proven to be valuable as a molecular programming model that enables researchers to understand the potential and limitations of computing with molecules, unencumbered by low-level details. These two levels of abstraction are linked; it is possible to "compile" CRN programs into nucleic acid systems that make the programs implementable in a test tube.

We design and analyze CRN algorithms for two purposes. First, we show how any semilinear function can be computed by CRNs, even when no "leader" species (i.e., initial species with constant but non-zero counts) is present. Our results improve earlier results of Chen et al. (2012) who showed that only semilinear functions are computable by error-free CRNs using leaders. Our new CRN construction can be done in expected time $O(n)$, which is faster than $O(n \log n)$ bound achieved by Chen et al. Second, we provide the most intuitive proofs of correctness and efficiency for three different CRNs computing Approximate Majority: Given a mixture of two types of species with an initial gap between their counts, a CRN computation must reach totality on the majority species with high probability. The CRNs of our interest have the ability to start with an initial gap of $\Omega(\sqrt{n \log n})$.

In the second part of this thesis, we study the problem of predicting the Minimum Free Energy secondary structure (the set of base pairs) of a given set of nucleic acid strands with no pseudoknots (crossing base pairs).We show that this problem is APX-hard which implies that there does not exist a polynomial time approximation scheme for this problem, unless P = NP. We also propose a new Monte-Carlo based method to efficiently estimate nucleic acid folding kinetics.

Lay Summary

Our contributions are motivated by an exciting challenge at the intersection of computer science and biochemistry: Can we program molecules to do interesting or useful computations? First, we study Chemical Reaction Networks (CRNs) – a valuable molecular model for programming the dynamics of interacting molecules in a well-mixed solution. We show how CRNs can compute functions that are unions of linear pieces, and provide a simple analysis of an elegant CRN that determines which of two molecular species is most populous in the mixture. Since CRN programs can be"compiled" into nucleic acid systems whose folding dynamics simulate the reactions, the second part of our thesis studies such systems. We present an efficient method to estimate the nucleic acid folding kinetics, and show that the structure prediction of multiple interacting strands is computationally intractable.

Preface

The author contributed to all major ideas and writing of all published and unpublished manuscripts that are the basis of this dissertation. In no instance, there was a student as a co-author. The author collaborated with the author's supervisor and mentors in all aspects of research including defining the problems, design and implementation of algorithms, proving the main claims and conducting experimental studies.

Part I of this thesis was resulted from collaboration with author's supervisor Dr. Anne Condon, as well as Dr. Dave Doty, Dr. David Kirkpatrick and Dr. Jan Manuch.

- Chapters 1, 2 and 5 were written by the author, but used selected content from publications that she co-authored [28, 33, 34].
- A version of Chapter 3 has been published in the proceedings of the 19th Annual International Conference on DNA Computing and Molecular Programming (2013) [33] and also the Journal of Natural Computing (2015) [34]. The author collaborated with the co-author, Dr. Dave Doty, in developing the algorithms, proving the lemmas, and writing the manuscripts.
- A version of Chapter 4 appears in the proceedings of the 23th Annual International Conference on DNA Computing and Molecular Programming (2017) [28]. The author was the primary investigator to lead all the discussions and provide the initial detailed proofs of each phase in the new proof strategy, and performed most of the experiments. Dr. David Kirkpatrick proposed the new tri-molecular CRN as an abstraction and intuition of the phases required to compute Approximate Majority. All the authors then equally contributed in further simplifying the proofs and writing the manuscript.

Part II of this thesis was resulted from collaboration with a number of co-authors: the author's supervisor Dr. Anne Condon, Dr. Bonnie Kirkpatrick, Dr. Alexandre Bouchard-Côté, Dr. Chris Thachuk, and Dr. Liangliang Wang.

- Chapters 6, and 9 were written by the author, but used selected content from publications that she co-authored [29, 47, 60].

- A version of Chapter 7 has been published in 31th International Conference on Machine Learning (ICML2013) [47]. The author collaborated with co-authors in the development and implementation of the new sampling method and writing the manuscript. The author also performed all the experiments on *Nucleic Acid Folding Pathways*. Dr. Liangliang Wang took the lead on experimental studies of another biological example in the original paper that was excluded from the thesis.

- The work presented in Chapter 8 [29] is conducted from the collaboration of the author with the author's supervisor Dr. Anne Condon, and Dr. Chris Thachuk. The author contributed to all aspect of research and specifically revised the earlier reduction model proposed by co-authors to conclude more strong results of APX-hardness, proved the reduction correctness and was one of the main authors. The results from this chapter is not published yet, but the manuscript is almost ready (pending on experimental study) and will be submitted to SIAM Review Journal.

Table of Contents

List of Tables

List of Figures

Acknowledgements

I would like to express my special appreciation and thanks to my research supervisor Professor Anne Condon, you have been a tremendous mentor and advisor for me during both my masters and doctoral studies and I could not have imagined having a better mentor during these years. Thank you for making time for me despite your busy schedules, supporting me through my though times, allowing and encouraging me to pursue different research directions, and teaching me how to be a research scientist. I was extremely fortunate to know you and work along side you.

I would also like to thank my committee members, Professor David Kirkpatrick, and Professor Holger Hoos. Thank you David for all those long and helpful discussions. You thought me how to think more critically, pay more attentions to the details and become a better researcher. Moreover, I would like to thank you for your insightful comments and suggestions that remarkably improved the writing of the first part of my thesis. Thank you Holger for all your help, support and valuable comments and feedback that enhanced the quality of my thesis.

Many thanks to my examining committee, Professor Will Evans and Professor Ruben Zamar and my external examiner, Professor Ho-Lin Chen, for their time and effort that they put into reading my thesis and for their insightful comments and feedback. Special thanks to my collaborators, Dr. Dave Doty, Dr. Jan Manuch, Dr. Bonnie Kirkpatrick, Dr. Chris Thachuk and Professor Alex Bouchard Cote. My research has greatly benefited from the helpful discussions and your productive suggestions. I would like to thank the current and former members of the BETA lab including Shu Yang, Nasim Zolakbaf, Hosna Jabbari, Mirela Andronescu, Baharak Rastegari, and Chris Thachuk for their friendship, support and help during my time in UBC. Thanks to the support staff at the UBC Computer Science Department, in particular Holly Kwan, Joyce Poon, Lara Hall and Michele Ng for being such a great and supportive people.

A special thanks to my family. Words cannot express how grateful I am to my parents, my only amazing sister Mehri, and my brothers for all their encouragements, and sacrifices that they have made on my behalf. I

would also like to thank my friends, in particular Elahe, Noushin, Anahita, Fatemeh, Neda, and Samaneh. Thank you for true friendship and being a part of my life. At the end I would like to express my appreciation to my beloved husband Ali who has always been my support in the gloomiest times and always offered me his love, and advice. Thank you for being such a wonderful partner.

Dedication

To my dear mom and my late beloved dad.

Part I

Leaderless Deterministic Computation, and Approximate Majority with Chemical Reaction Networks

Chapter 1

Introduction

Molecular programming encompasses the design of programmable molecular systems such as molecular robotics, nanoscale computing, and programmable assembly of nanoscale patterns and devices. It also encompasses languages and algorithms for programming these systems [30, 62, 82]. In the last two decades, theoretical and experimental studies in this field have shed light on integration of logical computation with biological systems [25, 46]. For instance, such an integration can let a certain curative agent be released *if* a specific condition is detected in a cell.

A key goal is to re-purpose the *descriptive* language of chemistry and physics, which describes how the natural world works, as a *prescriptive* language of programming, which prescribes how an artificially engineered system *should* work. When the programming goal is the manipulation of individual molecules in a well-mixed solution, the language of chemical reaction networks (CRNs) is an attractive choice.

A CRN (i.e., formally defined in Section 2.1) is a finite set of reactions such as $X + Y \rightarrow Z + W$ each describing a rule for transforming reactant molecules (e.g., $\{X, Y\}$) into product molecules (e.g., $\{Z, W\}$) [42, 57]. Figures 1.1 and 1.2 show two examples of CRNs. The underlying model describes how the amounts of molecular species evolve when molecules interact in a well-mixed solution. If an interaction (i.e., a collision between some molecules) is reactive, some molecules are consumed and others are produced. Thus, by *reaction*, we mean a reactive interaction. A configuration of a well-mixed solution is also defined as the current composition of the amounts of all the species.

CRNs may model the "amount" of a species as a real number, namely its concentration (average count per unit volume), or as a nonnegative integer (i.e., total count in solution, requiring the total volume of the solution to be specified as part of the system). Here, we focus on the latter integer counts model which is called "stochastic". The stochastic nature of the CRN model makes it better suited than the former model for analysis of systems in which some species counts may be low and may fluctuate significantly. In this model, the reactions *discretely* change the configuration of the system, and

are assumed to happen probabilistically. In fact, the model is probabilistic at two levels. First, which interaction occurs next is stochastically determined, reflecting the dynamics of collisions in a well-mixed solution [20, 42]. For instance, interactions whose reactants have high molecular counts are more likely to happen first than interactions whose molecular counts are smaller. Second, a reaction can have more than one outcome, and rate constants associated with reactions determine the relative likelihood of each outcome. For example, reactions (0'x) and (0'y) of Figure 1.2(c) are equally likely. A computation of a CRN $\mathcal{C}$ is a trace of the configurations from a given initial configuration to some final target configuration. We note that interactions occur in parallel, and time of a computation is typically measured as the total number of interactions divided by n, assuming that the total number of interacting molecules and volume remain fixed and are $\Theta(n)$.

The computational power of CRNs has been investigated with regard to simulating boolean circuits [65], neural networks [50], digital signal processing [55], and simulating bounded-space Turing machines with an arbitrary small, non-zero probability of error with only a polylogarithmic slowdown [11]. We note that Angluin et al. investigated the computational power of bi-molecular CRNs (involving only two reactants) under a different name known as the *population protocols* (PPs) model [9], in which agents interact in a pairwise fashion and may change state upon interacting. Agents and states of a PP naturally correspond to molecules and species of a CRN. CRNs are even efficiently Turing-universal, again with a small, nonzero probability of error over all time [94]. Using a theoretical model of DNA strand displacement systems (DSDs) [1], it was shown that any CRN can be transformed into a set of DNA complexes that approximately emulate the CRN [83, 92, 95]. The experimental successes to date, are small in scale, but in the future we can expect to be able to reliably implement much larger CRNs by real chemicals, as causes of error are addressed [84, 99].
In the context of CRNs and PPs, a substanstial amount of research has also been conducted on a central problem called Approximate Majority (AM) [9, 12, 74, 81]: in a mixture of two types of species where the gap between the counts of the majority and minority species is above some threshold, which species is in the majority? Approximate Majority is a significant problem because, for instance, it can be used as a subroutine when simulating other computational models [9]. For example, a comparison operation (comparing the counts of two species or agents) is a necessary component

[1] DSDs consist of a set of strand displacement reactions where intuitively, in each reaction, a strand displaces another strand from a complex [18].

to simulate the behaviour of a register machine and the AM protocol is an important subroutine to compute this operation [12]. Moreover, the system of some biological switches is also related to both the structure and kinetics of the AM computation [19, 24].

While the works mentioned above focus on the stochastic behaviour of chemical kinetics and may encounter some probability of error in the computations, the deterministic behaviour of CRNs, in the sense that the designed protocols progress to a correct configuration with absolutely no error, is also of great interest [10, 21].

In this part of the thesis, we make new contributions to design of CRNs for deterministic computation, and analysis of CRNs for Approximate Majority. We discuss these two subjects and their related work in Section 1.1 and 1.2 respectively. Next, we sum up our objectives and contributions in each context in Section 1.3. Finally, we end this introductory chapter by giving an outline of the following chapters that detail our contributions in this thesis part.

1.1 Deterministic Computation with CRNs

In this section, our focus is on CRNs with deterministic guarantees on their behavior. These CRNs have the property that they are guaranteed to reach a correct configuration, regardless of the order in which reactions occur. For example, the CRN with the reaction $X \to 2Y$ is guaranteed eventually to reach a configuration in which the count of Y is twice the initial count of X, i.e., computes the function $f(x) = 2x$, representing the input by the initial count x of species X and the output by the count y of species Y, once stable. Similarly, the reactions $X_1 \to 2Y$ and $X_2 + Y \to \varnothing$, under arbitrary choice of sequence of the two reactions, compute the function $f(x_1, x_2) = \max\{0, 2x_1 - x_2\}$.

Angluin et al. [10, 11] explored the computational behaviour of CRNs that are deterministic under the population protocols model. They showed that the input sets $S \subseteq \mathbb{N}^k$ decidable by deterministic CRNs (i.e., providing "yes" or "no" answers by the presence or absence of certain indicator species) are precisely the *semilinear* subsets of $\mathbb{N}^k$. Semilinear sets are defined formally in Section 3.1. Informally, they are finite unions of "periodic" sets, where the definition of "periodic" is extended in a natural way to multi-dimensional spaces, such as $\mathbb{N}^k$. We note that the semilinear predicate computation presented by Angluin et al. [10, 11] only depends on the input species and does not require any auxiliary "leader" species, i.e., species

$$X \to Y \qquad (1.1.1)$$
$$L \to Y \qquad (1.1.2)$$

(a)

$$X \to B + 2Y \qquad (1.1.1)$$
$$B + B \to B + K \qquad (1.1.2)$$
$$Y + K \to \varnothing \qquad (1.1.3)$$

(b)

Figure 1.1: CRNs to compute function $f(x) = x + 1$. CRN (a) starts with x copies of species X and one copy of leader species L. CRN (b) starts with only x copies of species X. All the reactions have rate constant 1.

present initially with constant but non-zero counts (described in more detail later).

Chen et al. [21] extended these results to function computation and showed that precisely the semilinear functions (functions f whose graph $\{ (\mathbf{x}, \mathbf{y}) \in \mathbb{N}^{k+l} \mid f(\mathbf{x}) = \mathbf{y} \}$ is a semilinear set) are deterministically computable by CRNs. We say a function $f : \mathbb{N}^k \to \mathbb{N}^l$ is *stably* (a.k.a., *deterministically*) computable by a CRN $\mathcal{C}$ if there are "input" species $X_1, \ldots, X_k$ and "output" species $Y_1, \ldots, Y_l$ such that, if $\mathcal{C}$ starts with $x_1, \ldots, x_k$ copies of $X_1, \ldots, X_k$, respectively, then with probability one, it reaches a "count-stable" configuration in which the counts of $Y_1, \ldots, Y_l$ are expressed by the vector $f(x_1, ..., x_k)$, and these counts never again change [21].

The method proposed by Chen et al. [21] uses some auxiliary "leader" species present initially, in addition to the input species. To illustrate their utility, suppose that we want to compute function $f(x) = x + 1$ with CRNs. Using the previous approach, we have an input species X (with initial count x), an output species Y (with initial count 0) and an auxiliary "leader" species L (with initial count 1). Fig 1.1a shows the reactions that compute $f(x)$.

However, it is experimentally difficult to prepare a solution with a single copy (or a small constant number) of a certain species. Chen et al. [21] asked whether it is possible to do away with the initial "leader" molecules, i.e., to require that the initial configuration contains initial counts $x_1, x_2, \ldots, x_k$ of input species $X_1, X_2, \ldots, X_k$, and initial count 0 of every other species. It is easy to "elect" a single leader molecule from an arbitrary initial number of copies using a reaction such as $L + L \to L$, which eventually reduces the count of L to 1. However, the problem with this approach is that, since L is a reactant in other reactions, there is no way in general to prevent L from participating in these reactions until the reaction $L + L \to L$ has reduced it

to a single copy.

Despite these difficulties, we answer the question affirmatively, showing that each semilinear function can be computed by a "leaderless" CRN, i.e., a CRN which starts with an initial configuration containing only the input species. To illustrate the question, consider the function $f(x) = x + 1$ once more. In order to compute the function without a leader (i.e., the initial configuration has x copies of X and 0 copies of every other species), the reactions in Figure 1.1b suffice: Reaction 1.1.1 produces x copies of B and $2x$ copies of Y. Reaction 1.1.2 consumes all copies of B except one, so reaction 1.1.2 executes precisely $x - 1$ times, producing $x - 1$ copies of K. Therefore reaction 1.1.3 consumes $x - 1$ copies of output species Y, eventually resulting in $2x - (x - 1) = x + 1$ copies of Y. Note that this approach uses a sort of leader election on the B molecules. We generalize this example and describe our leaderless CRN construction in Chapter 3.

We also refer the readers to Sections 1.3.1 for a more detailed discussion of our contributions.

1.2 The Approximate Majority Problem

Consider a mixture with n molecules, some of species X and the rest of species Y. Let x and y denote the number of copies of X and Y during a CRN computation. The *Approximate Majority* problem [9] is to reach *consensus* — a configuration in which all molecules are X ($x = n$) or all are Y ($y = n$), from an initial configuration in which $x + y = n$ and the gap $|x - y|$ is above some threshold. If initially $x > y$, the consensus should be X-majority ($x = n$), and if initially $y > x$ the consensus should be Y-majority. We focus on the case when initially $x > y$ since the CRNs that we analyze are symmetric with respect to X and Y.

Angluin et al. [12] proposed and analyzed the Single-B CRN of Figure 1.2(c) in the PP framework. Informally, reactions (0'x) and (0'y) are equally likely to produce B's (blanks) from X's or Y's respectively, while reactions (1') and (2') recruit B's to become X's and Y's respectively. When X is initially in the majority ($x > y$ initially), a reaction event is more likely to be (1') than (2'), with the bias towards (1') increasing as x gets larger. Angluin et al. showed *correctness*: if initially $x - y = \omega(\sqrt{n} \log n)$, then with high probability Single-B reaches X-majority consensus. They also showed *efficiency*: with "high" probability $1 - n^{-\Omega(1)}$, for any initial gap value $x - y$, Single-B reaches consensus within $O(n \log n)$ interactions. In addition, they proved correctness and efficiency in more general settings,

$$X + X + Y \to X + X + X \quad (1)$$
$$X + Y + Y \to Y + Y + Y \quad (2)$$

(a) Tri-molecular CRN

$$X + Y \to B + B \quad (0')$$
$$X + B \to X + X \quad (1')$$
$$Y + B \to Y + Y \quad (2')$$

(b) Double-B CRN

$$X + Y \overset{1/2}{\to} X + B \quad (0'x)$$
$$X + Y \overset{1/2}{\to} Y + B \quad (0'y)$$
$$X + B \to X + X \quad (1')$$
$$Y + B \to Y + Y \quad (2')$$

(c) Single-B CRN

Figure 1.2: A tri-molecular and two bi-molecular chemical reaction networks (CRNs) for Approximate Majority. Reactions (0'x) and (1'y) of Single-B have rate constant 1/2 while all other reactions have rate constant 1.

such as when some molecules in the initial configuration are B's, or in the presence of $o(\sqrt{n})$ Byzantine agents.

Surprisingly, while the Single-B CRN is quite simple, they provided a very complicated and lengthy proof and noted that "Unfortunately, while the protocol itself is simple, proving that it converges quickly appears to be very difficult". Motivated by their intriguing but complex approach, we aim to provide simpler and more intuitive proofs of correctness and efficiency, with the hope that simple techniques can be adapted to reason about CRNs for other problems. For this purpose, we analyze three different CRNs for Approximate Majority: a simple tri-molecular CRN, whose reactions involve just the two species X and Y that are present initially, and two bi-molecular CRNs, which we call Double-B and Single-B, that use an additional "blank" species B – see Figure 1.2. As noted earlier, the Single-B CRN is the same as that of Angluin et al. The Double-B CRN is symmetric even in the PP setting.

Several others have subsequently and independently studied the problem; Mertzios et al. [74] analyze Single-B when $x + y = n$ and $y \leq \epsilon n$. They use a biased random walk argument to show that for $\epsilon = 1/7$, Single-B

reaches consensus on X-majority with exponentially small error probability $1 - e^{-\Theta(n)}$. They conjecture that exponentially small probability holds if ϵ is any positive constant less than $1/2$. We show in Section 4.6 how to prove this conjecture using our analysis tools.

Perron et al. [81] also analyze Single-B when the initial gap is a constant fraction times n, showing correctness with all but exponentially small probability. Their results do not apply to smaller initial gaps.

Cruise and Ganesh [31] devise a family of protocols in network models where agents (nodes) can poll other agents in order to update their state. Their family of protocols provides a natural generalization of our tri-molecular CRN. Their analysis uses connections between random walks and electrical networks. We consider our proof of the tri-molecular CRN to be simpler than theirs, and we obtain high probability bounds on efficiency, while they reason about expected time.

Yet other work on Approximate Majority pertains to settings with different assumptions about the number of states per agent or with different interaction scheduling rules [5, 36, 74], or analyze more general multi-valued consensus problems [12, 14, 15].

Section 1.3.2 describes our contributions on the Approximate Majority problem.

1.3 Contributions

Here, we summarize our contributions for the subjects discussed in Sections 1.1 and 1.2.

1.3.1 Deterministic Computations with CRNs

In this regard, we mainly answer the question asked by Chen et al. [21], whether deterministic CRNs without a leader retain the same power, and make the following contributions.

1. We show that every semilinear function is deterministically computable by a CRN whose initial configuration contains only the input species $X_1, \ldots, X_k$, and zero counts of every other species, so long as $f(\mathbf{0}) = \mathbf{0}$.

2. We describe a leaderless CRN construction to compute any semilinear function. We use a similar framework to the construction of Chen et al. [21], decomposing the semilinear function into a finite union of affine partial functions (linear functions with an offset; defined formally

in Section 3.1). We show how to compute each affine function with leaderless CRNs, using a fundamentally different construction than the affine-function computing CRNs of Chen et al. [21].

3. We show that our direct stable leaderless computation of a semilinear function completes in expected time $O(n)$, where n is the total number of input molecules. This time bound is slower than the $O(\log^5 n)$ time bound achieved by Chen et al. [21], but faster than the $O(n \log n)$ bound achieved by the *direct* construction of Chen et al. [21] where both fast and direct constructions crucially use leaders.

1.3.2 Approximate Majority with CRNs

Our main contribution is that we provide the simplest and most intuitive proofs of correctness and efficiency for the CRNs listed in Figure 1.2 that compute Approximate Majority. Our detailed contributions are listed below.

1. We propose a new and simple tri-molecular CRN (see Figure 1.2(a)) for Approximate Majority.

2. We analyze the correctness and efficiency of the tri-molecular CRN using simple analysis tools such as biased random walks and Chernoff bounds.

3. We analyze the efficiency and correctness of bi-molecular CRNs of Figure 1.2 – Double-B and Single-B CRNs – by relating them to the tri-molecular CRN (see Section 4.3).

4. We show that the count of B (blank) species in Single-B CRN is tightly bounded by the count of Y species, assuming that initially $y < x$ and $b < n/2$, where n is the total count of molecules.

5. A bonus is that all of our results apply with high probability when the initial gap is $\Omega(\sqrt{n \log n})$, and thus are a factor of $\sqrt{\log n}$ stronger than Angluin et al.'s results [12]. However, we note that their results address a more general settings (such as starting from any arbitrary gap or tolerating $o(\sqrt{n})$ Byzantine agents) than ours and we suspect that the complexity of their approach may well be attributable to the difficulty of handling these cases that we do not address.

1.4 Outline

In Chapter 2 we provide the required background and preliminary definitions to understand chemical reaction networks kinetics. Next, Chapter 3 reports on our leaderless computation with CRNs. Section 3.1 first describes the CRN semantics for the deterministic computation. Second, Section 3.2 explains our leaderless deterministic construction to compute any semilinear function, along with its time complexity analysis. Then, in Chapter 4, we discuss the correctness and efficiency of our concerned Approximate Majority CRNs. We study the behaviour of the tri-molecular CRN in three different phases in Section 4.2. Later, we analyze the behaviour of Double-B and Single-B CRNs in Section 4.3.

Chapter 2

Background on Chemical Reaction Networks

Here, we present some preliminary concepts and definitions as needed to understand chemical reaction networks and their underlying kinetic model used throughout Chapters 3 and 4. We introduce additional concepts and definitions as needed in each corresponding chapter.

2.1 Chemical Reaction Networks

We model interacting molecules in a well-mixed solution, under fixed environmental conditions, such as temperature. Let $\Lambda = \{\Lambda_1, \Lambda_2, \ldots, \Lambda_m\}$ be a finite set of chemical species here and throughout. We write $\mathbb{N}^\Lambda$ to denote the set of vectors of $|\Lambda|$ nonnegative integers, with each coordinate "labeled" by a distinct element of Λ. Given $X \in \Lambda$ and $\mathbf{c} \in \mathbb{N}^\Lambda$, we refer to $\mathbf{c}(X)$ as the *count of* X *in* $\mathbf{c}$. We write $\mathbf{c} \leq \mathbf{c}'$ to denote that $\mathbf{c}(X) \leq \mathbf{c}'(X)$ for all $X \in \Lambda$. Given $\mathbf{c}, \mathbf{c}' \in \mathbb{N}^\Lambda$, we define the vector component-wise operations of addition $\mathbf{c}+\mathbf{c}'$, subtraction $\mathbf{c}-\mathbf{c}'$, and scalar multiplication $n\mathbf{c}$ for $n \in \mathbb{N}$. If $\Delta \subset \Lambda$, we view a vector $\mathbf{c} \in \mathbb{N}^\Delta$ equivalently as a vector $\mathbf{c} \in \mathbb{N}^\Lambda$ by assuming $\mathbf{c}(X) = 0$ for all $X \in \Lambda \setminus \Delta$.

Molecules can collide, or equivalently, *interact*. Given a finite set of chemical species Λ, we denote an interaction that simultaneously involves $\mathbf{r}_1$ copies of Λ_1, $\mathbf{r}_2$ copies of Λ_2, and so on by a vector $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m)$, and define the *order* of the interaction to be $r_1+r_2+\ldots+r_m$. Reactions with orders 1, 2, and 3 are called *unimolecular*, *bimolecular*, and *tri-molecular* respectively. In our research on CRNs, we assume that all interactions have the same order (either two or three). In fact, each interaction $\mathbf{r}$ has an associated rate constant $k_r > 0$. In our discussions in Chapters 3 and 4 we assume that all interaction rate constants k_r are 1. Some interactions are *non-reactive* (collisions resulting in no consumed or produced molecules), while others can trigger one or more *reactions*. A *reaction* over Λ is a triple $\alpha = \langle \mathbf{r}, \mathbf{p}, k_\alpha \rangle \in \mathbb{N}^\Lambda \times \mathbb{N}^\Lambda \times \mathbb{R}^+$, where $\mathbf{r} \neq \mathbf{p}$, specifying the stoichiome-

try of the reactants and products, respectively, and the *rate constant* k_α. The reaction rate constant determines the probability that the reaction is accomplished. We can write reaction α in the form

$$\alpha : r_1\Lambda_1 + r_2\Lambda_2 + \ldots + r_m\Lambda_m \xrightarrow{k_\alpha} p_1\Lambda_1 + p_2\Lambda_2 + \ldots + p_m\Lambda_m.$$

For instance, reaction $A + 2B \xrightarrow{1/2} A + 3C$ is a reaction with tuple $\langle(1,2,0), (1,0,3), \frac{1}{2}\rangle$.

A *(finite) chemical reaction network (CRN)* is a pair $\mathcal{C} = (\Lambda, R)$, where Λ is a finite set of chemical *species*, and R is a finite set of reactions over Λ, such that if $R(r)$ is the subset of R with reactant vector r, then $\sum_{\alpha \in R(r)} k_\alpha \leq 1 (= k_r)$. A *configuration* of a CRN $\mathcal{C} = (\Lambda, R)$ is a vector $\mathbf{c} \in \mathbb{N}^\Lambda$.

Given a configuration $\mathbf{c}$ and reaction $\alpha = \langle \mathbf{r}, \mathbf{p}, k_\alpha \rangle$, we say that α is *applicable* to $\mathbf{c}$ if $\mathbf{r} \leq \mathbf{c}$ (i.e., $\mathbf{c}$ contains enough of each of the reactants for the reaction to occur). If α is applicable to $\mathbf{c}$, then we write $\alpha(\mathbf{c})$ to denote the configuration $\mathbf{c} + \mathbf{p} - \mathbf{r}$ (i.e., the configuration that results from applying reaction α to $\mathbf{c}$). If $\mathbf{c}' = \alpha(\mathbf{c})$ for some reaction $\alpha \in R$, we write $\mathbf{c} \to_{\mathcal{C}} \mathbf{c}'$, or merely $\mathbf{c} \to \mathbf{c}'$ when $\mathcal{C}$ is clear from context.

2.2 Kinetic Model

The following model of stochastic chemical kinetics is widely used in quantitative biology and other fields dealing with chemical reactions between species present in small counts [42]. It ascribes probabilities to computation and execution sequences, and also defines the time of reactions, allowing us to study the computational complexity of the CRN computation in the following two chapters.

The kinetics of a CRN is described by a stochastic model of chemical kinetics [42, IIB] as follows. Given a fixed volume $v \in \mathbb{R}^+$ and current configuration $\mathbf{c} = (c_1, \cdots, c_m)$, where $c_1 = \mathbf{c}(\Lambda_1), c_2 = \mathbf{c}(\Lambda_2), \ldots, c_m = \mathbf{c}(\Lambda_m)$, the *propensity* of a particular interaction $\mathbf{r} = (r_1, r_2, \ldots, r_m)$ of order o is

$$\rho(\mathbf{c}, \mathbf{r}) = \left[\prod_{i=1}^{m} \binom{c_i}{r_i}\right] / v^{o-1}.$$

the propensity of reaction $\alpha = \langle \mathbf{r}, \mathbf{p}, k_\alpha \rangle$ is $\rho(\mathbf{c}, \alpha) = k_\alpha \rho(\mathbf{c}, \mathbf{r})$. For instance, the propensity of a unimolecular reaction $\alpha : X \xrightarrow{k_\alpha} \ldots$ in configuration $\mathbf{c}$ is $\rho(\mathbf{c}, \alpha) = k_\alpha \mathbf{c}(X)$. Likewise, the propensity of a bimolecular reaction $\alpha : X + Y \xrightarrow{k_\alpha} \ldots$, where $X \neq Y$, is $\rho(\mathbf{c}, \alpha) = k_\alpha \frac{\mathbf{c}(X)\mathbf{c}(Y)}{v}$ and the propensity

of a tri-molecular reaction $\alpha : 2X + Y \xrightarrow{k_\alpha} \ldots$ is $\rho(\mathbf{c}, \alpha) = k_\alpha \frac{\mathbf{c}(X)(\mathbf{c}(X)-1)\mathbf{c}(Y)}{2v^2}$.

Let $\rho(\mathbf{c}, R) = \sum_{\alpha \in R} \rho(\mathbf{c}, \alpha)$ be the sum of the propensities of all reactions applicable to configuration $\mathbf{c}$. When an interaction occurs, the probability that it is a reaction event is

$$\rho(\mathbf{c}, R)/\rho,$$

where $\rho = \sum_{\mathbf{r}} \rho(\mathbf{c}, \mathbf{r})$ is the sum of the propensities of all possible interactions $\mathbf{r}$ of orders $\{o_1, o_2, \ldots, o_l\}$ that are assumed to happen in a well-mixed solution. For example, if only interactions of orders $\{1, 2\}$ can happen in a solution with total molecular count n, then $\rho = \binom{n}{1}/v^0 + \binom{n}{2}/v^1$.
When a reaction event occurs, the probability that it is reaction $\alpha = \langle \mathbf{r}, \mathbf{p}, k_\alpha \rangle$ is

$$k_\alpha \rho(\mathbf{c}, \mathbf{r})/\rho(\mathbf{c}, R).$$

With respect to a given initial configuration $\mathbf{c}$, a *computation* is a finite or infinite sequence $I_1, I_2, \ldots$ where each I_i is either a non-reactive interaction or a reaction, chosen according to their probabilities. Formally, let $\mathbf{c}_0 = \mathbf{c}$. For $i \geq 1$, the probability that I_i is non-reactive interaction $\mathbf{r}$ is $\rho(\mathbf{c}_{i-1}, \mathbf{r})/\rho$, in which case we let $\mathbf{c}_i = \mathbf{c}_{i-1}$, and the probability that I_i is reaction $\alpha = (\mathbf{r}, \mathbf{p}, k_\alpha)$ is $k_\alpha \lambda(\mathbf{c}_{i-1}, \mathbf{r})/\rho$, in which case $\mathbf{c}_i$ is such that $\mathbf{c}_{i-1} \to \mathbf{c}_i$.

An *execution* (a.k.a., *execution sequence*) $\mathcal{E}$ is a finite or infinite subsequence of configurations $\mathbf{c}_0, \mathbf{c}_1, \ldots$ of the underlying *computation* where non-reactive interactions that result in no change of configuration are excluded. Therefore, for all $i \in \{1, \ldots, |\mathcal{E}|-1\}$, $\mathbf{c}_{i-1} \to \mathbf{c}_i$. If a finite execution sequence starts with $\mathbf{c}$ and ends with $\mathbf{c}'$, we write $\mathbf{c} \to^*_{\mathcal{C}} \mathbf{c}'$, or merely $\mathbf{c} \to^* \mathbf{c}'$ when the CRN $\mathcal{C}$ is clear from context. In this case, we say that $\mathbf{c}'$ is *reachable* from $\mathbf{c}$.

The propensity function then determines the evolution of the system as follows. At any moment, the *time* until the next interaction occurs is exponentially distributed with parameter ρ [42, IIIC] defined earlier. Then, the expected time until an interaction occurs is $\frac{1}{\rho}$. Accordingly, the expected time until the next reaction occurs is $1/\rho(c, R)$ (note that $\rho(\mathbf{c}, R) = 0$ if no reactions are applicable to $\mathbf{c}$).

The kinetic model is based on the physical assumption of well-mixedness in a diluted solution. Thus, we assume the *finite density constraint*, which stipulates that a volume required to execute a CRN must be proportional to the maximum molecular count obtained during execution [94]. In other words, the total concentration (molecular count per volume) is bounded. This realistically constrains the speed of the computation achievable by CRNs. We apply the kinetic model only to CRNs with configuration spaces

that are bounded for each start configuration, choosing the volume to be equal to the reachable configuration with the highest molecular count. In this thesis, the volume will always be within a constant multiplicative factor of the number of input molecules.

Chapter 3

Leaderless Deterministic Chemical Reaction Networks

In this chapter, we begin with preliminary mathematical definitions and different semantic interpretations of CRNs that are used throughout this chapter. We then provide our construction and expected time analysis to prove that a semilinear function is deterministically computed with leaderless CRNs. Finally, we summarize our results and discuss our contributions.

3.1 Preliminaries

Given a vector $\mathbf{x} \in \mathbb{N}^k$, let $\|\mathbf{x}\| = \|\mathbf{x}\|_1 = \sum_{i=1}^{k} |\mathbf{x}(i)|$, where $\mathbf{x}(i)$ denotes the ith coordinate of $\mathbf{x}$. A set $A \subseteq \mathbb{N}^k$ is *linear* if there exist vectors $\mathbf{b}, \mathbf{u}_1, \ldots, \mathbf{u}_p \in \mathbb{N}^k$ such that

$$A = \{\ \mathbf{b} + n_1\mathbf{u}_1 + \ldots + n_p\mathbf{u}_p \mid\ n_1, \ldots, n_p \in \mathbb{N}\ \}.$$

A is *semilinear* if it is a finite union of linear sets. If $f : \mathbb{N}^k \to \mathbb{N}^l$ is a function, define the *graph* of f to be the set $\{\ (\mathbf{x}, \mathbf{y}) \in \mathbb{N}^k \times \mathbb{N}^l \mid\ f(\mathbf{x}) = \mathbf{y}\ \}$. A function is *semilinear* if its graph is semilinear. A predicate $\phi : \mathbb{N}^k \to \{0, 1\}$ is *semilinear* if the set $\{\ \mathbf{x} \in \mathbb{N}^k \mid\ \phi(\mathbf{x}) = 1\ \}$ is a semilinear set.

We say a partial function $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$ is *affine* if there exist kl rational numbers $a_{1,1}, \ldots, a_{l,k} \in \mathbb{Q}$ and $l+k$ nonnegative integers $b_1, \ldots, b_l, c_1, \ldots, c_k \in \mathbb{N}$ such that, if $\mathbf{y} = f(\mathbf{x})$, then for each $j \in \{1, \ldots, l\}$, $\mathbf{y}(j) = b_j + \sum_{i=1}^{k} a_{j,i}(\mathbf{x}(i) - c_i)$, and for each $i \in \{1, \ldots, k\}$, $\mathbf{x}(i) - c_i \geq 0$. In matrix notation, there exist a $l \times k$ rational matrix $\mathbf{A}$ and vectors $\mathbf{b} \in \mathbb{N}^l$ and $\mathbf{c} \in \mathbb{N}^k$ such that $f(\mathbf{x}) = \mathbf{A}(\mathbf{x} - \mathbf{c}) + \mathbf{b}$.

