Analysis of Interacting Nucleic Acids in Dilute Solutions

Thesis by

Justin S. Bois

In Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy

California Institute of Technology
Pasadena, California

2007
(Defended April 30, 2007)

Acknowledgements

First and foremost, I thank Prof. Niles Pierce. He has provided me with a working environment where I am free to explore topics that spark my curiosity, all the while participating in an ambitious research program making nucleic acids do things I never imagined they could. He is a good scientific citizen, committed to ethically and passionately conducting meaningful research, providing quality instruction in the classroom, reaching out to the youth in the community, and improving the Institute as a whole. In the last several years, he has profoundly influenced my development as a scientist, leading by example with skill, enthusiasm, and integrity.

I have had the good fortune of being co-advised by (and being a teaching-assistant with) Prof. Zhen-Gang Wang. He has consistently provided clear, patient advice. His depth of understanding is apparent in his comments, and also in his probing questions. He has encouraged me to think deeply about scientific problems and to distill them to their essentials. He has very generously shared his talents with me, and I am very grateful.

My other two committee members have also been very helpful to me in the past couple of years. Prof. Erik Winfree and his group are close collaborators, also being creative with nucleic acids. I have had many fruitful discussions with him, finding them both challenging and enjoyable. Prof. Christina Smolke is also an expert on nucleic acid technology. I have enjoyed discussions with her and her group about our respective research, all of which I find very exciting. I also had the great pleasure of being a teaching assistant with her.

All of the work presented in this thesis is the product of collaboration with other talented scientists, and the Pierce Lab seems to be full of them! In particular, Robert Dirks and I worked closely on developing analysis tools for interacting strands. He laid the groundwork for almost all work in the Pierce Lab, including development of partition function algorithms. His experimental studies were done in conjunction with much of the theoretical and computational work in this thesis. All experimental results shown in this thesis were acquired by him. Joe Zadeh and I worked together to develop NUPACK (Chapter 4), with assistance from Marshall Pierce. Joe did the bulk of the coding to build the front end of the analysis web application, making the guts Robert and I built under the hood look pretty (and more importantly accessible) for the world. Suvir Venkataraman and Harry Choi conducted experiments that served to validate and extend ideas I developed on topological interactions in hybridization kinetics. Melinda Kirk greatly facilitates the work in the Pierce Lab, and certainly made my life easier. In addition to these individuals, all members of the Pierce Lab, past

and present, have contributed to my work here in one way or another, none of which are insignificant, and I am grateful.

Outside of the Pierce Lab, I also wish to thank the members of the Wang Group, especially Andy Spakowitz, for enlightening discussions. Joseph Schaeffer in the Winfree Lab is currently developing software for kinetic simulations of interacting nucleic acid strands and had a key role in writing our paper on their thermodynamics. We shared a floor of the Broad Center with the Elowitz, Mayo, and Phillips labs, all of whom have provided stimulating discussions.

I consider many of the aforementioned people to be good friends, and they have enhanced my experience at Caltech on a personal as well as professional level. Additionally, the four guys who seem to be involved in nearly every lunch (always Ernie's) and weekend I've had for years are Alex Brown, Aditya Khair, Rafael Verduzco, and Akira Villar. They are among my best friends and will undoubtedly continue to be. I also had the pleasure of playing on numerous athletic teams and am grateful to the members of Lucky Baldwin's F.C. and the Caltech Club Soccer Team and to the members of my Caltech intramural teams, particularly Ernie's A.F. and the Fuzzy Bunnies of Death (F-BOD) for the great times we've shared playing football and basketball.

Finally, I would like to thank my family. My sister Bridget, my brother Jeremy, and more recently my sister-in-law Alexa, have been supportive of me and are close friends. My parents, Steve and Nancy, have always unwaveringly provided me with more love and support than I could ever thank them for. It is the pleasure I derive from learning that has motivated me from the beginning of my schooling to the writing of this thesis. My mother was my first teacher—it began with her.

Abstract

Motivated by the growing demand for analysis tools for diverse natural and engineered DNA and RNA systems, we develop a general theory and set of computational algorithms to perform thermodynamic analysis of dilute reactive solutions and then apply these techniques to interacting nucleic acids. The theory correctly accounts for the effects of indistinguishability in partition function calculations for complexes of interacting strands. With partition functions in hand, the unique complex concentrations corresponding to thermodynamic equilibrium are obtained by solving a convex programming problem. Partition function and concentration information can then be used to calculate equilibrium base-pairing observables corresponding to experimentally measurable properties. The underlying physics and mathematical formulation of these problems lead to an interesting blend of approaches, including ideas from graph theory, group theory, dynamic programming, combinatorics, convex optimization, and Lagrange duality.

To make these analysis tools available to researchers worldwide, we present NUPACK, a web-based software suite for thermodynamic analysis of nucleic acids. Its efficacy is demonstrated in example calculations and the results are shown to be in agreement with experiment.

Finally, the thermodynamic properties of a DNA-based triggered self-assembly device [1] are analyzed using NUPACK and extensions of its tools. The computational results complement experimental studies, exposing novel properties about the system and dictating further research.

Contents

List of Figures

List of Notation

Following is a list of mathematical symbols used in this thesis with Greek symbols listed after Roman symbols. In some cases, symbols are multiply defined, such as k, which is the Boltzmann constant or an indexing variable. In cases like these, the use of the symbol is obvious in context and the most pervasive use is the one given in this list. Only symbols used throughout the thesis are presented. The page on which the symbol is first defined is given at right.

$*$	superscript indicating optimal value (or critical value in HCR)	13
$\_$	overbar indicating distinguishable particles or strands	6
$'$	superscript indicating all particles or strands in the system are distinguishable	10
$\dagger$	superscript indicating alternate variables for kinetics simulation	30
$\cdot$	dot indicating pairing of bases, as in $i \cdot j$	20
$[\text{H1}]_0^*$	critical HCR monomer concentration	64
$[\text{X}]_0$	total molarity of strand type X	36
A	stoichiometry matrix with entries A_{ij}	11
A_{ij}	number of particles or strands of type i in compound or ordered complex j	11
C_n	cyclic group of order n	7
D_n	dihedral group of order $2n$	7
F	constant volume free energy	9
F_{box}	constant volume free energy of a box	11
f	per-solvent constant volume free energy in units of kT	12
$f_A(i_A \cdot j_B)$	fraction of A strands that have base pair $i_A \cdot j_B$	29
$f_A(i_A)$	fraction of A strands that have base i paired	29
f_{unpaired}	fraction of bases that are unpaired in a solution	34
$\mathcal{G}_j$	permutation group acting on the set Y_j	5
G_j	free energy of HCR product j (see Figure 5.3 for detailed descriptions)	48
$G(s)$	free energy of secondary structure s	21
G_{box}	constant pressure free energy of a box	17
g	per-solvent constant pressure free energy in units of kT	17
h	Lagrange dual function	13

K	equilibrium constant (HCR equilibrium constants defined on page 61)	4
k	Boltzmann constant	8
$\mathcal{L}$	Lagrangian	13
L_j	total number of particles or strands in a compound or ordered complex j	5
$L_{\max}$	maximum complex size to be considered in a calculation	16
ℓ	average polymer length excluding monomers	68
M	total number of compounds or ordered complexes	3
M^0	total number of particles or strands	5
M_s	total number of solvent molecules in a box	11
m	vector of compound or ordered complex populations	5
m^0	vector of total number of particles or strands	5
N	strand length	22
$N_{\max}$	maximum number of bases in an ordered complex	26
n	number of HCR monomers in an HCR product	49
$n_{\max}$	number of HCR monomers in an HCR product	53
$n(m)$	accounting factor	6
PDI	polydispersity index	68
p	pressure	17
$p_k(i_n \cdot j_m)$	equilibrium probability of base pair $i_n \cdot j_m$ in ordered complex k	28
$p(n)$	equilibrium probability of an initiated HCR polymer of length n	64
Q	vector with entry j being Q_j	13
Q_j	volume-independent portion of partition function for compound or ordered complex j	9
Q_{box}	partition function of a box of interacting particles or strands	10
Q_{ref}	partition function defining reference free energy	11
s	secondary structure	21
T	temperature	3
T_p	polymerization temperature	63
V	volume of a system or “box”	3
$v(\pi)$	number of cyclic permutations, equal to periodic strand repeat in ordered complex	24
X	total mole fraction of HCR products	67
x	vector of compound or ordered complex mole fractions	12
x_i	mole fraction of species i (see Figure 5.3 for HCR products)	4
x^0	vector of total particle or strand mole fractions	12
x_Π	vector of mole fractions of supercompounds	15
Y_j	set of legal arrangements of particle identifiers	5
Z	partition function of a system	3