This definition of affine function may appear contrived; see [21] for an explanation of its various intricacies. For reading this chapter, the main utility of the definition is that it satisfies Lemma 3.2.2.

Note that by appropriate integer arithmetic, a partial function $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$ is affine if and only if there exist kl integers $n_{1,1}, \ldots, n_{k,l} \in \mathbb{Z}$ and $2l + k$ nonnegative integers $b_1, \ldots, b_l, c_1, \ldots, c_k, d_1, \ldots, d_l \in \mathbb{N}$ such that, if $\mathbf{y} =$

$f(\mathbf{x})$, then for each $j \in \{1, \ldots, l\}$, $\mathbf{y}(j) = b_j + \frac{1}{d_j} \sum_{i=1}^{k} n_{i,j}(\mathbf{x}(i) - c_i)$, and for each $i \in \{1, \ldots, k\}$, $\mathbf{x}(i) - c_i \geq 0$. Each d_j may be taken to be the least common multiple of the denominators of the rational coefficients in the original definition. We employ this latter definition, since it is more convenient for working with integer-valued molecular counts.

In this chapter, we use CRNs to decide subsets of $\mathbb{N}^k$ (for which we reserve the term "chemical reaction *decider*" or CRD) and to compute functions $f : \mathbb{N}^k \to \mathbb{N}^l$ (for which we reserve the term "chemical reaction *computer*" or CRC). In the next two subsections we define two semantic interpretations of CRNs that correspond to these two tasks. We use the term CRN to refer to either a CRD or CRC when the statement is applicable to either type.

These definitions differ slightly from those of Chen et al. [21], because ours are specialized to "leaderless" CRNs: those that can compute a predicate or function in which no species are present in the initial configuration other than the input species. In the terminology of Chen et al. [21], a CRN with species set Λ and input species set Σ is *leaderless* if it has an *initial context* $\sigma : \Lambda \to \mathbb{N}$ such that $\sigma(S) = 0$ for all $S \in \Lambda \setminus \Sigma$. The definitions below are simplified by assuming this to be true of all CRNs.

We also use the convention of Angluin, Aspnes, and Eisenstat [10] that for a CRD, all species "vote" yes or no, rather than only a subset of species as in [21], since this convention is convenient for proving time bounds.

3.1.1 Stable Decidability of Predicates

We now review the definition of stable decidability of predicates introduced by Angluin, Aspnes, and Eisenstat [10].[2] Intuitively, the set of species is partitioned into two sets: those that "vote" yes and those that vote no, and the system stabilizes to an output when a consensus vote is reached (all positive-count species have the same vote) that can no longer be changed (no species voting the other way can ever again be produced). It would be too strong to characterize deterministic correctness by requiring all possible executions to achieve the correct answer; for example, a reversible reaction such as $A \rightleftharpoons B$ could simply be chosen to run back and forth forever, starving any other reactions. In the more refined definition that follows, the

[2]Those authors use the term "stably *compute*", but we reserve the term "compute" for the computation of non-Boolean functions. Also, we omit discussion of the definition of stable computation used in the population protocols literature, which employs a notion of "fair" executions; the definitions are proven equivalent in [21].

determinism of the system is captured in that it is impossible to stabilize to an incorrect answer, and the correct stable output is always reachable.

A *(leaderless) chemical reaction decider* (CRD) is a tuple $\mathcal{D} = (\Lambda, R, \Sigma, \Upsilon)$, where (Λ, R) is a CRN, $\Sigma \subseteq \Lambda$ is the *set of input species*, and $\Upsilon \subseteq \Lambda$ is the set of *yes voters*, with species in $\Lambda \setminus \Upsilon$ referred to as *no voters*. An input to $\mathcal{D}$ will be an *initial configuration* $\mathbf{i} \in \mathbb{N}^{\Sigma}$ (equivalently, $\mathbf{i} \in \mathbb{N}^k$ if we write $\Sigma = \{X_1, \ldots, X_k\}$ and assign X_i to represent the i'th coordinate); that is, only input species are allowed to be non-zero. If we are discussing a CRN understood from context to have a certain initial configuration $\mathbf{i}$, we write $\#_0 X$ to denote $\mathbf{i}(X)$.

We define a global output partial function $\Phi : \mathbb{N}^{\Lambda} \dashrightarrow \{0, 1\}$ as follows. $\Phi(\mathbf{c})$ is undefined if either $\mathbf{c} = \mathbf{0}$, or if there exist $S_0 \in \Lambda \setminus \Upsilon$ and $S_1 \in \Upsilon$ such that $\mathbf{c}(S_0) > 0$ and $\mathbf{c}(S_1) > 0$. Otherwise, either $(\forall S \in \Lambda)(\mathbf{c}(S) > 0 \implies S \in \Upsilon)$ or $(\forall S \in \Lambda)(\mathbf{c}(S) > 0 \implies S \in \Lambda \setminus \Upsilon)$; in the former case, the *output* $\Phi(\mathbf{c})$ of configuration $\mathbf{c}$ is 1, and in the latter case, $\Phi(\mathbf{c}) = 0$.

A configuration $\mathbf{o}$ is *output stable* if $\Phi(\mathbf{o})$ is defined and, for all $\mathbf{c}$ such that $\mathbf{o} \to^* \mathbf{c}$, $\Phi(\mathbf{c}) = \Phi(\mathbf{o})$. We say a CRD $\mathcal{D}$ *stably decides* the predicate $\psi : \mathbb{N}^{\Sigma} \to \{0, 1\}$ if, for any initial configuration $\mathbf{i} \in \mathbb{N}^k$, for all configurations $\mathbf{c} \in \mathbb{N}^{\Lambda}$, $\mathbf{i} \to^* \mathbf{c}$ implies $\mathbf{c} \to^* \mathbf{o}$ such that $\mathbf{o}$ is output stable and $\Phi(\mathbf{o}) = \psi(\mathbf{i})$. Note that this condition implies that no incorrect output stable configuration is reachable from $\mathbf{i}$. We say that $\mathcal{D}$ *stably decides* a set $A \in \mathbb{N}^k$ if it stably decides its indicator function.

The following theorem is due to Angluin, Aspnes, and Eisenstat [10]:

Theorem 3.1.1 ([10])**.** *A set $A \subseteq \mathbb{N}^k$ is stably decidable by a CRD if and only if it is semilinear.*

The model they use is defined in a slightly different way; the differences (and those differences' lack of significance to the questions we explore) are explained in Chen et al. [21].

3.1.2 Stable Computation of Functions

We now define a notion of stable computation of *functions* similar to those above for predicates. Intuitively, the inputs to the function are the initial counts of input species $X_1, \ldots, X_k$, and the outputs are the counts of output species $Y_1, \ldots, Y_l$. The system stabilizes to an output when the counts of the output species can no longer change. Again determinism is captured in that it is impossible to stabilize to an incorrect answer and the correct stable output is always reachable.

A *(leaderless) chemical reaction computer (CRC)* is a tuple $\mathcal{C} = (\Lambda, R, \Sigma, \Gamma)$, where (Λ, R) is a CRN, $\Sigma \subset \Lambda$ is the *set of input species*, $\Gamma \subset \Lambda$ is the *set of output species*, such that $\Sigma \cap \Gamma = \varnothing$. By convention, we let $\Sigma = \{X_1, X_2, \ldots, X_k\}$ and $\Gamma = \{Y_1, Y_2, \ldots, Y_l\}$. We say that a configuration $\mathbf{o}$ is *output stable* if, for every $\mathbf{c}$ such that $\mathbf{o} \to^* \mathbf{c}$ and every $Y_i \in \Gamma$, $\mathbf{o}(Y_i) = \mathbf{c}(Y_i)$ (i.e., the counts of species in Γ will never change if $\mathbf{o}$ is reached). As with CRD's, we require initial configurations $\mathbf{i} \in \mathbb{N}^\Sigma$ in which only input species are allowed to be positive. We say that $\mathcal{C}$ *stably computes* a function $f : \mathbb{N}^k \to \mathbb{N}^l$ if for any initial configuration $\mathbf{i} \in \mathbb{N}^\Sigma$, $\mathbf{i} \to^* \mathbf{c}$ implies there exists $\mathbf{o}$ such that $\mathbf{c} \to^* \mathbf{o}$ and $\mathbf{o}$ is an output stable configuration with $f(\mathbf{i}) = (\mathbf{o}(Y_1), \mathbf{o}(Y_2), \ldots, \mathbf{o}(Y_l))$. Note that this condition implies that no incorrect output stable configuration is reachable from $\mathbf{i}$.

If a CRN stably decides a predicate or stably computes a function, we say the CRN is *stable* (a.k.a., *deterministic*).

If $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$ is a partial function undefined on some inputs, we say that a CRC $\mathcal{C}$ stably computes f if $\mathcal{C}$ stably computes f on all inputs $\mathbf{x} \in \operatorname{dom} f$, with no constraint on the behavior of $\mathcal{C}$ if it is given an input $\mathbf{x} \notin \operatorname{dom} f$.

3.1.3 Kinetic Model

Here, we employ the same kinetic model described in Section 2.2. However, we assume that the rate constants of all reactions are 1, and we adjust the model with this assumption. The reaction rate constants do not affect the definition of stable computation; they only affect the expected time analysis. Our expected time analyses remain asymptotically unaffected if the rate constants are changed (although the constants hidden in the big-O notation would change). Intuitively, we can assume that in Lemmas 3.1.2, 3.1.3, and 3.1.4, i.e., the main lemmas to analyze the expected time in this chapter, all the rate constants are replaced by the smallest one which only scales down the total propensity of reactions by a constant factor.

Given CRN $\mathcal{C} = (\Lambda, R)$ and configuration $\mathbf{c}$, recall from Section 2.2 that the expected time until the next reaction occurs is $\frac{1}{\rho(\mathbf{c}, R)}$.

Moreover, for the reactions in R', where $R' \subset R$, assuming that the reactions in $R \setminus R'$ either do not affect the reactants of reactions in R', or that if they do, those reactions are not applicable, then the expected time until the next reaction in R' occurs is $\frac{1}{\rho(\mathbf{c}, R')}$. This observation allows us to analyze different independent components of a CRN as if they were their own CRN, so long as they either run in parallel without affecting each others' reactants, or they run in series, but under the assumption that one

component has finished reacting (and therefore cannot affect reactant counts for the next component).

It is not difficult to show that if a CRN is stable and has a finite reachable configuration space from any initial configuration $\mathbf{i}$, then under the kinetic model (in fact, for any choice of rate constants), with probability 1 the CRN will eventually reach an output stable configuration [21].

We require the following lemmas later in our main theorems. Some of these are implicit or explicit in many earlier papers on stochastic CRNs, but we include their proofs for the sake of self-containment.

The lemmas are stated with respect to a certain "initial configuration" $\mathbf{c}$ that may not be the initial configuration of an actual CRN we define. This is because the lemmas are employed to argue about CRNs that are guaranteed to evolve to some configuration $\mathbf{c}$ that satisfies the hypothesis of the lemma, and we use the lemma to bound the expected time it takes for the CRN to complete a sequence of reactions, starting from $\mathbf{c}$. Therefore terms such as "applicable reaction" refer to being applicable from $\mathbf{c}$ and any configuration reachable from it, although some additional inapplicable reactions may have been applicable prior to reaching the configuration $\mathbf{c}$. We note that, from now on, if current configuration $\mathbf{c}$ is understood from the context, we may write $\#X$ to denote $\mathbf{c}(X)$.

Lemma 3.1.2. Let $\mathbf{c}$ be a configuration. Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a set of species with the property that, for all configurations reachable from $\mathbf{c}$, every applicable reaction in which any species in $\mathcal{A}$ appears is of the form $A_i \to B_1 + \ldots + B_l$, where each $B_{i'} \notin \mathcal{A}$ for $1 \leq i' \leq l$. Then starting from a configuration $\mathbf{c}$, in which $S = \sum_{i=1}^{m} \mathbf{c}(A_i) \leq L$, the expected time to reach from $\mathbf{c}$ to a configuration in which all A_i's disappear is $O(\log L)$.

Proof. Assume the hypothesis. After each relevant reaction occurs, the sum S is reduced by 1. Therefore no reactions can occur after (at most) L reactions have executed. If $\sum_{i=1}^{m} \#A_i = k$ (i.e., the sum of the propensities of each possible reaction) in an arbitrary configuration reachable from $\mathbf{c}$, the expected time for any reaction to occur is $\frac{1}{k}$. By linearity of expectation, the expected time for L reactions to execute is at most $\sum_{k=1}^{L} \frac{1}{k} = O(\log L)$. □

Lemma 3.1.3. Let $\mathbf{c}$ be a configuration. Let $\mathcal{A} = \{A_1, \ldots, A_m\}$ be a set of species with the property that, for all configurations reachable from $\mathbf{c}$, every applicable reaction in which any species in $\mathcal{A}$ appears is of the form $A_i + A_j \to A_p + B_1 + \ldots + B_l$, where each $B_{i'} \notin \mathcal{A}$ for $1 \leq i' \leq l$, and for all $i, j \in \{1, \ldots, m\}$, there is at least one reaction $A_i + A_j \to \ldots$ of this form. Then starting from a configuration $\mathbf{c}$ in which $S = \sum_{i=1}^{m} \mathbf{c}(A_i) \leq L$, with

volume $O(L)$, the expected time to reach a configuration in which none of the described reactions can occur is $O(L)$.

Proof. Assume the hypothesis. Let $c' \in \mathbb{N}$ be a constant such that the volume is at most $c'L$. After each relevant reaction occurs, the sum S is reduced by 1. Therefore no reactions can occur after (at most) $L-1$ reactions have executed. Now let $\rho(\mathbf{c}, \alpha_{ij})$ be the propensity of the reaction $A_i + A_j \to A_p + B_1 + \ldots + B_l$, which is equal to $\rho(\mathbf{c}, \alpha_{ji})$ as well, and if there is more than one reaction of that form, let $\rho(\mathbf{c}, \alpha_{ij})$ represent the rate of one of those reactions selected arbitrarily. Since A_i can react with A_j for any $i, j \in \{1, \ldots, m\}$, given that $\sum_{i=1}^{m} \#A_i = k$ in an arbitrary configuration reachable from $\mathbf{c}$, the expected time for the next reaction to occur is inversely proportional to the sum of the propensities of each possible reaction, i.e.,

$$\begin{aligned}
\sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \geq i}}^{m} \rho(\mathbf{c}, \alpha_{ij}) &= \frac{1}{2} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} \rho(\mathbf{c}, \alpha_{ij}) + \sum_{i=1}^{m} \rho(\mathbf{c}, \alpha_{ii}) \\
&= \frac{1}{2} \sum_{i=1}^{m} \sum_{\substack{j=1 \\ j \neq i}}^{m} \frac{\#A_i \#A_j}{c'L} + \sum_{i=1}^{m} \frac{\#A_i(\#A_i - 1)}{2c'L} \\
&= \frac{1}{2c'L} \left[\sum_{i=1}^{m} \sum_{j=1}^{m} \#A_i \#A_j - \sum_{i=1}^{m} \#A_i^2 \right] + \frac{1}{2c'L} \sum_{i=1}^{m} \#A_i(\#A_i - 1) \\
&= \frac{1}{2c'L} \left[\sum_{i=1}^{m} \#A_i \left(\sum_{j=1}^{m} \#A_j \right) - \sum_{i=1}^{m} \#A_i \right] \\
&= \frac{1}{2c'L}(k^2 - k)
\end{aligned}$$

so the expected time for the next reaction to occur is $\frac{2c'L}{k^2-k}$. By linearity of expectation, the expected time for (at most) $L-1$ reactions to execute is at most $\sum_{k=1}^{L-1} \frac{2c'L}{k^2-k} = 2c'L \sum_{k=1}^{L-1} (\frac{1}{k-1} - \frac{1}{k}) = 2c'L(1 - \frac{1}{L-1}) = O(L)$. □

Lemma 3.1.4. Let $\mathbf{c}$ be a configuration. Let $\mathcal{C} = \{C_1, \ldots, C_p\}$ and $\mathcal{A} = \{A_1, \ldots, A_m\}$ be two sets of species with the property that, for all configurations reachable from $\mathbf{c}$, every applicable reaction in which any species in $\mathcal{A}$ or $\mathcal{C}$ appears is of the form $C_i + A_j \to C_i + B_1 + \ldots + B_l$, where each $B_{i'} \notin \mathcal{A}$ for $1 \leq i' \leq l$. Then starting from a configuration $\mathbf{c}$ in which for all $i \in \{1, \ldots, p\}$, $\mathbf{c}(C_i) = \Omega(L)$, and $S = \sum_{i=1}^{m} \mathbf{c}(A_i) \leq L$, with volume $O(L)$, the expected time to reach a configuration in which all A_i's disappear

is $O(\log L)$.

Proof. Assume the hypothesis. Then the counts of each C_i do not decrease. (They may increase if some $B_l \in \mathcal{C}$, but this only strengths the conclusion.) Therefore this is similar to the proof of Lemma 3.1.2, since for each k, the expected time until the next reaction occurs when $\sum_{j=1}^{m} \#A_j = k$, in an arbitrary configuration reachable from $\mathbf{c}$, is within a constant of $\frac{1}{k}$. Thus by linearity of expectation, the expected time for (at most) L (i.e., $\sum_{i=1}^{m} \mathbf{c}(A_i) \leq L$) reactions to occur is at most $\sum_{k=1}^{L} \frac{1}{k} = O(\log L)$. □

3.2 Leaderless CRCs can Compute Semilinear Functions

To supply an input vector $\mathbf{x} \in \mathbb{N}^k$ to a CRN, we use an initial configuration with $\mathbf{x}(i)$ molecules of input species X_i. Throughout this section, we let $n = ||\mathbf{x}||_1 = \sum_{i=1}^{k} \mathbf{x}(i)$ denote the initial number of molecules in solution. Since all CRNs we employ have the property that they produce at most a constant multiplicative factor more molecules than are initially present, this implies that the volume required to satisfy the finite density constraint is $O(n)$.

Suppose the CRC $\mathcal{C}$ stably computes a function $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$. We say that $\mathcal{C}$ stably computes f *monotonically* if its output species are not consumed in any reaction.[3]

We show in Lemma 3.2.1 that affine partial functions can be computed in expected time $O(n)$ by a leaderless CRC. For its use in proving Theorem 3.2.4, we require that the output molecules be produced monotonically. If we used a direct encoding of the output of the function, this would be impossible for general affine functions. For example, consider the function $f(x_1, x_2) = x_1 - x_2$ where dom $f = \{ (x_1, x_2) \mid x_1 \geq x_2 \}$. By withholding a single copy of X_2 and letting the CRC stabilize to the output value $\#Y = x_1 - x_2 + 1$, then allowing the extra copy of X_2 to interact, the only way to stabilize to the correct output value $x_1 - x_2$ is to consume a copy of the output species Y. Therefore Lemma 3.2.1 computes f indirectly via an encoding of f's output that allows monotonic production of outputs, encoding the output value $\mathbf{y}(j)$ as the difference between the counts of two

[3]Its output species could potentially be reactants so long as they are catalytic, meaning that the stoichiometry of the species as a product is at least as great as its stoichiometry as a reactant, e.g. if Y is the output species, $X + Y \to Z + Y$ or $A + Y \to Y + Y$.

monotonically produced species Y_j^P and Y_j^C, a concept formalized by the following definition.

Let $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$ be a partial function. We say that a partial function $\hat{f} : \mathbb{N}^k \dashrightarrow \mathbb{N}^l \times \mathbb{N}^l$ is a *diff-representation* of f if dom $f =$ dom $\hat{f}$ and, for all $\mathbf{x} \in$ dom f, if $(\mathbf{y}_P, \mathbf{y}_C) = \hat{f}(\mathbf{x})$, where $\mathbf{y}_P, \mathbf{y}_C \in \mathbb{N}^l$, then $f(\mathbf{x}) = \mathbf{y}_P - \mathbf{y}_C$, and $\mathbf{y}_P = O(f(\mathbf{x}))$.[4] In other words, $\hat{f}$ represents f as the difference of its two outputs $\mathbf{y}_P$ and $\mathbf{y}_C$, with the larger output $\mathbf{y}_P$ possibly being larger than the original function's output, but is at most by a multiplicative constant larger.

The following lemma is the main technical result required for proving our main theorem, Theorem 3.2.4. It shows that every affine function can be computed (via a diff-representation) in time $O(n)$ by a leaderless CRC. The example in Figure 3.1 also clarifies the essence of the leaderless computation of affine functions.

Lemma 3.2.1. *Let $f : \mathbb{N}^k \dashrightarrow \mathbb{N}^l$ be an affine partial function with $f(\mathbf{0}) = \mathbf{0}$ if $\mathbf{0} \in$ dom f. Then there is a diff-representation $\hat{f} : \mathbb{N}^k \dashrightarrow \mathbb{N}^l \times \mathbb{N}^l$ of f and a leaderless CRC that monotonically stably computes $\hat{f}$ in expected time $O(n)$.*

Proof. If f is affine, then there exist kl integers $n_{1,1}, \ldots, n_{k,l} \in \mathbb{Z}$ and $2l + k$ nonnegative integers $b_1, \ldots, b_l, c_1, \ldots, c_k, d_1, \ldots, d_l \in \mathbb{N}$ such that, if $\mathbf{y} = f(\mathbf{x})$, then for each $j \in \{1, \ldots, l\}$, $\mathbf{y}(j) = b_j + \frac{1}{d_j} \sum_{i=1}^{k} n_{i,j}(\mathbf{x}(i) - c_i)$, and for each $i \in \{1, \ldots, k\}$, $\mathbf{x}(i) - c_i \geq 0$. Note in particular that since the range of f is $\mathbb{N}^l$, the value $b_j + \frac{1}{d_j} \sum_{i=1}^{k} n_{i,j}(\mathbf{x}(i) - c_i)$ must be an integer.

Define the CRC as follows. It has input species $\Sigma = \{X_1, \ldots, X_k\}$ and output species $\Gamma = \{Y_1^P, \ldots, Y_l^P, Y_1^C, \ldots, Y_l^C\}$.

There are three main components of the CRN, separately handling the c_i offset, the $n_{i,j}/d_j$ coefficient, and the b_j offset.

For a species S that stabilizes to a fixed count depending only on the input configuration, write $\#_\infty S$ to denote the eventual stable count of S (in the case of Y_j^P and Y_j^C, this will be the same as the total amount ever produced, since they are never consumed). The latter two components both make use of Y_j^C molecules to account for production of Y_j^P molecules in excess of $\mathbf{y}(j)$ to ensure that $\#_\infty Y_j^P - \#_\infty Y_j^C = \mathbf{y}(j)$, which establishes that the CRC stably computes a diff-representation of f. It is clear by inspection of the reactions that $\#_\infty Y_j^P = O(\mathbf{y}(j))$.

[4] By $\mathbf{y}_P = O(f(\mathbf{x}))$, we mean that there is a constant c such that $y_P \leq cf(\mathbf{x})$ for all $\mathbf{x} \in \mathbb{N}^k$.

For all input species X_i ($1 \leq i \leq k$), add the reaction

$$X_i \to C_{i,1} + B_1 + B_2 + \ldots + B_l + b_1 Y_1^P + b_2 Y_2^P + \ldots b_l Y_l^P \tag{3.2.1}$$

The first product $C_{i,1}$ will be used to handle the c_i offset, and the remaining products will be used to handle the b_j offsets. By Lemma 3.1.2, reaction (3.2.1) takes time $O(\log n)$ to complete.

We now describe the three components of the CRC separately.

c_i **offset:** Reaction (3.2.1) produces $\mathbf{x}(i)$ copies of $C_{i,1}$. We must reduce this number by c_i, producing $\mathbf{x}(i) - c_i$ copies of X_i', the species that will be used by the next component to handle the $n_{i,j}/d_j$ coefficient. A high-order reaction implementing this is $(c_i+1)C_{i,1} \to c_i C_{i,1} + X_i'$, since that reaction will eventually happen exactly $\mathbf{x}(i) - c_i$ times (stopping when $\#C_{i,1}$ reaches c_i). This is implemented by the following bimolecular reactions.

For each $i \in \{1, \ldots, k\}$ and $m, p \in \{1, \ldots, c_i\}$, if $m + p \leq c_i$, add the reaction

$$C_{i,m} + C_{i,p} \to C_{i,m+p}.$$

If $m + p > c_i$, add the reaction

$$C_{i,m} + C_{i,p} \to C_{i,c_i} + (m + p - c_i)X_i'.$$

By Lemma 3.1.3, these reactions complete in expected time $O(n)$.

Note that although the final reaction above may produce a large number if X_i' has a large number of products, it is straightforward to simulate any such reaction with products $P_1, \ldots, P_\ell$ with reactions having two products only, e.g., the first product is P_1', followed by reactions $P_i' \to P_{i+1}' + P_i$ for each $i \in \{1, \ldots, \ell - 2\}$, and $P_{\ell-1}' \to P_{\ell-1} + P_\ell$.

$n_{i,j}/d_j$ **coefficient:** For each $i \in \{1, \ldots, k\}$, add the reaction

$$X_i' \to X_{i,1} + X_{i,2} + \ldots + X_{i,l}$$

This allows each output to be associated with its own copy of the input. By Lemma 3.1.2, these reactions complete in expected time $O(\log n)$.

For each $i \in \{1, \ldots, k\}$ and $j \in \{1, \ldots, l\}$, add the reaction

$$X_{i,j} \to \begin{cases} n_{i,j} D_{j,1}^P, & \text{if } n_{i,j} > 0; \\ (-n_{i,j}) D_{j,1}^C, & \text{if } n_{i,j} < 0. \end{cases}$$

By Lemma 3.1.2, these reactions complete in expected time $O(\log n)$.

We must now divide $\#D^P_{j,1}$ and $\#D^C_{j,1}$ by d_j. This is accomplished by the high-order reactions $d_j D^P_{j,1} \to Y^P_j$ and $d_j D^C_{j,1} \to Y^C_j$. Similarly to the previous component, we implement these with the following reactions for $d_j \geq 1$.

We first handle the case $d_j > 1$. For each $j \in \{1, \ldots, l\}$ and $m, p \in \{1, \ldots, d_j - 1\}$, if $m + p \leq d_j - 1$, add the reactions

$$\begin{aligned} D^P_{j,m} + D^P_{j,p} &\to D^P_{j,m+p} \\ D^C_{j,m} + D^C_{j,p} &\to D^C_{j,m+p} \end{aligned}$$

If $m + p > c_i$, add the reactions

$$\begin{aligned} D^P_{j,m} + D^P_{j,p} &\to D^P_{j,m+p-d_j} + Y^P_j \\ D^C_{j,m} + D^C_{j,p} &\to D^C_{j,m+p-d_j} + Y^C_j \end{aligned}$$

By Lemma 3.1.3, these reactions complete in expected time $O(n)$.

When $d_j = 1$, we only have the following unimolecular reactions.

$$\begin{aligned} D^P_{j,1} &\to Y^P_j \\ D^C_{j,1} &\to Y^C_j \end{aligned}$$

By Lemma 3.1.2, these reactions complete in expected time $O(\log n)$.

These reactions will produce $\frac{1}{d_j} \sum_{n_{i,j}>0} n_{i,j}(\mathbf{x}(i) - c_i)$ copies of Y^P_j and $-\frac{1}{d_j} \sum_{n_{i,j}<0} n_{i,j}(\mathbf{x}(i) - c_i)$ copies of Y^C_j. Therefore, letting $\#_{\text{coef}} Y^P_j$ and $\#_{\text{coef}} Y^C_j$ denote the number of copies of Y^P_j and Y^C_j eventually produced just by this component, it holds that $\#_{\text{coef}} Y^P_j - \#_{\text{coef}} Y^C_j = \frac{1}{d_j} \sum_{i=1}^{k} n_{i,j}(\mathbf{x}(i) - c_i)$.

b_j **offset:** For each $j \in \{1, \ldots, l\}$, add the reaction

$$B_j + B_j \to B_j + b_j Y^C_j \tag{3.2.2}$$

By Lemma 3.1.3, these reactions complete in expected time $O(n)$.

Reaction (3.2.1) produces b_j copies of Y^P_j for each copy of B_j produced, which is $\sum_{i=1}^{k} \mathbf{x}(i)$. Reaction (3.2.2) occurs precisely $(\sum_{i=1}^{k} \mathbf{x}(i)) - 1$ times. Therefore reaction (3.2.2) produces precisely b_j fewer copies of Y^C_j than reaction (3.2.1) produces of Y^P_j. This implies that when all

$$\mathbf{Y} = \mathbf{f}(\mathbf{X}) = \frac{\mathbf{3}}{\mathbf{2}}(\mathbf{X} - \mathbf{5}) + \mathbf{4}$$

- **Initial configuration:** $\#X = n, 0$ any other species
- **Reactions:**

make copies for parallel computations: $X \to C + B + 4Y$ $\#\mathbf{C}=\#\mathbf{B}=\mathbf{n},\#\mathbf{Y}=\mathbf{4}n$

1) *offset* 5:

$(5+1)C \to 5C + X'$ $\#\mathbf{C}=\mathbf{5},\#\mathbf{X'}=\mathbf{n}-\mathbf{5}$

2) *coefficient* $\frac{3}{2}$:

$$\begin{cases} X' \to 3D & \#\mathbf{D}=\mathbf{3}(\mathbf{n}-\mathbf{5}) \\ 2D \to Y & \#\mathbf{Y}=\frac{\mathbf{3}}{\mathbf{2}}(\mathbf{n}-\mathbf{5}) \end{cases}$$

3) *offset* 4:

$B + B \to B + 4Y^C$ $\#\mathbf{B}=\mathbf{1},\#\mathbf{Y^C}=\mathbf{4(n-1)}$

4) *combining steps 1, 2, and 3*:

$Y + Y^C \to \varnothing$

$\#\mathbf{Y}=\frac{\mathbf{3}}{\mathbf{2}}(\mathbf{n}-\mathbf{5})+\mathbf{4n}-\mathbf{4}(\mathbf{n}-\mathbf{1})=\frac{\mathbf{3}}{\mathbf{2}}(\mathbf{n}-\mathbf{5})+\mathbf{4}$

Figure 3.1: An example illustrating of the essential steps of our leaderless construction for the deterministic computation of partial affine functions – Lemma 3.2.1.

copies of Y_j^C are eventually produced by reaction (3.2.2), the number of Y_j^P's produced by reaction (3.2.1) minus the number of Y_j^C's produced by reaction (3.2.2) is b_j.

□

We require the following lemma, proven in Chen et al. [21].

Lemma 3.2.2 ([21])**.** *Let $f : \mathbb{N}^k \to \mathbb{N}^l$ be a semilinear function. Then there is a finite set $\{f_1 : \mathbb{N}^k \dashrightarrow \mathbb{N}^l, \ldots, f_m : \mathbb{N}^k \dashrightarrow \mathbb{N}^l\}$ of affine partial functions, where each* dom f_i *is a linear set, such that, for each $\mathbf{x} \in \mathbb{N}^k$, if $f_i(\mathbf{x})$ is defined, then $f(\mathbf{x}) = f_i(\mathbf{x})$, and $\bigcup_{i=1}^m$ dom $f_i = \mathbb{N}^k$.*

We require the following theorem, due to Angluin, Aspnes, and Eisenstat [11, Theorem 5], which states that any semilinear predicate can be

decided by a CRD in expected time $O(n)$.

Theorem 3.2.3 ([11])**.** *Let $\phi : \mathbb{N}^k \to \{0,1\}$ be a semilinear predicate. Then there is a leaderless CRD $\mathcal{D}$ that stably decides ϕ (so long as some positive number of molecules are initially present), and the expected time to reach an output-stable configuration is $O(n)$.*

The following is the main theorem of our work in this chapter. It shows that semilinear functions can be computed by leaderless CRCs in linear expected time.

Theorem 3.2.4. *Let $f : \mathbb{N}^k \to \mathbb{N}^l$ be a semilinear function with $f(\mathbf{0}) = \mathbf{0}$[5]. Then there is a leaderless CRC that stably computes f in expected time $O(n)$.*

Proof. The CRC will have input species $\Sigma = \{X_1, \ldots, X_k\}$ and output species $\Gamma = \{Y_1, \ldots, Y_l\}$. By Lemma 3.2.2, there is a finite set $F = \{f_1 : \mathbb{N}^k \dashrightarrow \mathbb{N}^l, \ldots, f_m : \mathbb{N}^k \dashrightarrow \mathbb{N}^l\}$ of affine partial functions, where each dom f_i is a linear set, such that, for each $\mathbf{x} \in \mathbb{N}^k$, if $f_i(\mathbf{x})$ is defined, then $f(\mathbf{x}) = f_i(\mathbf{x})$. We compute f on input $\mathbf{x}$ as follows. Since each dom f_i is a linear (and therefore semilinear) set, by Theorem 3.2.3 we compute each semilinear predicate $\phi_i =$ "$\mathbf{x} \in$ dom f_i and $(\forall i' \in \{1, \ldots, i-1\})$ $\mathbf{x} \notin$ dom $f_{i'}$?" by separate parallel CRD's each stabilizing in expected time $O(n)$. (The latter condition ensures that for each $\mathbf{x}$, precisely one of the predicates is true, in case the domains of the partial functions have nonempty intersection.) Here we are relying on the fact that Boolean combinations (union, intersection, complement) of semilinear sets are semilinear [43].

By Lemma 3.2.1, for each $i \in \{1, \ldots, m\}$, there is a diff-representation $\hat{f}_i$ of f_i that can be stably computed by parallel CRC's. Assume that for each $i \in \{1, \ldots, m\}$ and each $j \in \{1, \ldots, l\}$, the jth pair of outputs $\mathbf{y}_P(j)$ and $\mathbf{y}_C(j)$ of the ith function is represented by species $\hat{Y}^P_{i,j}$ and $\hat{Y}^C_{i,j}$. We interpret each $\hat{Y}^P_{i,j}$ and $\hat{Y}^C_{i,j}$ as an "inactive" version of "active" output species $Y^P_{i,j}$ and $Y^C_{i,j}$.

For each $i \in \{1, \ldots, m\}$, for the CRD $\mathcal{D}_i = (\Lambda, R, \Sigma, \Upsilon)$ computing the predicate ϕ_i, let L^1_i represent any species in Υ, and L^0_i represent any species in $\Lambda \setminus \Upsilon$, and that once $\mathcal{D}_i$ reaches an output stable configuration, where b is the output of $\mathcal{D}_i$. By Theorem 5 of Angluin et al. [11], if the total count of L^1_i and L^0_i molecules is $\Omega(n)$ (which can be enforced by adding

[5] It is easy to see that no leaderless CRN could reach an output stable state with positive count of output species Y from an initial state with no molecules, since it would need to contain reaction(s) of the form $\varnothing \to A$ for some species A from which (unbounded counts of) Y could be produced.

both L_i^1 and L_i^0 as products of the initial reactions of the input molecules: $X_i \to L_i^1 + L_i^0 + \ldots$), then the correct vote can be spread through the population of L_i^b molecules (for $b \in \{0,1\}$) in time $O(n)$.

Add the following reactions for each $i \in \{1,\ldots,m\}$ and each $j \in \{1,\ldots,l\}$:

$$L_i^1 + \hat{Y}_{i,j}^P \to L_i^1 + Y_{i,j}^P + Y_j \tag{3.2.3}$$
$$L_i^0 + Y_{i,j}^P \to L_i^0 + M_{i,j} \tag{3.2.4}$$
$$M_{i,j} + Y_j \to \hat{Y}_{i,j}^P \tag{3.2.5}$$

The latter two reactions implement the reverse direction of the first reaction – using L_i^0 as a catalyst instead of L_i^1 – using only bimolecular reactions. Also add the reactions

$$L_i^1 + \hat{Y}_{i,j}^C \to L_i^1 + Y_{i,j}^C \tag{3.2.6}$$
$$L_i^0 + Y_{i,j}^C \to L_i^0 + \hat{Y}_{i,j}^C \tag{3.2.7}$$

and

$$Y_{i,j}^P + Y_{i,j}^C \to K_j \tag{3.2.8}$$
$$K_j + Y_j \to \varnothing \tag{3.2.9}$$

That is, a "yes" answer for function i activates the ith output and a "no" answer deactivates the ith output. Eventually each CRD stabilizes so that precisely one i has L_i^1 present, and for all $i' \neq i$, $L_{i'}^0$ is present, which takes time $O(n)$ by Theorem 5 of Angluin et al. [11]. We now claim that at this point, all outputs for the correct function $\hat{f}_i$ will be activated and all other outputs will be deactivated. The reactions enforce that at any time, $\#Y_j = \#K_j + \sum_{i=1}^m (\#Y_{i,j}^P + \#M_{i,j})$. In particular, $\#Y_j \geq \#K_j$ and $\#Y_j \geq \#M_{i,j}$ at all times, so there will never be a K_j or $M_{i,j}$ molecule that cannot participate in the reaction of which it is a reactant. Eventually $\#Y_{i,j}^P$ and $\#Y_{i,j}^C$ stabilize to 0 for all but one value of i (by reactions (3.2.4), (3.2.5), (3.2.7)), and for this value of i, $\#Y_{i,j}^P$ stabilizes to $\mathbf{y}(j)$ and $\#Y_{i,j}^C$ stabilizes to 0 (by reaction (3.2.8)). Eventually $\#K_j$ stabilizes to 0 by the last reaction. Eventually $\#M_{i,j}$ stabilizes to 0 since L_i^0 is absent for the correct function $\hat{f}_i$. This ensures that $\#Y_j$ stabilizes to $\mathbf{y}(j)$.

It remains to analyze the expected time to stabilization. Let $n = \|\mathbf{x}\|$. By Lemma 3.2.1, the expected time for each affine function computation to complete is $O(n)$. Since the $\hat{Y}_{i,j}^P$ are produced monotonically, the most $Y_{i,j}^P$ molecules that are ever produced is $\#_\infty \hat{Y}_{i,j}^P$. Since we have m computations

in parallel, the expected time for all of them to complete is $O(nm) = O(n)$ (since m depends on f but not n). We must also wait for each predicate computation to complete. By Theorem 3.2.3, each of these predicates takes expected time $O(n)$ to complete, so all of them complete in expected time $O(mn) = O(n)$.

At this point, the L_1^i leaders must convert inactive output species to active, and $L_0^{i'}$ (for $i' \neq i$) must convert active output species to inactive. By Lemma 3.1.4, reactions (3.2.3), (3.2.4), (3.2.6), and (3.2.7) complete in expected time $O(\log n)$. Once this is completed, by Lemma 3.1.3, reaction (3.2.5) completes in expected time $O(n)$. Reaction (3.2.8) completes in expected time $O(n)$ by Lemma 3.1.3. Once this is done, reaction (3.2.9) completes in expected time $O(n)$ by Lemma 3.1.3. □

3.3 Conclusion

In this chapter, we have answered an open question of Chen, Doty, and Soloveichik [21], who showed that a function $f : \mathbb{N}^k \to \mathbb{N}^l$ is deterministically computable by a stochastic chemical reaction network (CRN) if and only if the graph of f is a semilinear subset of $\mathbb{N}^{k+l}$. Their proposed construction crucially used auxiliary *leader* species. The authors asked whether deterministic CRNs without a leader can still compute semilinear functions. We have affirmatively answered this question and showed that every semilinear function is deterministically computable by a CRN which starts with an initial configuration containing only the input species and zero counts of every other species, so long as $f(\mathbf{0}) = \mathbf{0}$.

Chen et al. [21] provided, for every semilinear function f, a direct construction of a CRN that computes f (using leaders) in expected time $O(n \log n)$, where n is the number of molecules present initially. They then combined this direct, error-free construction in parallel with a fast ($O(\log^5 n)$) but error-prone CRN that uses a leader to compute *any* computable function (including semilinear), using the error-free computation to change the answer of the error-prone computation only if the latter is incorrect. This combination speeds up the computation from expected time $O(n \log n)$ for the direct construction to expected time $O(\log^5 n)$ for the combined construction.

Since we have assumed no leaders may be supplied in the initial configuration, and since the problem of computing arbitrary computable functions without a leader has remained a major open problem [11], this trick does not work for speeding up our construction. However, we have shown that

with some care in the choice of reactions, the direct stable computation of a semilinear function can be done in expected time $O(n)$, improving upon the $O(n \log n)$ bound of the direct construction of [21].

Chapter 4

Simplifying Analyses of Chemical Reaction Networks for Approximate Majority

In this chapter, we describe the details of our techniques to investigate the efficiency and correctness of the CRNs shown in Figure 1.2. We start by explaining the preliminary semantics of the Approximate Majority CRNs and our analysis tools utilized throughout in Section 4.1. We then analyze the behaviour of the tri-molecular CRN in Section 4.2. Next, we analyze the bi-molecular CRNs of Figure 1.2 (Single-B and Double-B CRNs) in Section 4.3 by showing that they are essentially simulations of the tri-molecular CRN. We end this chapter by describing our experimental results and reviewing our outcomes in this work.

4.1 Preliminaries

We employ the kinetic model described in Section 2.2 for our analysis[6].

We note that for the CRNs that we analyze in this chapter, there is some order o such that for every reaction $(\mathbf{r}, \mathbf{p}, k_\alpha)$ of R, $r_1 + r_2 + \ldots + r_m = p_1 + p_2 + \ldots p_m = o$. Thus the number of interacting molecules does not change over time. Moreover, according to the finite density constraint, we'll assume that the volume of the solution is proportional to the initial number of molecules.

We consider a system in which the initial molecular count is n, and so the molecular count in each subsequent configuration is also n. Therefore, the total molecular count in each configuration is n and, without loss of generality, we assume that volume v, which remains fixed, is also equal to

[6]Here, we note that the reaction rate constants defined for the CRNs shown in Figure 1.2 are very important in the proof of their correctness and efficiency. It is an area of future work (briefly discussed in Section 5.2.2) to understand how the reaction rate constants can be accurately approximated by DSDs.

n.

Now, suppose that the well-mixed solution is in configuration $\mathbf{c}$. Recall from Section 2.2, the time until the next interaction happens is exponentially distributed with parameter $\rho = \sum_{\mathbf{r}} \rho(\mathbf{c}, \mathbf{r})$. Accordingly, the time T_k for k interactions is at most the sum of k exponentially distributed random variables, each with parameter ρ, with expected value and variance $\mathbb{E}[T_k] = k/\rho$ and $\mathrm{Var}[T_k] = k/\rho^2$ respectively. With the assumptions that all the interactions are of order o and $v = n$, we conclude

$$\rho = \binom{n}{o} / v^{o-1} = \Theta(n) \leq n. \tag{4.1.1}$$

Thus, by Chebyshev's inequality, we have that

$$\mathbb{P}[|T_k - \mathbb{E}[T_k]| \geq h\sqrt{\mathrm{Var}[T_k]}] = \mathbb{P}[|T_k - \Theta(k/n)| \geq h\Theta(\sqrt{k}/n)] \leq 1/h^2.$$

If for example $k = n$, then by setting $h = \sqrt{n}$ we see that the time for n interactions is $O(1)$ with probability at least $1 - 1/n$. Thus we may use the number of interactions, divided by n, as a proxy for time. More generally, the time for $k = \Omega(n)$ interactions is $O(k/n)$ with probability at least $1-1/n$.

4.1.1 Analysis Tools

We will use the following well-known property of random walks, Chernoff tail bounds on functions of independent random variables, and Azuma's inequality.

Lemma 4.1.1 (Asymmetric one-dimensional random walk [38](XIV.2))**.** *If we run an arbitrarily long sequence of independent trials, each with success probability at least p, then the probability that the number of failures ever exceeds the number of successes by b is at most $(\frac{1-p}{p})^b$.*

Lemma 4.1.2 (Chernoff tail bounds [26])**.** *If we run N independent trials, with success probability p, then S_N, the number of successes, has expected value $\mu = Np$ and, for $0 < \delta < 1$,*
(a) $\mathbb{P}(S_N \leq (1-\delta)\mu) \leq \exp(-\frac{\delta^2\mu}{2})$, and
(b) $\mathbb{P}(S_N \geq (1+\delta)\mu) \leq \exp(-\frac{\delta^2\mu}{3})$.

Lemma 4.1.3 (Azuma's inequality [73])**.** *Let $Q_1, \ldots, Q_k$ be independent random variables, with Q_r taking values in a set A_r for each r. Suppose that the (measurable) function $f : \Pi A_r \to R$ satisfies $|f(x) - f(x')| \leq c_r$*

whenever the vectors x and x' differ only in the rth coordinate. Let Y be the random variable $f(Q_1, \ldots, Q_k)$. Then , for any $t > 0$,

$$\mathbb{P}[|Y - E[Y]| \geq t] \leq 2\exp\Big(-2t^2/\sum_{r=1}^{k} c_r^2\Big).$$

4.2 Approximate Majority Using Tri-molecular Reactions

In this section we analyze the behaviour of the tri-molecular CRN of Figure 1.2(a). Intuitively, its reactions sample triples of molecules and amplify the majority species by exploiting the facts that (i) every triple must have a majority of either X or Y, and (ii) the ratio of the number of triples with two X-molecules and one Y-molecule to the number of triples with 2 Y-molecules and one X-molecule, is exactly the ratio of X-molecules to Y-molecules.

Our main goal is to prove the following:

Theorem 4.2.1. *For any constant $\gamma > 0$, there exists a constant c_γ such that, provided the initial molecular count of X exceeds that of Y by at least $c_\gamma\sqrt{n \lg n}$ [7], the tri-molecular CRN reaches a consensus of X-majority, with probability at least $1 - n^{-\gamma}$, in at most $c_\gamma n \lg n$ interactions.*

Recall that we denote by x and y the random variables corresponding to the molecular count of X and Y respectively. We divide the computation into a sequence of three, slightly overlapping and possibly degenerate, phases, where c_γ, d_γ and e_γ are constants depending on γ:

phase 1 $c_\gamma/2\sqrt{n \lg n} < x - y \leq n(d_\gamma - 2)/d_\gamma$. It ends as soon as $y \leq n/d_\gamma$.

phase 2 $e_\gamma \lg n < y < 2n/d_\gamma$. It ends as soon as $y \leq e_\gamma \lg n$.

phase 3 $0 \leq y < 2e_\gamma \lg n$. It ends as soon as $y = 0$.

Of course the assertion that a computation can be partitioned in such a way that these phases occur in sequence holds only with sufficiently high probability. To facilitate this argument, as well as the subsequent efficiency analysis, we divide both phase 1 and phase 2 into $\Theta(\lg n)$ stages, defined by integral values of t and s, as follows:

[7] In this chapter, when we use notation lg to refer to $\log_2$.

- A typical stage in phase 1 starts with $x \geq y + 2^t\sqrt{n \lg n}$ and ends with $x \geq y + 2^{t+1}\sqrt{n \lg n}$, where $\lg c_\gamma \leq t \leq (\lg n - \lg \lg n)/2 + \lg((d_\gamma - 2)/(2d_\gamma))$.
- A typical stage in phase 2 starts with $y \leq n/2^s$ and ends with $y \leq n/2^{s+1}$, where $\lg d_\gamma \leq s \leq \lg n - \lg \lg n - \lg e_\gamma - 1$.

Our proof of correctness (the computation proceeds through the specified phases as intended) and our timing analysis (how many interactions does it take to realize the required number of reaction events) exploit the simple and familiar tools set out in the previous section, taking advantage of bounds on the probability of reactions (1) and (2) that hold throughout a given phase/stage:

(a) [**Low probability of unintended phase/stage completion**] The relative probability of reactions (1) and (2) is determined by the relative counts of X and Y. This allows us to argue, using a biased random walk analysis (Lemma 4.1.1 above), that, with high probability, there is no back-sliding; when the computation leaves a phase/stage it is always to a higher indexed phase/stage (cf. Corollaries 1, 2 and 3, below).

(b) [**High probability of intended phase/stage completion within a small number of reaction events**] Within a fixed phase/stage the computation can be viewed as a sequence of independent trials (choice of reaction (1) or (2)) with a fixed lower bound on the probability of success (choice of reaction (1)). This allows us to establish, by a direct application of Chernoff's upper tail bound Lemma 4.1.2, an upper bound, for each phase/stage, on the probability that the phase/stage completes within a specified number of reaction events (cf. Corollaries 4, 5 and 6, below).

(c) [**High probability that the reaction events occur within a small number of molecular interactions**] Within a fixed phase/stage the choice of reaction events, among interactions, can be viewed as a sequence of independent trials with a fixed lower bound on the probability of success (the interaction corresponds to a reaction event). Thus our timing analysis (proof of efficiency) is another direct application of Chernoff's upper tail bound (Lemma 4.1.2) (cf. Corollary 7, below).

Lemma 4.2.2. *At any point in the computation, if $x - y = \Delta$ then the probability that $x - y \leq \Delta/2$ at some subsequent point in the computation is less than $(1/e)^{\Delta^2/(2n+2\Delta)}$.*

Proof. Since $x - y > \Delta/2$ up to the point when we first have $x - y \leq \Delta/2$, it follows that $x \geq n/2 + \Delta/4$ and $y \leq n/2 - \Delta/4$. We can view the change in $x - y$ as a random walk, starting at Δ, with success (an increase in $x - y$) probability p satisfying $p \geq 1/2 + \Delta/(4n)$.

It follows from Lemma 4.1.1 that reaching a configuration where $x - y \leq \Delta/2$ (which entails an excess of $\Delta/2$ failures to successes) is less than $(\frac{1}{1+\Delta/n})^{\Delta/2}$ which is at most $(1/e)^{\Delta^2/(2n+2\Delta)}$. □

Corollary 4.2.3. *In stage t of phase 1, $x - y$ reduces to $2^{t-1}\sqrt{n \lg n}$ with probability less than $1/n^{2^{2t-2}}$.*

Lemma 4.2.4. *At any point in the computation, if $y = n/k$ then the probability that $y > 2n/k$ at some subsequent point in the computation is less than $(2/(k-2))^{n/k}$.*

Proof. Since $y \leq 2n/k$ up to the point when we first have $y > 2n/k$, we can view the change in y as a random walk, starting at n/k, with success (a decrease in y) probability p satisfying $p \geq 1 - 2/k$.

It follows from Lemma 4.1.1 that reaching a configuration where $y > 2n/k$ (which entails an excess of n/k failures to successes) is less than $(2/(k-2))^{n/k}$. □

Corollary 4.2.5. *In stage s of phase 2, y increases to $n/2^{s-1}$ with probability less than $(2/(2^s - 2))^{n/2^s}$.*

Corollary 4.2.6. *In phase 3, y increases to $2e_\gamma \lg n$ with probability less than $(2e_\gamma \lg n/(n - 2e_\gamma \lg n))^{e_\gamma \lg n}$.*

Lemma 4.2.7. *At any point in the computation, if $x - y = \Delta \leq n/2$ then, assuming that $x - y$ never reduces to $\Delta/2$, the probability that $x - y$ increases to 2Δ within at most λn reaction events is at least $1 - \exp(-\frac{(\lambda-2)\Delta^2}{\lambda(2n+\Delta)})$.*

Proof. We view the choice of reaction as an independent trial with success corresponding to reaction (1), and failure to reaction (2). We start with $x - y = \Delta$ and run until either $x - y = \Delta/2$ or we have completed λn reactions. By Lemma 4.1.2, the probability that we complete λn reactions with fewer than $\lambda n/2 + \Delta/2$ successes, which is necessary under our assumptions if we finish with $x - y < 2\Delta$, is at most $\exp(-\frac{(\lambda-2)\Delta^2}{\lambda(2n+\Delta)})$.

□

Corollary 4.2.8. *In stage t of phase 1, assuming that $x - y$ never reduces to $2^{t-1}\sqrt{n \lg n}$, the probability that $x - y$ increases to $2^{t+1}\sqrt{n \lg n}$ within at most λn reaction events is at least $1 - \exp(-\frac{(\lambda-2)2^{2t} \lg n}{3\lambda})$.*

Lemma 4.2.9. *At any point in the computation, if $y = n/k$ then, assuming that y never increases to $2n/k$, the probability that y decreases to $n/k - r$ within $f(n) > 2r$ reaction events is at least $1 - \exp(-\Theta(f(n))$.*

Proof. We view the choice of reaction as an independent trial with success corresponding to reaction (1), and failure to reaction (2). We start with $y = n/k$ and run until either $y = n/k - r$ or we have completed $f(n)$ reaction events. (We assume, by Lemma 4.2.4, that $y < 2n/k$, and so $p > 1 - 2/k$, throughout.)

By Lemma 4.1.2, the probability that we complete $f(n)$ reactions with fewer than $(f(n)+r)/2$ successes, which is necessary under our assumptions if we finish with $y > \frac{n}{k} - r$, is at most

$$\exp(-\frac{[f(n)(k-2)/k - (f(n)+r)/2]^2}{2f(n)(k-2)/k}),$$

which is at most $\exp(-\Theta(f(n))$, when $f(n) > 2r$. □

Corollary 4.2.10. *In stage s of phase 2, assuming that y never increases to $n/2^{s-1}$, y decreases to $n/2^{s+1}$, ending stage s, in at most $\lambda n/2^s$ reaction events, with probability at least $1 - \exp(-\Theta(\lambda n/2^s))$.*

Corollary 4.2.11. *In phase 3, assuming that y never increases to $2e_\gamma \lg n$, y decreases to 0, ending phase 3 (and the entire computation), in at most $\lambda \lg n$ reaction events, with probability at least $1 - \exp(-\Theta(\lambda \lg n))$.*

Lemma 4.2.12. *If during some sequence of m interactions the total propensity of all reactions is at least p then the probability that the sequence gives rise to fewer than $mp/(2n)$ reaction events is no more than $\exp(-mp/(8n))$.*

Proof. Recall from Section 2.2 that if the total propensity of all reactions in a given configuration $\mathbf{c}$ is at least p then the probability that an interaction results in a reaction is $\frac{\rho(\mathbf{c},R)}{\rho} \geq p/n$ (see Equation 4.1.1). Hence, by Lemma 4.1.2, the probability that a sequence of m interactions gives rise to fewer than $mp/(2n)$ reaction events is no more than $\exp(-mp/(8n))$. □

Corollary 4.2.13.
(i) The λn reaction events of each stage of phase 1 occur within $(8/3)d_\gamma \lambda n$ interactions, with probability at least $1 - \exp(-\lambda n/4)$

(ii) The $\lambda(n/2^s)$ reaction events of stage s of phase 2 occur within $(16/3)\lambda n$ interactions, with probability at least $1 - \exp(-\lambda n/2^{s+2})$; and
(iii) The $\lambda \lg n$ reaction events of phase 3 occur within $(8/3)\lambda n \lg n$ interactions, with probability at least $1 - \exp(\lambda \lg n/4)$.

Proof. It suffices to observe the following lower bounds on the propensity of reaction (1) alone in individual phases/stages:
(i) in phase 1, $x > y \geq n/d_\gamma$, so the propensity of reaction (1) is greater than $3n/(4d_\gamma)$;
(ii) in stage s of phase 2, $x > n(1 - 2^{s-1})$ and $y \geq n/2^{s+1} \geq (\lg n)/2$, so the propensity of reaction (1) is at least $3n/2^{s+3}$;
(iii) in phase 3, $x \geq n - \lg n$ and $y \geq 1$, so the propensity of reaction (1) is at least 3/4. □

Finally, we prove Theorem 4.2.1 using the pieces proved until now.

of Theorem 4.2.1. (i) [Correctness] It follows directly from Corollaries 1 and 4 (respectively, 2 and 5, 3 and 6) that phase 1 (respectively phase 2, phase 3) completes in the intended fashion, within at most $\lambda n \lg n$ (respectively, λn, $\lambda \lg n$) reaction events, with probability at least $1 - \exp(-\Theta(c_\gamma \lg n))$ (respectively, $1 - \exp(-\Theta(\lambda n/d_\gamma))$, $1 - \exp(-\Theta(\lambda \lg n))$).
(ii) [Efficiency] It is immediate from Corollary 4.2.13 that the required number of reaction events in phase 1 (respectively, phase 2, phase 3) occur within $\Theta(\lambda n \lg n)$ interactions, with probability at least $1 - \exp(-\Theta(\lambda \lg n))$.
□

4.3 Approximate Majority Using Bi-molecular Reactions

Here we show correctness and efficiency of the Double-B and Single-B CRNs, essentially by showing that the abstraction of both CRNs corresponds to the tri-molecular CRN of the previous section.

Here and throughout, we denote by b the random variable corresponding to the molecular count of B.

4.3.1 The Double-B CRN

In this section we analyse the behaviour of the Double-B CRN of Figure 1.2(b). We will show that:

Theorem 4.3.1. *For any constant $\gamma > 0$, there exists a constant c_γ such that, provided (i) the initial molecular count of X and Y together is at least $n/2$, and (ii) the count of X exceeds that of Y by at least $c_\gamma\sqrt{n \lg n}$, the computation reaches a consensus of X-majority, with probability at least $1 - n^{-\gamma}$, in at most $c_\gamma n \lg n$ interactions.*

Comparing with Theorem 4.2.1, it becomes clear that the role of the molecule B is simply to facilitate the simulation. The sense in which Double-B can be seen as simulating the earlier tri-molecular CRN is that we can analyse its behaviour in three phases that directly parallel those of our tri-molecular analysis. We measure progress throughout in terms of the change in the molecular counts $\hat{x}$, defined as $x + b/2$, and $\hat{y}$, defined as $y + b/2$, noting that reaction (0') leaves these counts unchanged and reactions (1') and (2') change $\hat{x}$ and $\hat{y}$ at exactly half the rate that the corresponding tri-molecular reactions (1) and (2) change x and y. In each phase, we note that the relative propensity of reaction (1') to that of (2'), equals or exceeds the relative propensity of reaction (1) to that of (2) in the tri-molecular CRN, and we argue that the total propensity of reactions (1') and (2') is at least some constant fraction of the total propensity of reactions (1) and (2). This allows us to conclude that Double-B (i) takes at most twice as many reaction events as the tri-molecular CRN to complete each corresponding phase/stage and (ii) these reaction events occur within a number of interactions that is at most some constant multiple of the number of interactions needed to realize the required reaction events in the tri-molecular CRN.

Bounds on b, the molecular count of B

We start by setting out bounds on b, the molecular count of molecule B, specifically $y/2\alpha \leq b \leq n/2$, where $\alpha \geq 20$, that, with high probability, hold after the first $\Theta(n)$ interactions (see Lemmas 4.3.2 and 4.3.8), and continue to hold thereafter. These bounds allow us to establish property (ii) above. Our bounds are summarized in Lemma 4.3.3 below, and its proof is a straightforward application of Chernoff bounds. We note that the connection between the number of reaction events and the number of interactions in each interval I used in Lemmas 4.3.2 and 4.3.3 is shown in Lemma 4.3.8.

Lemma 4.3.2. *Let I be the interval of the first $l = \frac{1}{2}y_0$ reaction events in a computation of Double-B, where y_0 is the initial molecular count of Y. Let b_0, and b_e denote the initial and final values of b in this interval (similarly for y). If $b_0 < y_0/\alpha$ (even if $b_0 = 0$), then there exists a constant*

f_γ such that if $y_0 \geq f_\gamma \lg n$, then $y_e/\alpha \leq b_e \leq 10n/21$ with probability at least $1 - 1/n^\gamma$.

Proof. Note that $x-y$ does not change by reaction (0'), and by Lemma 4.2.2, it never reaches $(x_0 - y_0)/2$ through reactions (1') and (2'), which result in $x - y \geq 0$ in interval I.

Assuming that $b \leq y/\alpha$ over the course of interval I, the probability that a reaction event would be equal to reaction (0') is at least $\frac{\alpha}{\alpha+2}$ (see computations below).

$$\mathbb{P}[\text{reaction (0') occurs}] = \frac{xy}{xy + xb + yb} \geq \frac{xy}{xy + 2x\frac{y}{\alpha}} \geq \frac{\alpha}{\alpha + 2}.$$

Therefore, by the Chernoff lower tail bound 4.1.2, there are at least $\frac{3\alpha}{4(\alpha+2)}l$ reaction events of type (0'), among $l = 1/2y_0$ reaction events, with probability at least $1 - 1/n^\gamma$ provided that $y_0 \geq f_\gamma \lg n$. Thus, at most $\frac{\alpha+8}{4(\alpha+2)}l$ reactions events are of types (1') and (2').

We note that the number of B molecules increases via reaction (0') and decreases via reactions (1') and (2'). The number of Y molecules also increases via reaction (2') and decreases via reaction (0'). So, we can compute b_e and y_e as follows.

$$b_e \geq b_0 + \frac{6\alpha}{4(\alpha+2)}l - \frac{\alpha+8}{4(\alpha+2)}l, \quad \text{and } y_e \leq y_0 - \frac{3\alpha}{4(\alpha+2)}l + \frac{\alpha+8}{4(\alpha+2)}l,$$

With a simple computation, it is clear that $b_e \geq \frac{y_e}{\alpha}$ even if $b_0 = 0$. Moreover, since $x > y$ in interval I, we have $y_0 < n/2$ and subsequently $l < n/4$ which never let b reach $10n/21$. □

Lemma 4.3.3. *Let I be any interval of ϵy_0 reaction events of a computation of Double-B, where $\epsilon \leq \frac{1}{\alpha+1}$, $\alpha \geq 20$, and y_0 is the value of y at the beginning of I. Let b_0, b_e, b_{min}, b_{max} denote the initial, final, minimum, and maximum values of b in the interval I (similarly for y). Then for any $\gamma > 0$, there is a constant l_γ such that if $y_0 \geq l_\gamma \lg n$, then the following bounds hold with probability at least $1 - 1/n^\gamma$.*

(a) *(Lower bounds) If $b_0 \geq \frac{y_0}{\alpha}$ then $b_{\min} \geq \frac{y_{\max}}{2\alpha}$ and $b_e \geq \frac{y_e}{\alpha}$.*

(b) *(Upper bounds) If $b_0 \leq 10n/21$ then $b_e \leq 10n/21$ and $b_{\max} \leq n/2$.*

Proof. We prove the claims (a) and (b) as follows.

Lower bounds: We can validate that both lower bounds hold when $b_0 \geq \frac{2}{\alpha}y_0$. In fact, we can simply consider the minimum value $b_0 - \epsilon y_0$

for $b_{\min}$ and b_e and the maximum value $y_{\max} = y_0 + \epsilon y_0$ for $y_{\max}$ and y_e. So, we just focus on the situation where $b_0 < \frac{2}{\alpha} y_0$. Using this bound on b together with the assumptions that $x \geq y$ (Lemma 4.2.2), and $\epsilon \leq \frac{1}{\alpha+1}$, we obtain that a reaction event is equal to reaction (0') with probability at least $\frac{\alpha-1}{\alpha+5}$ (see computation below).

$$\mathbb{P}[\text{Reaction (0') occurs}] = \frac{xy}{xy + xb + yb} \geq \frac{y}{y + 2b} \geq \frac{y_0 - \epsilon y_0}{y_0 + \frac{2}{\alpha} y_0 + 3\epsilon y_0} \geq \frac{\alpha - 1}{\alpha + 5}.$$

Therefore, by the Chernoff lower tail bound (Lemma 4.1.2), there are at least $\frac{3(\alpha-1)}{4(\alpha+5)} \epsilon \hat{y_0}$ reaction events of types (0'), among ϵy_0 reaction events, with probability at least $1 - 1/n^{\gamma}$ on the condition that $y_0 \geq l_{\gamma} \lg n$. As a result, at most $\frac{\alpha+23}{4(\alpha+5)}$ reaction events are equal to either reaction (1') or (2') in Figure 1.2. Therefore, we can prove the correctness of the lower bounds as follows.

case (i) $\frac{y_{\max}}{2\alpha} \leq b_{\min}$. Throughout interval I, at most $\frac{\alpha+23}{4(\alpha+5)} \epsilon y_0$ (reactions (1') and (2')) B molecules are consumed and at most $\frac{\alpha+23}{4(\alpha+5)} \epsilon y_0$ (reactions (2')) Y molecules are produced. So,

$$b_{\min} \geq b_0 - \frac{(\alpha + 23)}{4(\alpha + 5)} \epsilon y_0, \text{and } \; y_{\max} \leq y_0 + \frac{(\alpha + 23)}{4(\alpha + 5)} \epsilon y_0$$

that confirm the inequality for all $\frac{y_0}{\alpha} \leq b_0 \leq \frac{2y_0}{\alpha}$.

case (ii) $\frac{y_e}{\alpha} \leq b_e$. At the end of the interval I, in addition to the maximum consumption and production of B and Y molecules described in case (i), at least $\frac{6(\alpha-1)}{4(\alpha+5)} \epsilon y_0$ (reaction(0')) B molecules are produced and at least $\frac{3(\alpha-1)}{4(\alpha+5)} \epsilon y_0$ (reaction (0')) Y molecules are consumed as well. So,

$$b_e \geq b_0 + \frac{5\alpha - 29}{2(\alpha + 5)} \epsilon y_0, \text{ and } \; y_e \leq y_0 - \frac{\alpha - 13}{(\alpha + 5)} \epsilon y_0$$

which result in $\frac{y_e}{\alpha} \leq b_e$ for all $\frac{y_0}{\alpha} \leq b_0 \leq \frac{2y_0}{\alpha}$.

Upper bounds: Note that $y_0 \leq n/2$, since $x > y$ and $x + y \leq n$ through the interval. It is easy to verify that while $b_0 \leq \frac{9n}{21}$, both upper bounds completely hold even if b increases to its maximum of $b_0 + 2\epsilon y_0$ ($b_e \leq b_{\max} \leq 10n/21 \leq n/2$). Moreover, note that over the course of I, even if $b_0 = 10n/21$, we have $b_{\max} \leq 10n/21 + 2\epsilon y_0 \leq 11n/21$ and subsequently $x + y \geq 10n/21$.

We now consider the case where $b_0 > \frac{9n}{21}$. The value of b increases via reaction (0') and decreases via reactions (1') and (2'). Then, the relative probability of reaction (0') to that of reactions (1') and (2') can be computed as follows.

$$\frac{\mathbb{P}[\text{Reaction (0') occurs}]}{\mathbb{P}[\text{Reaction (1') or (2') occurs}]} = \frac{xy}{(x+y)b} \leq \frac{(10/42)^2}{(10n/21)(9n/21)} \leq \frac{2}{5}$$

Therefore, the probability that a reaction event would be equal to reaction (0') is less than 2/7.

By the Chernoff upper tail bound (Lemma 4.1.2), we can then conclude that there at most $11/35\epsilon y_0$ reaction events are equal to reaction (0') (i.e., increasing b) with probability at most $\exp(-\Theta(n))$ assuming that $y = \Theta(n)$. Interchangeably, it means that the number of reaction events equal to reactions (1') or (2') (i.e., decreasing b) is also at least $24/35\epsilon y_0$. Therefore, with high probability, the net change in b after ϵy_0 reaction events is negative ($(22/35 - 24/35)\epsilon y_0 < 0$) and it gives us $b_e \leq \frac{10n}{21}$. Also, as the total increase in b is at most $\frac{22}{35}\epsilon y_0$ throughout interval I, b will be bounded by $b_0 + \frac{22}{35}\epsilon\hat{y_0} \leq n/2$ as well. □

Lemma 4.3.4. *Let I be any interval of $l = cy_0$ reaction events in the computation of the Double-B CRN, where $c \leq 1/2$, y_0 is the initial value of y at the beginning of I, and $x \geq y$ throughout the interval. If $y_0 \geq w \lg n$, then for any $\gamma > 0$, interval I will happen within $\frac{16\gamma}{(1-c)w}n$ interactions with probability at least $1 - 1/n^\gamma$.*

Proof. Using Lemma 4.2.12, it is sufficient to show that the total propensity of Double-B reactions over the course of interval I is $\frac{1-c}{2}y_0$. Among $l = \epsilon y_0$ reaction events, the number of Y molecules can decrease by at most l, so $y \geq y_0 - l$. On the other hand, since $x \geq y$, we have $n = x+y+b \leq x+x+b$ leading to $x+b/2 \geq n/2$ throughout this interval. Putting these observations together, we obtain:

$$\rho(c, R) = \frac{xy + xb + yb}{n} \geq \frac{y(x + b/2)}{n} \geq 1/2(y_0 - l) = \frac{1}{2}(1 - c)y_0.$$

□

Double-B simulates the tri-molecular CRN

To understand computations of the Double-B CRN as simulations of the tri-molecular CRN, we again identify three phases, this time expressed in

terms of $\hat{x}$ and $\hat{y}$. (Note that (i) $\hat{x} - \hat{y} = x - y$ and (ii) $\hat{y} = 0$ implies both $y = 0$ and $b = 0$):

phase 1 $c_\gamma/2\sqrt{n \lg n} < \hat{x} - \hat{y} \leq n(d_\gamma - 2)/d_\gamma$. It ends as soon as $\hat{y} \leq n/d_\gamma$.

phase 2 $e_\gamma \lg n < \hat{y} < 2n/d_\gamma$. It ends as soon as $\hat{y} \leq e_\gamma \lg n$.

phase 3 $0 \leq \hat{y} < 2e_\gamma \lg n$. It ends as soon as $\hat{y} = 0$.

As before, we divide both phase 1 and phase 2 into $\Theta(\lg n)$ stages, defined by integral values of t and s, as follows:

- A typical stage in phase 1 starts with $\hat{x} \geq \hat{y} + 2^t\sqrt{n \lg n}$ and ends with $\hat{x} \geq \hat{y} + 2^{t+1}\sqrt{n \lg n}$, where $\lg c_\gamma \leq t \leq (\lg n - \lg \lg n)/2 + \lg((d_\gamma - 2)/(2d_\gamma))$.
- A typical stage in phase 2 starts with $\hat{y} \leq n/2^s$ and ends with $\hat{y} \leq n/2^{s+1}$, where $\lg d_\gamma \leq s \leq \lg n - \lg \lg n - \lg e_\gamma - 1$.

The proof of Theorem 4.3.1 completely parallels that of Theorem 4.2.1, with $\hat{x}$ and $\hat{y}$ substituted for x and y. For correctness of Double-B, it suffices to note that (i) reaction (0') does not change either $\hat{x}$ or $\hat{y}$, (ii) reaction (1') increases $\hat{x}$ by 1/2 and decreases $\hat{y}$ by 1/2, and (iii) reaction (2') increases $\hat{y}$ by 1/2 and decreases $\hat{x}$ by 1/2. Thus twice as many reactions (1') and (2') suffice to achieve the same results as reactions (1) and (2) of the tri-molecular CRN.

The efficiency of Double-B follows in a similar way from the earlier analysis of the tri-molecular CRN presented in Corollary 4.2.13. There we observed that it sufficed to bound from below the propensity of reaction (1). For the corresponding analysis of Double-B, we observe that in all corresponding phases/stages the propensity of reaction (1') is up to a constant factor the same as that of reaction (1). This follows immediately from the upper bound $(n/2)$ on b, which ensures that the molecular count of X is at least $n/4$, and the lower bound $(y/292)$ on b, which ensures that the molecular count of B is at least a constant fraction of that of Y. The constant e_γ that is used in demarking the end of phase 2 and the start of phase 3 will now depend on the constant f_γ of Lemma **??**, in order to ensure that this lower bound on b holds throughout phase 2 with high probability.

4.3.2 The Single-B CRN

Here, we study the behaviour of Single-B, originally proposed by Angluin et al. [12] and shown in Figure 1.2(b). We will show that:

Theorem 4.3.5. *For any constant $\gamma > 0$, there exists a constant $c_\gamma > \gamma$ such that, provided (i) the initial molecular count of X and Y together is at least $n/2$, and (ii) the count of X exceeds that of Y by at least $c_\gamma\sqrt{n \lg n}$, the Single-B CRN reaches a consensus of X-majority, with probability at least $1 - n^{-\gamma}$, in $c_\gamma n \lg n$ interactions.*

Comparing the Double-B and Single-B CRNs, we notice the only difference is that reaction (0') is replaced by probabilistic reactions (0'x) and (0'y) which are equally likely. Intuitively, this replacement keeps the behaviour of the Single-B and Double-B CRNs essentially the same, as reactions (0'x) and (0'y), on average, have no effect on $\hat{x}$ and $\hat{y}$. However, its advantage is to never let Single-B CRN reach B-majority consensus[8].

In order to analyze the behaviour of Single-B, we also consider three phases similar to those for Double-B and tri-molecular CRNs. The progress of the protocol is also measured in terms of random variables $\hat{x}$ and $\hat{y}$. Reactions (0'x) and (1') increase $\hat{x}$ by $1/2$ and decrease $\hat{y}$ by $1/2$, and reactions (0'y) and (2') decrease $\hat{x}$ by $1/2$ and increase $\hat{y}$ by $1/2$. We recall that we assume x and y are substituted by $\hat{x}$ and $\hat{y}$ when we refer to lemmas of Section 4.2.

phase 1 $(c_\gamma - \gamma)/2\sqrt{n \lg n} < \hat{x} - \hat{y} \leq n(d_\gamma - 2)/d_\gamma$. It ends as soon as $\hat{y} \leq n/d_\gamma$.

phase 2 $e_\gamma \lg n < \hat{y} < 2n/d_\gamma$. It ends as soon as $\hat{y} \leq e_\gamma \lg n$.

phase 3 $0 \leq \hat{y} < 2e_\gamma \lg n$. It ends as soon as $\hat{y} = 0$.