z	single particle partition function	3
ΔG	change in free energy for formation of an HCR product (see Figure 5.6)	51
Λ	set of population vectors satisfying conservation of mass	11
λ	vector of Lagrange multipliers	13
μ_i	chemical potential of species i	4
ξ	HCR parameter, $\xi \equiv K_1 x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}$	61
Π	maximal subset of distinct circular permutations with respect to strand type	23
Π_{A_k}	subset of Ψ that contains compounds with stoichiometry A_k	15
$\overline{\Pi}$	set of circular permutations of strand identifiers	23
π	a strand ordering	23
ρ^*	overlap concentration	26
$\phi(d)$	Euler's totient function acting on d	25
$\phi^{\mathrm{even}}, \phi^{\mathrm{odd}}$	fraction of monomers in polymers with n even and odd, respectively	70
$\phi_\ell^{\mathrm{even}}, \phi_\ell^{\mathrm{odd}}$	fraction of polymers with n even and odd, respectively	70
Ψ	set of allowed compounds or ordered complexes	5
Ψ^0	set of particle or strand types	4
Ψ^{super}	set of supercompounds	16
Ω	ensemble of secondary structures	22

Chapter 1

Introduction

DNA is the primary genetic storage medium for life. RNA plays a more varied role in biology, participating in storage, regulation, catalysis and synthesis [2]. The capabilities of these versatile molecules rely on a few fundamental features resulting from their composition. DNA and RNA are polynucleotides, where each nucleotide consists of a negatively charged phosphate group, a sugar, and one of four types of base: A, C, G, and T for DNA, with U replacing T for RNA. Sequences of these four bases are responsible for coding the genetic information. Nucleic acid energetics are dominated by the formation of base pairs between complementary bases (Watson-Crick pairs C-G and A-T (DNA) or A-U (RNA) and wobble pairs G-T (DNA) or G-U (RNA)), each base participating in at most one base pair. Strands are directional (with the beginning denoted $5'$ and the end denoted $3'$) and base pairing occurs in an anti-parallel fashion (e.g., $5'$-GCTCA-$3'$ is the reverse-complement of $5'$-TGAGC-$3'$, allowing complete base pairing to yield a familiar DNA double helix). In the most general scenario, base pairing within a single DNA or RNA strand, or between multiple strands that have arbitrary degrees of complementarity, is possible. These interstrand and intrastrand interactions, being pairwise (as opposed to spread over multiple units on the polymer chains, as for proteins) allow for the high fidelity and efficiency of DNA transcription and replication (and ultimately translation) and for RNAs to fold into specific structures for their diverse tasks [3].

In addition to enabling their varied roles in biology, the unique structural properties of nucleic acids make them attractive materials for engineering nanoscale structures and devices. The well-defined pairwise nature of the dominant interactions allows coarsening of the state space describing the physics of the system to include only base pairs. This enables *in silico* study of base pairing by fast algorithms, which in turn makes nucleic acid strands a programmable material. By appropriately designing the sequence of bases in each strand, synthetic nucleic acid systems can be programmed to self-assemble into complex structures and to implement dynamic mechanical tasks. The field of nucleic acid nanotechnology is devoted to exploring and developing these capabilities for applications in molecular robotics, fabrication, computation, biosensing, electronics, and medicine. A myriad of novel systems have been designed and constructed including Holliday junctions, cubes, lattices, octahedra, tic-tac-toe games, tweezers, walkers, nanotubes, triggered self-assembly devices, programmable two-dimensional patterns, DNA logic circuits, and a host of other devices [4–7].

As nucleic acids in biological processes continue to be studied, the diversity of their roles becomes more and more apparent. Similarly, nucleic-acid based nanotechnology is a burgeoning field with new applications continually emerging. To keep pace with progress in these areas of research, we have developed theoretical and computational analysis tools to characterize base pairing in nucleic acid-based systems. To date, most applications have dealt with the folding of single RNA or single-stranded DNA (ssDNA) strands [8–13]. Our work extends these tools to enable study of multiple interacting strands, a hallmark of DNA nanotechnology applications and also important in RNA regulatory processes in biology [14, 15]. Although biology typically operates in the crowded environment of the cell, and nucleic acid-based nanotechnology will undoubtedly be entering the cell in coming years, the new techniques we have developed are restricted to dilute solutions. The tools are still useful, as many experiments of biological importance are done *in vitro* and technological devices continue to be developed in buffer solutions. For technological applications, the analysis tools complement sequence design [16] and are crucial for expanding capabilities for developing new, more complex devices.

This thesis describes theoretical and computational analysis tools to study interacting nucleic acid strands and is best read sequentially, as every chapter uses results from chapters coming before it. Read this way it tells a story, summarized as follows. A general theory describing the thermodynamics of interacting particles is developed (Chapter 2) and applied to nucleic acid systems to enable extension of existing tools for studying base pairing in single strands to multiple interacting strands (Chapter 3). These newly developed tools are shared with researchers worldwide through NUPACK, a web-based tool for analysis of interacting nucleic acids (Chapter 4). The tools are then used, in conjunction with experiments, to characterize the synthetic nucleic acid-based system known as hybridization chain reaction [1] (Chapter 5).

Chapter 2

Thermodynamics of dilute mixtures

The work presented in this chapter and in Chapter 3 is largely based on [17], R. M. Dirks, J. S. Bois, J. M. Schaeffer, E. Winfree, N. A. Pierce, Thermodynamic analysis of interacting nucleic acid strands, *SIAM Rev.*, **49**, 65–88 (2007). Reprinted with permission from the Society of Industrial and Applied Mathematics.

2.1 Introduction

The characterization of equilibrium of dilute mixtures is one of the classic problems in statistical thermodynamics, typically encountered in a student's first course on the subject. The usual first example is a monatomic ideal gas. The partition function for M atoms[1] of a monatomic ideal gas in a fixed volume V is [18]

$$Z(M,V,T) = \frac{(z(T))^M}{M!},$$

where $z(T)$ is the single-particle partition function at temperature T and the $M!$ accounts for the indistinguishability of particles. This expression is valid when the temperature is high enough such that there are so many quantum states available that the chances that any two particles are in the same one is vanishingly small. Alternatively stated, the prevailing energy model dictates that any given particle is indistinguishable from any other and that the particles do not interact with each other.

A lecture or two after the monatomic ideal gas is introduced, the student computes a partition function for a homonuclear diatomic ideal gas. When the temperature is high enough that nuclear spin states need not be considered, the single-compound partition function $z(T)$ is computed as if the two nuclei were distinguishable and then divided by a "symmetry factor" of two to account for the rotational symmetry of the molecule.

These corrections are typically described as "overcounting" corrections in the partition function [18–20]. When the like particles are considered distinguishable, the same physical state (in which like particles cannot be distinguished) is counted too many times in the partition function. While this explanation is conceptually pleasing and indeed very useful, it is still subject of study and debate [21–23]. A more rigorous framework

[1] We use the variable M to denote number of atoms as opposed to the traditional N. This is to maintain consistency with notation in other parts of this thesis and published work applying the present development to nucleic acid solutions.

with precise language ensures correct application of distinguishability corrections to more complicated systems. In Sections 2.2 and 2.3, such a framework is provided. While it may be unnecessarily complicated for describing such simple systems as diatomic ideal gases, the application to nucleic acids in Chapter 3 demonstrates its utility.