Similar to the Double-B and tri-molecular CRNs, we divide both phase 1 and phase 2 into $\Theta(\lg n)$ stages, defined by integral values of t and s, as follows:

- A typical stage in phase 1 starts with $\hat{x} \geq \hat{y} + 2^t\sqrt{n \lg n}$ and ends with $\hat{x} \geq \hat{y} + 2^{t+1}\sqrt{n \lg n}$, where $\lg(c_\gamma - \gamma) \leq t \leq (\lg n - \lg\lg n)/2 + \lg((d_\gamma - 2)/(2d_\gamma))$.
- A typical stage in phase 2 starts with $\hat{y} \leq n/2^s$ and ends with $\hat{y} \leq n/2^{s+1}$, where $\lg d_\gamma \leq s \leq \lg n - \lg\lg n - \lg e_\gamma - 1$.

The proof that Single-B is correct and efficient parallels the phase analysis of the tri-molecular CRN, if we make the following adjustment steps.

[8] We note that although the B-majority consensus is reachable in the Double-B CRN, the probability of such an event is very small (i.e., $n^{\Theta(-\lg(n))}$) with our initial configuration setting. The bound is computed with a simple biased random walk analysis in $\lg n$ stages.

1. Establish that the initial gap between $\hat{x}$ and $\hat{y}$, with high probability, is not decreased by more than $\gamma\sqrt{n \lg n}$ within n reaction events – see Lemma 4.3.6. Thus, it assures us that $\hat{x} \geq \hat{y}$ in the starting interval required to get b bounded. Moreover, with the assumption that initially $\hat{x} - \hat{y} \geq c_\gamma\sqrt{n \lg n}$, with high probability, phase 1 starts with $\hat{x} - \hat{y} \geq (c_\gamma - \gamma)\sqrt{n \lg n}$.

2. Employ Lemmas 4.3.2 and 4.3.3 to provide the lower-bound $(y/2\alpha)$, where $\alpha \geq 20$, and upper-bound $(n/2)$ on b after the first $\Theta(n)$ interactions. In order to verify the proof of lemma for Single-B, it's only needed to adjust the constants according to propensities of reactions (0'x) and (0'y). We are able to provide a tighter upper bound on b with respect to variable y[9]. In fact, we can show that inequality $\frac{y}{2\alpha} \leq b \leq 2\alpha y$ hold at any point of the Single-B computation although it doesn't hold for the Double-B computation – see Lemmas 4.3.7 and 4.3.8.

3. Show that Lemmas 4.2.2 and 4.2.7 and their corresponding corollaries also prove the correctness and efficiency of phase 1 in the Single-B CRN. Utilizing the lower bound on b, the ratio of total propensity of reactions (0'x) and (1') to that of reactions (0'y) and (2') is lower than the relative propensity of reaction (1) to that of reaction (2) in the tri-molecular CRN by at most a small constant – See Lemma 4.3.9. Therefore, the analysis of phase 1 of Single-B parallels that of the tri-molecular CRN.

4. Modify Lemmas 4.2.4 and 4.2.9 so that they also verify the correctness and efficiency of phases 2 and 3 in the single-B CRN. Referring back to the lower bound on b, we note that the ratio of total propensity of reactions (0'y) and (2') to that of reactions (0'x) and (1'), is greater than the ratio of the propensity of reaction (2) to that of reaction (1) in the tri-molecular CRN, by at most a small constant. It is straightforward to consider this small constant and revise Lemmas 4.2.4 and 4.2.9 and their related corollaries, to make the analysis of phases 2 and 3 of Single-B also parallel to those of the tri-molecular CRN – See Lemmas 4.3.10 and 4.3.11.

5. Employ Lemma 4.2.12 to complete the proof of efficiency. Using the

[9]We note that, the derived b bounds in Lemma 4.3.3 are sufficient for the proof of correctness and efficiency of our two bi-molecular CRNs. However, a tighter upper bound on b may be useful when the Single-B protocol is used as a part of other CRNs.

upper bound on b, which confirms that $x \geq n/4$ and the lower bound on b, which confirms $b \geq y/292$, we can conclude that the total propensities of reactions (0'x), (0'y), (1'), and (2') is at least some constant fraction of the total propensities of reaction (1) and (2) in tri-molecular CRN. Therefore, the total number of interactions in Single-B is at most some constant multiple the required number of interactions in the tri-molecular CRN.

Lemma 4.3.6. *Starting from $\hat{x} - \hat{y} \geq c_\gamma \sqrt{n \lg n}$, where $c_\gamma > \gamma$, $\hat{x} - \hat{y}$ reduces to $(c_\gamma - \gamma)\sqrt{n \lg n}$ within $d = n$ reaction events with probability less than $1/n^{(\gamma^2)}$.*

Proof. Starting from $\hat{x} - \hat{y} \geq c_\gamma \sqrt{n \lg n}$, the probability that $\hat{x} - \hat{y}$ increases is at least as much as the probability that it decreases. As a worst case scenario, we can view the changes in $\hat{x} - \hat{y}$, as an unbiased random walk which starts at $c_\gamma \sqrt{n \lg n}$ and its expected translation distance is $\sqrt{n}$ within n reaction events [38](XIV). We now define event M_d as the one where $\hat{x} - \hat{y}$ decreases in total by at most $\gamma \sqrt{n \lg n}$ within $d = n$ reaction events. Let $Q_1, \ldots, Q_r$ denote independent random variables where $0 \leq r \leq d$ taking values in set $A_r = [1, -1]$. The independent random variables Q_r satisfy the conditions of Azuma's inequality (Lemma 4.1.3) with $c_r = 2$, the expected change $\sqrt{n}$ (assuming an unbiased random walk), and function $Y = f(Q_1, \ldots, Q_d) = \max_{1 \leq r \leq d} |\sum_{i=1}^{r} Q_i|$ which gives us the maximum translation distance over d reaction events. Now, using Azuma's inequality, we can infer that $\mathbb{P}[|Y - \sqrt{n}| \geq \gamma \sqrt{n \lg n}] \leq 1/n^{\gamma^2}$. It means that in our unbiased random walk the maximum distance from the origin is not more than $\gamma \sqrt{n \lg n}$ with high probability which leads to $\mathbb{P}[M_d] \leq 1 - 1/n^{\gamma^2}$. □

Lemma 4.3.7. *Let I be any interval of ϵy_0 reaction events of a computation of Single-B, where $\epsilon \leq \frac{1}{\alpha+1}$, $\alpha \geq 20$, and y_0 is the initial value of y at the beginning of I. Let b_0, b_e, b_{min}, b_{max} denote the initial, final, minimum, and maximum values of b in the interval I (similarly for y). Then for any $\gamma > 0$, there is a constant l_γ such that if $y_0 \geq l_\gamma \lg n$, then the following bounds hold with probability at least $1 - 1/n^\gamma$.*

(a) *(Upper bounds) If $b_0 \leq \min(n/2, \alpha y_0)$ then $b_{\max} \leq 2\alpha y_{\min}$ and $b_e \leq \alpha y_e$.*

Proof. We prove tighter upper bounds on b as follows.

Upper bounds: It is easy to verify that while $b_0 \leq (\alpha - 1)y_0$, upper bounds $b_{\max} \leq 2\alpha y_{\min}$ and $b_e \leq \alpha y_e$ completely hold even if y reaches to its minimum of $y_0 - \epsilon y_0$ and b increases to its maximum of $b_0 + \epsilon y_0$ within ϵy_0

reaction events. Therefore, for simplicity we just consider the case where $b_0 > (\alpha - 1)y_0$. The number of B molecules can only increase via reactions (0'x) and (0'y). We can bound the probability that a reaction event would be equal to one of these two reactions as follows.

$$\begin{aligned}\mathbb{P}[\text{Reaction 0'x or 0'y ocurrs}] = \frac{xy}{xy + xb + yb} &\leq \frac{y}{y + b} \\ &\leq \frac{y_0 + \epsilon y_0}{y_0 + (\alpha - 1)y_0 + 2\epsilon y_0} \leq \frac{1 + \epsilon}{\alpha - 2\epsilon}.\end{aligned}$$

With an application of the Chernoff upper tail bound 4.1.2, we then conclude that the probability of having more than $\frac{4(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$ reaction events of types (0'x) and (0'y), is at most $1/n^\gamma$, provided that $y_0 \geq l_\gamma \lg n$. Also, as reactions (0'x) and (0'y) have the same rate, there are at most $\frac{2(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$ reaction events of type (0'x) with probability at least $1 - 1/n^\gamma$. Thus, the number of reaction events equal to either reaction (1') or (2') is at least $\epsilon y_0 - \frac{4(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$. So, we can prove the upper bounds of our claim as follows.

case (i) $b_{\max} \leq 2\alpha y_{\min}$. Since at most $\frac{4(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$ (reactions (0'x) and (0'y)) B molecules are produced and at most $\frac{2(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$ (reaction (0'x)) Y molecules are consumed, we have:

$$b_{\max} \leq b_0 + \frac{4(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0, \text{and } y_{\min} \geq y_0 - \frac{2(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$$

which lead to $b_{\max} \leq 2\alpha y_{\min}$ for all $(\alpha - 1)y_0 < b_0 \leq \alpha y_0$.

case(ii) $b_e \leq \alpha y_e$. We note that, at the end of the interval I, at least $\epsilon y_0 - \frac{4(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$ (reactions (1') and (2')) B molecules are also consumed. Therefor, we get

$$b_e \leq b_0 + \frac{8(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0 - \epsilon y_0, \text{and } y_e \geq y_0 - \frac{2(1+\epsilon)}{3(\alpha-2\epsilon)}\epsilon y_0$$

which confirm $b_e \leq \alpha y_e$.

□

Lemma 4.3.8. *With high probability, ϵy_0 reaction events of interval I discussed in Lemma 4.3.3 happens in $\Theta(n)$ interactions.*

Proof. Using Lemma 4.2.12, it is sufficient to show that the total propensity of Single-B reactions over the course of interval I is $\Omega(y_0)$. Among $l = \epsilon y_0$ reaction events, the number of B and Y molecules can decrease by at most l, so $y \geq y_0 - l$. On the other hand, since $x \geq y$, we have $n = x+y+b \leq x+x+b$ leading to $x+b/2 \geq n/2$ throughout this interval. Putting these observations together, we obtain:

$$\rho(c, R) = \frac{xy + xb + yb}{n/2} \geq \frac{y(x + b/2)}{n/2} \geq y_0 - l = \Omega(y_0).$$

□

Lemma 4.3.9. *At any point in the computations, assuming that the $\hat{x} - \hat{y} \geq \Delta/2$, the probability that $\hat{x} - \hat{y}$ increases is at least $1/2 + \Theta(\Delta/n)$.*

Proof. Let p denote the probability of a success ($\hat{x} - \hat{y}$ increases) and q denote the probability of a failure ($\hat{x} - \hat{y}$ increases). We note that reactions (0'y) and (2') decrease the gap between $\hat{x}$ and $\hat{y}$, and reactions (0'x) and (1') increase the gap. So, given that $x \leq n$, and $y/292 < b$, we can compute the probability that $\hat{x} - \hat{y}$ increases on a reaction event as follows.

$$1)\ \frac{q = \mathbb{P}[\hat{x} - \hat{y} \text{ decreases}]}{p = \mathbb{P}[\hat{x} - \hat{y} \text{ increases}]} = \frac{1/2xy + yb}{1/2xy + xb}$$
$$\leq 1 - \frac{(\hat{x} - \hat{y})b}{1/2xy + xb} \leq 1 - \frac{(\Delta/2)b}{x(1/2y + b)} \leq 1 - \Theta(\Delta/n),$$
$$2)\ q + p = 1$$

So, with a simple calculation, equations 1 and 2 result in $p \geq 1/2+\Theta(\Delta/4n)$. □

Lemma 4.3.10. *At any point in the computation, if $\hat{y} = n/k$ then the probability that $\hat{y} > 2n/k$ at some subsequent point in the computation is less than $(1 - \Theta(1))^{n/k}$.*

Proof. Let p denote the probability of a success ($\hat{y}$ decreases) and q denote the probability of a failure ($\hat{y}$ increases). We note that reactions (0'y) and (2') increase $\hat{y}$, and reactions (0'x) and (1') decrease it. So, assuming that $x \leq n$, $\hat{x} - \hat{y} \geq n - n/4k$, and $y < 292b$, we can compute the ratio q/p on a

reaction event as follows.

$$1)\ \frac{q = \mathbb{P}[\hat{y} \text{ increases}]}{p = \mathbb{P}[\hat{y} \text{ decreases}]} = \frac{1/2xy + yb}{1/2xy + xb}$$
$$\leq 1 - \frac{(\hat{x} - \hat{y})b}{1/2xy + xb} \leq 1 - \frac{(n - 4n/k)b}{n(1/2y + b)} \leq 1 - \Theta(1)$$

By Lemma 4.1.1, we conclude that reaching a configuration where $y > 2n/k$ (which entails an excess of n/k failures to successes) is less than $(1 - \Theta(1))^{n/k}$. □

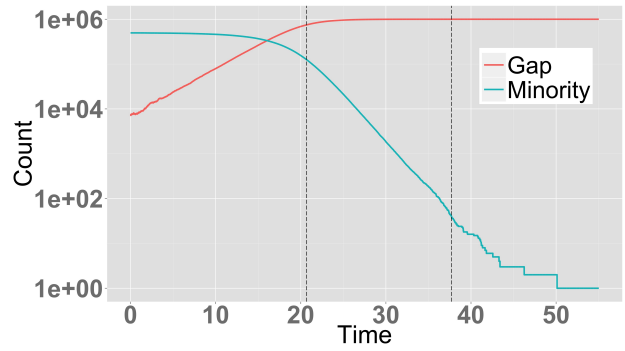
Lemma 4.3.11. *At any point in the computation, if $\hat{y} = n/k$ then, assuming that $\hat{y}$ never increases to $2n/k$, the probability that $\hat{y}$ decreases to $n/k - r$ within $f(n) > \Theta(r)$ reaction events is at least $1 - exp(-\Theta(f(n))$.*

Proof. The proof is completely parallel to the proof of Lemma 4.2.9. We only need to compute the probability of a success ($\hat{y}$ decrease). By Lemma 4.3.10, $q/p = 1 - \Theta(1)$. So, considering $p + q = 1$, it's straightforward to obtain $p \geq \frac{1}{2} + \Theta(1)$. □

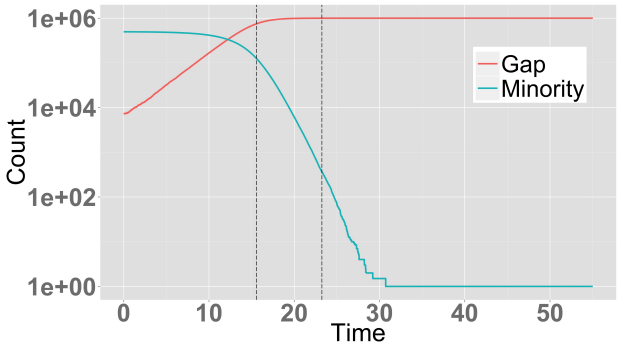
4.4 Experimental Results

Figure 4.1 illustrates the progress of computations of each of our CRNs in each of the three phases, on a single run. In the first phase, the gap $x-y$ (red line) increases steadily. Once the gap is sufficiently high, phase 2 starts and the count of y for the tri-molecular CRN, and $\hat{y}$ for the bi-molecular CRNs, decrease steadily. In the last phase, as the counts of y and $\hat{y}$ are small, there is more noise in the evolution of y and $\hat{y}$, but they do reach 0. The rate of convergence is faster for Double-B than Single-B, stemming from the fact that blanks are produced at double the rate. Figure 4.2 shows how expected time and probability of correctness (success rate) increase as a function of the volume n, for each of the CRNs. A fit to the data of that figure shows that the expected times of the tri-molecular, Double-B and Single-B CRNs grow as $3.4 \ln n$, $2.4 \ln n$, and $4.0 \ln n$ respectively[10]. For $n \geq 100$, the tri-molecular CRN has at least 99% probability of correctness and the bi-molecular CRNs have at least 97% percent probability of correctness. These probabilities all approach 1 as n gets larger.

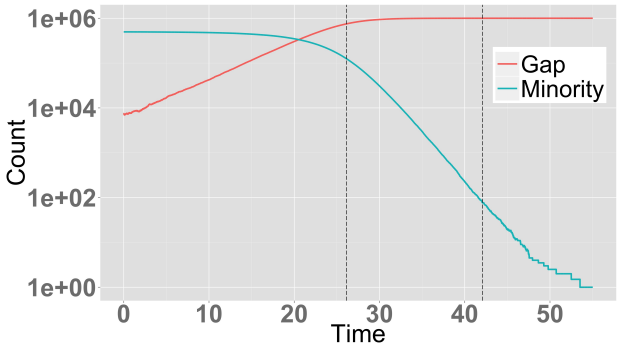
[10]We note that the x-axis is in the logarithmic scale in Figure 4.2(a). So, to compute the growing rates, we only need to find the slope a of each set of data points according to equation $y \approx a \ln x$.



(a)

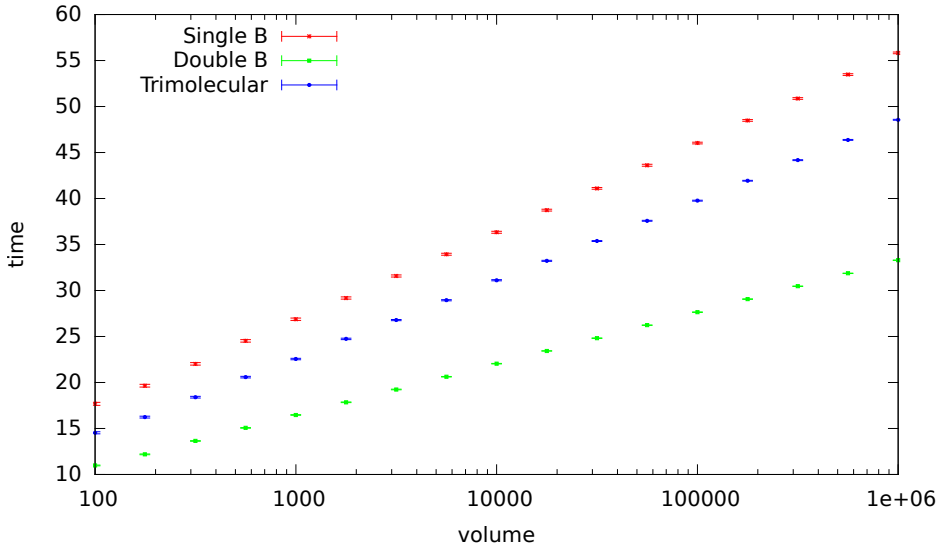


(b)

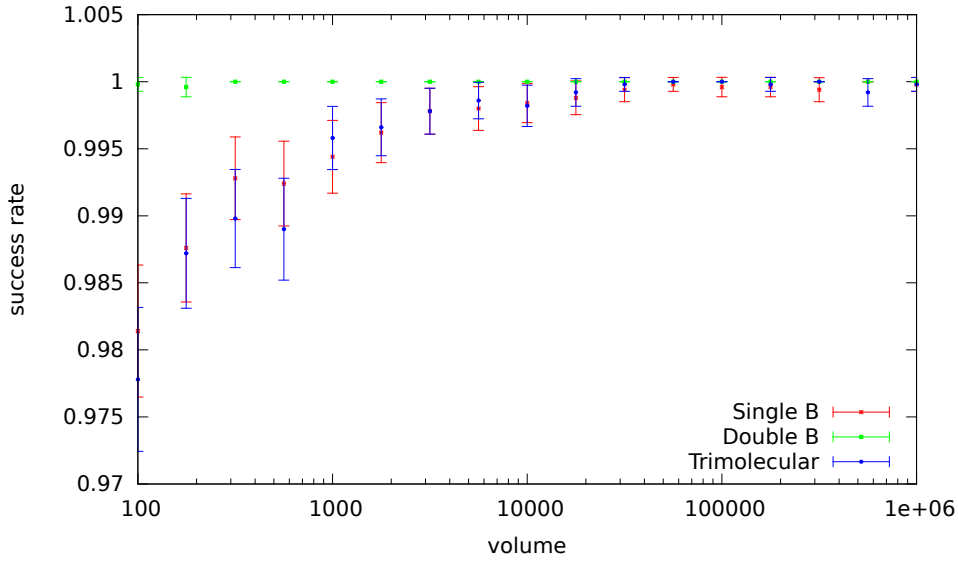


(c)

Figure 4.1: The gap $x - y$ (red line) and minority (count y for tri-molecular CRN and $\hat{y}$ for bi-molecular CRNs) (blue line), as a function of time, of sample runs of the (a) tri-molecular, (b) Double-B, and (c) Single-B CRNs. The initial count is $n = 10^6$, the initial gap $x - y$ is $2\sqrt{n \lg n}$ and parameters c_γ, d_γ and e_γ are set to 2, 8, and 2 respectively. The vertical dotted lines demark the transition between phases 1, 2 and 3.



(a)



(b)

Figure 4.2: Comparison of the time (a) and success rate, i.e., probability of correctness (b) of Single-B, Double-B and the tri-molecular CRN for Approximate Majority. Each point in the plot is an average over 5,000 trials. The initial configuration has no B's and the imbalance between X's and Y's is $\sqrt{n \ln n}$. Plots show 99% confidence intervals.

4.5 Application

As one application of our analysis methods, we show how to resolve a conjecture of Mertzios et al. [74] for the Single-B CRN (when initially $x + y = n$):

Conjecture: *For any constant $\epsilon > 0$, if initially $x - y = \epsilon n$ then the CRN reaches consensus on X-majority with probability at least $1 - \exp(-\Theta(n))$.*

Mertzios et al. [74] proved that the conjecture holds for Single-B when $\epsilon > 5n/7$. We first argue that Mertzios et al.'s conjecture holds for the tri-molecular CRN, for any $\epsilon > 0$. As in previous sections, we model the evolution of $x - y$ as a sequence of stages, where the ith stage starts when $x - y \geq 2^i \epsilon n$ for the first time, and ends when $x - y = 2^{i+1}\epsilon n$. Lemma 4.2.2 shows that, once the ith stage starts, the probability that $x - y$ reduces to $2^{i-1}\epsilon n$ is at most $\exp(-\Theta(n))$. Lemma 4.2.7 shows that, assuming that $x - y$ does not reduce to $2^{i-1}\epsilon n$, the stage does indeed end, i.e. $x - y$ reaches $2^{i+1}\epsilon n$, with probability $1 - \exp(-\Theta(n))$. In these applications of Lemmas 4.2.2 and 4.2.7, the constant in the Θ depends on ϵ. Since consensus to X-majority is reached in $O(\lg n)$ stages, the overall success probability is at least $1 - \exp(-\Theta(n))$.

To prove Mertzios et al.'s conjecture for Single-B, we show that if $x - y = \Delta \geq \epsilon n$ (and $x + y + b = n$) then for some sufficiently large constant k that depends on ϵ, the probability that $x - y$ increases to $\min\{2\Delta, n\}$ within kn reaction events is at least $1 - \exp(-\Theta(n))$ (where the constant in the Θ depends linearly on ϵ^2/k). From this property, the conjecture follows by modeling the evolution of $x - y$ as a sequence of stages exactly as for the tri-molecular CRN.

Let #(0'x), #(0'y), #(1') and #(2') be random variables denoting the number of reactions of type (0'x), (0'y), (1') and (2') respectively at any point in the sequence of kn reaction events. Azuma's inequality 4.1.3 tells us that throughout the computation, with probability at least $1 - \exp(-\Theta(n))$,

$$|\#(0'\text{x}) - \#(0'\text{y})| \leq \Delta/6. \tag{4.5.1}$$

So we'll assume in what follows that (4.5.1) holds. Then at any point in the computation,

$$x - y = \Delta + (\#(1') - \#(2')) + (\#(0'\text{x}) - \#(0'\text{y})) \geq 5\Delta/6 + (\#(1') - \#(2')).$$

A slight variant of Lemma 4.2.2 shows that the random walk #(1') − #(2') reduces to $-2\Delta/3$ at some point in the computation with probability at most $\exp(-\Theta(n))$. Similarly, a variant of Lemma 4.2.7 then tells us that,

assuming that #(1') − #(2') never reduces to $-2\Delta/3$, and thus $x - y$ never reduces to $\Delta/2$, with probability at least $1 - \exp(-\Theta(n))$, $x - y$ increases to $\min\{2\Delta, n\}$ within $k'n$ reactions of type #(1') or #(2'), for a sufficiently large constant k'. To complete the argument, note that any sequence of $kn = 2k'n + n$ reaction events of Single-B must contain at least $k'n$ reactions of types #(1') or #(2'). Otherwise the reaction sequence would contain at least $k'n + n + 1$ reactions of types #(0'x) and #(0'y), and these reactions would consume $k'n + n + 1$ X's and Y's in total. This is more than the sum of the number of X's and Y's present at the start of the sequence of reactions (at most n) plus the number of X's and Y's produced during the sequence of reactions (at most $k'n$), a contradiction.

4.6 Conclusion

Angluin, Aspnes, and Eisenstat proposed a simple population protocol for Approximate Majority and proved correctness and $O(\log n)$ time efficiency with high probability, given an initial gap of size $\omega(\sqrt{n}\log n)$ when the total molecular count in the mixture is n. Motivated by their intriguing but complicated proof, we have provided simpler, more intuitive proofs of correctness and efficiency for three different CRNs for Approximate Majority. Key to our approach has been to first analyze a tri-molecular CRN with just two reactions and two species. We have then showed how two bi-molecular CRNs, including that of Angluin et al., are essentially simulations of the tri-molecular CRN. Our results improve on those of Angluin et al. in that they hold even with an initial gap of $\Omega(\sqrt{n\log_2 n})$. Our analysis approach, which leverages the simplicity of a tri-molecular CRN to ultimately reason about bi-molecular CRNs, may be also useful in analyzing other CRNs.

Chapter 5

Summary and Future Work

5.1 Summary

In Chapter 3 we have proposed a new design to deterministically compute semilinear functions with CRNs using no leader species. We were motivated by the intriguing question of Chen et al. [21] who asked about the possibility of such computations. Similar to Chen et al.'s framework [21], we have also decomposed the semilinear function into a finite union of affine partial functions. We have then provided leaderless CRNs to compute each affine function. Our CRN construction differs from the affine-function computing CRNs of Chen et al. [21] in that we only use the input species (and no leader species) to compute the offset and coefficients of each affine partial function. Lemma 3.2.1, is in fact our primary technical contribution. Next, in order to decide which affine function should be applied to a given input, we have employed the leaderless semilinear predicate computation of Angluin et al. [11]; this latter part of the construction is actually identical to the construction of Chen et al. [21], but we have included it because our time analysis is different. Assuming n is the number of molecules present initially, we have proved that our construction ends in expected time $O(n)$ which is faster than the $O(n \log n)$ expected time bound on the direct construction (with the use of leaders) of Chen et al. [21], but slower than the $O(\log^5 n)$ expected time bound on the fast construction of Chen et al. [21] which relied heavily on the use of a fast, error-prone CRN that computes arbitrary computable functions, and which crucially uses a leader.

In Chapter 4, first we have analyzed our tri-molecular CRN, shown in Figure 1.2(a) which computes Approximate Majority. We have studied the CRN in three phases. Recall that x and y denote the number of copies of X and Y during a CRN computation. In the first phase we have modeled the evolution of the gap $x-y$ as a sequence of random walks with increasing bias of success (i.e., increase in $x - y$). Similarly, in the second phase we model the evolution of the count of y as a sequence of random walks with increasing bias of success (decrease in y). We have used a simple biased random walk analysis to show that these walks make forward progress with high probabil-

ity, thereby ensuring correctness. To show efficiency of each random walk, we have modeled it as a sequence of independent trials, observed a natural lower bound on the probability of progress, and applied Chernoff bounds. In the third and last phase we have modeled the "end game" as y decreases from $\Theta(\log n)$ to 0, and applied the random walk analysis and Chernoff bounds a final time to show correctness and efficiency, respectively. For the Double-B CRN, we have showed that with high probability, after a short initial start-up period and continuing almost until consensus is reached, the number of B's (blanks) is at least proportional to y and is at most $n/2$, in which case reaction events are reactions (1') or (2') with probability $\Omega(1)$. Moreover, blanks are in a natural sense a proxy for $X + Y$ (an interaction between X and Y), and so reactions (1') and (2') behave exactly like the two reactions of our tri-molecular CRN. Essentially the same argument applies to Single-B. However, we have been also able to provide a tighter upper bound (i.e., proportional to y) on the number of blanks. We didn't concern ourselves with smaller initial gaps. But note that even with no initial gap we can still expect efficiency, since the expected number of interactions until a gap of $\sqrt{n \log n}$ is reached is $O(n \log n)$. This would be true even if there were no bias in favour of reaction (1') as x, the majority species, increases. We suspect that the complexity of Angluin et al.'s proof stems from the case when the initial gap is small ($O(\sqrt{n \log n})$), and the fact that they show efficiency with high probability, rather than expected efficiency for the cases with small enough gaps.

5.2 Future Work

We discuss our future work pertaining to the first part of this thesis.

5.2.1 Deterministic Computation With CRNs

We identify two general directions for future work in the context of deterministic computations.

Time Complexity The clearest shortcoming of our leaderless CRC, compared to the leader-employing CRC of Chen et al. [21], is its time complexity. Our CRC takes expected time $O(n)$ to complete with n input molecules, versus $O(\log^5 n)$ for the CRC of Theorem 4.4 of Chen et al. [21]. However, we do obtain a modest speedup ($O(n)$ versus $O(n \log n)$), compared to the *direct* construction of Theorem 4.1 of Chen et al. [21]. The indirect construction of

Theorem 4.1 of Chen et al. [21] relied heavily on the use of a fast, error-prone CRN that computes arbitrary computable functions crucially using a leader. The major open question is, for each semilinear function $f : \mathbb{N}^k \to \mathbb{N}^l$, is there a leaderless CRC that stably computes f on input of size n in expected time $t(n)$, where t is a sublinear function? Belleville et al. [16] very recently showed that, a wide range of semilinear functions and predicates, satisfying some conditions, require linear time to be deterministically computed. However, the optimal computing time of the other semilinear functions and predicates not satisfying those conditions, still remains as an open question.

If this is not possible for all semilinear functions, another interesting open question is to precisely characterize the class of functions that can be stably computed by a leaderless CRC in polylogarithmic time. For example, the class of linear functions with positive integer coefficients (e.g., $f(x_1, x_2) = 3x_1 + 2x_2$) has this property since they are computable by $O(\log n)$-time unimolecular reactions such as $X_1 \to 3Y, X_2 \to 2Y$. However, most of the CRN programming techniques used to generalize beyond such functions seem to require some bimolecular reaction $A+B \to \ldots$ in which it is possible to have $\#A = \#B = 1$, making the expected time at least n just for this reaction.

Tolerance to Imprecise Inputs Despite the fact that removing the assumption of initial leader species makes the model more realistic, it remains an unrealistic aspect of the model. We have assumed the ability to prepare an initial state with precisely specified counts of input molecules. It is certainly equally as difficult to prepare a solution with 999,999 molecules of X, ensuring that the solution does not contain 1,000,000 molecules, as to ensure that the solution contains 1 molecule of leader L and not 2. However, it is not clear how to properly formalize the question, "What computations can CRNs do when initial states can only be approximately specified?" If we imagined, for instance, being able to prepare counts only to within k bits of precision for some constant k, then at most 2^k different values of a given input could be specified.

Rather than discussing errors of specification and approximate initial counts, there is an alternative way to formalize the idea that with large amounts of molecules, we lose control over individual counts. This is to use the continuous (mass-action) model, in which the amount of a species is given by a nonnegative real number indicating its average count per unit volume in an infinite volume solution. Even with the ability to specify a precise initial state (vector of real-valued concentrations), without control

over the rates of reactions, only continuous piecewise linear functions can be computed [22]. Because these functions are continuous (in fact, uniformly continuous, i.e., rates of change of the output with respect to input are bounded by a constant), a small error in specifying an initial state provably leads only to a small error in reporting the output. Therefore, continuous CRNs are in a sense already robust to imprecise inputs, merely because they can only compute functions that are naturally "error-tolerant".

Contrast this to the case of the discrete model and the semilinear functions they compute, such as the function $f(n) = n$ if n is even and $f(n) = 0$ otherwise. Here, a small change in the input causes an arbitrarily large change in the correct output, and hence an arbitrarily large error in the reported output if the initial state is specified incorrectly. Thus, given the theoretical ability of discrete CRNs to compute functions lacking the natural robustness of continuous functions to small errors in inputs, it remains an open problem to formalize how such a CRN might compute such nonrobust functions in a robust way.

5.2.2 Approximate Majority

There are several ways in which we can extend our results. Angluin et al. [12] analyze settings in which (i) some agents (molecules) have Byzantine, i.e., adversarial, behaviour upon interactions with others, (ii) some molecules are "activated" (become eligible for reaction) by epidemic spread of a signal, and (iii) there are three or more species present initially and the goal is to reach consensus on the most populous species (multi-valued consensus). We believe that our techniques can be generalized to these settings.

There are other ways in which we might generalize our results, motivated by practicalities of molecular systems. When a CRN is "compiled" to a DNA strand displacement system, it may be that the DNA strand displacement reaction rate constants closely approximate, but are not exactly equal to, the CRN reaction rates. It could be helpful to describe how the initial gap needed to guarantee correct and efficient computations for Approximate Majority with high probability depends on the uncertainty in the rate constants. It could also be useful to analyze variants of the CRNs analyzed here, or other CRNs, in which some or all of the reactions are reversible. For example, if the blank-producing reaction (0') of Double-B is reversible, the CRN appears to still be both correct and efficient, while having the additional nice property that stable B-majority is no longer possible. Again, we believe that our analyses can generalize to these scenarios.

Another interesting problem to investigate, is the application of Approx-

imate Majority in solving and analyzing the *fast probabilistic leader election* problem using CRNs, i.e., in volume n with an initial configuration with n copies of species X and no other species initially present, produce a single copy of a species L in time $O(\log n)$ with high probability. We have suggested such a leader election CRN using Approximate Majority and our experimental results are very promising. However, the proof of its correctness and efficiency is still very challenging. Our hope is that our new analysis of Approximate Majority would be a big step toward simplifying the analysis of our proposed leader election CRN.

Part II

On Prediction of Nucleic Acid Folding Pathways and Structures

Chapter 6

Introduction

RNA and DNA are both nucleic acids; the role of DNA in the cell is to serve as an information storage channel available to cellular computing networks [93]. RNA molecules are involved in many cellular functions, including DNA transcription into mRNA, translation of RNA into proteins by tRNA [61], control of the plasmid copy number in *Escherichia coli* [41] and gene regulation via mechanisms that degrade mRNA [13]. The possible splice isoforms of the mRNA transcript are also partly regulated by RNA molecules [103]. In order to determine RNA/DNA functions, it is an important step to recognize their structures [4, 80]. Moreover, nucleic acid sequences are the basis of DNA computing and molecular programming for construction of nano-devices such as DNA origami [40, 48] and DNA strand displacement (DSD) systems [87]. As dicussed in the first part of this thesis, DSDs are also promising components to pysically implement even hypothetical chemical reaction networks [95], which abstract details about displacements. Therefore, identifying RNA and DNA structures is also a fundamental step in the design, programming, and verification of these systems. Determining the nucleic acid structure using experimental methods, such as NMR and X-ray crystallography, is expensive, time-consuming and in some cases impractical [37]. Therefore, computational methods are widely used to help understand the structure and function of DNA and RNA molecules.

Nucleic acid folding pathways describe how RNA and DNA molecules fold in on themselves via intra-molecular interactions. RNA/DNA molecules dynamically move through a sequence of intermediate structures, when folding into their functional structures, i.e., three-dimensional shapes. In some cases these intermediate structures also contribute to the biological function of the molecule. For example, the function of the flavinmononucleotide riboswitch depends on this molecule's ability to change its structure [106]. In other examples, a molecule may be bistable, i.e., have two stable functional structures [39, 106]. The folding pathways of DNA molecules can help determine gene transcription rates or control the DNA strand displacement kinetics [112]. In fact, DSDs are designed so that a sequence of strand displacements follows a DNA folding pathway.

While it would be ideal to have the full description of nucleic acid folding dynamics, nucleic acid secondary structure – a set of bonds formed between nucleotides in an RNA/DNA molecule (see Section 6.1) – already provides a level of description that yields much insight into its thermodynamics and folding kinetics [100, 112]. Therefore, we also focus on nucleic acid secondary structure in our research.

In recent years, computational methods for nucleic acid folding pathways simulation have received increasing attention, since they can be very helpful in designing nano-scale machines that have potential health applications [104, 112] and they can also provide significant insights into RNA/DNA folding dynamics. For example, there is a designed RNA that can detect a cancer mutation and activate the cell death pathway [104]. In another example, designed RNAs were used to map simultaneous RNA expression patterns in intact biological samples [27].

Motivated by all of these roles of nucleic acid structures and folding pathways, in this part of the thesis, we aim to contribute to computational methods helpful for improving the folding pathway and structure prediction of nucleic acids. In the remainder of this chapter, we first provide some background on nucleic acids. We then describe related work on nucleic acid folding pathway and structure prediction and introduce the problems that are our focus in sections 6.2 and 6.3 respectively. We continue to summarize our objectives and contributions in Section 6.4 and provide an outline of the remaining chapters in Section 6.5.

6.1 Background on Nucleic Acids: RNA/DNA Molecules

A single DNA or RNA strand is a sequence of nucleotide bases, which we represent using the character set $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{T}\}$ or $\{\texttt{A}, \texttt{C}, \texttt{G}, \texttt{U}\}$ respectively[11], with the left end of the sequence corresponding to the $5'$ end of the strand and the right end corresponding to the $3'$ end. Here and throughout, let n denote the length of RNA/DNA sequences. If sequences are indexed consecutively starting from 1, we can represent a base pair as a tuple (i, j), such that $i < j - 1$, which specifies that the base at position i in the sequence is paired with the base at position j (where j is not consecutive with i). A secondary structure is simply a matching of strand positions that agrees with the *Watson-Crick* base pairs [32], namely C–G, A–T for DNA and C–G,

[11] The letters A, C, G, U, and Tstand for Adenine, Cytosine, Guanine, Uracil, and Thymine nucleotides respectively.

A–U for RNA where no position is in two pairs. That is, if (i, j) and (i', j') are in the structure then i, j, i' and j' are all distinct. We note that wobble pairs, i.e., non-Watson-Crick G–U pairs, are also frequent in RNA structures and we consider them in our analysis of RNA kinetics. The secondary structure is *pseudoknot-free*, if no base pairs cross, and more formally, if neither $i < i' < j < j'$ nor $i' < i < j' < j$, for all tuples (i, j) and (i', j'). If a secondary structure is not pseudoknot-free, it is said to have a *pseudoknot*. Figure 6.1 shows planar representations of two possible pseudoknot-free secondary structures for an RNA sequence of 23 nucleotides. From now on, when we use structure, we refer exclusively to secondary structure.

Each secondary structure is composed of motifs, which have an associated free energy value. The free energy of a secondary structure then is calculated as the sum of its motifs' energies.

At first, the free energy of a secondary structure was computed for a simple "base pair" energy model [77, 78, 114] in which the free energy of the structure was only dependent on the number and types of its base pairs. However, since the mid-1990s, more sophisticated thermodynamic energy models have been developed that account for entropic loop penalties, stacked pairs and other structural motifs [64, 68–70]. For example, in Figure 6.1, the energy of the right structure is -8.70 kcal/mol, using the Turner energy model [68].
RNA/DNA molecules tend to fold into their minimum free energy (MFE) structure at equilibrium [101]. A nucleic acid partition function is also defined as a sum of the free energy over all possible structures[12].

6.2 Nucleic Acid Folding Pathways

We start this section with an informal description of nucleic acid folding pathways and population kinetics. We then briefly discuss available methods and challenges in computing the population kinetics. The readers can find the formal and detailed description of population kinetics estimation in Chapter 7.
Throughout, we only discuss RNA sequences as our arguments about DNA folding pathways follow the same principle.

A *folding pathway*, from an initial structure to a final one, is defined as a sequence of secondary structures where each successive secondary structure differs from the previous one by a single base pair. Figure 6.2 shows a

[12]The partition function algorithm proposed by Mathews [69] can also predict a MFE structure with highly probable base pairs.

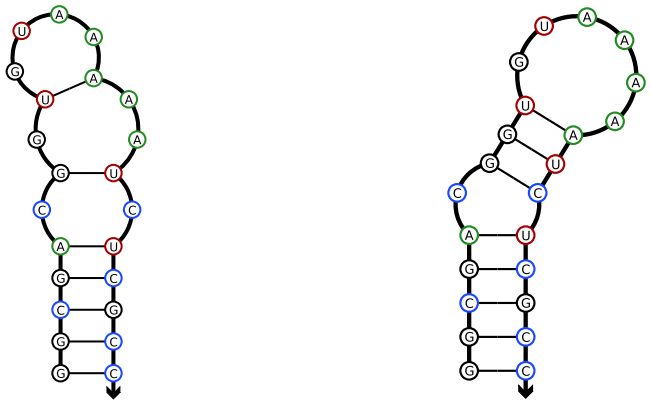


Figure 6.1: Two RNA secondary structures drawn using NUPACK [110]. These are two out of the many possible secondary structures.

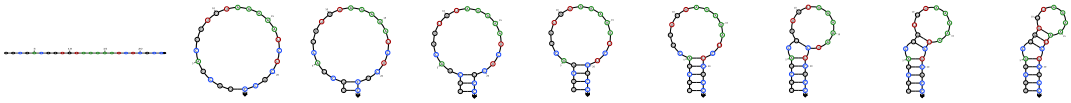


Figure 6.2: An example of a folding pathway for the RNA sequence in Figure 6.1. Each structure has been drawn using NUPACK [110].

folding pathway for the RNA sequence represented in Figure 6.1 starting from its unfolded structure and ending at its minimum free energy (MFE) structure. If the holding times, i.e., the elapsed times in transitioning from one structure to the next, are also included in a folding pathway, the obtained sequence is called a *folding pathway trajectory*. If we were to imagine a very large number of copies of the same RNA, say all initially in the unfolded state, and observe a folding pathway for each copy, then we could compute the fraction of copies occupying each secondary structure at each time point. The vector of these fractions over time is the *approximate population kinetics* of the RNA molecule (with respect to the initial unfolded state). For a given time point, the *exact* population kinetics are essentially the diffusion limit of the folding pathway simulation at that time point, i.e., when the number of copies of the RNA is taken to infinity [76].

RNA population kinetics were initially researched as a way to enhance prediction of the functional secondary structure of RNA molecules [2, 67, 96]. Structure prediction could be improved by predicting structures that are very common in the population kinetics but that are not necessarily the

thermodynamically most favourable, i.e., MFE structure [88]. Considering the dependency between the quality of MFE-based secondary structure prediction and the quality of the thermodynamic energy parameters used, it is also the case that the quality of population kinetics prediction depends on the kinetic rates used by the predictive model. Work that relies on or predicts these rates are grouped into two broad categories: 1) work at thermodynamic equilibrium without force acting on the system [23, 53, 112] and 2) work at a thermodynamic non-equilibrium where a force *is* acting on the system [66, 102].

Methods such as the Multistrand Simulator [90] and Kinfold [39] can simulate folding pathways of RNA/DNA molecules in the examples described as our motivations. For a given input RNA molecule and initial structure, these methods model the RNA folding process by continuous-time Markov chains (CTMCs). A model of CTMC can be thought of as a timed random walk on a directed graph. There are probabilities of transition associated with the edges of the graph, and the holding time is exponentially distributed with a rate depending on the current node. To approach inference in CTMCs representing RNA folding process, these two methods perform Monte-Carlo simulations (or execute the Doob-Gillespie algorithm [42]) of folding pathway trajectories, where the underlying state space is the set of all possible secondary structures for the input molecule and kinetic rates determine holding times, and transition probabilities from one secondary structure to another. Since it seems quite difficult to obtain the necessary kinetic rates experimentally, in these simulators the kinetic transition rates are derived from nearest neighbor thermodynamic parameters such as the Turner energy model [68].

The simulation model, Kinfold, of Flamm et al. [39], has all secondary structures of an input sequence in its state space, and we refer to it as the *full model*. The full model can also be used to infer exact or approximate population kinetics (see Section 7.1). However, inferring the population kinetics from the full model is slow, in part because of the size of the underlying state space. For an RNA sequence of length n, there can be as many as $\binom{n}{2k}$ ways to create a secondary structure of k base pairs, and thus $O(3^n)$ possible secondary structures. This means that the size of the state space for the full model will be exponentially related to the length of RNA, which even renders the Monte Carlo simulations (or the Doob-Gillespie algorithm [42]), non-practical for longer RNAs (e.g., with length > 30) or multi-stranded RNA sequences (i.e., an RNA sequence composed of multiple interacting strands). This difficulty prompted us to investigate an alternative for the classic Monte Carlo methods in order to efficiently estimate the population

kinetics of longer RNAs. We discuss our objectives and contributions on this matter in Section 6.4.1.

6.3 Nucleic Acid Secondary Structure Prediction

As explained earlier, computational methods are widely used to obtain a better understanding of the structure and function of nucleic acids. In this area, a central challenge has been reliable prediction of nucleic acid secondary structure. In both biological and molecular computing contexts, thermodynamic analyses are widely used to predict secondary structures. Much work has focused on prediction of pseudoknot-free secondary structures, since such structures are common in both biological and designed systems and since pseudoknot-free structures are easier to handle algorithmically [54, 63, 71]. Here and throughout, we consider a method to be efficient if its running time is bounded by a fixed-degree polynomial in the total length of the strands in the multi-set.

In what follows, we briefly summarize significant contributions on the development of algorithms for predicting the pseudoknot-free secondary structure of a single nucleic acid strand, or of multiple interacting strands. Table 6.1 also presents a summary of the time complexity of pseudoknot-free secondary structure and partition function prediction (our contribution in this thesis is shown in bold). When the input has multiple strands, we separate the cases where the number of strands is bounded by a fixed constant c, and when the number of strands is unbounded, i.e., can grow with the input size.

For single strands with length n, $O(n^3)$ dynamic programming algorithms have long been used to efficiently predict minimum free energy (MFE) pseudoknot-free secondary structures, first for a simple base pair model and later for more sophisticated energy models [77, 78, 114]. However, very recently, Bringmann et al. [17] proposed a truly sub-cubic algorithm to predict MFE secondary structures for a simple base pair energy model. Dynamic programming methods can also be used to efficiently calculate the partition function for a given strand, making it possible to compute the probability of base pair formation in equilibrium [72].

In addition to prediction of secondary structure of single strands, there has also been much interest in prediction of complexes that result when base pairs form between two or more strands. Such predictions can be used to understand the affinity of binding between a nucleic acid oligonucleotide and its potential target in biological processes such as RNA interference [107].

Input Type	MFE	Partition Function
Single Strand	$O(n^3)$ [77, 78, 114]	$O(n^3)$ [72]
Multiple Strands, Bounded ($\leq c$)	?	$O(n^3(c-1)!)$ [32]
Multiple Strands, Unbounded	**APX-hard**	?

Table 6.1: Computational complexity of predicting nucleic acid MFE pseudoknot-free secondary structures and partition functions, when the input is a single strand, multiple strands with a constant bound c on the number of strands, and multiple strands where the number of strands can grow with the input length n. In each case, n is the total number of bases in the input strand(s). We note that, for a single strand, a new work by Bringmann et al. [17] presents an exact sub-cubic algorithm using a simple base pair model. The bold term shows our contribution and the question marks show that the complexity of the corresponding problems is as yet unresolved.

Prediction of multi-stranded secondary structures is also important, because methods for biomolecular programming and construction of nano-devices, such as algorithmic self-assembly, DNA strand displacement systems and DNA origami, are based on the formation of such complexes [40, 87]; prediction methods, such as that provided by NUPACK, [111] can guide the design of such programs and devices and be very useful in physical implementation of chemical reaction networks studied in part I of the thesis.

An energy model for single-stranded secondary structure formation can be extended to obtain a model for multi-stranded complex formation by (i) charging an additional strand association penalty, typically a constant times the number of strands involved in the complex, and (ii) accounting for rotational symmetries [32] . We explain a simple extension of the base pair energy model to operate on multi-stranded nucleic acids in Section 8.1.1. Predicting MFE pseudoknot-free secondary structures formed from two (or any constant number) of strands with respect to a model that only accounts for strand association penalties is a straightforward extension of dynamic programming algorithms for single strands [8, 109, 113]. However, it is not clear how such algorithms can efficiently account for rotational symmetries that can arise when two or more indistinguishable strands interact [32]. Nevertheless, Dirks et al. [32] showed how to efficiently calculate the partition function for a constant number of interacting molecules that form pseudoknot-free structures, by showing how rotational symmetry could be

accounted for, while simultaneously addressing algorithmic overcounting issues that arise in partition function calculation. However, the partition function calculation method of Dirks et al. requires a separate dynamic programming computation on all possible orderings of strands that interact to form a single complex. As a result, the method does not run in polynomial time when the number of participating strands grows with the overall input size (total length of strands). This situation can arise, for example, in DNA strand displacement systems. Also, surprisingly, while the partition function for a constant number of interacting strands can be calculated efficiently, it is not known how to efficiently calculate the MFE pseudoknot-free secondary structure of a constant number of interacting strands.
In summary, exact and efficient dynamic programming algorithms are well known for predicting the MFE pseudoknot-free secondary structure of a single nucleic acid strand. However, all the available methods for computing MFE pseduoknot-free secondary structure formed from a set of nucleic acid strands are either heuristic and therefore are not guaranteed to find the optimum structure or require an exponential runtime to find the exact solution. Thus, we are motivated to answer the following basic question that has remained open after over three decades of work on computational pseudoknot-free secondary structure prediction of nucleic acids: given a set of nucleic acid strands, is it possible to efficiently compute a MFE pseudoknot-free secondary structure that they can form? We discuss our objectives and contributions in more detail in Section 6.4.2.

6.4 Objectives and Contributions

Next, we give a summary of our objectives and contributions on the two different problems discussed in this part of the thesis.

6.4.1 Nucleic Acid Folding Pathways

To enhance the functional structure prediction, i.e., common structures that are not necessarily the MFE ones, of RNA molecules, it would be beneficial to have an accurate estimation of their population kinetics. RNA population kinetics indeed provide information about the probability of reaching different secondary structures at a given time. Considering an RNA molecule with a combinatorial state space, the computation of population kinetics, and hence probabilistic inference, would be difficult or impossible with the existing methods. So, we first aim to find an approach that accurately and efficiently computes the probability that an RNA molecule beginning in

one secondary structure, x, will transition in the given time, t, to a target structure, y. For this purpose, we focus on the computation of transition probabilities in general CTMCs having combinatorial state spaces, because not only this could approximate RNA population kinetics, but also it could be beneficial to many other applications as diverse as queueing theory, pylogenetics, and chemical reaction networks [51, 75].Our contributions on this matter are as follows.

1. For CTMC problems with countably infinite states, where classical methods such as matrix exponentiation are not applicable, the main solution approach has been Markov chain Monte Carlo methods for both the holding times and sequences of visited states. We propose a modified Monte Carlo method, where the holding times are marginalized analytically. Our method approaches inference in CTMCs with weak assumptions on the state space and can approximate transition probabilities as well as estimate CTMC parameters for this general class of processes.

2. We confirm our results by conducting experiments on an important example of CTMCs with combinatorial state space (i.e., a set of all secondary structures): nucleic acid folding pathways. We verify on real RNA sequences that our method outperforms the classic Monte Carlo approach for estimating the transition probabilities that marginalize over folding pathways and provide a model for the kinetics of an RNA molecule interacting chemically with itself.

6.4.2 Multi-stranded Nucleic Acid Structure Prediction

We note that MFE secondary structure prediction is still of particular interest to researchers for a better understanding of the functional nucleic acid structure. While efficient thermodynamics-based approaches are well known for prediction of pseudoknot-free secondary structures of single strands, the problem of predicting pseudoknot-free secondary structures of multiple interacting strands is not studied well enough.

Given a set of nucleic acid strands and a positive integer k, let Multi-Pkf-SSP be the problem of determining whether the strands can form a pseudoknot-free secondary structure with at least k base pairs.

In this thesis, we mainly prove that for the base pair energy model, the Multi-Pkf-SSP problem is computationally intractable. We note that hardness results can be valuable even for the simple base pair energy model;

it would seem unlikely that the prediction problem becomes easier if the energy model is more sophisticated.

Our contributions, in more detail, are as follows.

1. We show that Multi-Pkf-SSP is NP-hard, meaning that the existence of an polynomial time method for MFE pseudoknot-free secondary structure prediction of a multi-set of strands would imply all problems in the complexity class NP, which includes problems that are widely believed to be intractable, would have polynomial time algorithms.

2. Our hardness proof of Multi-Pkf-SSP uses a reduction from a variant of 3-dimensional matching (3DM), already known to be NP-hard, and employs code word designs with high pairwise edit distance [91]. We noticed an error in the proof of high pairwise distance between words that were padded (i.e., expanded by inserting a character in regular positions) given by Schulman and Zukerman [91]. So, as another contribution we fixed the issue and will ask them to publish an erratum.

3. Given the NP-hardness result above, we also prove that there is no efficient method to find a pseudoknot-free secondary structure whose energy is a close estimate of the energy of the MFE structure unless $\mathsf{NP} \neq \mathsf{P}$. Specifically, if there is a *polynomial time approximation scheme (PTAS)* that could find a pseudoknot-free secondary structure whose free energy closely approximates that of the MFE for any given multi-set of strands, then again $\mathsf{NP} = \mathsf{P}$. A PTAS is a polynomial time algorithm that receives as input an instance of an optimization problem and an arbitrary parameter $\epsilon > 0$, and returns an output whose value (in our case, the number of base pairs in the MFE structure) is within a factor $1 - \epsilon$ of the value of the optimal solution. The running time of a PTAS could be dependent on ϵ, but it must be polynomial in the input size for every fixed ϵ. Formally, we show that the optimization problem of finding the MFE structure for a multi-set of nucleic acid strands is hard for the complexity class APX (shown as "APX-hard" in Table 6.1), the class of NP optimization problems that have constant factor approximation algorithms.

6.5 Outline

We address the formal definition of RNA folding models, population kinetics, continuous time Markov chains (CTMCs) and our efficient estimation of CTMC transition probabilities in Chapter 7, Sections 7.1 and 7.2. We note that we only describe our methodology with respect to a simple CTMC setup that expresses RNA folding pathways. However, our algorithm can be extended to other situations, where some or all the parameters of the CTMC are unknown, but as it is not directly related to the scope of this dissertation we do not discuss them here[13]. Finally, in Section 7.3, we provide the experimental results on approximating population kinetics of some real RNAs employing our new method.

Chapter 8, Section 8.1 first provides preliminary definitions, problem statements, and an overview of some useful theorems to resolve the computational complexity of the MFE structure prediction for multi-stranded nucleic acid systems. Later, in Section 8.2, we outline the string properties and designs required for our NP-hardness proof. We further provide a polynomial time reduction from a variant of 3DM to Multi-Pkf-SSP in Section 8.3, and prove its correctness in Section 8.4. In Section 8.5, we also infer that an optimization version of the problem is hard for the complexity class APX.

[13]The extended version of our method can be found in the published manuscript [47].

Chapter 7

Approximating Nucleic Acid Population Kinetics

In this chapter, we first provide some background and formal definitions on RNA folding models. Equipped with this background, in Section 7.2, we present an effective method for approximating transition probabilities in general CTMCs defined over combinatorial state spaces. In Section 7.3, we then support the efficiency of our method by applying it to RNA folding models, as a powerful example of CTMCs, to estimate the population kinetics of some real RNA molecules. Later, in Section 7.4, we give a summary of our results and contributions.

7.1 Preliminaries

Here and throughout, we will restrict our attention to RNA pseudoknot-free secondary structures, a large and important class of structures[14].

A *folding pathway* of m steps consists of a sequence of secondary structures $\sigma = i_1, i_2, ..., i_m$ where each successive secondary structure differs from the previous one by a single base pair, i.e. the Hamming distance between i_{j-1} and i_j is exactly 1 for all $2 \leq j \leq m$. A *folding pathway trajectory* of m steps is a sequence of tuples $(s_0, t_0), (s_1, t_1), ..., (s_m, t_m)$ where s_i is the secondary structure at time i and t_i is the holding time for structure s_i. The collection of structures in a folding pathway or trajectory is not necessarily distinct. In other words, one structure can appear more than once along the folding pathway or trajectory.

The free energy of a secondary structure i, $E(i)$, is computed using the Turner energy model [68] and an $O(n^3)$ dynamic programming algorithm [72].

[14]We note that all models and methods in this chapter can trivially be extended to DNA sequences. Moreover, they all can also trivially apply to the case of pseudoknots by simply including psudoknotted structures in the model and using free energies of pseudoknots to derive kinetic rates. However, our emperical results depend non-trivially on the assumption of pseudoknot-free structures.

The equilibrium probability that an RNA molecule will occupy a particular secondary structure is

$$k_i = \frac{e^{-E(i)/(\rho\tau)}}{Z_k}$$

where ρ is the gas constant, τ is the temperature, $Z_k = \sum_{j \in U} e^{-E(j)/(\rho\tau)}$ is the partition function [72], and U is the set of all possible secondary structures for the molecule. The vector k is a distribution since the sum of the k_i's is one, and is known as the Boltzmann distribution.

For example $\tau = 310.15$ K is roughly body temperature. The gas constant is $\rho = 1.985 \times 10^3$ kcal/(K mol), making $\rho\tau = 6.1565 \times 10^5$ kcal/mol. The gas constant is related to the Boltzmann constant as $\rho = N_A k_B$ where k_B is the Boltzmann constant and N_A is Avogadro's number. Since $\rho\tau$ and E have the same units (i.e., kcal/mol), this makes the Boltzmann distribution unit-less.

7.1.1 RNA Folding Models

Flamm et al. [39] introduced the RNA folding model that was inspired by chemical reaction systems modelled using continuous-time Markov chains (CTMCs). A CTMC is defined via its infinitesimal generator matrix, i.e., an array of numbers representing the rate a CTMC moves between states, and an initial distribution over its states. Like discrete Markov chains, many CTMCs have a stationary distribution.

Assume that an RNA molecule is given and let U be the space of pseudoknot-free secondary structures for that molecule. Let $N(i) \subset U$ be the neighborhood of i where $\{i\} \cap N(i) = \varnothing$ and symmetry is preserved (if $j \in N(i)$ then $i \in N(j)$ for all $i \neq j, i, j \in U$). For the full model defined directly below, the neighborhood of i includes every structure that differs from i by a single base pair.

We say that secondary structures i and j are connected if there is a path of structures from i to j where each subsequent structure is a neighbor of the previous one, i.e. $i = s_1, s_2, ..., s_{\ell-1}, s_\ell = j$ such that $s_\ell \in N(s_{\ell-1})$ for all ℓ.

The Full Model [39]. Kawasaki introduced an 'inverse-symmetric' transition rule, i.e., one for which $K_{ij} = K_{ji}^{-1}$ for $i \neq j$. The Kawasaki transition

rule gives the infinitesimal generator matrix for $i \neq j$ as

$$K_{ij} = \begin{cases} e^{(E(i)-E(j))/(2\rho\tau)} & \text{if } j \in N(i) \\ 0 & \text{if } j \notin N(i) \end{cases} \tag{7.1.1}$$

and

$$K_{ii} = -\sum_{j \neq i} K_{ij}.$$

The neighborhood of i is every structure that differs from i by exactly one base pair. Notice that although Flamm et al. also included as neighbors a pair of structures that differ by shifted base pairs, we do not and only consider basic and well-known kinetic moves (i.e., base pair removal and addition). This means that every pair of secondary structures is connected. We use the 'full model' to refer to the CTMT with infinitesimal matrix K and with the initial state, x, being that distribution over the structures that the RNA molecule begins in. Typically x is the point-mass on either the open-chain or the minimum free energy state. The concepts introduced in what follows apply to any other initial state distribution.

Informally, the *population kinetics* at a given time t is a vector whose jth entry is the probability that the CTMT is in state (secondary structure) j at time t. Formally, for a CTMT with initial distribution x and infinitesimal generator matrix Q, the population kinetics is given by $x\exp(tQ)$. In the case that $Q = K$, as $t \to \infty$ the population kinetics converges to the stationary distribution given by the Boltzmann distribution k (this follows from the fact that $k\exp(tK) = k,$ for all $t \geq 0 \Leftrightarrow kK = 0$).

The Subset Model [98]. The main challenge with obtaining the matrix K is that the size of the state space U is exponential in the number of bases in an RNA molecule. For this reason, some approximations take a subset of connected structures $S \subseteq U$ [75], such that the initial (unfolded) state is in S. The choice of S is an important decision and we briefly discuss it in Chapter 9. We also follow the approach proposed by Kirkpatrick et al.[60] to verify the quality of the chosen set. Some models design a CTMC on the subset of nodes S that converges to the Boltzmann distribution normalized on the subset S of structures [98]. Next, we describe such a model.

We now employ the Kawasaki rule as described in Tang et al. [98] to infer the infinitesimal rate matrix L on the subset S.

$$L_{ij} = \begin{cases} K_{ij} & \text{if } i \in S \text{ and } j \in N(i) \cap S \\ 0 & \text{otherwise} \end{cases}$$

and

$$L_{ii} = -\sum_{j \neq i} L_{ij} \geq K_{ii},$$

where K is the rate matrix in Equation 7.1.1. Notice that since $L_{ii} \geq K_{ii}$ the process L will have slower holding times on average than does the process K. More discussion of the holding times will follow.

The process with infinitesimal rate matrix L converges to the distribution on S which is proportional to the Boltzmann distribution

$$\ell_i = \frac{e^{-E(i)/(\rho\tau)}}{Z_\ell}$$

where $Z_\ell = \sum_{j \in S} e^{-E(j)/(\rho\tau)}$.

7.1.2 Population Kinetic Calculations

There are two ways to calculate the population kinetics of an RNA molecule: either from the transition probabilities of the corresponding CTMT or from a Monte Carlo simulation.

For an arbitrary finite CTMC with infinitesimal generator matrix Q, the *transition probability* $P^Q(t)_{ij}$ is defined to be the probability that the CTMT transitions to state j at time t, given that it is in state i. The corresponding matrix of transition probabilities $P^Q(t)$ satisfies the ODE

$$\frac{dP^Q(t)}{dt} = QP^Q(t), \quad t \geq 0.$$

This ODE can be solved using an ODE solver or matrix exponentiation (see Norris [76] 2.1.1 for a proof):

$$P^Q(t) = \exp(tQ) := \sum_{j=1}^{\infty} \frac{(tQ)^j}{j!}. \tag{7.1.2}$$

From the matrix exponential, we can immediately obtain the exact population kinetics which are the vector $x \exp(tQ)$, where x is the initial distribution of the CTMC.

Computing Equation 7.1.2 can be done through matrix exponentiation methods such as spectral decomposition or the eigenvalues using the Lanczos algorithm [89]. However, these matrix exponentiation algorithms have exponential running time, since $|U|$, and thus the size of the matrix is exponential. Munsky and Khammash provide an alternative method for approximating

the matrix exponential [75].

A second way to compute the population kinetics is the Monte Carlo simulation of folding pathway trajectories from process Q. A single sample of such a simulation tracks the progress of a single RNA molecule as it moves across the secondary structure state space U. Monte Carlo simulation [42] is accomplished using the exponential holding-time distribution and the embedded Markov chain [45]:

(1) The holding time t_i is exponentially distributed with parameter $-Q_{ii}$.

(2) The *embedded (discrete) Markov chain* has transition matrix $H_{ij} = -Q_{ij}/Q_{ii}$ when $Q_{ii} < 0$ whereas $H_{ii} = 0\ \forall i$. The self-loop transition probability H_{ii} is zero to represent that self-transitions are invisible in continuous time.

Initially s_i is distributed according to the initial distribution x. Steps (1)-(2) yield a single tuple, (s_i, t_i), and a trajectory is made of m tuples, $\theta = (s_0, t_0), (s_1, t_1), ..., (s_m, t_m)$. Let Θ be the set of trajectories sampled. Given these $|\Theta|$ samples, approximate population kinetics at a given time t can be computed by calculating the fraction of samples in which the RNA molecule is in a given secondary structure at time t. Recall that the population kinetics is a vector and its i'th component is approximated as

$$(\tilde{p}(t))_i = (1/|\Theta|) \cdot \left| \left\{ s_k = i | \sum_{j \leq k-1} t_k \leq t < \sum_{j \leq k} t_k \right\} \right|$$

The approximation $\tilde{p}(t)$ converges to the exact population kinetics, $x \exp(tQ)$, as the number of samples grows [76]. This Monte Carlo simulation is the one performed by the RNA folding software Kinfold [39] with process K and with both Metropolis-Hastings and Kawasaki versions of process K.

7.2 Efficient Continuous-Time Markov Chain Estimation

In leveraging the modelling capabilities of CTMCs, the bottleneck is typically the computation of the *transition probabilities*: the conditional probability that a trajectory ends in a given end state, given a start state and a time interval. This computation involves the marginalization over the uncountable set of end point conditioned paths. We propose an efficient Monte Carlo method to approach inference in CTMCs with weak assumptions on

the state space. Our method can approximate transition probabilities as well as estimate CTMC parameters for this general class of processes. More precisely, we are interested in countably infinite state space CTMCs that satisfy the following two criteria. First, we require the construction of a certain type of potential on the state space. We describe this potential in more detail in Section 7.2.1, and show in Section 7.3 that such potentials can be easily constructed for RNA folding models, as a real example of CTMCs. Second, the CTMC should be explosion-free to avoid pathologies (i.e., it is required to have a finite number of transitions in any finite time interval with probability one).

In contrast, classical uniformization methods assume that there is a fixed bound on all the rates [44], a much stronger condition than our explosion-free assumption. Other approaches, based on performing Markov chain Monte Carlo (MCMC), relax the bounded rate assumption [85, 86], but they have a running time that depends linearly on the size of the state space in the sparse case and quadratically in the dense case.

Assume that every trajectory (or path) is considered as a particle, then particle-based methods offer an interesting complementary approach, because they have a time complexity per particle that depends on the imputed *number of transitions between the two end points* instead of on the size of the state space.

In the simplest case, one can implement this idea using a proposal distribution equal to the generative process over paths initialized at the start point. The weight of a particle is then equal to one if the end point of the generated path coincides with the observed end point, and zero otherwise. We call this proposal the *forward sampling* proposal and it exactly corresponds to the simple Monte Carlo process explained in the previous section for the approximation of RNA population kinetics.

Unfortunately, the forward sampling method has two serious limitations. First, the requirement of imputing holding times between each transition means that the proposal distribution is defined over a potentially high-dimensional continuous space. This implies that large numbers of particles are required in practice. Second, in problems where each state has a large number of successors, the probability of reaching the end state can become extremely small.

Now, we present our efficient particle-based Monte Carlo approach method where the holding times are marginalized analytically. For convenience, we call our approach *Time Integrated Path Sampling, or TIPS*.

7.2.1 Methodology

In this thesis, we only describe the simplest framework in which our method can be applied and our main contributions can be understood: computing the probability that a CTMC with known rate parameters occupies state $y \in \mathcal{X}$ at time t given that it occupies state $x \in \mathcal{X}$ at time 0, where $\mathcal{X}$ is a countable set of states. This simple framework is still very influential and completely represents the RNA folding models that are of particular interest to us. However, we note that our method is not limited to this setup and can be extended to more complicated CTMC frameworks that fall out of the scope of this thesis - see Hajiaghayi et al. [47] for more details.

Notation. Let $\nu(x, y)$ denote the transition probability from state $x \in \mathcal{X}$ to state $y \in \mathcal{X}$ given that a state jump occurs (i.e. $\sum_{y:y\neq x} \nu(x, y) = 1, \nu(x, x) = 0$). Let $\lambda(x)$ denote the rate of the exponentially-distributed holding time at state x ($\lambda : \mathcal{X} \to [0, \infty)$).[15] We only require efficient pointwise evaluation of $\lambda(\cdot), \nu(\cdot, \cdot)$ and efficient simulation from $\nu(x, \cdot)$ for all $x \in \mathcal{X}$. We start by assuming that ν and λ are fixed, and discuss their estimation later. We define some notation for paths sampled from this process. Let $X_1, X_2, \ldots$ denote the list of visited states with $X_i \neq X_{i+1}$, called the *jump chain*, and $H_1, H_2, \ldots$, the list of corresponding *holding times*. The model is characterized by the following distributions: $X_{i+1}|X_i \sim \nu(X_i, \cdot)$, $H_i|X_i \sim F(\lambda(X_i))$, where $F(\lambda(X_i)) = 1 - \exp(-\lambda\nu(X_i, \cdot))$, is the exponential distribution CDF with rate λ. Given a start state $X_1 = x$, we denote by $\mathbb{P}_x$ the probability distribution induced by this model. Finally, we denote by N the number of states visited, counting multiplicities, in the interval $[0, t]$, i.e. $(N = n) \equiv (\sum_{i=1}^{n-1} H_i \leq t < \sum_{i=1}^{n} H_i)$.

Overview of the inference method Using the simple setup introduced above, the problem we try to solve is to approximate $\mathbb{P}_x(X_N = y)$, which we approach using an importance sampling method[16] [35]. Each proposed particle consists of a sequence (a list of variable finite length) of states $\mathcal{X}^*$ starting at x and ending at y. More formally, we have

$$\mathcal{X}^* = (x_1, \ldots, x_n) \mid n \in \mathbb{N}, x_1 = x, x_n = y \text{ and } x_i \in \mathcal{X}.$$

[15] Note that this is a reparameterization of the infinitesimal generator matrix $Q_{x,y}$, with $Q_{x,x} = -\lambda(x)$, and $Q_{x,y} = \lambda(x)\nu(x, y)$ for $x \neq y$.

[16] Intuitively, when the target distribution is difficult to sample from, importance sampling can be used as an alternative approach. This sampling method specifies a new probability density function as the *proposal* distribution and draw samples from this distribution rather than drawing them directly from the target distribution.

In fact, we marginalize the holding times, hence avoiding the difficulties involved with sequentially proposing times constrained to sum to the time t between the end points.

Concretely, our method is based on the following elementary property:

Proposition 7.2.1. *If we let $\pi(x^*) = \gamma(x^*)/\mathbb{P}_x(X_N = y)$, where,*

$$\gamma(x^*) = \left(\prod_{i=1}^{n-1} \nu(x_i, x_{i+1})\right) \times \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right), \tag{7.2.1}$$

for $x^* \in \mathcal{X}^*$ and zero otherwise, where $n = |x^*|$, $X^* = (X_1, \cdots, X_N)$, and H_i's are sampled independently according to $F(\lambda(X_i))$, then π is a normalized probability mass function.

Proof. We have, for any $x^* = (x_1, x_2, \ldots, x_n) \in \mathcal{X}^*$,

$$\begin{aligned}
&\frac{\left(\prod_{i=1}^{n-1} \nu(x_i, x_{i+1})\right) \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right)}{\mathbb{P}_x(X_N = y)} \\
&= \frac{\mathbb{P}_x(X_1 = x_1, \ldots, X_n = x_n)}{\mathbb{P}_x(X_N = y)} \\
&\quad \times \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i \middle| X_1 = x_1, \ldots, X_n = x_n\right) \\
&= \frac{1}{\mathbb{P}_x(X_N = y)} \\
&\quad \times \mathbb{P}_x\left(\sum_{i=1}^{n-1} H_i \leq T < \sum_{i=1}^{n} H_i, X_1 = x_1, \ldots, X_n = x_n\right) \\
&= \mathbb{P}_x(X^* = x^* | X_N = y).
\end{aligned}$$

Since the right hand side is a conditional distribution,

$$\pi(x^*) = \mathbb{P}_x(X^* = x^* | X_N = y),$$

is indeed a normalized probability mass function. □

As our notation for γ and π suggests, we use this result as follows. First, we define an importance sampling algorithm that targets the unnormalized density $\gamma(x^*)$ via a proposal $\tilde{\mathbb{P}}(X^* = x^*)$. Let us denote the k-th particle

produced by this algorithm by $x^*(k) \in \mathcal{X}^*$, $k \in \{1, \ldots, K\}$, where the number of particles K is an approximation accuracy parameter. Each of the K particles is sampled independently according to the proposal $\tilde{\mathbb{P}}$. Second, we exploit the fact that the sample average of the unnormalized importance weights $w(x^*(k)) = \gamma(x^*(k))/\tilde{\mathbb{P}}(X^* = x^*(k))$ generated by this algorithm provide a consistent estimator for the normalizer of γ. Finally, by Proposition 7.2.1, this normalizer coincides with the quantity of interest here, $\mathbb{P}_x(X_N = y)$ – see Algorithm 1 followed by Algorithms 2, and 3. The only formal requirement on the proposal is that $\mathbb{P}_x(X^* = x^*) > 0$ should imply $\tilde{\mathbb{P}}(X^* = x^*) > 0$. However, to render this algorithm practical, we need to show that it is possible to define efficient proposals, in particular proposals such that $\mathbb{P}_x(X^* = x) > 0$ if and only if $\tilde{\mathbb{P}}(X^* = x^*) > 0$ (in order to avoid particles of zero weight). We also need to show that γ can be evaluated point-wise efficiently, which we establish in Proposition 7.2.3.