For reactive systems, after partition functions are calculated (with distinguishability corrections in place) for the species populating a dilute mixture, the problem of finding the chemical equilibrium—the population of each type of species—remains. For each chemical reaction $\sum_i \nu_i \, i = 0$ (where ν_i is the stoichiometric coefficient of species i and the sum is over the species present in the mixture), the equilibrium condition is

$$\sum_i \nu_i \, \mu_i = 0, \tag{2.1}$$

where μ_i is the chemical potential of species i [24]. For dilute mixtures, this results in equilibrium expressions for each reaction,

$$K(T) = \prod_i x_i^{\nu_i}, \tag{2.2}$$

where $K(T)$ is the equilibrium constant and x_i is the mole fraction of species i at equilibrium.

When there are many reactions in the mixture, the resulting nonlinear system of algebraic equations is very difficult to solve. To address this problem, the conditions for chemical equilibrium are developed from an optimization theoretic perspective in Section 2.4. We then show that although not amenable to reacting ideal gases at constant pressure, the chemical equilibrium of a dilute solution with reactive solute may be determined very efficiently by solving a strictly convex optimization problem.

The work in Sections 2.2 and 2.3 is a generalization of the *Distinguishability Correction Theorem* derived by Robert Dirks and published in [17], which was applied to complexes of nucleic acid strands. The differences between the present development and that of Dirks are discussed in Section 3.4.2. The work in Section 2.4 was also published in [17].

2.2 Definitions

We begin the mathematical treatment of dilute mixtures by codifying the language used to describe their components. The thermodynamic system consists of particles and compounds in a closed container (or "box") of finite volume V. Following are the formal definitions of a *particle* and a *compound* and associated properties of each. The terms defined in this section are used in this and subsequent chapters.

Definition 2.1. A *particle* is an irreducible unit of a specified *type*. Two particles of different types are always distinguishable from each other.

We define the set of all particle types in the system to be Ψ^0. The number of particles in the system is given by $m^0 \in \mathbb{Z}_{\geq 0}^{|\Psi^0|}$, where m_i^0 is the number of particles of type i, and the total number of particles is $M^0 \equiv \sum_{i \in \Psi^0} m_i^0$. A particle may have a unique *identifier* (in $\{1, 2, \ldots, M^0\}$) associated with it which serves to distinguish it from other particles, including those of like type.

Definition 2.2. A given *compound* j consists of $L_j \geq 1$ interacting particles and is defined by the energy model describing the interactions among its constituent particles (self-interactions included). The energy model dictates the number, positions, and types of its constituent particles and the state space they occupy and the energetics with which they interact. A compound does not have any interactions with other compounds.

Of course the compounds cannot be strictly noninteracting, lest thermodynamic equilibrium never be reached. By saying the compounds are noninteracting, we mean that they interact enough for energy and particle exchange to establish equilibrium, but the interactions are weak enough such that intercompound forces are negligible [18]. If the energetic interactions between particles are considered edges and the particles vertices, a compound is defined to be connected in a graph theoretic sense and are not connected to any other compound (they are independent).

In the case of distinguishable particles, we can assign a set of particle identifiers to compound j, with each distinguishable particle getting a unique identifier. The set Y_j is the set of all arrangements of particle identifiers such that the appropriate identifiers correspond to the correct type of particle in the compound (i.e., a given identifier may only be matched with one type of particle).

The energy model defines an equivalence relation used to compare elements of Y_j. Thus, compound j has associated with it a permutation group $\mathcal{G}_j$ which acts on the set Y_j by permuting the identifiers such that the resulting re-labeled compounds belong to the same equivalence class as the identity element of $\mathcal{G}_j$. For a given $y \in Y_j$, all permutations in this equivalence class (called an orbit of y under $\mathcal{G}_j$ acting on Y_j, $\mathrm{orb}_{\mathcal{G}_j}(y) = \{g(y) \in Y_j : g \in \mathcal{G}_j\}$), are of course equivalent to y under the energy model.

It is important to note that the group $\mathcal{G}_j$ derives from symmetries in the *energy model* and not from, for example, spatial orientations of the particles or the particular energetics of a given state (a state being a particular point in the phase space associated with the energy model), though these may coincide with the energy model symmetry. To clarify this point, consider a compound consisting of three particles, two of type A and one of type B in which each A particle interacts with the B particle, as depicted in Figure 2.1. Although the spatial arrangement of the particles may be symmetric, only symmetry in the energy model affects the composition of the group $\mathcal{G}_j$. In the example in Figure 2.1, we could have $|\mathcal{G}_j| = 2$ for (a) and $|\mathcal{G}_j| = 1$ for (b) and (c). Note that for all three, $|Y_j| = 2$.

The contents of the system are defined by Ψ, the set of compounds in a system composed from indistinguishable (unlabeled) particles in Ψ^0, with $\Psi^0 \subseteq \Psi$. The *population* of compounds in the system is given by $m \in \mathbb{Z}_{\geq 0}^{|\Psi|}$ with m_j being the number of compounds of type j in the system.

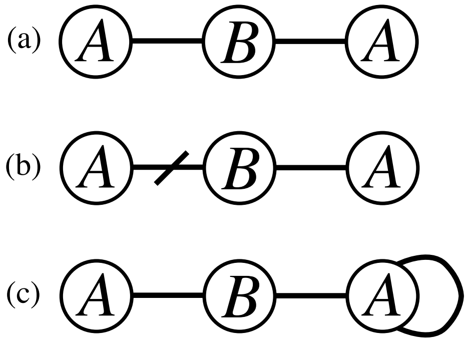


Figure 2.1: An example compound consisting of two A particles and a B particle. The interactions specified by the energy model are depicted as thick lines connecting the particles. (a) The energy model is symmetric. (b) Even though the spatial orientation is symmetric, the energy model is not. The energy model describing the interaction of B with the left A (marked by a tick) is different from that with the right A. Note that the tick identifies a different energy *model*, not different energetics in a particular state of the compound. (c) Another example of an asymmetric energy model in which one of the A molecules has a self-interaction and the other does not.

2.3 Effects of indistinguishable particles

In this section, we formally derive the effects of the indistinguishability of particles on the thermodynamics of a system of independent compounds at equilibrium and on results from computer simulations.

2.3.1 Particle accounting and indistinguishability for a convenient case

It is often convenient to compute a single-compound partition function as if the particles in the compound were distinguishable and then correct for the indistinguishability of particles of like types *ex post facto* [18]. The independence of compounds is then invoked to determine the partition function of the entire system. Here, I codify this process, which is commonly done in the study of ideal gases and dilute solutions.

Consider two systems with varying forms of indistinguishability. “System 1” is the physical system in which all particles of like type are indistinguishable. “System 2” is identical to system 1, except all particles are distinguishable, and the specific (distinguishable) compound that a given particle belongs to is fixed, as is its position in the compound (as “position” pertains to the energy model and not necessarily to position in space). Thus, each compound in system 2 has a unique subset of particle identifiers associated with it.

For a given state of system 1, let the *accounting factor* $\overline{n}(m)$ be the number of equivalent states in system 2. In this context, equivalence means that particles of given types interact in system 2 exactly as they do in system 1. The fact that $\overline{n}(m)$ can be greater than unity is a result of multiple possible arrangements of particle identifiers. In what follows, variables marked with an over-line refer to system 2.

Theorem 2.1. $\overline{n}(m) = \prod_{j\in\Psi} m_j!\,|\mathcal{G}_j|^{m_j}$.

Proof. Consider one of the $\overline{n}(m)$ states of system 2. We can attain another state equivalent to that in system 1 by exchanging the positions of two given compounds of like type in state space, which is equivalent to exchanging the subsets of particle identifiers of the two compounds. In fact, all possible permutations of the particle identifier subsets among compounds of like type gives a state equivalent to that of system 1. Note that a permutation of subsets is not a permutation of the elements in a subset, but a permutation of subsets associated with given compounds. For a compound of type j, there are $m_j!$ permutations of particle identifier subsets. There are therefore $\prod_{j \in \Psi} m_j!$ total ways to arrange them.

Consider now an arbitrary compound k of type j in system 1. We wish to calculate $\overline{n}_k$, the number of equivalent states of the corresponding compound in system 2. Recall that in system 2 compound k has a fixed subset of identifiers and in evaluating $\overline{n}_k$, we consider all legal permutations of identifiers (those that preserve the matching of an identifier with its specified particle type).