Algorithm 1 : **TIPS**(x, y, t)

Input: Two end point states x and y and a given time t
Output: The sample average of unnormalized importance weights, s/K

```
s ← 0
for k = 1, 2, ..., K do
    (L, p̃, p) ← propose(x, {y})
    Q̌ ← Q̌(L) {See Section 'Analytic jump integration'}
    n = |L| {The length of the list of states L}
    s ← s + p × (exp(tQ̌))_{1,n}/p̃
end for
return s/K
```

Algorithm 2 : **propose**(x, A)

Input: A start state x, and a set of target states A
Output: A sampled path L, proposal transition probability $\tilde{p}$, and jump chain transition probability p

$(L, \tilde{p}, p) \leftarrow$**proposeHittingPath**$(x, A,$ false$)$
$n \sim \text{Geo}(\cdot, \beta)$ {Geometric with support 1, 2, ...}
$\tilde{p} \leftarrow p \times \text{Geo}(n; \beta)$ {Multiply by geometric probability mass function}
for $i = 2, 3, \ldots, n$ **do**
  $x' \leftarrow$ **last**(L) {Last state visited in the list L}
  $(L', \tilde{p}', p') \leftarrow$**proposeHittingPath**$(x', A,$ true$)$
  $\tilde{p} \leftarrow \tilde{p} \times \tilde{p}'$
  $p \leftarrow p \times p'$
  $L \leftarrow L \circ L'$ {Concatenation of the two lists}
end for
return $(L, \tilde{p}, p)$

Algorithm 3 : **proposeHittingPath**(x, A, b)

Input: A start state x, a set of target states A, and a boolean variable b to whether sample at least one state or not
Output: A sampled path L, proposal transition probability $\tilde{p}$, and jump chain transition probability p

$p \leftarrow 1$
$\tilde{p} \leftarrow 1$
$L \leftarrow$ **list**(x) {Creates a new list containing the point x}
for $i = 1, 2, \ldots$ **do**
  if $x \in A$ and (not(b) or $i > 1$) **then**
    return $(L, \tilde{p}, p)$
  end if
  $x'|x \sim \tilde{\mathbb{P}}(\cdot | X_{i-1} = x)$
  $\tilde{p} \leftarrow \tilde{p} \times \tilde{\mathbb{P}}(X_i = x' | X_{i-1} = x)$
  $p \leftarrow p \times \nu(x, x')$
  $L \leftarrow L \circ x'$
  $x \leftarrow x'$
end for
return $(L, \tilde{p}, p)$

Proposal distributions. Our proposal distribution is based on the idea of simulating the jump chain, i.e., of sequentially sampling from ν until y is reached. However this idea needs to be modified for two reasons. First, (*), since the state is countably infinite in the general case, there is a potentially positive probability that the jump chain sampling procedure will never hit y. Even when the state is finite, it may take an unreasonably large number of steps to reach y. Second, (**), forward jump chain sampling assigns zero probability to paths visiting y more than once.

We address (*) by using a user-specified *potential* $\rho^y : \mathcal{X} \to \mathbb{N}$ centered at the target state y (see Lemma 7.2.2 for the conditions we impose on ρ^y). For example, we used the Hamming distance, i.e., the distance between the base pairs of the current structure and the target structure, for RNA kinetics applications. Informally, the fact that this distance favours states which are closer to y is all that we need to bias the sampling of our new jump process towards visiting y.

How do we bias the proposal sampling of the next state? Let $D(x) \subset \mathcal{X}$ be the set of states that decrease the potential from x. The proposed jump-chain transitions are chosen with probability

$$\begin{aligned}&\tilde{\mathbb{P}}(X_{i+1} = x_{i+1} | X_i = x_i) = \\ &(\alpha^y_{x_i}) \left(\frac{\nu(x_i, x_{i+1}) \mathbf{1}\{x_{i+1} \in D(x_i)\}}{\sum_{x'_{i+1} \in D(x_i)} \nu(x_i, x'_{i+1})} \right) \\ &+ (1 - \alpha^y_{x_i}) \left(\frac{\nu(x_i, x_{i+1})(1 - \mathbf{1}\{x_{i+1} \in D(x_i)\})}{\sum_{x'_{i+1} \notin D(x_i)} \nu(x_i, x'_{i+1})} \right),\end{aligned} \tag{7.2.2}$$

where $\alpha^y_x = \max\{\alpha, \sum_{x'_{i+1} \in D(x_i)} \nu(x_i, x'_{i+1})\}$. We note that $\alpha > 1/2$ is a tuning parameter. We briefly discuss the sensitivity of our method to this parameter in Chapter 9. Lemma 7.2.2 guarantees that our proposal hits the target end point y with probability one.

Point (**) can be also easily addressed by simulating a geometrically-distributed number of excursions where the first excursion starts at x, and the others at y, and each excursion ends at y. We let β denote the parameter of this geometric distribution, a tuning parameter, which we also discuss in Chapter 9.

Lemma 7.2.2. *Assuming* $\alpha^y_x = \max\{\alpha, \sum_{x'_{i+1} \in D(x_i)} \nu(x_i, x'_{i+1})\}$, *under weak conditions, our proposal mechanism in Equation 7.2.2 will hit target* y *in finite time with probability one.*

Proof. We show that our proposal mechanism hits the target end point y

with probability one under the following assumptions:

1. The potential $\rho^y(x)$ takes the value zero if and only if $x = y$.
2. The potential changes by one in absolute value for all state transitions:

$$|\rho^y(z) - \rho^y(x)| = 1 \text{ for all } z \text{ such that } \nu(x, z) > 0$$

3. For all states $x \neq y$, there is always a way to propose a state that results in a decrease in potential:

For all $x \in \mathcal{X}, x \neq y,$ there exists z such that $\nu(x, z) > 0$ and $\rho^y(z) < \rho^y(x)$.

To simplify the notation, we will drop the y superscript for the remainder of this proof.

To prove that the process always hits y, it is sufficient to show that the sequence $\rho(X_n)$ is a supermartingale[17], which in our case reduces to showing that $\mathbb{E}[\rho(X_{i+1})|X_i] \leq \rho(X_i)$.

Note that the last condition ensures that the normalizer $\sum_{x_2' \in D(X_1)} \nu(X_1, x_2')$ is always positive, hence our expression of the proposal mechanism is always well defined. Note that technically, we should also require $0 < \mathbb{P}_x(\rho^y(X_2) < \rho^y(x)) \leq 1$ to ensure that the second normalizer, $\sum_{x_2' \notin D(X_1)} \nu(X_1, x_2')$, is also positive, but if this is not the case, the proposal mechanism can always be replaced by ν in these cases without changing the conclusion of the result proven here.

Using the second condition, we have:

$$\begin{aligned}\mathbb{E}[\rho(X_{i+1})|X_i] &= \alpha_{X_i}(\rho(X_i) - 1) + (1 - \alpha_{X_i})(\rho(X_i) + 1) \\ &= 1 - 2\alpha_{X_i} + \rho(X_i) \\ &\leq \rho(X_i).\end{aligned}$$

Finally, since the supermartingale $\rho(X_n)$ is non-negative, $\tilde{\mathbb{P}}(N < \infty) = 1$, we conclude that the process always hits y. □

Analytic jump integration. Now, we describe how the unnormalized density $\gamma(x^*)$ defined in Equation (7.2.1) can be evaluated efficiently for any given path $x^* \in \mathcal{X}^*$.

[17] A supermartingale is a sequence of real-valued random variables $X_0, X_1, X_2, \ldots$ with the property that for each X_i, $\mathbb{E}[X_i] < \infty$ and $\mathbb{E}[X_i|X_0, \ldots, X_{i-1}] \leq X_{i-1}$ [45](Chapter 12).

It is enough to show that we can compute the following integral for $H_i|X^* \sim F(\lambda(X_i))$ independently conditionally on X^*:

$$\begin{aligned}
&\mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq t < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right) = \\
&\int \cdots \int_{h_i>0:\sum_{i=1}^n h_i = t} g(h_1, h_2, \ldots, h_n)\, \mathrm{d}h_1\, \mathrm{d}h_2 \ldots\, \mathrm{d}h_n, \\
&\text{where} \quad g(h_1, h_2, \ldots, h_n) = \\
&\left\{\prod_{i=1}^{n-1} f(h_i; \lambda(x_i))\right\} (1 - F(h_n; \lambda(x_n))),
\end{aligned} \tag{7.2.3}$$

and where f is the exponential density function. Unfortunately, there is no efficient closed form for this high-dimensional integral, except for special cases (for example, if all rates are equal) [3]. This integral is related to those needed for computing convolutions of non-identical independent exponential random variables. While there exists a rich literature on numerical approximations to these convolutions, these methods either add assumptions on the rate multiplicities (e.g. $|\{\lambda(x_1), \ldots, \lambda(x_N)\}| = |(\lambda(x_1), \ldots, \lambda(x_N))|$), or are computationally intractable [6].

We propose to do this integration using the construction of an auxiliary, finite state CTMC with a $n+1$ by $n+1$ rate matrix $\check{Q}$ (to be defined shortly). The states of $\check{Q}$ correspond to the states visited in the path $(x_1, x_2, \ldots, x_n)$ with multiplicities plus an extra state s_{n+1}. All off-diagonal entries of $\check{Q}$ are set to zero with the exception of transitions going from x_i to x_{i+1}, for $i \in \{1, \ldots, n\}$. More specifically, $\check{Q}$ is

$$\begin{bmatrix}
-\lambda(x_1) & \lambda(x_1) & 0 & \cdots & 0 & 0 \\
0 & -\lambda(x_2) & \lambda(x_2) & \cdots & 0 & 0 \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\
0 & 0 & 0 & \cdots & -\lambda(x_n) & \lambda(x_n) \\
0 & 0 & 0 & \cdots & 0 & 0
\end{bmatrix}. \tag{7.2.4}$$

This construction is motivated by the following property.

Proposition 7.2.3. *For any finite proposed path $(x_1, x_2, \ldots, x_n)$, if $\check{Q}$ is defined as in Equation (7.2.4), then*

$$\left(\exp(t\check{Q})\right)_{1,n} = \mathbb{P}\left(\sum_{i=1}^{n-1} H_i \leq t < \sum_{i=1}^{n} H_i \middle| X^* = x^*\right) \tag{7.2.5}$$

where $\exp(A)$ *denotes the matrix exponential of* A.[18]

Proof. Let $\check{X}_1, \check{X}_2, \ldots$ and $\check{H}_1, \check{H}_2, \ldots$ denote the states and holding times respectively of a CTMC with rate matrix $\check{Q}$. The states take values in $\{1, 2, \ldots, n+1\}$, and we let $\check{\mathbb{P}}_1$ denote the path probabilities under this process conditioned on starting at $X_1 = 1$. Let $\check{N}$ be defined similarly to N (the random number of states visited):

$$\begin{aligned}(\check{N} = n) &\equiv \left(\sum_{i=1}^{n-1} \check{H}_i \leq t < \sum_{i=1}^{n} \check{H}_i\right) \\ &\equiv \left\{\omega \in \check{\Omega} : \sum_{i=1}^{n-1} \check{H}_i(\omega) \leq t < \sum_{i=1}^{n} \check{H}_i(\omega)\right\}.\end{aligned}$$

Here, $\check{\Omega}$ is an auxiliary probability space used to define the above random variables:

$$\begin{aligned}\check{X}_i &: \check{\Omega} \to \mathcal{X} \\ \check{H}_i &: \check{\Omega} \to [0, \infty).\end{aligned}$$

For all $i \in \{2, \ldots, n+1\}$, only state $i-1$ has a positive rate of transitioning to state i, therefore $(\check{X}_i = j) \subset (\check{X}_{i-1} = j-1)$ for all j (*). Using equation 7.1.2

[18] Multiplicities of the rates in $\check{Q}$ greater than one will break diagonalization-based methods of solving $\exp(t\check{Q})$, but other efficient matrix exponentiation methods such as the squaring and scaling method are still applicable in these cases.

and applying (*) inductively yield:

$$\begin{aligned}
\left(\exp(t\check{Q})\right)_{1,n} &= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n, \check{X}_{\check{N}-1} = n-1\right) \\
&\vdots \\
&= \check{\mathbb{P}}_1\left(\check{X}_{\check{N}} = n, \check{X}_{\check{N}-1} = n-1, \ldots, \check{X}_1 = n - \check{N} + 1\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n, \check{X}_1 = 1, \check{X}_2 = 2, \ldots, \check{X}_n = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \check{\mathbb{P}}_1\left(\check{X}_1 = 1, \check{X}_2 = 2, \ldots, \check{X}_n = n | \check{N} = n\right) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \prod_{i=2}^{n} \check{\mathbb{P}}(\check{X}_i = i | \check{X}_{i-1} = i-1) \\
&= \check{\mathbb{P}}_1\left(\check{N} = n\right) \\
&= \int \int \cdots \int_{h_i > 0 : h_1 + h_2 + \cdots + h_n = t} g(h_1, h_2, \ldots, h_n)\, \mathrm{d}h_1\, \mathrm{d}h_2 \ldots\, \mathrm{d}h_n.
\end{aligned}$$

□

7.3 Experimental Results

Here, we will use our method, TIPS, to approximate RNA population kinetics. More specifically, we compare the accuracy of the transition probability estimates given by our method (TIPS) to those obtained by forward sampling (FS). We used the RNA molecules shown in Table 7.1.

Sequence	Length	$\lvert U\rvert$	$\lvert S\rvert$
1AFX	12	70	-
1XV6	12	48	-
RNA21	21	~ 1100	657
HIV	23	~ 1500	266

Table 7.1: Biological RNA sequences obtained from the RNA STRAND database [7]. Symbols U and S correspond to the set of secondary structures obtained from the full model and subset model, respectively.

For each method (TIPS and FS) and molecule, we first approximated the probability $\mathbb{P}_x(X_N = y)$ that beginning in its unfolded structure x, the molecule would end, after folding time t, in its MFE structure y. We then

computed, as a reference, the probability of this transition using an expensive matrix exponential. Computing the matrix exponential on the full state space was only possible for the RNAs of no more than 12 nucleotides. For the longer RNAs, we used an RNA subset model – see Section 7.1.1 – and restricted the state space to a connected subset S of secondary structures. While our method scales to longer RNAs, we wanted to be able to compare against forward sampling and to the true value obtained by matrix exponentiation.

We note that Figure 7.1 shows the results on the subset model of RNA21 and HIV molecules, and Figure 7.1 shows the results on the full model of 1AFX and 1XV6.

We ran the experiments with a range of number of particles, $\{5^1, 5^2, \cdots, 5^6\}$, for 30 replicates on folding times from $\{0.125, 0.25, \cdots, 8\}$[19]. Here, we compare the performance of the two methods by looking at the absolute log error of the estimate $\hat{p}$ (i.e., $\text{error}(\hat{p}) = |\log \hat{p} - \log \mathbb{P}_x(X_N = y)|$) over all replicates.

Figures 7.1a, 7.1d, 7.2a, and 7.2d show the performance of the FS and TIPS methods on selective folding times, $\{0.25, 1, 4\}$. Figures 7.1b, 7.1e, 7.2b, and 7.2e show the CPU times (in milliseconds) corresponding to the minimum number of particles required to satisfy the certain accuracy level, $I = \{\hat{p} : \text{error}(\hat{p}) < 1.0\}$ on all the folding times.

The variances of FS and TIPS weights, for $5^6 = 15625$ particles, are also computed and compared on different folding times (see Figures 7.1c, 7.1f, 7.2c, and 7.2f). Note that the variance is shown in log scale in these figures.

The graphs show that our novel method TIPS outperforms FS in estimating the probability of transition from x to y in shorter folding times, since it needs many fewer particles (and correspondingly faster CPU times) than FS to be able to precisely estimate the probability. For instance, for the RNA21 molecule with folding time 0.25, FS cannot satisfy the accuracy level I given above, even with 15625 particles, however TIPS only needs 5 particles with 16 ms of CPU time to reach the same accuracy level. Similarly, the variance of our method is smaller by a larger margin.

For longer folding times in Figure 7.1, the performance of the TIPS and FS methods are comparable (in terms of the obtained errors and CPU times), slightly in favour of forward sampling. For example, for the HIV23 molecule with folding time 4.0, TIPS and FS require 5 and 25 particles, and

[19]Folding time, in this context, is a dimensionless quantity, meaning that it can be scaled. See the discussion about dimensions for the Boltzmann distribution in Section 7.1.

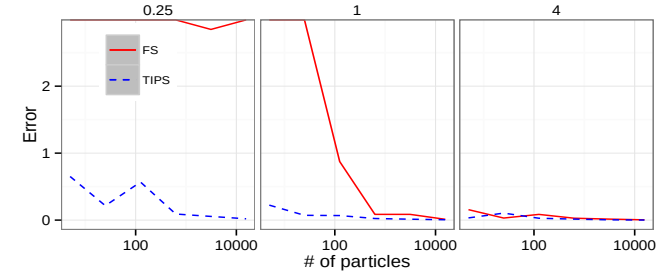
CPU times, 12 ms and 5 ms, respectively to satisfy I.

We note that the reason why FS can still perform reasonably well for longer folding times is that we picked the final end point to be the MFE structure, which has high probability under the stationary distribution. For low probability targets, FS will often fail to produce even a single hitting trajectory, whereas each trajectory sampled by our method will hit the target by construction.

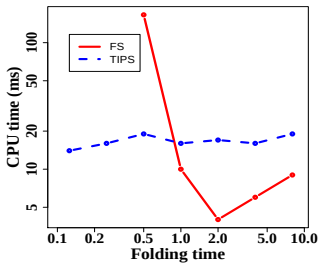
7.4 Conclusion

We have presented an efficient method for approximating transition probabilities and posterior distributions over parameters in countably infinite CTMCs. We have demonstrated on real RNA molecules that our method is competitive with existing methods for estimating the transition probabilities which marginalize over folding pathways and provide a model for the kinetics of a single strand of RNA interacting chemically with itself.

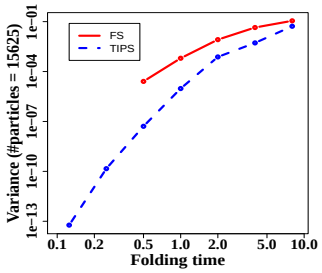
What makes our method particularly attractive in large or countably infinite state space CTMCs is that our method's running time per particle is independent of the size of the state space. The running time does depend cubically on the number of imputed jumps, so we expect that our method will be most effective when the typical number of transitions between two observations or imputed latent states is moderate (no more than approximately a thousand with current architectures). The distribution of the jump chain should also be reasonably concentrated to ensure that the sampler can proceed with a moderate number of particles. We have shown the realistic examples on RNA folding pathways where these conditions are empirically met.



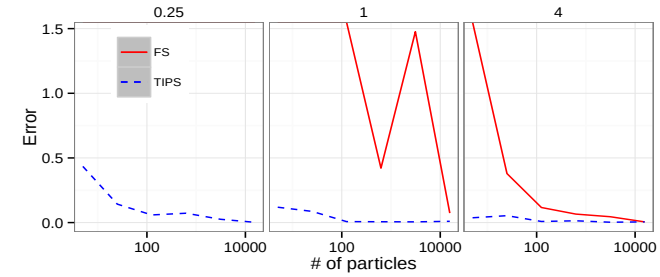
(a) RNA21- error



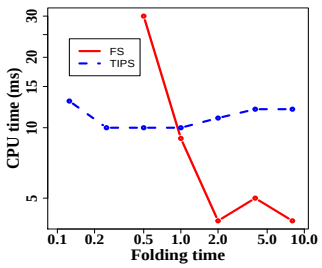
(b) RNA21- CPU time (ms)



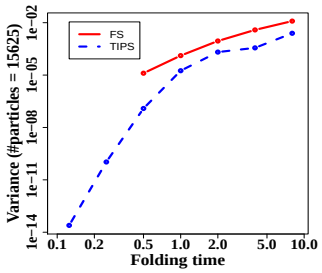
(c) RNA21- variance



(d) HIV23 - error

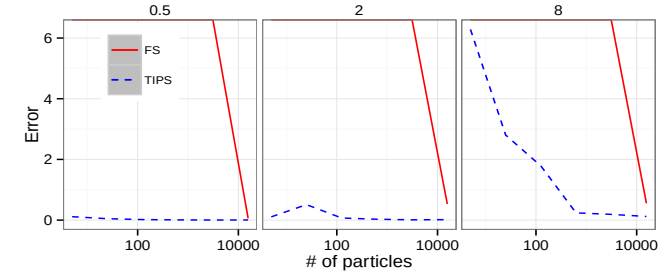


(e) HIV23 - CPU time (ms)

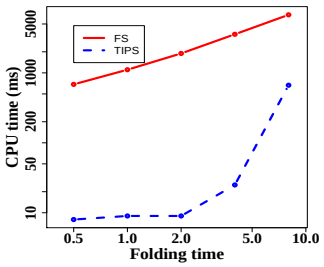


(f) HIV23 - variance

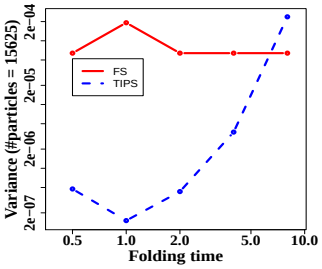
Figure 7.1: Performance of our method (TIPS) and forward sampling (FS) on RNA21 and HIV23 molecules with their *subset* state space. The relative errors of the estimates vs. folding times, $\{0.25,1,4\}$, are shown (Figures (a) and (d)) along with the CPU times corresponding to the minimum number of particles required to satisfy the accuracy level I in milliseconds (Figures (b) and (e)) and the variance of TIPS and FS estimations (Figures (c) and (f)) on folding times, $\{0.125, 0.25, \cdots, 8\}$.



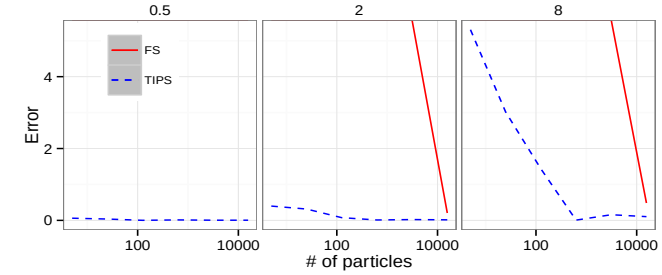
(a) 1AFX- error



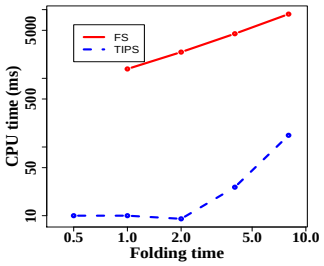
(b) 1AFX- CPU time (ms)



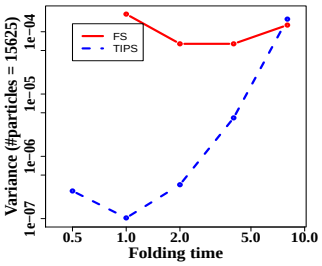
(c) 1AFX- variance



(d) 1XV6 - error



(e) 1XV6 - CPU time (ms)



(f) 1XV6 - variance

Figure 7.2: Performance of our method (TIPS) and forward sampling (FS) on 1AFX and 1XV6 molecules with their *full* state space. The relative errors of the estimates vs. folding times, {0.5,2,8} are shown (Figures (a) and (d)) along with the CPU times corresponding to the minimum number of particles required to satisfy the accuracy level I in milliseconds (Figures (b) and (e)) and the variance of TIPS and FS estimations (Figures (c) and (f)) on folding times, $\{0.5, 1, \cdots, 8\}$.

Chapter 8

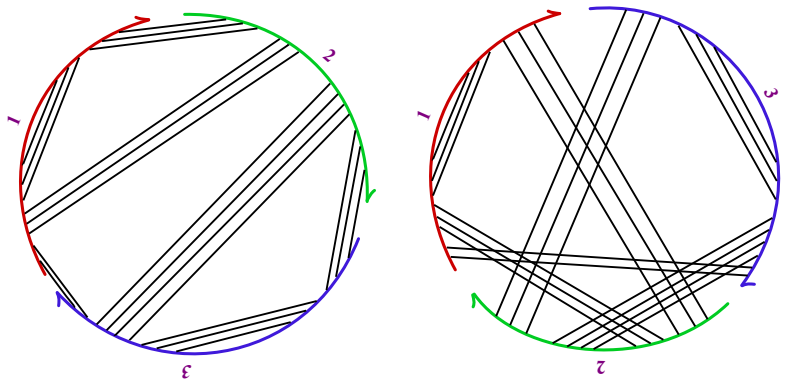
Hardness of Multi-stranded Nucleic Acid MFE Secondary Structure Prediction

In the previous chapter, we proposed an efficient method to estimate the population kinetics of RNA/DNA molecules that can contribute to a more accurate prediction of their functional structure. Here, we discuss the computational hardness of MFE structure prediction for a set of nucleic acids strands, as another important method for a better understanding of nucleic acid functions.

8.1 Preliminaries

We review some basic terminology and prior work in order to precisely formulate the problem description and proof techniques. We employ the properties described in Section 6.1 as our basis for single-stranded nucleic acids and assume that only Watson-Crick base pairs can form between nucleotide bases. Nevertheless, an RNA/DNA sequence can be composed of multiple strands. In this case, we define a secondary structure formed between multiple interaction strands as follows. Base pairing between two strands occurs in an antiparallel format. That is, the Watson-Crick complement (sometimes referred to as the 'reverse-complement') of strand $x = 5' - x_1 \cdots x_n - 3'$ is the strand $3' - y_1 \cdots y_n - 5' \equiv 5' - y_n \cdots y_1 - 3'$ where (x_i, y_i) is a Watson-Crick base pair. For example, the reverse-complement of $5' - \texttt{ACTCG} - 3'$ is $5' - \texttt{CGAGT} - 3'$. Throughout, for simplicity, we will use the term *complement* to mean reverse-complement or Watson-Crick complement and denote the complement of x by $\bar{x}$.

Similar to the single-stranded model, the secondary structure formed by m interacting strands is a set of Watson-Crick base pairs. To specify the



(a) (b)

Figure 8.1: a) Polymer graph representation of the pseudoknot-free secondary structure for the strand set {1, 2, 3} with ordering {123}, b) second polymer graph for the same set of strands with ordering {132}.

secondary structure, we assign identifiers from 1 to m to the strands, and each base is named by a strand identifier and a position on the corresponding strand. For instance if base i in strand s pairs with base j in strand t, where $s \leq t$ and $i < j - 1$ if $s = t$, the base pair is denoted as (i_s, j_t). Schematically, a multi-stranded secondary structure can be represented as a polymer graph by ordering and depicting the directional ($5'$ to $3'$) strands around the circumference of a circle and connecting the base pairs with straight lines. The number of different ways to position m strands on a circle corresponds to the set of circular permutations is $(m-1)!$ (e.g., {123} and {132} are the only orderings for three strands 1, 2, and 3 — see Figure 8.1) [32]. If there exists a polymer graph for a given secondary structure, corresponding to a circular permutation without crossing lines, then the secondary structure is called *pseudoknot-free* — see Figure 8.1a. A secondary structure consists of one or more *complexes* that correspond to the connected components in the polymer graph representation.

8.1.1 The Simple Energy Model

Here, we employ a very simple extension of the "base pair" free energy model for secondary structures [78]. In that model, the energy of each base pair is -1, and the overall free energy of a secondary structure is the total energy contribution of its base pairs. This means that a higher number of base pairs in a secondary structure of a single strand corresponds to a lower free

energy.

In a system consisting of multiple interacting strands, there is an entropic penalty for strands to associate via base pairing (i.e., a penalty for reducing the number of complexes) [32]. In this simplified energy model, we define the strand association penalty to be $K_{\text{assoc}} \geq 0$. Thus, for a pseudoknot-free secondary structure S consisting of m strands, $l \leq m$ complexes, and p base pairs, the free-energy of S is defined as $\mathbf{E}(S) = p(-1) + (m - l)K_{\text{assoc}}$. For example, the secondary structure in Figure 8.1(a) has free energy $21(-1) + (3-1)K_{\text{assoc}} = -21+2K_{\text{assoc}}$. Therefore, given a set of strands $\{s_1, \ldots, s_m\}$, an *optimal* pseudoknot-free secondary structure S_i has the property that $\mathbf{E}(S_i) \leq \mathbf{E}(S_j)$ for all $S_j \in \mathcal{S}(s_1, \ldots, s_m)$ where $\mathcal{S}(s_1, \ldots, s_m)$ is the set of all pseudoknot-free secondary structures of $s_1, \ldots, s_m$.

Since there can be a tradeoff between the number of base pairs and the number of complexes, then it is possible under this model for an optimal pseudoknot-free secondary structure to have less than the maximum number of possible base pairs. However, our proofs have been constructed so that pseudoknot-free MFE secondary structures will have a maximum number of base pairs for any reasonable value of the constant K_{assoc}. We will proceed with our problem definitions under the assumption that $K_{\text{assoc}} = 0$ and formally argue later that the results hold for all constants $K_{\text{assoc}} \geq 0$.

8.1.2 Problem definitions

We now formally define the main problem of interest in this chapter.

Problem 1. Multi-Pkf-SSP
Instance: Given m nucleic acid strands and a positive integer k.
Question: Is there a pseudoknot-free secondary structure of the m strands containing at least k base pairs?

To show hardness of our problem, we will describe a polynomial-time reduction from a restriction of the 3-dimensional matching problem to Multi-Pkf-SSP. A 3-dimensional matching is defined as follows. Let X, Y, and Z be finite, disjoint sets, and let $\mathcal{T}$ be a subset of $X \times Y \times Z$. That is, $\mathcal{T}$ consists of triples (x, y, z) such that $x \in X$, $y \in Y$, and $z \in Z$. Now $\mathcal{M} \subseteq \mathcal{T}$ is a *3-dimensional matching* if the following holds: for any two distinct triples $(x_i, y_j, z_k) \in \mathcal{M}$ and $(x_a, y_b, z_c) \in \mathcal{M}$, we have $x_i \neq x_a$, $y_j \neq y_b$, and $z_k \neq z_c$.

For convenience in our construction, we use a restriction of the 3-dimensional matching problem, called 3dm(3), that requires each element to appear in

(a) (b)

Figure 8.2: An instance of the restricted 3-dimensional matching problem (3DM(3)) where $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3\}$, $Z = \{z_1, z_2, z_3\}$. (a) The set of permitted triples, $\mathcal{T} = \{(x_1, y_2, z_2), (x_2, y_1, z_1), (x_2, y_3, z_2), (x_3, y_3, z_3)\}$. (b) A valid matching $\mathcal{M} \subseteq \mathcal{T}$.

at most three triples of $\mathcal{T}$.

Problem 2. 3DM(3)
Instance: Given $\mathcal{T} \subseteq X \times Y \times Z$, where $|X| = |Y| = |Z| = n$ and each element of X, Y and Z appears in at most 3 triples of $\mathcal{T}$.
Question: Does there exist a matching $M \subseteq T$, with $|M| = n$?

Theorem 8.1.1 (Garey & Johnson (1979) [1])**.** 3DM(3) *is* NP*-complete.*

We note that the MULTI-PKF-SSP problem is a decision problem that determines whether there exists an output structure with at least k base pairs or not. This problem can be turned into an optimization problem with a slight modification. We name this variant of the problem MAX-MULTI-PKF-SSP.

Problem 3. MAX-MULTI-PKF-SSP
Instance: Given m nucleic acid strands.
Question: Determine a pseudoknot-free secondary structure of the m strands with maximum number of base pairs.

An optimization problem is in APX if it has a constant factor approximation algorithm, i.e., an efficient method that can determine a solution within some fixed multiplicative factor of an optimal solution. A problem is APX-hard if for some constant c, a c-approximation algorithm for the problem would imply that NP = P. One way to prove a problem is APX-hard is to

show an approximation-preserving reduction from a known APX-hard problem. We derive our hardness result for the MAX-MULTI-PKF-SSP problem by a reduction from the MAX-3DM(3) problem, an optimization variant of 3DM(3).

Problem 4. MAX-3DM(3)
Instance: Given $\mathcal{T} \subseteq X \times Y \times Z$, where $|X| = |Y| = |Z| = n$ and each element of X, Y and Z appears in at most 3 triples of $\mathcal{T}$.
Question: Find a maximum size 3-dimensional matching $M \subseteq \mathcal{T}$.

Kann [58] had previously shown MAX-3DM(3) is APX-hard by showing that it is NP-hard to decide whether an arbitrary instance of the problem has a matching of size n or a matching of size at most $(1-\epsilon_0)n$, for some $\epsilon_0 > 0$ [59]. Nutov & Beniaminy [79] gave a $(1 - 1/e)$ approximation algorithm, demonstrating that MAX-3DM(3) is in APX.

Theorem 8.1.2 (Kann (1994) [58] and Nutov & Beniaminy [79])**.** MAX-3DM(3) *is* APX*-complete.*

8.2 String Designs and their Properties

In this section we show how to design strings with properties that are useful in our reduction. We follow standard string notation: for a string $a = a_1 \ldots a_n$ we denote its i^{th} character (or symbol) by a_i and its length by $|a| = n$; for any symbol B, we let B^l denote a string of length l consisting of only B's. The following related string properties are of particular interest to us.

1. A *pairwise sequence alignment*, or simply *alignment*, of strings a and b is a pair of strings (a', b') with $|a'| = |b'|$, where a' and b' are obtained from a and b respectively by the insertion of zero or more copies of a special *gap* symbol. Moreover, for any i, not both a'_i and b'_i are gap symbols and if neither a'_i nor b'_i is the gap symbol then $a'_i = b'_i$. The alignment can alternatively be considered as a sequence of aligned pairs $(a'_i, b'_i), 1 \leq i \leq |a'|$. A pair is a *gap* pair if either a'_i or b'_i is a gap symbol. We also define an *optimal alignment* of a and b as a pairwise alignment of a and b with a minimum number of gap pairs, amongst all possible alignments.

2. A *longest common subsequence* between strings a and b is a longest subsequence common to the two strings. We denote the length of

such a subsequence by $\mathbf{LCS}(a, b)$. A longest common subsequence corresponds to an optimal alignment of a and b and $\mathbf{LCS}(a, b)$ is equal to the total number of gap-free pairs of symbols in the alignment.

3. The *insertion-deletion distance* $\mathbf{d_{LCS}}(a, b)$ between strings a and b is the minimum number of insertions and deletions of symbols needed to convert a into b (or equivalently to convert b to a). Equivalently, the insertion-deletion distance between a and b is equal to the number of gap pairs in an optimal alignment of a and b.

The insertion-deletion distance and length of the longest common subsequence of two strings are related by the following known result.

Theorem 8.2.1 ([49])**.** *Given two strings a and b, where $|a| = n$ and $|b| = n'$, then $\mathbf{d_{LCS}}(a, b) = k$ if and only if $\mathbf{LCS}(a, b) = \frac{(n+n'-k)}{2}$.*

Note that if a and b are equi-length strings, then k is an even number.

In the next theorem, we provide a set of strings with a technique that employs a greedy codeword design used also in Justesen [56] and Schulman and Zuckerman [91].

Theorem 8.2.2. *Let $w > 0$ and $\delta > 0$. For any n, a set of at least wn equi-length strands over the alphabet $\{\mathtt{A}, \mathtt{T}\}$, each of length $k \log_2 n$ for some constant k (that depends on w and δ), can be designed in $2^{O(\log_2 n)}$ time, such that the insertion-deletion distance between any pair in the set is at least $\delta \log_2 n$. Moreover, all strands in the set have at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{A}$'s and at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{T}$'s.*

Proof. We construct the desired set using a greedy algorithm that is specified in terms of a quantity $t = \Theta(\log_2 n)$ that we determine later. From $\{\mathtt{A}, \mathtt{T}\}^t$, first put the two strings $\mathtt{A}^t$ and $\mathtt{T}^t$ in the set (we will remove them at the end). Once $i \geq 2$ strings are in the set, choose any string from $\{\mathtt{A}, \mathtt{T}\}^t$ whose insertion-deletion distance from all i strings already in the set is at least $\delta \log_2 n$, and add it to the set. Continue until no more strings can be chosen with the desired insertion-deletion distance. Finally, remove the strings $\mathtt{A}^t$ and $\mathtt{T}^t$. This algorithm runs in time $2^{O(\log_2 n)}$.

The number of strings in $\{\mathtt{A}, \mathtt{T}\}^t$ that have insertion-deletion distance at most $2d$ from a given string s is at most $\binom{t}{d}^2 2^d$ (see proof of Lemma 2 of Schulman and Zukerman [91]). If $d = \lceil \delta \log_2 n/2 \rceil$, then our set has the desired property that the insertion-deletion distance between any pair in the set is at least $\delta \log_2 n$. Furthermore all strings in the set, once $\mathtt{A}^t$ and $\mathtt{T}^t$ are removed, must have at least $\lceil \delta \log_2 n/2 \rceil$ $\mathtt{A}$'s and at least $\lceil \delta \log_2 n/2 \rceil$

T's; otherwise, their insertion-deletion distance from $\mathtt{A}^t$ and $\mathtt{T}^t$, would be less than $\delta \log_2 n$.

The number of strings in the set before removal of $\mathtt{A}^t$ and $\mathtt{T}^t$ is at least $wn + 2$ if we choose t so that

$$2^t/(\binom{t}{d}^2 2^d) \geq 2^{t/2} \geq wn + 2.$$

These inequalities hold if t is a sufficiently large constant times $\log_2 n$. (For the second inequality, we simply need that $t \geq 1 + 2\log_2 w + 2\log_2 n$. For the first inequality, from Stirling's formula we have that $\binom{t}{d} < (t\,e/d)^d$, and so the inequality holds if $d \log_2(t\,e/d) \leq t/2$. This in turn holds if $t = \eta d$ $(= \eta\lceil \delta \log_2 n/2 \rceil)$ where we choose constant η so that $\eta e \leq 2^\eta$.)

Finally, since the strings $\mathtt{A}^t$ and $\mathtt{T}^t$ are removed and all other strings have insertion-deletion distance at least $\delta \log_2 n$ from strings $\mathtt{A}^t$ and $\mathtt{T}^t$, all strands in the set have at least $\delta \log_2 n$ A's and at least $\delta \log_2 n$ T's. □

Our design also makes use of a *padding* function. Let ρ^5 denote the padding function that, applied to a string, inserts five A's (called padded A's) at the start of, and between, every pair of symbols in the string.

Definition 8.2.3 (padding function ρ^5)**.** Let $a = a_1 a_2 \ldots a_n$ be a string. Then $\rho^5(a) = \mathtt{A}^5 a_1 \mathtt{A}^5 a_2 \ldots \mathtt{A}^5 a_n$.

If $\mathbf{d_{LCS}}(a, b) = k$ then $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b))$ may be less than k. To illustrate why, first consider a modified padding function ρ^1, defined as $\rho^1(a_1 a_2 \ldots a_n) = \mathtt{A}^1 a_1 \mathtt{A}^1 a_2 \ldots \mathtt{A}^1 a_n$. If we choose $a = a_1 a_2 a_3 a_4 a_5 = \mathtt{AATATT}$, and $a_c = \mathtt{TTATAA}$ (a_c is the real complement of a), then $\mathbf{d_{LCS}}(a, a_c) = 6$ whereas $\mathbf{d_{LCS}}(\rho^1(a), \rho^1(a_c)) = 4$. This appears to contradict an assertion in Lemma 2 of Schulman and Zukerman [91]. Adapting this example, it is the case that if

$$a' = a_1^5 a_2^5 \ldots a_5^5 = \mathtt{A}^5\mathtt{A}^5\mathtt{T}^5\mathtt{A}^5\mathtt{T}^5\mathtt{T}^5$$

and a'_c would be the real complement of a', then $\mathbf{d_{LCS}}(a', a'_c) = 30$, while $\mathbf{d_{LCS}}(\rho^5(a'), \rho^5(a'_c)) = 24$.

We next show a general lower bound on $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b))$ in terms of $\mathbf{d_{LCS}}(a, b)$.

Lemma 8.2.4. *Let a and b be equi-length strings over $\{\mathtt{A}, \mathtt{T}\}$. If $\mathbf{d_{LCS}}(a, b) = k$ then $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b)) \geq \frac{k}{2}$.*

Proof. Suppose that $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b)) < \frac{k}{2}$. Let n be the length of a and b. We will obtain a contradiction to the hypothesis of the lemma that

$\mathbf{d_{LCS}}(a, b) = k$. Throughout, when referring to characters in the padded strings $\rho^5(a)$ and $\rho^5(b)$, we'll denote the characters of the original strings a and b by $\mathtt{A}_o$ and $\mathtt{T}_o$ and the padded $\mathtt{A}$'s by $\mathtt{A}_p$.

Let $\mathcal{A}$ be an optimal alignment of $\rho^5(a)$ and $\rho^5(b)$. Each pair of characters in alignment $\mathcal{A}$ has one of four types: *original*, with two original characters; *padded*, with two padded characters; *mixed*, with one A_o and one A_p, or *gap*, with one gap symbol. Let x, y and u denote, in order, the counts of original, padded and mixed pairs, respectively. To prove the lemma, we first establish various bounds on these counts.

First, it must be that $x \leq n - \frac{k}{2}$: since $\mathbf{d_{LCS}}(a, b) = k$, if x were greater than $n - \frac{k}{2}$ we would be able to use the alignment $\mathcal{A}$ to obtain an alignment of a and b with less than k gap pairs.

Second, using Theorem 8.2.1 and our assumption that $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b)) < \frac{k}{2}$, we have that $\mathbf{LCS}(\rho^5(a), \rho^5(b)) \geq 6n - \lfloor \frac{k}{4} \rfloor$, and so

$$x + y + u = \mathbf{LCS}(\rho^5(a), \rho^5(b)) \geq 6n - \lfloor \tfrac{k}{4} \rfloor. \tag{8.2.1}$$

Third, we'll obtain a lower bound on x. Note that $2y + u$ is bounded by the total number of $\mathtt{A}_p$ characters, and so is at most $10n$. Therefore $y + \lceil \frac{u}{2} \rceil \leq 5n$. Substituting this inequality into Equation 8.2.1, we have that

$$x \geq n - \lfloor \tfrac{k}{4} \rfloor - \lfloor \tfrac{u}{2} \rfloor. \tag{8.2.2}$$

From the fact that $x \leq n - \frac{k}{2}$ and inequality 8.2.2 we also have that the number of mixed pairs u is at least $\frac{k}{2}$.

Now partition the mixed pairs into two types: sloppy and tight. A mixed pair p is *sloppy* if, among the first five pairs to the right of p, there is at least one gap pair containing a T_o or A_p character. If p is not sloppy, we call it *tight*. If p is tight, let p' be the first pair to the right of p that is not a padded pair. Such a pair p' must exist, since our padding function is such that any A_p character is eventually followed by an original character. Pair p' is either a gap pair containing A_o or is a mixed pair, in which case it also contains A_o. In either case, because exactly five A_p's separate any two original characters, if the A_o character of pair p is in string a then the A_o character of pair p' is in string b and vice versa. In what follows, we refer to p' as p's *partner*. Note that p' may itself be a tight pair.

Each sloppy pair contributes one to $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b))$. Since we are assuming that $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b)) < \frac{k}{2}$, less than $\frac{k}{2}$ of the mixed pairs are sloppy.

Thus, at least $u - \frac{k}{2} + 1$ of the mixed pairs are tight. Using these tight

pairs, we now convert alignment $\mathcal{A}$ to another alignment $\mathcal{A}'$ with at least $n-\frac{k}{2}+1$ original pairs, obtained as follows. Starting from the leftmost pair of alignment $\mathcal{A}$ and working towards the right, find the first tight mixed pair p of $\mathcal{A}$ and its partner p'. Remove p, p' and all of the intervening (padded) pairs between them from the alignment, and instead pair each padded character from the removed pairs with a gap, and pair the A_o character of p with the A_o character of p' (recall that one of these A_o characters is in string a and the other is in string b). Repeat, starting from the pair just to the right of p', until the rightmost end of $\mathcal{A}$ is reached.

Let d be the number of new original pairs obtained in this manner. Then d is at least $\lfloor \frac{u}{2} \rfloor - \lceil \frac{k}{4} \rceil + 1$: this lower bound is achieved when all partners are themselves tight mixed pairs. Therefore, the number of original pairs in alignment $\mathcal{A}'$ is

$$x + d \geq n - \lfloor \tfrac{k}{4} \rfloor - \lfloor \tfrac{u}{2} \rfloor + \lfloor \tfrac{u}{2} \rfloor - \lceil \tfrac{k}{4} \rceil + 1 = n - \tfrac{k}{2} + 1.$$

As noted earlier, any alignment of $\rho^5(a)$ and $\rho^5(b)$ has at most $n-\frac{k}{2}$ original pairs since $\mathbf{d_{LCS}}(a,b) = k$, and so we have a contradiction. Thus, our assumption that $\mathbf{d_{LCS}}(\rho^5(a), \rho^5(b)) < \frac{k}{2}$ is false, and the lemma is true. □

We next define the *unpairedness* of a secondary structure for a strand or pair of strands, and show that sets of padded strings with high insertion-deletion distance are useful in obtaining strands whose optimal structures have high unpairedness.

Definition 8.2.5. Let a and b be strands and let $S(a)$ and $S(a,b)$ be secondary structures for strand a and pair (a,b) respectively. The *unpairedness* of $S(a)$ or $S(a,b)$ is the number of bases that are not paired in $S(a)$ or $S(a,b)$, respectively. We note that the base pairs of (a,b) includes both inter-molecular and intra-molecular pairs.

Lemma 8.2.6. *Let a' and b' be any strands over the alphabet $\{\mathtt{A},\mathtt{T}\}$, let $a = \rho^5(a')$, let $b = \rho^5(b')$, and let s be any substrand of a or $\bar{a}$. Let $S(s)$, $S(a,b)$, $S(\bar{a},\bar{b})$ and $S(a,\bar{b})$ be any pseudoknot-free secondary structures for s, (a,b), $(\bar{a},\bar{b})$ and $(a,\bar{b})$, respectively. Then*

1. *The unpairedness of $S(s)$ is at least $\frac{1}{3}|s|$.*
2. *The unpairedness of $S(a,b)$ is at least $\frac{2}{3}(|a|+|b|)$.*
3. *The unpairedness of $S(\bar{a},\bar{b})$ is at least $\frac{2}{3}\left(|\bar{a}|+|\bar{b}|\right)$.*
4. *The unpairedness of $S(a,\bar{b})$ is at least $\frac{1}{3}\mathbf{d_{LCS}}(a,b)$.*

Proof. To show part 1, first suppose that s is a substrand of $a = \rho^5(a')$. All bases that are paired in the secondary structure $S(s)$ are within substrand s. If $|s| \leq 2$, then no bases of s are paired in $S(s)$, given our assumption that consecutive bases in a strand cannot form a base pair, and so part 1 holds. If $|s| \geq 3$, the number of (intra-molecular) base pairs of $S(s)$ is at most the number of T's in s. If $3 \leq |s| \leq 6$ then s can have at most one T, and thus at most one base pair, so s has at least $|s| - 2$ unpaired bases and again part 1 holds. Suppose that $|s| \geq 7$. Because s is a substrand of a padded strand, the number of T's in s is at most $\lceil 2|s|/7 \rceil$: this maximum is achieved if $|s| = 7$ and s both starts and ends with a T. Even if all of the T's of s are paired to A's, the number of unpaired A's is still at least $\lfloor 3|s|/7 \rfloor \geq |s|/3$ since $|s| \geq 7$. The argument when s is a substrand of $\bar{a}$ is obtained by replacing A's with T's in the argument for a substrand of a.

Similarly, the total number of T's in $S(a, b)$ is at most $(|a|+|b|)/6$ and so the unpairedness is at least $4(|a|+|b|)/6$. The argument for the unpairedness of $S(\bar{a}, \bar{b})$ is obtained by replacing A's with T's in the argument for $\{a, b\}$.

Finally, the inter-molecular base pairs of $S(a, \bar{b})$ correspond to a common subsequence of strands a and b, and thus the number of such base pairs is at most $\mathbf{LCS}(a, b) = n - \frac{\mathbf{d_{LCS}}(a,b)}{2}$ by Theorem 8.2.1. Therefore the total number of bases in both a and $\bar{b}$ that do not form inter-molecular base pairs of $S(a, \bar{b})$ is at least $\mathbf{d_{LCS}}(a, b)$. Now consider any substructure of $S(a, \bar{b})$ within some maximal substrand s of either a or $\bar{b}$ that has no inter-molecular base pairs. The unpairedness of this substructure is at least $\frac{1}{3}|s|$, by part 1 of this Lemma. Thus, over all substrands that do not contain inter-molecular base pairs, at least a fraction $\frac{1}{3}$ of bases are unpaired (not involved in intra-molecular base pairs). Since the total length of such substrands is at least $\mathbf{d_{LCS}}(a, b)$, the unpairedness of $S(a, \bar{b})$ is at least $\frac{1}{3}\mathbf{d_{LCS}}(a, b)$. □

Definition 8.2.7. A set $\mathcal{S}$ of strands is *k-robust* if the following properties hold:

1. All strands of $\mathcal{S}$ have the same length.
2. All strands of $\mathcal{S}$ have at least k A's and at least k T's.
3. For any a and b in the set, the unpairedness of optimal structures for a, $\bar{a}$, (a, b), $(\bar{a}, \bar{b})$, and $(a, \bar{b})$ is at least k.

Theorem 8.2.8. *Let $w > 0$. For any n, a $\log_2 n$-robust set of at least wn strands, each of length $p \log_2 n$ for some constant p, can be designed in $2^{O(\log_2 n)}$ time.*

Proof. Using Theorem 8.2.2, for any $w > 0$ and $\delta = 6$ we can obtain, in time $2^{O(\log_2 n)}$, a set $\mathcal{S}'$ of at least wn strands, each of length $k \log_2 n$ for some constant k, such that the insertion-deletion distance between any pair of strands in $\mathcal{S}'$ is at least $6 \log_2 n$. Moreover, all strands in $\mathcal{S}'$ have at least $3 \log_2 n$ `A`'s and at least $3 \log_2 n$ `T`'s. This latter property implies that the strands in $\mathcal{S}'$ have length at least $6 \log_2 n$.

Apply the padding function ρ^5 to strands in $\mathcal{S}'$ to obtain a new set $\mathcal{S}$. Note that the strands in $\mathcal{S}$ have length $6k \log_2 n$, which must be at least $36 \log_2 n$. Lemma 8.2.4 shows that the insertion-deletion distance between any pair of strands in $\mathcal{S}$ is at least $\delta \log_2 n/2 = 3 \log_2 n$. Lemma 8.2.6 then shows that if a and b are any two strands in the set $\mathcal{S}$, the unpairedness of the optimal structure of a, or its complement, or of (a, b), $(a, \bar{b})$ or $(\bar{a}, \bar{b})$, is at least $\min\{\frac{1}{3}|a|, \frac{2}{3}(|a| + |b|), \frac{1}{3}\mathbf{d_{LCS}}(a, b)\}$. Given that $|a|$ and $|b|$ are at least $36 \log_2 n$ and that $\mathbf{d_{LCS}}(a, b) = 3 \log_2 n$, this lower bound is at least $\log_2 n$. Therefore, the unpairedness of the set $\mathcal{S}$ is at least $\log_2 n$, as desired. □

8.3 The Reduction

We show a polynomial time, many-one reduction from 3DM(3) to MULTI-PKF-SSP (3DM(3) $\leq_T^P$ MULTI-PKF-SSP). Given an instance $I = (X, Y, Z, \mathcal{T})$ of 3DM(3), where $m = |\mathcal{T}|$ and $n = |X| = |Y| = |Z|$, we construct an instance I' of MULTI-PKF-SSP as follows.

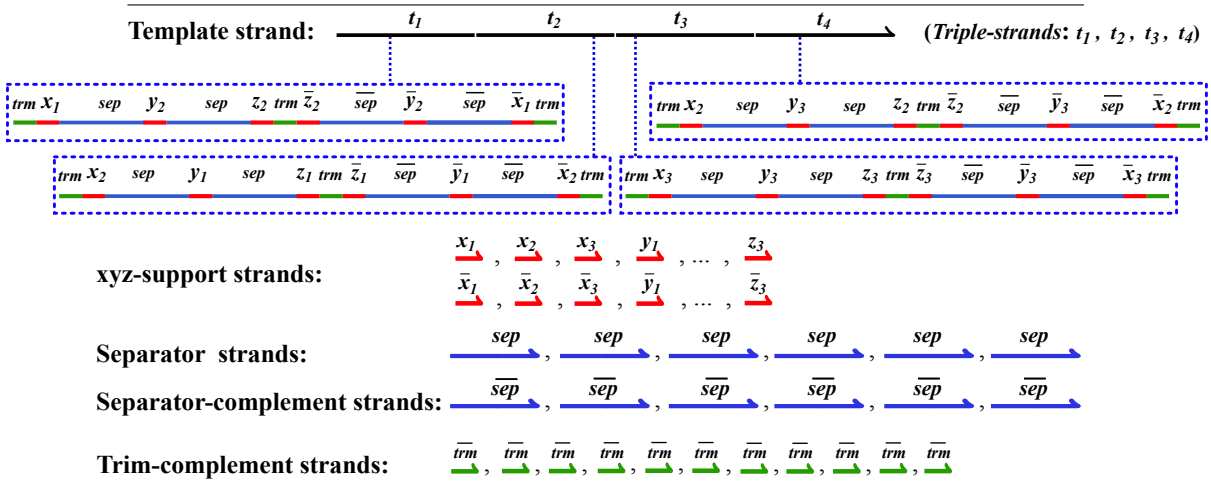
Domains used in strands of I':

The strands of the resulting instance, I', consist of the following domains.

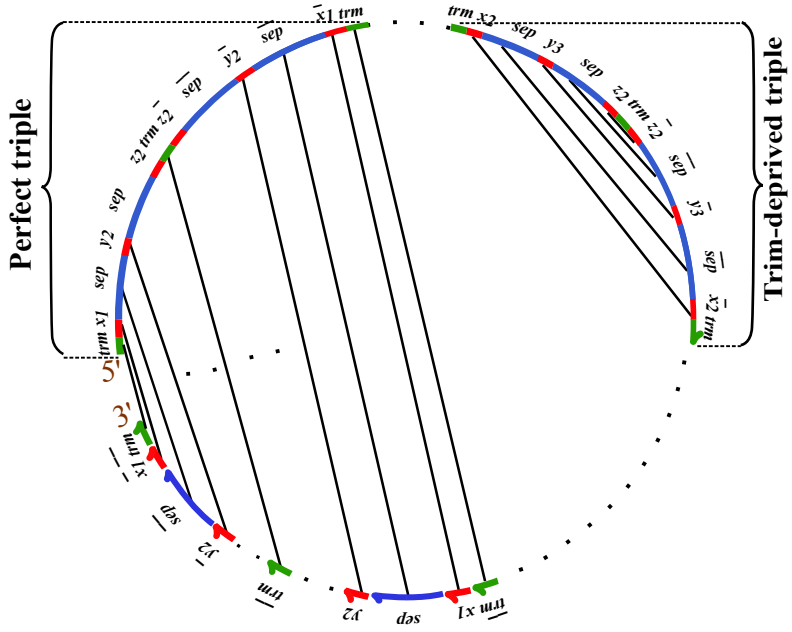
- One domain for each $x \in X$, $y \in Y$, and $z \in Z$ and one domain for each complement. Where no confusion arises, we use x, $\bar{x}$, y, $\bar{y}$, z, and $\bar{z}$ to refer to these domains.
- A *separator* and a *separator-complement* domain, denoted by `Sep` and $\overline{\texttt{Sep}}$.
- A *trim* domain and a *trim-complement* domain, denoted by `Trm` and $\overline{\texttt{Trm}}$ respectively.

Strands of I':

The resulting instance, I', consists of the following strands; the strands are described as a sequence of domains.



(a) Strands



(b) MFE structure

Figure 8.3: The resulting set of strands, specified at the domain level (part a), and the MFE structure (part b), when reducing from the 3DM instance of Figure 8.2. In the MFE structure, triple-strand t_1 labeled as *perfect triple* represents that the triple (x_1, y_2, z_2) is in the solution of the 3DM(3) instance. However, triple-strand t_4 denoted as *trim-deprived triple* shows that the triple (x_2, y_3, z_2) is not selected in the solution.

- *Template strand*: one strand that is the concatenation of *triple-strands*. There is one triple-strand for each triple $(x, y, z) \in \mathcal{T}$, which is the concatenation of the domains `Trm`, x, `Sep`, y, `Sep`, z, `Trm`, $\bar{z}$, $\overline{\texttt{Sep}}$, $\bar{y}$, $\overline{\texttt{Sep}}$, $\bar{x}$, `Trm` in that order. We call the substrands x `Sep` y `Sep` z and $\bar{z}$ $\overline{\texttt{Sep}}$ $\bar{y}$ $\overline{\texttt{Sep}}$ $\bar{x}$ of a triple-strand the $5'$ and $3'$ *flanks*, respectively.

  We call the `Trm` domains at the ends of the triple-strand the end-trims and the `Trm` domain at the center of the triple-strand the center-trim.

- *Separator (-complement) support strands*: there are $2n$ strands consisting of just the domain `Sep` and another $2n$ strands consisting of just the domain $\overline{\texttt{Sep}}$.

- `xyz`*-support strands*: for each x, y and z domain there is one strand consisting of just that domain and one for its complement, for a total of $6n$ strands.

- *Trim-complement strands*: there are $2m + n$ strands consisting of just the domain $\overline{\texttt{Trm}}$ (these strands are the complement of the trim domains `Trm`).

We refer to the `xyz`-support strands and the separator and separator-complement support strands collectively as the support strands.

This completes the description of the reduction at the domain level of detail. Figure 8.3a shows the resulting Multi-Pkf-SSP instance, specified at the domain level, after a reduction from the 3DM(3) instance depicted in Figure 8.2.

The MFE structure of the resulting set of strands is partially depicted in Figure 8.3b. The binding of the `xyz`-supports and separator supports x_1, `Sep`, y_2, `Sep`, z_2, their complements, and trim-complement strands to the substrand labeled as "perfect triple", denotes that the triple (x_1, y_2, z_2) is selected in the solution of the 3DM(3) instance. The other triple-strand that is depicted is a "trim-deprived triple" — i.e., a triple where at least one of its trim domains is unbound — as the triple (x_2, y_3, z_2) does not appear in the solution from Figure 8.2 (right). Intuitively, there is a trim-complement strand available to bind with each of the $2m$ end-trim domains at the ends of all triple-strands, and in addition the number of xyz-support, separator supports and additional trim-complement strands is necessary and sufficient to have n "perfect triples" in an optimal secondary structure when the 3DM(3) instance has a perfect matching of size n.

Sequence design for I':

To complete the reduction, we specify a sequence design for each domain of I'. For the x, y, and z domains, we use the set of sequences of Theorem 8.2.8 with $w = 3$, since we need $3n$ domains (plus their complements) in total. Let E ($= \Theta(\log_2 n)$) be the length of these domains. The trim domains consist only of the base `G`, and are also of length E. Formally, let $\texttt{Trm} = \texttt{G}^E$. The `Sep` domain is $\texttt{A}^{6E}$, and the $\overline{\texttt{Sep}}$ domain is the complement of the `Sep` domain, namely $\texttt{T}^{6E}$.

The sequence design has the property that there are an equal number of `A` and `T` bases overall: for every x, y, z or separator domain in a triple-strand, or x, y, z, or separator strand there is another domain or strand that is its complement. The total number of `C`'s in trim-complement strands is $(2m + n)E$. The total number of `G`'s in end-trims and center-trims is $3mE$. Since $m \geq n$, the total number of `G`'s is at least as great as the total number of `C`'s. Therefore, under the assumption that only Watson-Crick base pairs can form, the maximum number of base pairs is limited to the total number of `A` (or `T`) bases plus the total number of `C` bases. Let P denote this quantity.

The instance I' is comprised of the strands of I' and the positive integer P.

Lemma 8.3.1. *Instance I' can be constructed in time polynomial in n.*

Proof. Instance I' has one template strand, $2n$ separator supports and $2n$ separator-complement supports $6n$ `xyz`-support strands, and $2m + n$ trim-complement strands, for a total of $2m + 11n + 1$ strands. The template strand has $13m$ domains and the other strands have one domain each, for a total of $15m + 11n$ domains.

Since every domain in the construction has length $\Theta(\log_2 n)$, instance I' is of size polynomial in n overall. The sequences can also be designed in polynomial time: The sequence design of separator and trim domains is trivial, and the sequences for the x, y, z domains can be designed in time polynomial in n by Theorem 8.2.8. □

8.4 Reduction Correctness

We show that if the given instance I of 3DM(3) has a perfect matching then the optimal secondary structure formed from strands in I' is a single complex that has P base pairs–the maximum number of base pairs that can be formed from the strands of I', whereas if the optimal matching of I has size $n - i$ then the optimal structure has only $P - \Omega(iE)$ base pairs.

Lemma 8.4.1. *If I has a perfect matching, then the strands of I' can form a pseudoknot-free secondary structure, consisting of a single complex and P base pairs, with n perfect triple-strands.*

Proof. Here, in the reduced instance I', bases in the n triple-strands corresponding to the perfect matching can be bound to the support strands x, `Sep`, y, `Sep`, z, their complements, and three trim-complement strands and form n perfect triple-strands. The end-trims of the remaining triple-strands can also be bound to two trim-complement strands while their complementary $5'$ and $3'$ flanks are paired together to make trim-deprived triples. Therefore, as all `A`'s and `C`'s are paired in this single (connected) complex, the number of base pairs is P which is optimal. □

We next consider the case that the optimal matching of I has size at most $n - i$. Let $\mathrm{Opt}(I')$ denote the optimal pseudoknot-free structure of the reduced instance I'. We establish properties that must hold true of $\mathrm{Opt}(I')$ and conclude that when the optimal matching of I has size at most $n - i$, then $\mathrm{Opt}(I')$ has $P - \Omega(iE)$ base pairs.

With respect to a given structure, we say that a domain is *bound* if at least one of its bases forms a base pair. A domain d in a triple-strand (as part of the template strand) is *connected to* a non-template strand s if there is a sequence of non-template strands $s_1, s_2, \ldots, s_j$ where $s_j = s$, such that d forms a base pair with s_1, s_1 forms a base pair with s_2, and so on up to s_{j-1} forming a base pair with $s_j = s$.

We partition the triple-strands into four types, depending on the structure they form in $\mathrm{Opt}(I')$.

- *Perfect triples*: the triple-strand binds to the set of non-template strands that are complementary to the triple-strand domains. This set of non-template strands contains two `Sep`'s, two $\overline{\texttt{Sep}}$'s, three $\overline{\texttt{Trm}}$'s and six `xyz`-support strands in total. The set of perfect triples corresponds to a matching of instance I.

- *Trim-deprived triples*: at least one trim of a triple-strand is unbound.

- *Hogger triples*: these are triple-strands which are not trim-deprived, and moreover, the ten domains in the flanks of a hogger triple are bound to, or connected to, at least eleven support strands in total.

- *Flawed triples*: none of the above. In particular, flawed triples are not trim-deprived.

Each triple-strand belongs to exactly one of the above four types. Note also that because a hogger or a flawed triple is not trim-deprived, the support domains that are bound to or connected to the $5'$ flank (or $3'$ flank) of its triple-strand cannot bind to other domains on the template strand, or a pseudoknot would form.

Lemma 8.4.2. *The total number of trim-deprived and flawed triples in Opt(I') is at least $(m-n)+i/11$.*

Proof. There are m triple-strands overall. Let p be the number of perfect triples; each of these triple-strands has 10 support strands bound to it. Let h be the number of hogger triples; there are at least 11 support strands bound or connected to each. There are $6n$ `xyz`-supports and $4n$ separator and separator-complement strands in total, so $10p + 11h \leq 10n$ and $h \leq 10(n-p)/11$. Note also that since the optimal matching of I has size at most $n-i$, the number of perfect triples p must be at most $n-i$ and so $n-p \geq i$.

The total number of triple-strands is m, so the number of triple-strands that are neither perfect nor hogger is

$$m - p - h \geq m - p - 10(n-p)/11 = m - n + (n-p)/11 \geq (m-n) + i/11.$$

□

Lemma 8.4.3. *Either Opt(I') has at least $m-n+i/22$ trim-deprived triples, or at least $i/22$ flawed triples.*

Proof. Suppose that the number of trim-deprived triples is less than $m - n + i/22$. By Lemma 8.4.2, the total number of trim-deprived and flawed triples is at least $(m-n) + i/11$. Subtracting, we have that the number of flawed triples is at least $i/22$. □

We now adapt our notion of unpairedness from Section 8.2 to `ACT`-unpairedness. Let a and b be strands and let $S(a)$ and $S(a,b)$ be secondary structures for strand a and pair (a,b) respectively. The `ACT`-unpairedness of $S(a)$ or $S(a,b)$ is the number of `A`, `C` and `T` bases that are not paired in $S(a)$ or $S(a,b)$, respectively.

Lemma 8.4.4. *If the number of trim-deprived triples in Opt(I') is at least $m-n+i/22$, then at least $iE/22$ `C`'s are unpaired in Opt(I'), and so Opt(I') has `ACT`-unpairedness $\Omega(iE)$.*

Proof. Each trim-deprived triple forms at most $2E$ `CG` base pairs, with the `G`s being in the trims (center-trim and end-trims) of the triple-strand and the `C`s being in trim-complement strands. Triple-strands that are not trim-deprived form at most $3E$ `CG` base pairs. There are no other `CG` base pairs. So, the total number of `CG` base pairs is at most

$$(m - n + i/22)2E + (m - (m - n + i/22))3E = (2m + n - i/22)E.$$

The total number of trim-complement strands is $2m + n$, each containing E `C`s. So, the number of unpaired `C` bases in trim-complements is at least $iE/22$. □

In order to show that many flawed triples cause $\mathrm{Opt}(I')$ to have high `ACT`-unpairedness, we first derive some useful properties about flawed triples. In what follows, we let $L_f = x\ \mathtt{Sep}_{xy}\ y\ \mathtt{Sep}_{yz}\ z$ and $R_f = \bar{z}\ \overline{\mathtt{Sep}}_{yz}\ \bar{y}\ \overline{\mathtt{Sep}}_{xy}\ \bar{x}$ denote the sequences on the $5'$ and $3'$ flanks of a flawed triple. Let $B_f(5')$ and $B_f(3')$ be the sets of support strands that are bound to, or connected to domains of L_f and R_f respectively, in the structure $\mathrm{Opt}(I')$. Since a flawed triple has at most ten support strands bound to it in total, either $B_f(5') \leq 5$ or $B_f(3') \leq 5$. In the following lemmas, for concreteness, we suppose that $B_f(5') \leq 5$; the argument when $B_f(3') \leq 5$ is obtained by replacing domains and strands with their complements and bases `A` and `T` with each other. Let $\mathrm{Opt}(L_f)$ be the substructure of $\mathrm{Opt}(I')$ formed by the bases in L_f and the strands in $B_f(5')$.

Lemma 8.4.5. *Let $L_f = x\ \mathtt{Sep}_{xy}\ y\ \mathtt{Sep}_{yz}\ z$ be the left flank of a flawed triple with respect to structure $\mathrm{Opt}(I')$. Suppose that there are $l \geq 2$ bonds between x, y or z domain of L_f and either $\mathtt{Sep}_{xy}$ or $\mathtt{Sep}_{yz}$. Then $\mathrm{Opt}(L_f)$ has `ACT`-unpairedness at least $5(l-1)$.*

Proof. As $\mathtt{Sep}_{xy}$ and $\mathtt{Sep}_{yz}$ contain only `A`'s, they can only bind with `T`'s of L_f. Our sequence design ensures that there are at least five padded `A`'s between any two successive `T`'s of x, y or z. Therefore, in order to avoid pseduoknots, if there are l bonds between x, y, or z and a `Sep` domain, at least $5(l-1)$ padded `A`'s remain unpaired. □

Lemma 8.4.6. *Suppose that in $\mathrm{Opt}(L_f)$, $B_f(5') \leq 5$ and the `ACT`-unpairedness of L_f is less than $(\log_2 n)/3$. Then the following must hold.*

1. *Each `Sep` domain of L_f is bound to a $\overline{\mathtt{Sep}}$-support domain.*
2. *Each x, y and z domain of L_f is bound to an `xyz`-support domain.*

As a consequence, each x, y, and z domain of L_f is bound to a distinct xyz*-support of $B_f(5')$, each* Sep *domain of L_f is bound to a distinct* $\overline{\texttt{Sep}}$ *support of $B_f(5')$, and $B_f(5')$ contains exactly three* xyz*-supports and two* $\overline{\texttt{Sep}}$ *supports.*

Proof. Suppose to the contrary that the first condition does not hold, i.e., one of L_f's Sep domains is not bound to a $\overline{\texttt{Sep}}$ support. The total number of T's that can bind to the Sep domain is at most $5.5E$, accounted for as follows: at most $3E/6$ T's in the x, y, and z domains of L_f plus at most $5E$ in the remaining support strands, if they are five xyz-support strands. Thus at least $E/2$ of the $6E$ A's in the Sep domain are unpaired. Since $E \geq \log_2 n$, we get a contradiction to the hypothesis of the lemma. Thus the first condition must hold.

Next suppose that the first condition holds but that the second does not; specifically that the x domain of L_f is not bound to an xyz-support domain (the argument is similar for the y or z domains). Recall that domain x contains at least $\log_2 n$ T's, since by design the domains comprise a $\log_2 n$-robust set. At least $2(\log_2 n)/3$ of the T's must be paired, or the hypothesis of the lemma that the ACT-unpairedness of L_f is less than $(\log_2 n)/3$ would not be true. Since the first condition of the lemma holds, the Sep domain adjacent to x on the $5'$ flank is bound to a $\overline{\texttt{Sep}}$ strand. Therefore domain x cannot have bonds to domain y or z, or to the Sep domain between y and z, or a pseudoknot would form. Also, the T's in domain x cannot bind to $\overline{\texttt{Sep}}$ strands, since $\overline{\texttt{Sep}}$'s are composed only of T's. If there were at least $(\log_2 n)/3$ bonds between x and $\texttt{Sep}_{xy}$, Lemma 8.4.5 would imply that x has ACT-unpairedness at least $5((\log_2 n)/3 - 1) \geq \log_2 n$, again contradicting the hypothesis of the lemma.

Therefore, at least $(\log_2 n)/3$ T's of x must form intramolecular bonds with A's that are also in the x domain. The total length of substrands of x that have either unpaired bases or intramolecular base pairs must be at least $3(\log_2 n)/3$: this lower bound is met if each T, say at position i of x is bound to an A that is either at position $i-2$ or $i+2$ (since we assume that no base pair can form between consecutive bases). Part 1 of Lemma 8.2.6 therefore implies that x has ACT-unpairedness at least $(\log_2 n)/3$, once again contradicting the hypothesis of the lemma. We conclude that the second condition of the lemma must hold.

Since both conditions hold, it cannot be that two of the x, y, and z domains of L_f are bound to the same xyz-support of $B_f(5')$, or a pseudoknot would form with bonds between a Sep of L_f and a $\overline{\texttt{Sep}}$ support. Similarly, it cannot be that both Sep's have bonds to the same $\overline{\texttt{Sep}}$. Hence, each

Sep domain of L_f is bound to a distinct $\overline{\texttt{Sep}}$ support of $B_f(5')$, and $B_f(5')$ contains exactly three xyz-supports and two $\overline{\texttt{Sep}}$ supports, completing the proof of the Lemma. □

Lemma 8.4.7. *Suppose that in Opt(L_f), $B_f(5') \leq 5$.* ACT*-unpairedness of L_f is less than $(\log_2 n)/3$. Then for any constant $\alpha < 1/7$, the* ACT*-unpairedness of Opt(L_f) is at least $\alpha \log_2 n$.*

Proof. Let $\alpha < 1/7$. Suppose to the contrary that the ACT-unpairedness of $\mathrm{Opt}(L_f)$ is less than $\alpha \log_2 n$. By Lemma 8.4.6, $B_f(5')$ must contain three xyz-supports, say a, b, and c, with a bound to x, b bound to y, and c bound to x.

We first show that in $\mathrm{Opt}(L_f)$, there can be at most $\alpha \log_2 n/5$ bases between a Sep domain of L_f and one of the domains x, y, or z adjacent to the Sep domain. Otherwise, by Lemma 8.4.5, at least $\alpha \log_2 n$ bases of a would be unpaired, and we get a contradiction. Similarly, there can be at most $\alpha \log_2 n/5$ bases between a $\overline{\texttt{Sep}}$ domain of L_f and one of the domains a, b, or c adjacent to the $\overline{\texttt{Sep}}$ domain.