Because all particles are distinguishable, the stabilizer[2] of any given $y \in Y_j$ in $\mathcal{G}_j$ is the identity element, and thus $|\mathrm{stab}_{\mathcal{G}_j}(y)| = 1 \; \forall \; y \in Y_j$. By the orbit-stabilizer theorem [25], the number of elements in the orbit of any given $y \in Y_j$ under $\mathcal{G}_j$ is $|\mathrm{orb}_{\mathcal{G}_j}(y)| = |\mathcal{G}_j|/|\mathrm{stab}_{\mathcal{G}_j}(y)| = |\mathcal{G}_j|$. Thus, each of the equivalence classes into which Y_j is divided by the equivalence relation induced by $\mathcal{G}_j$ is of the same size, $|\mathcal{G}_j|$. Therefore, for the specified permutation of particle identifiers in compound k in system 2 (or any other arbitrary legal permutation), there are $|\mathcal{G}_j|$ permutations that are equivalent under the energy model, giving $\overline{n}_k = |\mathcal{G}_j|$.

Since the compounds by definition do not interact, the multiplication principle of combinatorics [26] gives that $\overline{n}(m)$ is the product of the number of ways to arrange the identifier subsets and the number of ways to arrange the identifiers in each complex. Thus, there are $\overline{n}(m) = \prod_{j \in \Psi} m_j! \, |G_j|^{m_j}$ states in system 2 corresponding to a given state in system 1. □

As an example, consider a compound j consisting of four identical particles. Figure 2.2 shows the partition of the set Y_j into orbits under the action of various $\mathcal{G}_j$. If the energy model considers all rotations and reflections[3] of a given state to be equivalent, then $\mathcal{G}_j$ is the dihedral group D_4.[4] Since in this case $|\mathcal{G}_j| = |D_4| = 8$, each equivalence class has eight elements and therefore $\overline{n}_j = 8$. However, if only rotations are considered equivalent, the group is the cyclic group C_4, and $\overline{n}_j = |\mathcal{G}_j| = |C_4| = 4$.

2.3.2 Implications of indistinguishability on thermodynamic potentials

We now investigate the consequences of particle indistinguishability on the thermodynamic potentials. Let $\Gamma(m, V, E)$ be the number of states of system 1 with energy E and population m. Since $\overline{n}(m)$ is independent of energy,

$$\overline{\Gamma}(m, V, E) = \overline{n}(m) \, \Gamma(m, V, E).$$

[2] The stabilizer of a $y \in Y_j$ in $\mathcal{G}_j$ is the subset of $\mathcal{G}_j$ whose elements act on y to map it to itself, or $\mathrm{stab}_{\mathcal{G}_j}(y) = \{g \in \mathcal{G}_j : g(y) = y\}$.

[3] These need not necessarily be *physical* rotations and reflections, but may be, depending on the energy model.

[4] We use D_n to denote the dihedral group of order $2n$.

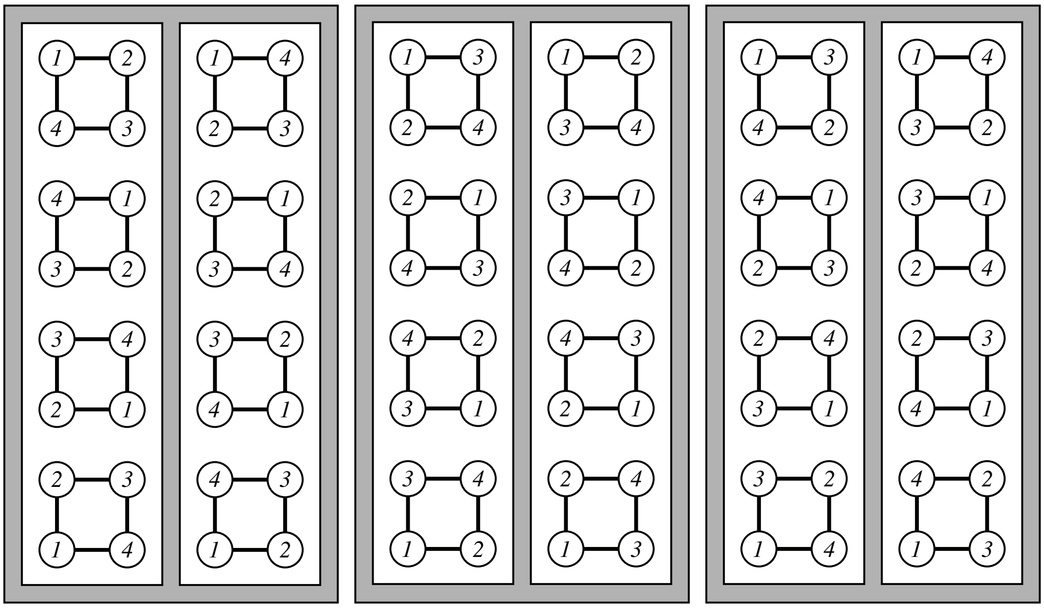


Figure 2.2: A example compound where all particles are of like type. Numbers represent particle identifiers. Each white box contains elements of Y_j that are in the same equivalence class (orbit) under the cyclic group C_4. Each shaded box contains elements that are in the same equivalence class under the dihedral group D_4.

The partition function is the Boltzmann-weighted sum over the energies available to the system.

$$\begin{aligned}\overline{Z}(m,V,T) &= \sum_E \overline{\Gamma}(m,V,E)e^{-E/kT} = \overline{n}(m)\sum_E \Gamma(m,V,E)e^{-E/kT} = \overline{n}(m)\,Z(m,V,T) \\ &= \left(\prod_{j\in\Psi} m_j!\,|\mathcal{G}_j|^{m_j}\right) Z(m,V,T), \qquad (2.3)\end{aligned}$$

where k is the Boltzmann constant.

If we consider the special case where the system is a single compound of type j, (2.3) gives

$$Z(V,T) = \overline{Z}_j(V,T)/|\mathcal{G}_j|. \qquad (2.4)$$

By virtue of the fact that compounds are by definition independent, the free energy of the system is additive in the single-compound free energies and the partition function is multiplicative in the single-compound partition functions $Z_j(V,T)$.

$$\overline{Z}(m,V,T) = \prod_{j\in\Psi}(\overline{Z}_j(V,T))^{m_j}. \qquad (2.5)$$

Combining (2.3), (2.4), and (2.5),

$$Z(m, V, T) = \prod_{j \in \Psi} \frac{1}{m_j!} \left(\frac{\overline{Z}_j(V, T)}{|\mathcal{G}_j|} \right)^{m_j} . \tag{2.6}$$

We can make inferences from this equation about the functional form of $Z_j(V, T)$. According to (2.4) and (2.6), the partition function of the system is

$$Z(m, V, T) = \prod_{j \in \Psi} \frac{(Z_j(V, T))^{m_j}}{m_j!}.$$

In the thermodynamic limit of a large system with $m_j \gg 1 \,\forall\, j \in \Psi$, the free energy of the box is

$$\frac{F(m, V, T)}{kT} = -\log Z(m, V, T) = \sum_{j \in \Psi} \left[m_j (\log m_j - \log Z_j(V, T) - 1) + \mathcal{O}(\log m_j) \right],$$

where Stirling's approximation has been applied.[5] The free energy must be a first-order homogeneous function[6] of its extensive variables, V and m [24]. Therefore,

$$Z_j(V, T) = V\, Q_j(T),$$

where $Q_j(T)$ is the volume-independent portion of the partition function for compound j (with dimension of inverse volume). Note that only $Q_j(T)$ depends on the $\mathcal{G}_j$, so

$$Q_j(T) = \frac{\overline{Q}_j(T)}{|\mathcal{G}_j|} \tag{2.7}$$

by (2.4). We thus have

$$Z(m, V, T) = \prod_{j \in \Psi} \frac{(V\, Q_j(T))^{m_j}}{m_j!} = \prod_{j \in \Psi} \frac{1}{m_j!} \left(\frac{V\, \overline{Q}_j(T)}{|\mathcal{G}_j|} \right)^{m_j} . \tag{2.8}$$

The resulting free energy (ignoring the $\mathcal{O}(\log m_j)$ terms) is[7]

$$\frac{F(m, V, T)}{kT} = \sum_{j \in \Psi} m_j \left(\log \left(\frac{m_j}{V} \right) - \log Q_j(T) - 1 \right). \tag{2.9}$$

[5] Stirling's approximation is that for large n, $\log n! = n \log n - n + \mathcal{O}(\log n)$.

[6] A function f is a first-order homogeneous function of its dependent variables x if $f(a\, x) = a\, f(x) \;\forall\; a \in \mathbb{R}_{>0}$.

[7] It seems awkward to have two logarithms of quantities that are not nondimensional in (2.9). We choose to write it as shown to illustrate the separation of volume and temperature dependence in the free energy.

2.3.3 Particle accounting and indistinguishability in computer simulations

When performing computer simulations considering individual particles, all particles are necessarily distinguishable because the properties of each one occupy specific addresses in the computer's memory. Correct accounting of distinguishability is therefore required to connect the results of a computer simulation to a physical system. In a similar treatment as Section 2.3.1, consider "system 3", which is identical to system 1, except all particles are distinguishable and the subset of identifiers assigned to a given compound are not fixed. Variables describing quantities in system 3 are marked with a prime, and its accounting factor is $n'(m^0)$.

Lemma 2.1. $n'(m^0) = \prod_{i\in\Psi^0} m_i^0!$.

Proof. We construct a state of system 3 by placing the particles of appropriate types in the same arrangement as in the state specified by system 1. We then count the number of ways to label the particles with unique identifiers to give $n'(m^0)$, since all particles are distinguishable. The number of ways to assign the identifiers for particles of type i is simply $m_i^0!$. Thus, by the multiplication principle, there are a total of $n'(m^0) = \prod_{i\in\Psi^0} m_i^0!$ states in system 3 equivalent to a given state in system 1. □

As a result of Lemma 2.1, when we are considering a given single compound in system 3, all elements in Y_j are in the same equivalence class. In the example in Figure 2.2, $n'_j(m) = |Y_j| = 4! = 24$.

Treating system 3 similarly as in Section 2.3.2, we get

$$Z'(m, V, T) = n'(m)\, Z(m, V, T) = \left(\prod_{i\in\Psi^0} m_i^0!\right) Z(m, V, T). \tag{2.10}$$

An important consequence is that a computer simulation in which all particles are distinguishable produces the same equilibrium behavior as the physical system. The overall free energy is shifted by a constant $-kT \sum_{i\in\Psi^0} \log m_i^0!$, which can be absorbed into the reference state of the free energy for the simulation.

2.4 Concentration determination in dilute solutions

Using the assumed ability to either compute, experimentally determine, or look up the partition function for a single compound as a starting point, we now turn to the problem of analyzing the equilibrium composition of a box of volume V containing m_i^0 particles of type $i \;\forall\; i \in \Psi^0$ that form compounds in the finite set Ψ in a dilute solution.

2.4.1 Partition function of the box

As we proceed to compute the partition function for the box, $Q_{\text{box}}(m^0, V, T)$, we first investigate the contributions of the solvent. A compound is defined to be a species that is independent of all others. Naturally, a compound will interact with the solvent in a solution, as it is in direct contact with the solvent molecules. Therefore, we assume the interactions of a compound with the surrounding solvent are included in the energy model describing the compound and therefore in its partition function $Q_j(T)$. Also implicit in this assumption is that the solvent acts on the compound in a constant way, i.e., the compound-solvent interactions do not depend on the position of the compound in the box and the solvent is at constant density. Finally, while not necessarily implicit in our definition of a compound, we assume that the compounds contribute negligibly to the volume of the box (the solution is dilute), and therefore the number of solvent molecules in the box, M_s, is related to the volume by $M_s = \rho_s V$, where ρ_s is the number density of the solvent. Given this proportionality, we may rewrite the partition function as $Z(m, M_s, T)$ and redefine $Q_j(T)$ to include ρ_s such that it is now dimensionless, enabling us to rewrite (2.8) as

$$Z(m, M_s, T) = \prod_{j\in\Psi} \frac{(M_s\, Q_j(T))^{m_j}}{m_j!}. \tag{2.11}$$

With this equation in hand, we can write the partition function for the box.

$$Q_{\text{box}}(m^0, M_s, T) = Q_{\text{ref}} \sum_{m\in\Lambda} Z(m, M_s, T) = Q_{\text{ref}} \sum_{m\in\Lambda} \prod_{j\in\Psi} \frac{(M_s\, Q_j(T))^{m_j}}{m_j!}. \tag{2.12}$$

Here, Q_{ref} is chosen to set the reference state of the free energy and Λ is the set of population vectors m satisfying the conservation of mass constraint $A\, m = m^0$, where $A \in \mathbb{Z}_{\geq 0}^{|\Psi^0|\times|\Psi|}$ has entries A_{ij} denoting the number of particles of species i in compound j. The free energy of the box is given by $F_{\text{box}} = -kT \log Q_{\text{box}}$. It is often convenient to define F_{box} to be zero when all particles are contained in the box and are in a ground state defined to have zero energy, so we specify $Q_{\text{ref}} \equiv \prod_{i\in\Psi^0} (m_i^0!/M_s^{m_i^0})$.

The probability of population vector m at equilibrium is

$$p(m, M_s, T) = Q_{\text{box}}^{-1}\, Q_{\text{ref}}\, Z(m, M_s, T) \tag{2.13}$$

and the expected value of each population m_j is[8]

$$\langle m_j \rangle = \sum_{m\in\Lambda} m_j\, p(m, M_s, T). \tag{2.14}$$

The probability distribution for each m_j is then found by calculating

$$p_j(n) = \sum_{\substack{m\,\in\,\Lambda \\ \text{s.t. } m_j\,=\,n}} p(m, M_s, T) \tag{2.15}$$

[8] The convention that the particles in the box can form the set of compounds Ψ implies $m_j > 0\; \forall\, j \in \Psi$ for at least one $m \in \Lambda$. Hence, $\langle m_j \rangle > 0\; \forall\, j \in \Psi$.

for each value n taken by m_j in the set Λ. For a box containing a small number of particles, Q_{box} and the equilibrium population distributions can be evaluated explicitly. For a large box containing a large number of particles, explicit enumeration of all population vectors m in Λ is no longer feasible.

2.4.2 Concentration determination in the thermodynamic limit

For large systems, the distributions of extensive thermodynamic variables are Gaussian with variance scaling as the mean [24]. Hence, for large numbers of interacting particles, the distribution of populations, $p(m, M_s, T)$, is a $|\Psi|$-dimensional Gaussian with variance proportional to $\langle m_j \rangle$ in coordinate j. By (2.13), the distribution of $Z(m, M_s, T)$ is a rescaling of $p(m, M_s, T)$ and hence Gaussian, so the sum of (2.12) may be approximated by a product of the height $Q_{\text{ref}}\, Z(\langle m \rangle, M_s, T)$ and the width $\langle m_j \rangle^{1/2}$ in each coordinate $j \in \Psi$:

$$Q_{\text{box}} \approx Q_{\text{ref}}\, Z(\langle m \rangle, M_s, T) \prod_{j \in \Psi} \langle m_j \rangle^{\frac{1}{2}} .$$

Substituting for $Z(\langle m \rangle, M_s, T)$ and applying Stirling's approximation, we obtain the free energy[9]

$$\frac{F_{\text{box}}(\langle m \rangle, V, T)}{kT} = -\log Q_{\text{ref}} + \sum_{j \in \Psi} \left\{ \langle m_j \rangle \left[\log\left(\frac{\langle m_j \rangle}{M_s} \right) - \log Q_j(T) - 1 \right] + \mathcal{O}\left(\log \langle m_j \rangle \right) \right\}, \quad (2.16)$$

analogously to (2.9). The contribution to F_{box} by each compound $j \in \Psi$ scales as $\langle m_j \rangle \log \langle m_j \rangle$, while the error in this contribution resulting from neglecting the width of the distribution and from using Stirling's approximation is only $\mathcal{O}(\log \langle m_j \rangle)$. Hence, for large systems, F_{box} can be accurately calculated by replacing (2.12) with $Q_{\text{box}} \approx Q_{\text{ref}}\, Z(\langle m \rangle, M_s, T)$. On a per-solvent basis, the free energy is then

$$f(\langle x \rangle, T) \equiv \frac{F_{\text{box}}}{M_s kT} \approx f_{\text{ref}} + \sum_{j \in \Psi} \langle x_j \rangle \left(\log \langle x_j \rangle - \log Q_j(T) - 1 \right), \quad (2.17)$$

where $\langle x_j \rangle \equiv \langle m_j \rangle / (M_s + \sum_{k \in \Psi} \langle m_k \rangle) \approx \langle m_j \rangle / M_s$ is the equilibrium concentration[10] of complex species $j \in \Psi$ and $f_{\text{ref}} \equiv -M_s^{-1} \log Q_{\text{ref}} = \sum_{i \in \Psi^0} x_i^0 (1 - \log x_i^0)$. The sharply peaked Gaussian population distributions allow us to equate $\langle m \rangle$ with the population vector m that maximizes $Z(m, M_s, T)$ subject to conservation of mass. Alternatively, we may equate $\langle x \rangle$ with the concentrations $x \approx m/M_s$ that minimize $f(x, T)$ while conserving total particle concentrations $x^0 \approx m^0/M_s$.