Since L_f is the flank of a flawed triple, either $a \neq \bar{x}$, $b \neq \bar{y}$, or $c \neq \bar{z}$. First suppose that $a \neq \bar{x}$. Since the set of domains is $\log_2 n$-robust, there can be at most $E - \log_2 n$ base pairs between a and x. By the argument in the previous paragraph, x has at most $\alpha(\log_2 n)/5$ bases to $\texttt{Sep}_{xy}$. Similarly, if $\overline{\texttt{Sep}}_{ab}$ is the separator complement between a and b, then a has at most $\alpha(\log_2 n)/5$ bases to $\overline{\texttt{Sep}}_{ab}$. If a has base pairs with $\texttt{Sep}_{xy}$, then x cannot have base pairs with $\overline{\texttt{Sep}}_{ab}$ and vice versa, in order to avoid pseudoknots. Therefore, either a or x has at least $\log_2 n - \alpha(\log_2 n)/5 \geq 34(\log_2 n)/35$ bases that are either unpaired or form intramolecular bonds. By Lemma 8.2.6, either a or x has unpairedness at least $11(\log_2 n)/35 \geq (\log_2 n)/4$, proving the lemma. The argument when $c \neq \bar{z}$ is similar to that when $a \neq \bar{x}$.

Finally, suppose that $a = \bar{x}$ and $c = \bar{z}$ but $b \neq \bar{y}$. As noted earlier, b has at most $\alpha(\log_2 n)/5$ bonds with each $\overline{\texttt{Sep}}$ adjacent to it. Also, at least $\log_2 n$ bases of b are not paired with y, since the set of domains is $\log_2 n$-robust. Of these, at most $\alpha log_2 n$ can be unpaired, or again we get a contradiction. Therefore, b has at least $\log_2 n - 2\alpha(\log_2 n)/5 - \alpha \log_2 n = \log_2 n - 7\alpha(\log_2 n)/5$ bonds to the Sep's adjacent to y, and so b has at least $\frac{1}{2}(\log_2 n - 7\alpha(\log_2 n)/5)$ bonds to $\texttt{Sep}_{xy}$.

Moreover, $\overline{\texttt{Sep}}_{ab}$ must have at least $6E - \alpha \log_2 n(12/5)$ base pairs with $\texttt{Sep}_{xy}$. This is because $\overline{\texttt{Sep}}_{ab}$ has at most $\alpha(\log_2 n)/5$ bases with each of a and b, and $\overline{\texttt{Sep}}_{ab}$ has at most $3\alpha \log_2 n$ bases paired with x. To see why

the latter assertion holds, note that otherwise at least $3\alpha \log_2$ bases of a are not paired with any strand other than a and thus by Lemma 8.2.6, at least $\alpha \log_2 n$ bases of a are unpaired, which again is a contradiction. Therefore, $\overline{\texttt{Sep}}_{ab}$ has at most $\alpha \log_2(2/5+3)$ pairs in total with a, x, and b, and since at most $\alpha \log_2 n$ bases of $\overline{\texttt{Sep}}_{ab}$ can be unpaired, $\overline{\texttt{Sep}}_{ab}$ has at least $6E - \alpha \log_2 n(2/5+3-1) = 6E - \alpha \log_2 n(12/5)$ base pairs with $\texttt{Sep}_{xy}$.

Therefore the total number of bases that are paired with bases of $\texttt{Sep}_{xy}$ is at least $\frac{1}{2}(\log_2 n - 7\alpha(\log_2 n)/5)$ (with b) plus $6E - \alpha \log_2 n(12/5)$ (with $\overline{\texttt{Sep}}_{ab}$). The total is

$$6E + \log_2 n(1/2 - 7\alpha/10 - \alpha(12/5)) \geq 6E + \log_2 n(1/2 - \alpha(31/10)).$$

Since $\alpha \leq 1/7$, this quantity is greater than $6E$, again a contradiction since the length of $\texttt{Sep}_{xy}$ is $6E$. □

Lemma 8.4.8. *If the optimal matching of I has size at most $n - i$, then Opt(I') has $P - \Omega(iE)$ base pairs.*

Proof. By Lemma 8.4.3, Opt(I') either has at least $m - n + i/22$ trim-deprived triples, or at least $i/22$ flawed triples.

First suppose that Opt(I') has at least $m - n + i/22$ trim-deprived triples. Then by Lemma 8.4.4, the strands of Opt(I') have ACT-unpairedness $\Omega(iE)$. Similarly, if Opt(I') has at least $i/22$ flawed triples, then by Lemma 8.4.7, each flawed triple has ACT-unpairedness $\Omega(\log_2 n) = \Omega(E)$, since $E = \Theta(\log_2 n)$. Again, the total ACT-unpairedness is $\Omega(iE)$.

Recall that all A's, C's and T's must be paired in order for the total number of base pairs to be P. Since the total ACT-unpairedness is $\Omega(iE)$, it must be that the number of base pairs in Opt(I') is at most $P - \Omega(iE)$. □

Theorem 8.4.9. Multi-Pkf-SSP *is* NP*-complete.*

Proof. Let I be any instance of Multi-Pkf-SSP, i.e, m nucleic acid strands and a positive integer k. Given a secondary structure S for I, we can check in time polynomial in the total length of the strands whether S is a valid, pseudoknot-free secondary structure and whether it has k base pairs. Therefore, Multi-Pkf-SSP is in NP.

Moreover, in the last section we provided a a polynomial time reduction from any instance I of 3dm(3) to an instance I' of Multi-Pkf-SSP. The optimal structure Opt(I') has P base pairs if I has a perfect matching, by Lemma 8.4.1, and Opt(I') has less than P base pairs if I does not have a perfect matching (by Lemma 8.4.8), where P is the total number of A, T and C bases of the strands of instance I'.

Putting these together, we can conclude that Multi-Pkf-SSP is NP-complete. □

Until now, we have only considered the number of base pairs in the MFE structure under the assumption that there is no penalty for strand association, i.e., $K_{\text{assoc}} = 0$. Our construction has the property that structure $\text{Opt}(I')$ is a single complex when I has a perfect matching. When $K_{\text{assoc}} > 0$ the penalty to bring the $2m + 11n + 1$ strands into a single complex is $(2m + 11n)K_{\text{assoc}}$. However, the number of base pairs formed is at least E, the total length of the xyz-support strands, where $E = \Theta(\log_2 n)$. Thus, for any positive constant K_{assoc} the value of E can be scaled by a constant to ensure that a single domain binding is always favourable, even when decreasing the number of complexes by one.

8.5 Approximability

We proved that the Multi-Pkf-SSP problem is NP-complete in Theorem 8.4.9. Given this result, it is natural to investigate that if there is a polynomial-time algorithm to approximate the optimal secondary structure of multi-stranded systems. In this section we show that the Max-Multi-Pkf-SSP problem is APX-hard as well — see Theorem 8.5.2. This result asserts that there exists no PTAS for this problem, unless P = NP.

To show the hardness result, we first verify that our reduction from Max-3dm(3), which itself is APX-hard by Theorem 8.1.2, to Max-Multi-Pkf-SSP is a PTAS–reduction, i.e., an approximation-preserving reduction which transforms one optimization problem into another one. For this purpose, we map instances of Max-3dm(3) to instances of Max-Multi-Pkf-SSP with the same polynomial time construction used for reducing 3dm(3) to Multi-Pkf-SSP. We then prove that this construction also maps a solution of Max-Multi-Pkf-SSP to a solution of Max-3dm(3) using Lemma 8.5.1.

Lemma 8.5.1. *Our reduction from an instance I of* Max-3dm(3) *to an instance I' of* Max-Multi-Pkf-SSP *yields that*

- *if I has a matching of size n then $|Opt(I')| = P$;*
- *if I has a matching of size at most $(1-\epsilon_0)n$ then $|Opt(I')| \leq P - \alpha\epsilon_0 nE$ where $\alpha > 0$ is a constant.*

Proof. This lemma directly follows from Lemmas 8.4.1 and 8.4.8. □

Given an overview of our approach, we formally prove the main result of this section in what follows.

Theorem 8.5.2. Max-Multi-Pkf-SSP *is* APX*-hard.*

Proof. Let's be more specific about the values of P and E. Theorem 8.2.8 assures that parameter E, the length of each xyz-support domain, is $\Theta(\log_2 n)$. Using our sequence design and Lemma 8.3.1, we also get that instance I' includes $\Theta(n) + \Theta(m)$ domains of length $\Theta(E)$. Since we are working with instances of Max-3dm(3), we know that the number of triples in the instance is $m \leq 3n$. From all of our assumptions we can conclude that $P = \Theta(n \log_2 n)$.

We now apply Lemma 8.5.1 to show APX-hardness of Max-Multi-Pkf-SSP. Suppose by contradiction that for some $\epsilon > 0$, there is a $(1-\epsilon)$–approximation algorithm for this problem. Then,

- if I of Max-3dm(3) has a matching of size n, on instance I' of Max-Multi-Pkf-SSP the algorithm returns a solution with value at least $(1-\epsilon)|\text{Opt}(I')| = (1-\epsilon)P$;
- if I has a matching of size at most $(1-\epsilon_0)n$, on instance I' the algorithm returns a solution with value at most $|\text{Opt}(I')| \leq P - \alpha\epsilon_0 nE$.

Therefore, if

$$P - \alpha\epsilon_0 nE < (1-\epsilon)P \tag{8.5.1}$$

the algorithm can distinguish between the cases where I has a matching of size n or of size at most $(1-\epsilon_0)n$. By our current assumptions about P and E, equation 8.5.1 holds if

$$\epsilon < \frac{\alpha\epsilon_0 nE}{P}. \tag{8.5.2}$$

This contradicts the APX-hardness of Max-3dm(3) (Theorem 8.1.2). □

8.6 Conclusion

A basic question that has remained open from over three decades of work on computational pseudoknot-free secondary structure prediction of nucleic acids is: can we efficiently compute the minimum free energy (MFE) pseudoknot-free secondary structure for a multi-set of DNA or RNA strands? We have shown that this problem is NP-hard, and is therefore computationally intractable, unless P = NP. A natural question then is whether solutions to the problem can be efficiently approximated, if P ≠ NP. Unfortunately,

there is a limit to the accuracy of any such method. We have shown that the optimization problem of finding the MFE structure for a multi-set of nucleic acid strands is hard for the complexity class APX, the class of NP optimization problems that have constant factor approximation algorithms. The result implies that there does not exist a polynomial time approximation scheme for this problem, unless P = NP. Given these results, it suggests that heuristic methods, such as stochastic local search, and randomized algorithms should be investigated for structure prediction of multiple interacting strands.

Chapter 9

Summary and Future Work

9.1 Summary

Many problems of practical interest rely on continuous-time Markov chains (CTMCs) defined over combinatorial state spaces, rendering the computation of transition probabilities, and hence probabilistic inference, difficult or impossible with existing methods. For these problems, where classical methods are not applicable, the main alternative has been particle Markov chain Monte Carlo methods. In Chapter 7, we have proposed an efficient particle-based Monte Carlo method, called TIPS, to approach inference in CTMCs with weak assumptions on the state space using an importance sampling approach. Our method requires a user-specified potential function centered at the target end point and satisfying some certain conditions. We have defined our proposal sampling based on this potential function and proved that our proposal hits the target with probability one. We have also showed that our method (TIPS) outperforms the forward sampling method on nucleic acid folding pathways which is an important examples of CTMCs and demonstrated that in a range of realistic inferential setups, our scheme dramatically reduces the variance of the Monte Carlo approximation.

In Chapter 8, we have showed that, while efficient thermodynamics-based approaches are well known for prediction of pseudoknot-free secondary structures of single strands, the problem of predicting pseudoknot-free secondary structures of multiple interacting strands is computationally intractable unless $\mathsf{P} = \mathsf{NP}$. Our proof uses a polynomial time reduction from a variant of 3-dimensional matching to our problem Multi-Pkf-SSP. To provide this reduction, we have designed our sequences employing code word designs with high pairwise edit distance of Schulman and Zukerman [91]. However, we encountered an issue in their proof and fixed it in Lemma 8.2.4. Moreover, we have also proved that there are no polynomial time algorithms to provide an approximation for the MFE structure of a set of nucleic-acid strands unless $\mathsf{P} = \mathsf{NP}$ and therefore the problem is APX-hard as well.

9.2 Future Work

Our work in this part of thesis can be extended as follows.

9.2.1 Nucleic Acid Folding Pathways

Parameter Tuning One caveat of our results is that our method, TIPS, was sensitive to a range of values of the tuning parameters α and β. For example, we simply tried different values for parameter α and found that the accuracy of our sampling in RNA folding pathways was susceptible to the setting of this parameter (see Figure 9.1).

We believe that the behavior of our method is sensitive to α, β, because the sampled jump chains are typically longer in RNA folding pathways. Intuitively, for longer folding times, the transition probabilities are more influenced by the low probability particles or paths, as these low probability paths comprise a greater percent of all possible paths. This means that any setting of α that heavily biases the sampled paths to be from the region just around x and y will need to sample a large number of paths in order to approximate the contribution of paths with a low probability. This situation is analogous to the well-known problems in importance sampling of mismatches between the proposal and actual distributions. Similar sampling considerations apply to parameter β which controls the number of excursions from y. If β is too restrictive, again, paths will be sampled that do not well reflect the actual probability of excursions. Parameter tuning is therefore an important area of future work. It might be possible to use some automated tuners [52, 105] or to approach the problem by essentially creating mixtures of proposals each with its own tuning parameters.

Subset Selection In Section 7.1.1, we mentioned that the choice of subset S, a subset of secondary structures for a given nucleic acid, is an important decision, but we didn't argue how to choose the connected set S to *optimize* the accuracy of a given subset model. In fact, there are many possible choices of S. For example in Tang et al. [97], subset S is randomly sampled according to the Boltzmann distribution of the structures. In Kirkpatrick et al. [60], instead, subset S is a mixture of suboptimal structures (i.e., those with the closest free energy to the MFE structure) and structures sampled from the Boltzmann distribution, which is also used as our selection model in Section 7.3. As another example, Wolfinger et al. [108] use the set of

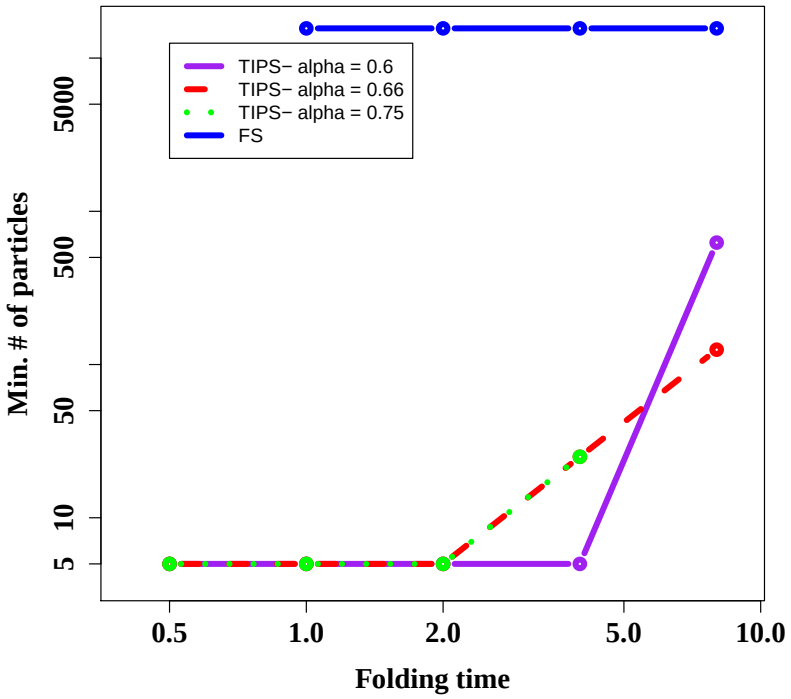


Figure 9.1: Tuning parameter α. Performance of our method (TIPS) using different values of α compared to forward sampling (FS) for estimating the folding pathway of the 1XV6 molecule on its *full* state space. The minimum number of particles required to perform each sampling method is shown on the y-axis.

local minima or metastable structures[20] and their connecting structures, known as saddle points, as subset S. Finding the best choice of S of a given size m that would minimize the inaccuracy of a particular subset model is still an intractable and open problem. However, we believe the optimal subsets have some tractable and explorable properties (such as connectivity, probability, diversity, inclusion of saddle points, etc.), and understanding these properties can be very helpful for modelling RNA folding pathways on subset S. As a future work, we are interested to study the impact of these properties in the quality of a subset.

9.2.2 Multi-stranded Nucleic Acid Secondary Structure Prediction

Computational Experiments We proved our results using a simple energy model. Although it would seem unlikely to have an easier prediction problem if a more complicated energy model is employed, it is still valuable to provide some computational experiments using a realistic energy model. For example, we are interested to run the following useful experiment: 1) consider an arbitrary 3DM(3) matching instance, 2) reduce it using our sequence design algorithm in Section 8.2, and 3) use some available software such as NUPACK [110] for predicting the MFE structure of multiple interacting strands under the use of the Turner energy model. This way, e.g., for the small 3DM(3) instance shown in Figure 8.2, we will end up with 42 strands with a total length of 4427. Unfortunately, there is no feasible way to conduct such an experiment at this stage, because there is no software capable of handling so many large sequences. We also thought about simplifying our experiment by calculating the MFE structure using a fixed order of strands. However, the number of sequences and their total length are too large to be handled by any existing software suites. Therefore, a possible future work can be our contribution for extending the NUPACK source code to support larger multi-stranded nucleic acid systems.

Another variant of the multi-stranded prediction problem In our work, we studied the hardness of the general prediction problem where there is no restriction (except pseudoknot-free constraint) on the multiple interacting strands. However, we can consider another variant of the problem where the goal is to find the MFE structure of a restricted mutli-set of

[20] Formally, structure x is called a local minimum if $E(x) \leq E(y)$ for all $y \in N(x)$ where $E(x)$ is the free energy and $N(x)$ is the set of all neighbours of x.

strands, e.g., a set of strands containing only specific types of nucleotides or a set with arbitrary edit distances (with no lower bound) between its strands. Then, this alternative problem may be more manageable and the hardness result may be avoided. We can investigate the complexity of this new problem as another area of future work.

Bibliography

[1] *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co. New York, New York, NY, USA.

[2] J. P. Abrashams, M. V. D. Berg, E. V. Batenburg, and C. Pleij. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. *Nucleic Acids Research*, 18(10):3035–3035, May 1990.

[3] M. Akkouchi. On the convolution of exponential distributions. *Journal of the Chungcheong Mathematical Society*, 21(4):502–510, 2008.

[4] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Molecular Biology of the Cell*. Garland Science, 4th edition, 2002.

[5] D. Alistarh, J. Aspnes, D. Eisenstat, R. Gelashvili, and R. L. Rivest. Time-space trade-offs in population protocols. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2560–2579, 2017.

[6] S. Amari and R. Misra. Closed-form expressions for distribution of sum of exponential random variables. *IEEE Transactions on Reliability*, 46(4):519–522, 1997.

[7] M. Andronescu, B. V., H. Hoos, and A. Condon. RNA STRAND: the RNA secondary structure and statistical analysis database. *BMC Bioinformatics*, 9:340, 2008.

[8] M. Andronescu, Z. C. Zhang, and A. Condon. Secondary structure prediction of interacting RNA molecules. *Journal of Molecular Biology*, 345(5):987–1001, February 2005.

[9] D. Angluin, J. Aspnes, Z. Diamadi, M. Fischer, and R. Peralta. Computation in networks of passively mobile finite-state sensors. *Distributed Computing*, 18:235–253, 2006. Preliminary version appeared in PODC 2004.

[10] D. Angluin, J. Aspnes, and D. Eisenstat. Stably computable predicates are semilinear. In *PODC 2006: Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 292–299, New York, NY, USA, 2006. ACM Press.

[11] D. Angluin, J. Aspnes, and D. Eisenstat. Fast computation by population protocols with a leader. *Distributed Computing*, 21(3):183–199, September 2008. Preliminary version appeared in DISC 2006.

[12] D. Angluin, J. Aspnes, and D. Eisenstat. A simple population protocol for fast robust approximate majority. *Distributed Computing*, 21(2):87–102, July 2008.

[13] D. P. Bartel. MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116:281–297, 2004.

[14] L. Becchetti, A. Clementi, E. Natale, F. Pasquale, R. Silvestri, and L. Trevisan. Simple dynamics for plurality consensus. *Distributed Computing*, pages 1–14, 2016.

[15] L. Becchetti, A. E. F. Clementi, E. Natale, F. Pasquale, and L. Trevisan. Stabilizing consensus with many opinions. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 620–635, 2016.

[16] A. Belleville, D. Doty, and D. Soloveichik. Hardness of computing and approximating predicates and functions with leaderless population. *Proceedings of the 44th International Colloquium on Automata, Languages and Programming [to appear]*, 2017.

[17] K. Bringmann, F. Grandoni, B. Saha, and V. V. Williams. Truly Subcubic Algorithms for Language Edit Distance and RNA-Folding via Fast Bounded-Difference Min-Plus Product. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 375–384, October 2016.

[18] L. Cardelli. Strand algebras for DNA computing. *Natural Computing*, 10(1):407–428, 2011.

[19] L. Cardelli and A. Csiksz-Nagy. The Cell Cycle Switch Computes Approximate Majority. *Scientific Reports*, 2:srep00656, September 2012.

[20] L. Cardelli, M. Kwiatkowska, and L. Laurenti. Programming discrete distributions with chemical reaction networks. In *Rondelez Y., Woods D. (eds) DNA Computing and Molecular Programming, Lecture Notes in Computer Science*, volume 9818, pages 35–51. Springer, Cham, 2016.

[21] H.-L. Chen, D. Doty, and D. Soloveichik. Deterministic function computation with chemical reaction networks. In *DNA 18: Proceedings of The 18th International Meeting on DNA Computing and Molecular Programming*, volume 7433 of *Lecture Notes in Computer Science*, pages 25–42. Springer, 2012.

[22] H.-L. Chen, D. Doty, and D. Soloveichik. Rate-independent computation in mass-action chemical reaction networks. In *ITCS 2014: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, pages 313–326, 2014.

[23] S.-J. Chen. RNA folding: Conformational statistics, folding kinetics, and ion electrostatics. *Annual Review of Biophysics*, 37(1):197–214, 2008. PMID: 18573079.

[24] Y.-J. Chen, N. Dalchau, N. Srinivas, A. Phillips, L. Cardelli, D. Soloveichik, and G. Seelig. Programmable chemical controllers made from DNA. *Nature Nanotechnology*, 8(10):755–762, October 2013.

[25] Y.-J. Chen, B. Groves, R. A. Muscat, and G. Seelig. DNA nanotechnology from the test tube to the cell. *Nature Nanotechnology*, 10(9):748–760, September 2015.

[26] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[27] H. M. T. Choi, J. Y. Chang, L. A. Trinh, J. E. Padilla, S. E. Fraser, and N. A. Pierce. Programmable in situ amplification for multiplexed imaging of mRNA expression. *Nature Biotechnology*, 28:1208– 1212, 2010.

[28] A. Condon, M. Hajiaghayi, D. Kirkpatrick, and J. Manuch. Simplifying analyses of chemical reaction networks for approximate majority. *Proceedings of the 23th International Conference on DNA Computing and Molecular Programming (DNA23) [to appear]*, 2017.

[29] A. Condon, M. Hajiaghayi, and C. Thachuk. Predicting minimum free energy structures of multi-stranded nucleic acid systems is APX-Hard.

[30] M. Cook, D. Soloveichik, E. Winfree, and J. Bruck. Programmability of chemical reaction networks. In *Algorithmic Bioprocesses*, pages 543–584. Springer Berlin Heidelberg, 2009.

[31] J. Cruise and A. Ganesh. Probabilistic consensus via polling and majority rules. *Queueing Systems*, 78(2):99–120, 2014.

[32] 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, January 2007.

[33] D. Doty and M. Hajiaghayi. Leaderless Deterministic Chemical Reaction Networks. In *Proceedings of the 19th International Conference on DNA Computing and Molecular Programming*, volume 8141 of *DNA 19*, pages 46–60, 2013.

[34] D. Doty and M. Hajiaghayi. Leaderless deterministic chemical reaction networks. *Natural Computing*, 14(2):213–223, June 2015.

[35] A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer, 2001.

[36] M. Draief and M. Vojnovic. Convergence speed of binary interval consensus. *SIAM Journal on Control and Optimization*, 50(3):1087–1109, 2012.

[37] B. Felden. RNA structure: experimental analysis. *Current Opinion in Microbiology*, 10(3):286–291, June 2007.

[38] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. Wiley, New York, 3rd edition, 1968.

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

[40] K. Fujibayashi, R. Hariadi, S. H. Park, E. Winfree, and S. Murata. Toward reliable algorithmic self-assembly of DNA tiles: A fixed-width cellular automaton pattern. *Nano Letters*, 8(7):1791–1797, July 2008.

[41] K. Gerdes and E. G. H. Wagner. RNA antitoxins. *Current Opinion in Microbiology*, 10(2):117–124, April 2007.

[42] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[43] S. Ginsburg and E. H. Spanier. Semigroups, Presburger formulas, and languages. *Pacific Journal of Mathematics*, 16(2):285–296, 1966.

[44] W. K. Grassmann. Transient solutions in markovian queueing systems. *Computers and Operations Research*, 4:47–100, 1977.

[45] G. R. Grimmett and D. R. Stirzaker. *Probability and Random Process*. Oxford University Press, New York, 2nd edition, 1992.

[46] B. Groves, Y.-J. Chen, C. Zurla, S. Pochekailov, J. L. Kirschman, P. J. Santangelo, and G. Seelig. Computing in mammalian cells with nucleic acid strand exchange. *Nature Nanotechnology*, 11(3):287–294, March 2016.

[47] M. Hajiaghayi, B. Kirkpatrick, L. Wang, and A. Bouchard-Ct. Efficient continuous-time markov chain estimation. *ICML*, 2014.

[48] D. Han, S. Pal, J. Nangreave, Z. Deng, Y. Liu, and H. Yan. DNA origami with complex curvatures in three-dimensional space. *Science*, 332(6027):342, 2011.

[49] D. S. Hirschberg. *Pattern matching algorithms*, chapter Serial Computations of Levenshtein Distances, pages 123–142. Oxford University Press, 1997.

[50] A. Hjelmfelt, E. D. Weinberger, and J. Ross. Chemical implementation of neural networks and Turing machines. *Proceedings of the National Academy of Sciences of the United States of America*, 88(24):10983–10987, 1991.

[51] J. P. Huelsenbeck and F. Ronquist. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8):754–755, 2001.

[52] F. Hutter, H. H. Hoos, K. Leyton-Brown, and T. Stützle. ParamILS: an automatic algorithm configuration framework. *Journal of Artificial Intelligence Research*, 36:267–306, October 2009.

[53] H. Isambert. The jerky and knotty dynamics of RNA. *Methods*, 49(2):189–96, 2009.

[54] J. A. Jaeger, D. H. Turner, and M. Zuker. Predicting optimal and suboptimal secondary structure for RNA. *Methods in Enzymology*, 183:281–306, 1990.

[55] H. Jiang, M. Riedel, and K. Parhi. Digital signal processing with molecular reactions. *IEEE Design and Test of Computers*, 29(3):21–31, 2012.

[56] J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652 – 656, 1972.

[57] N. G. v. Kampen. *Stochastic processes in physics and chemistry*. North-Holland, Amsterdam; New York, 1992.

[58] V. Kann. Maximum bounded 3-dimensional matching is MAX SNP-complete. *Information Processing Letters*, 37(1):27–35, January 1991.

[59] V. Kann. Maximum bounded h-matching is max SNP-complete. *Information Processing Letters*, 49(6):309–318, March 1994.

[60] B. Kirkpatrick, M. Hajiaghayi, and A. Condon. A new model for approximating RNA folding trajectories and population kinetics. *Computational Science & Discovery*, 6(1):014003, January 2013.

[61] A. Korostelev and H. F. Noller. The ribosome in focus: new structures bring new insights. *Trends in Biochemical Sciences*, 32:434–441, 2007.

[62] M. R. Lakin, D. Parker, L. Cardelli, M. Kwiatkowska, and A. Phillips. Design and analysis of DNA strand displacement devices using probabilistic model checking. *Journal of the Royal Society, Interface*, 9(72):1470–1485, July 2012.

[63] R. Lorenz, S. H. Bernhart, C. Hner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler, and I. L. Hofacker. ViennaRNA package 2.0. *Algorithms for Molecular Biology : AMB*, 6:26, November 2011.

[64] Z. J. Lu, D. H. Turner, and D. H. Mathews. A set of nearest neighbor parameters for predicting the enthalpy change of RNA secondary structure formation. *Nucleic Acids Research*, 34(17):4912–4924, October 2006.

[65] M. O. Magnasco. Chemical kinetics is Turing universal. *Physical Review Letters*, 78(6):1190–1193, 1997.

[66] P. Maragakis, F. Ritort, C. Bustamante, M. Karplus, and G. E. Crooks. Bayesian estimates of free energies from nonequilibrium work data in the presence of instrument noise. *The Journal of Chemical Physics*, 129(2), 2008.

[67] H. M. Martinez. An RNA folding rule. *Nucleic Acids Research*, 12(1 Pt 1):323–334, January 1984.

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

[69] D. H. MATHEWS. Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization. *RNA*, 10(8):1178–1190, 2004.

[70] D. H. Mathews, M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker, and D. H. Turner. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 101(19):7287–7292, May 2004.

[71] D. H. Mathews and D. H. Turner. Prediction of RNA secondary structure by free energy minimization. *Current Opinion in Structural Biology*, 16(3):270–278, June 2006.

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

[73] C. McDiarmid. On the method of bounded differences. *London Society Lecture Note Series*, 141:148–188, 1989.

[74] G. B. MertziosSotiris, E. Nikoletseas, C. L. Raptopoulos, and P. G. Spirakis. Determining majority in networks with local interactions and very small local memory. *Distributed Computing*, 30(1):1–16, 2017.

[75] B. Munsky and M. Khammash. The finite state projection algorithm for the solution of the chemical master equation. *Journal of Chemistry Physics*, 124(4):1–13, 2006.

[76] J. Norris. *Markov Chains*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 1997.

[77] R. Nussinov and A. B. Jacobson. Fast algorithm for predicting the secondary structure of single-stranded RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 77(11):6309 – 6313, November 1980.

[78] R. Nussinov, G. Pieczenik, J. Griggs, and D. Kleitman. Algorithms for loop matchings. *SIAM Journal on Applied Mathematics*, 35(1):68–82, July 1978.

[79] Z. Nutov and I. Beniaminy. A (1-1/e)-approximation algorithm for the generalized assignment problem. *Operations Research Letters - ORL*, 34(3), 2006.

[80] T. Pan. How RNA function requires structure. *Nature Structural and Molecular Biology*, 5(7):540–541, July 1998.

[81] E. Perron, D. Vasudevan, and M. Vojnovic. Using three states for binary consensus on complete graphs. In *Proceedings of the 28th IEEE Conference on Computer Communications (INFOCOM)*, pages 2527–2535, 2009.

[82] A. Phillips and L. Cardelli. A programming language for composable DNA circuits. *Journal of the Royal Society Interface*, 6 Suppl 4:S419–436, August 2009.

[83] L. Qian, E. Winfree, and J. Bruck. Neural network computation with DNA strand displacement cascades. *Nature*, 475(7356):368–372, 2011.

[84] L. Qian and E. Winfree. A simple DNA gate motif for synthesizing large-scale circuits. *Journal of The Royal Society Interface*, pages 71–89, February 2011.

[85] V. Rao and Y. W. Teh. Fast MCMC sampling for Markov jump processes and continuous time Bayesian networks. In *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*, pages 619–626, Corvallis, Oregon, 2011.

[86] V. Rao and Y. W. Teh. MCMC for continuous-time discrete-state systems. *NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems*, pages 701–709, 2012.

[87] P. W. K. Rothemund. Folding DNA to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, March 2006.

[88] G. R.R., L. J.C., and C. J.J. The accuracy of ribosomal RNA comparative structure models. *Current Opinion in Structural Biology*, 12(3):301–310, 2002.

[89] Y. Saad. *Numerical Methods for Large Eigenvalue Problems*. Halsted Press, New York, NY, 1992.

[90] J. Schaeffer. *The Multistrand Simulator: Stochastic Simulation of the Kinetics of Multiple Interacting DNA Strands*. PhD thesis, California Institute of Technology, 2012.

[91] L. Schulman and D. Zuckerman. Asymptotically good codes correcting insertions, deletions, and transpositions. *IEEE Transactions on Information Theory*, 45(7):2552–2557, November 1999.

[92] G. Seelig, D. Soloveichik, D. Y. Zhang, and E. Winfree. Enzyme-free nucleic acid logic circuits. *Science*, 314(5805):1585–1588, 2006.

[93] J. A. Shapiro. Genome Informatics: The Role of DNA in Cellular Computations. *Biological Theory*, 1(3):288–301, September 2006.

[94] D. Soloveichik, M. Cook, E. Winfree, and J. Bruck. Computation with finite stochastic chemical reaction networks. *Natural Computing*, 7(4):615–633, 2008.

[95] D. Soloveichik, G. Seelig, and E. Winfree. DNA as a universal substrate for chemical kinetics. *Proceedings of the National Academy of Sciences of the United States of America*, 107(12):5393–5398, 2010.

[96] M. Tacker, W. Fontana, P. F. Stadler, and P. Schuster. Statistics of RNA melting kinetics. *European Biophysics Journal*, 23(1):29–38, April 1994.

[97] X. Tang, B. Kirkpatrick, S. Thomas, G. Song, and N. M. Amato. Using motion planning to study RNA folding kinetics. *Journal of Computational Biology*, 12(6):862–881, 2005.

[98] X. Tang, S. Thomas, L. Tapia, and N. M. Amato. Tools for simulating and analyzing RNA folding kinetics. In *Proceedings of the 11th annual international conference on Research in computational molecular biology*, RECOMB'07, pages 268–282, Berlin, Heidelberg, 2007. Springer-Verlag.

[99] C. Thachuk, E. Winfree, and D. Soloveichik. Leakless DNA Strand Displacement Systems. In *DNA Computing and Molecular Programming*, volume 9211 of *Lecture Notes in Computer Science*, pages 133–153. Springer International Publishing, August 2015.

[100] D. Thirumalai, N. Lee, S. A. Woodson, and D. Klimov. Early events in RNA folding. *Annual Review of Physical Chemistry*, 52(1):751–762, 2001.

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

[102] I. Tinoco, D. Collin, and P. T. X. Li. The effect of force on thermodynamics and kinetics: unfolding single RNA molecules. *Biochemical Society Transactions*, 32(Pt 5):757–760, 2004.

[103] S. Valadkhan. The spliceosome: caught in a web of shifting interactions. *Current Opinion in Structural Biology*, 17:310–315, 2007.

[104] S. Venkataraman, R. M. Dirks, C. T. Ueda, and N. A. Pierce. Selective cell death mediated by small conditional RNAs. *Proceedings of the National Academy of Sciences*, 107(39):16777–16782, 2010.

[105] Z. Wang, S. Mohamed, and N. de Freitas. Adaptive hamiltonian and riemann monte carlo samplers. In *International Conference on Machine Learning (ICML)*, 2013.

[106] J. K. Wickiser, W. C. Winkler, R. R. Breaker, and D. M. Crothers. The speed of RNA transcription and metabolite binding kinetics operate an FMN riboswitch. *Molecular Cell*, 18:49–60, 2005.

[107] R. C. Wilson and J. A. Doudna. Molecular mechanisms of RNA interference. *Annual Review of Biophysics*, 42:217–239, 2013.

[108] M. T. Wolfinger, W. A. Svrcek-Seiler, C. Flamm, I. L. Hofacker, and P. F. Stadler. Exact folding dynamics of RNA secondary structures. *Journal of Physics A: Mathematical and General*, 37:4731–4741, 2004.

[109] S. Wuchty, W. Fontana, I. L. Hofacker, and P. Schuster. Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, 49(2):145–165, February 1999.

[110] J. N. Zadeh, C. D. Steenberg, J. S. Bois, B. R. Wolfe, M. B. Pierce, A. R. Khan, R. M. Dirks, and N. A. Pierce. NUPACK: analysis and

design of nucleic acid systems. *Journal of Computational Chemistry*, 32:170–173, 2011.

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

[112] D. Y. Zhang and E. Winfree. Control of DNA strand displacement kinetics using toehold exchange. *Journal of the American Chemical Society*, 131(47):17303–17314, 2009.

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

[114] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, July 1984.