The equilibrium concentrations for the compounds can therefore be determined by solving the optimization problem:

$$\min_x \quad f(x, T) \quad (2.18a)$$

[9] In (2.16), we have written F_{box} as a function of $\langle m \rangle$, though it appears as a function of m^0 since Q_{box} is a function only of m^0. As we will later prove, $\langle m \rangle$ is uniquely determined from m^0, so we interchange m^0 for $\langle m \rangle$ for clarity.

[10] This is a dimensionless concentration, equal to a mole fraction.

$$\text{subject to } \; Ax = x^0 \tag{2.18b}$$

for $f(x,T) : \mathbb{R}^{|\Psi|}_{>0} \to \mathbb{R}$, where the constraint enforces conservation of mass. Expressions (2.13) and (2.14) indicate that the equilibrium concentrations are strictly positive.

2.4.3 Convexity and duality

We now seek an efficient, globally convergent algorithm for solving (2.18) to determine the equilibrium concentration of each species of compound. The constraint is linear so the feasible set is convex and the free energy is a strictly convex function of the concentrations [27], as can be observed by noting that the Hessian of $f(x,T)$ is a diagonal positive definite matrix with entries $[\nabla^2_x f(x,T)]_{jj} = x_j^{-1}$. Hence, (2.18) has at most one solution x^* [28].

Defining the Lagrange multipliers $\lambda \in \mathbb{R}^{|\Psi^0|}$ to enforce mass conservation, the Lagrangian is

$$\mathcal{L}(x,\lambda) = f_{\text{ref}} + x^T(\log x - \log Q - \mathbf{1}) + \lambda^T(x^0 - Ax).$$

Here, and in subsequent expressions, we adopt the convention that $\log x$ and e^x denote the termwise logarithm and exponential of a vector x; $\mathbf{1}$ denotes a vector of ones of the appropriate length. The vector $Q \in \mathbb{R}^{|\Psi|}_{>0}$ contains the partition functions $Q_j(T)$ and we do not show the explicit temperature dependence to avoid clutter. The corresponding dual function has the form

$$h(\lambda) = \inf_x \mathcal{L}(x,\lambda) = f_{\text{ref}} + \lambda^T x^0 - Q^T e^{A^T\lambda},$$

and the dual problem corresponding to (2.18) is the unconstrained optimization problem

$$\max_{\lambda} \;\; h(\lambda) \tag{2.19}$$

with $h(\lambda) : \mathbb{R}^{|\Psi^0|} \to \mathbb{R}$.

Suppose the primal problem (2.18) has optimal value p^* and the dual problem (2.19) has optimal value d^*. For a convex primal problem, if the constraints satisfy the strong Slater conditions (full row rank for A in addition to feasibility) then strong duality holds ($p^* = d^*$) and the Karush-Kuhn-Tucker (KKT) optimality conditions

$$\nabla_x \mathcal{L}(x^*,\lambda^*) = \log x^* - \log Q - A^T\lambda^* = 0 \tag{2.20a}$$

$$A\,x^* = x^0 \tag{2.20b}$$

are necessary and sufficient for x^* and λ^* to be primal and dual optimal, respectively [28, 29]. The constraint matrix A has full row rank because $A_{ij} = \delta_{ij}$ for $j \in \Psi^0$. Primal feasibility is verified by letting $x_j = \epsilon$ for $j \in \Psi \backslash \Psi^0$ and $x_i = x_i^0 - \epsilon \sum_{j \in \Psi \backslash \Psi^0} A_{ij}$ for $i \in \Psi^0$ with $\epsilon > 0$ sufficiently small.

By the following lemma, the Hessian of $h(\lambda)$ is negative definite, so the dual problem (2.19) is strictly concave with at most one solution λ^*. The strong Slater conditions further ensure that d^* is finite and that there exists a corresponding finite λ^* [28, 29].

Lemma 2.2. *The Hessian $\nabla^2 h(\lambda)$ is real symmetric negative definite.*

Proof. The Hessian entries are given by

$$[\nabla^2 h(\lambda)]_{kl} = -\sum_{j \in \Psi} A_{kj}\, A_{lj}\, Q_j \, \exp\left\{ \sum_{i \in \Psi^0} \lambda_i A_{ij} \right\} \qquad \forall\, k, l \in \Psi^0, \tag{2.21}$$

so the Hessian is real and symmetric by inspection. The Hessian is negative definite if $y^T \nabla^2 h\, y < 0$ for $y \neq 0$. We note that $\nabla^2 h = -R^T R$ where $R \in \mathbb{R}_{\geq 0}^{|\Psi| \times |\Psi^0|}$ has entries $R_{ji} = A_{ij}[Q_j \, \exp\{\sum_{i \in \Psi^0} \lambda_i A_{ij}\}]^{1/2}$. Hence, $y^T \nabla^2 h\, y = -y^T R^T R\, y = -\|Ry\|^2$, which is negative provided R has linearly independent columns. A has full row rank so R has full column rank and hence $\nabla^2 h(\lambda)$ is negative definite. □

We now show that λ^* fully determines x^* so we are free to solve the dual problem (2.19) instead of the primal one (2.18). This is advantageous because the number of compound species $|\Psi|$ can be large even when the number of particle types $|\Psi^0|$ is small. The dual solution λ^* satisfies $\nabla h(\lambda^*) = 0$ or

$$A\, e^{A^T \lambda^* + \log Q} = x^0. \tag{2.22}$$

The first KKT condition (2.20a) gives an explicit representation for $x^* \in \mathbb{R}_{>0}^{|\Psi|}$ in terms of λ^*

$$x^* = e^{A^T \lambda^* + \log Q}, \tag{2.23}$$

and referring to (2.22) we see that the second KKT condition (2.20b) is also satisfied. Equating $\langle x \rangle \approx x^*$, the (positive) concentrations corresponding to thermodynamic equilibrium represent the unique solution to (2.18).

Any globally convergent unconstrained optimization algorithm applied to the dual problem (2.19) will suffice to find λ^*. We consider the equivalent dual problem $\min_\lambda(-h(\lambda))$ and apply a trust-region method with a Newton dog-leg step [30] that exploits the symmetric positive definiteness of $\nabla^2(-h(\lambda))$ by using Cholesky decomposition for the Newton matrix inversions.[11] For this problem, the trust-region method converges globally (in arbitrary-precision arithmetic) with quadratic local convergence [30, 31].[12]

[11] Implementation details are provided as comments in the file `concentrations.c` and related files in the NUPACK 2.0 source code distribution, which is freely available for research purposes at `nupack.org`.

[12] This result follows for a function $f(\lambda)$ that is twice continuously differentiable and bounded below with $\nabla^2 f(\lambda)$ Lipschitz continuous and $\|\nabla^2 f(\lambda)\| \leq \beta$ on the level set $\mathcal{S} \equiv \{\lambda \,|\, f(\lambda) \leq f(\lambda_0)\}$, where λ_0 is the initial guess [30, 31]. In our case, $f(\lambda)$ is infinitely differentiable. The strong Slater conditions ensure that $f(\lambda^*)$ is finite and furthermore that λ^* is finite [28, 29]. By Lemma 2.2, the Hessian $\nabla^2 f(\lambda)$ is positive definite so the outward normal derivative df/dn on the ball $B(\lambda^*, \epsilon) = \{\lambda^* + \epsilon u \,|\, \|u\| = 1\}$ satisfies a

2.4.4 Relation to standard equilibrium expressions

The equilibrium conditions derived in this section are equivalent to the standard expressions defining chemical equilibrium, as given by (2.1) and (2.2). To see this, note that for $j \in \Psi^0$, $A_{ij} = \delta_{ij}$, so $\lambda_i^* = \log(x_i^*/Q_i) = \mu_i/kT$, which is the dimensionless chemical potential for particle j in solution. Substitution of these expressions into the remaining $|\Psi| - |\Psi^0|$ equations for x_j^* in (2.20a) and results in

$$\log x_j^* - \log Q_j - \sum_{i \in \Psi^0} A_{ij}\, \mu_i/kT = \mu_j/kT - \sum_{i \in \Psi^0} A_{ij}\, \mu_i/kT = 0\, \forall\, j \in \Psi \backslash \Psi^0,$$

which is equivalent to (2.1) for the chemical reaction $j - \sum_{i \in \Psi^0} A_{ij}\, i = 0$ describing the construction of compound j from its constituent particles, where, A_{ij} is a stoichiometric coefficient. Using (2.23), the equilibrium expression (2.20a) may also be written as

$$x_j = K_j \prod_{i \in \Psi^0} x_i^{A_{ij}}\, \forall\, j \in \Psi \backslash \Psi^0, \quad \text{with } K_j = \frac{Q_j}{\prod_{i \in \Psi^0} Q_i^{A_{ij}}},$$

which is equivalent to (2.2).

2.4.5 Speed of the algorithm

The most expensive step of each iteration of the trust-region method is the calculation of the Hessian using (2.21)[13], the time complexity of which is $\mathcal{O}(|\Psi|\, |\Psi^0|^2)$.

The size of the optimization problem may be reduced by collapsing compounds with the same stoichiometry into a single "supercompound." In other words, each compound j has associated with it a column in the matrix A, $A_j \in \mathbb{Z}_{\geq 0}^{|\Psi^0|}$, and compounds j and k have the same stoichiometry if $A_j = A_k$. Let Π_{A_k} by the subset of Ψ that contains compounds with stoichiometry A_k. By (2.23),

$$\left\langle x_{\Pi_{A_k}} \right\rangle \equiv \sum_{k \in \Pi_{A_k}} \langle x_k \rangle = e^{A_k^T \lambda^*} \sum_{k \in \Pi_{A_k}} Q_k \text{ and } \frac{\langle x_k \rangle}{\left\langle x_{\Pi_{A_k}} \right\rangle} = \frac{Q_k}{\sum_{k \in \Pi_{A_k}} Q_k}.$$

We can thus define a supercompound for each subset $\Pi_{A_k} \subseteq \Psi$ with partition function $Q_{\Pi_{A_k}} = \sum_{k \in \Pi_{A_k}} Q_k$ and recast the set Ψ in terms of these supercompounds. We call the resulting set Ψ^{super}. The concentrations of the supercompounds, $x_\Pi \in \mathbb{R}_{>0}^{|\Psi^{\text{super}}|}$ are then computed using the same trust-region methods. The

bound $df/dn \geq \delta(n) \geq \delta_0 > 0$, where the uniform bound follows from the continuity of $f'(\lambda)$ on the compact set $B(\lambda^*, \epsilon)$. The normal derivative continues to increase as we proceed outward from the ball along any normal $n \in \mathbb{R}^{|\Psi^0|}$, so the distance $s(n)$ from λ^* to the boundary of $\mathcal{S}$ satisfies $s(n) \leq \epsilon + [f(\lambda_0) - f(\lambda^*)]/\delta_0$. Hence, the level set $\mathcal{S}$ is bounded. The continuity of the Hessian entries $[\nabla^2 f(\lambda)]_{ij}$ ensures that they are bounded on the closure of the bounded set $\mathcal{S}$ (say, $\max_{\lambda \in \mathcal{S}} [\nabla^2 f(\lambda)]_{ij} \leq \alpha,\ \forall i, j \in \Psi^0$). Hence, $\operatorname{tr}(\nabla^2 f(\lambda)) = \sigma^T \mathbf{1} \leq \alpha |\Psi^0| \equiv \beta$, where $\sigma(\lambda) : \mathbb{R}^{|\Psi^0|} \to \mathbb{R}_{>0}^{|\Psi^0|}$ denotes the eigenvalues of $\nabla^2 f(\lambda)$. For a symmetric positive definite Hessian we have $\|\nabla^2 f(\lambda)\| = \sigma_{\max}$ and hence $\|\nabla^2 f(\lambda)\| \leq \beta$ on $\mathcal{S}$.

[13] The second most expensive step is Cholesky decomposition, whose time complexity is $\mathcal{O}(|\Psi^0|^3)$ [32].

concentration for compound j in subset Π_{A_k} is

$$\langle x_j \rangle = \left\langle x_{\Pi_{A_k}} \right\rangle \frac{Q_j}{Q_{\Pi_{A_k}}}. \tag{2.24}$$

In principle, a compound can contain arbitrarily many particles, so in practice we limit the size of Ψ. This can be achieved, for example, by specifying Ψ to contain only those compounds that are expected to be physically significant. For example, we could consider all possible compounds with L particles for $1 \leq L \leq L_{\max}$.[14] In this case, the total number of supercompounds is then given by[15]

$$|\Psi^{\text{super}}| = \begin{pmatrix} L_{\max} + |\Psi^0| \\ |\Psi^0| \end{pmatrix} - 1. \tag{2.25}$$

Hence, we typically have $|\Psi^0| \ll |\Psi^{\text{super}}|$, and thus the time complexity scales linearly with the size of the problem (which is essentially the number of compounds).

2.5 Application to reacting ideal gas mixtures

The algorithm described in the previous section has great utility for dilute solutions at constant volume. In this section, we investigate its applicability to a mixture of reacting ideal gases.

2.5.1 Reacting ideal gas mixture in a fixed volume

The treatment and algorithm described in the Section 2.4 is easily applied to a reacting ideal mixture in a box with constant volume, provided an increase in pressure resulting from the reactions does not affect the ideality of the gas mixture. In all equations, the volume of the box, V, is substituted for M_s. The resulting optimization problem is completely analogous (*cf.* (2.9), (2.17), and (2.18)),

$$\min_c \quad f(c) = f_{\text{ref}} + \sum_{j \in \Psi} c_j (\log c_j - \log Q_j(T) - 1)$$

$$\text{subject to} \quad Ac = c^0,$$

with $c \equiv \langle m \rangle / V$ and $c^0 \equiv \left\langle m^0 \right\rangle / V$.

[14] For many physical systems, significant concentrations will be observed only for small compounds due to the entropic cost of particle association. For such systems, an effective strategy is to start with a small value of $L_{\max}$, calculate the equilibrium concentrations, increment $L_{\max}$, and then recalculate the concentrations to check that there are no significant changes, repeating this process if necessary. This strategy will not work for crystals and polymerization reactions for which there is a substantial nucleation barrier (requiring a critical complex size to be achieved before further aggregation becomes energetically favorable).

[15] This is equivalent to the number of ways to distribute $L_{\max}$ indistinguishable balls amongst $|\Psi^0| + 1$ distinct urns [33].

2.5.2 Reacting ideal gas mixture at constant pressure

For reactions in a dilute incompressible solution, the changes in concentrations of compounds have negligible effect on the pressure or volume of the system. Thus, the same equilibrium concentrations are achieved if the system is held at constant pressure as when held at constant volume. This is not true in the case of an ideal gas. When the pressure p is held constant, which is often the case for reacting gas mixtures of interest, the thermodynamic potential to be minimized is

$$\frac{G_{\text{box}}(\langle m\rangle, p, T)}{kT} = \frac{F_{\text{box}}}{kT} + \frac{pV}{kT} = -\log Q_{\text{ref}} + \sum_{j\in\Psi} \langle m_j\rangle \left[\log\left(\frac{\langle m_j\rangle}{M}\right) - \log\left(\frac{kT}{p}\, Q_j(T)\right)\right], \quad (2.26)$$

where we have used the fact that $pV/kT = M \equiv \sum_{j\in\Psi} \langle m_j\rangle$. (Recall that for the ideal case, $Q_j(T)$ has dimension of inverse volume.) The corresponding optimization problem is

$$\min_m \;\; g(m) \equiv G_{\text{box}}(m, p, T)/kT \quad (2.27a)$$

$$\text{subject to } \; Am = m^0, \quad (2.27b)$$

where the explicit dependence of g on p and T (which are constant) is dropped for convenience. We note that $g(m)$ is a convex function, with proof given in Appendix A.

Analogously to the development in Section 2.4.3, we can define Lagrange multipliers $\lambda \in \mathbb{R}^{|\Psi^0|}$ to enforce mass conservation and write the Lagrangian as

$$\mathcal{L}(m, \lambda) = g_{\text{ref}} + m^T \left[\log\left(\frac{m}{M}\right) - \log\left(\frac{kT}{p}\, Q\right)\right] + \lambda^T(m^0 - Am).$$

Given that $g(m)$ is convex and the set of constraints are linear in (2.27b), the primal problem is convex. The strong Slater conditions are satisfied as in Section 2.4.3, so strong duality holds and the KKT conditions

$$\nabla_m \mathcal{L}(m^*, \lambda^*) = \log\left(\frac{m^*}{M}\right) - \log\left(\frac{kT}{p}\, Q\right) - A^T\lambda^* = 0 \quad (2.28)$$

$$Am^* = m^0$$

are again necessary and sufficient conditions for m^* and λ^* to be primal and dual optimal, respectively.

We now compute the dual function, $h(\lambda) = \inf_m \mathcal{L}(m, \lambda)$. Since $\mathcal{L}(m, \lambda)$ is the sum of convex and affine functions of m, it is a convex function of m, so the infimum must satisfy (2.28). Thus,

$$\frac{m}{M} = \frac{kT}{p}\, Q\, e^{A^T\lambda}. \quad (2.29)$$

The system of equations defined by (2.29) may be written in matrix form as $Bm = 0$ where the entries of B

are given by

$$B_{ij} = \begin{cases} 1 - \frac{kT}{p}\, Q_j\, e^{\lambda^T A j} & \text{for } i = j \\ -\frac{kT}{p}\, Q_j\, e^{\lambda^T A j} & \text{for } i \neq j \end{cases} .$$

Equations analogous to (2.13) and (2.14) can be derived for this system, so $m > 0$. Therefore, (2.29) holds when B is singular, or $\det(B) = 0$. To compute the determinant, we perform operations to get B in upper triangular form. Starting with row $|\Psi|$ and working up to row 2, from each row i subtract row $(i-1)$. Row 1 remains unchanged and the entries for rows 2 through $|\Psi|$ are given by

$$B_{ij} \sim \begin{cases} 1 & \text{for } i = j \\ -1 & \text{for } i = j+1 \\ 0 \quad \text{otherwise} \end{cases} ,$$

where $\sim$ denotes equivalence for matrices with equal determinants. Now, starting with column $|\Psi| - 1$ and working left to column 1, to each column j add column $j+1$. As a result, for $i \geq 2$, $B_{ij} = \delta_{ij}$, and the entries in row 1 are

$$B_{1j} \sim \begin{cases} 1 - \frac{kT}{p} \sum_{k\in\Psi} Q_k\, e^{\lambda^T A_k} & \text{for } j = 1 \\ -\frac{kT}{p} \sum_{k=j}^{|\Psi|} Q_k\, e^{\lambda^T A_k} & \text{for } j > 1 \end{cases} .$$

Thus, the determinant of B is given by

$$\det(B) = 1 - \frac{kT}{p} \sum_{j\in\Psi} Q_j\, e^{\lambda^T A_j},$$

meaning $\inf_m \mathcal{L}(m, \lambda)$ is defined for λ such that

$$1 - \frac{kT}{p} \sum_{j\in\Psi} Q_j\, e^{\lambda^T A_j} = 0. \tag{2.30}$$

Using this result and (2.29), the dual function is

$$h(\lambda) = \lambda^T m^0 \quad \text{such that (2.30) holds.}$$

Because the dual function is only defined for λ satisfying (2.30), we cannot apply the unconstrained dual optimization to solve the primal problem, as we could with dilute solutions. However, because strong duality holds, we can still solve the constrained dual problem, which is of lower dimension than the linearly-constrained primal problem, but has a nonlinear constraint.

$$\max_{\lambda} \;\; h(\lambda) = \lambda^T m^0$$

$$\text{subject to } \; 1 - \frac{kT}{p} \sum_{j \in \Psi} Q_j \, e^{\lambda^T A_j} = 0.$$

Nevertheless, this problem is much more difficult to solve than its dilute solution analog.

Chapter 3

Thermodynamics of interacting nucleic acids in dilute solutions

3.1 Introduction

Being ultimately motivated by the study of interacting nucleic acid strands, this chapter explores the application of the theory and techniques of Chapter 2 to dilute solutions of nucleic acids. As mentioned in Section 2.3.1, it is convenient to compute the single-compound partition functions as if the particles in the compound were distinguishable and then correct for indistinguishability using (2.7). The essential task to apply these techniques is to use the prevailing energy model to identify what comprises a particle and what comprises a compound, and what group $\mathcal{G}_j$ is associated with each compound. As we will see, the "particles" of Chapter 2 are strands, the "compounds" are *ordered complexes* (defined in Section 3.3), and the "supercompounds" of Section 2.4.5 are *complexes* (defined in Section 3.4.4).

This chapter proceeds as follows. First, we give some background on the energy model describing single strands. This model is then extended to ordered complexes of multiple strands. We then discuss how we can deduce the appropriate group $\mathcal{G}_j$ for a given ordered complex j. Knowing $\mathcal{G}_j$, we can use existing algorithms to compute $\overline{Q}_j(T)$ and then use (2.7) to get the partition function of each ordered complex. With the partition functions in hand, we can compute equilibrium concentrations. We then describe how the results of these calculations can be used to compute base-pairing observables. Finally, the application of the results of Section 2.3.3 to stochastic simulations of hybridization of interacting strands is discussed. We withhold sample calculations until Chapter 4.

As was the case for Chapter 2, much of this work is published in [17].

3.2 Single-stranded secondary structure energy models and algorithms

The *secondary structure* of a nucleic acid strand in a particular physical conformation is simply the set of base pairs present in the molecule (with the convention that $i \cdot j$ denotes that base i is paired to base j). In

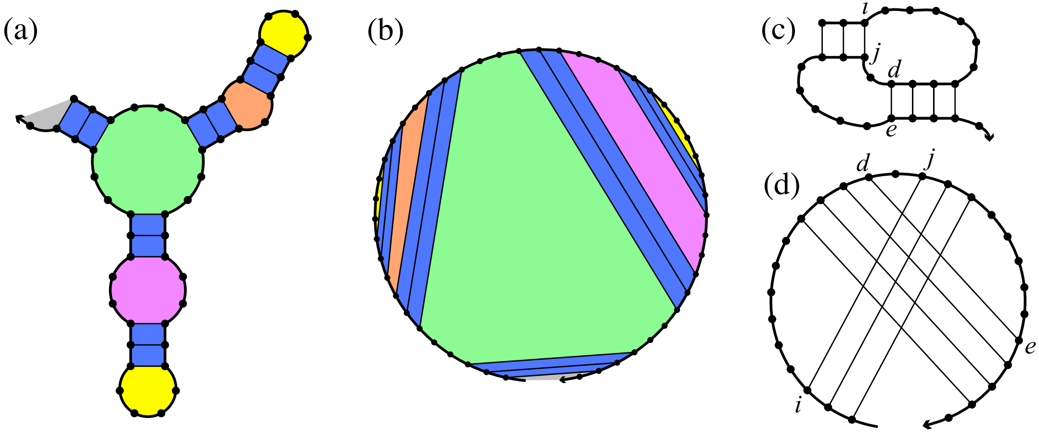


Figure 3.1: Secondary structure model for a single nucleic acid strand. (a) A sample secondary structure with the backbone depicted as a directed thick line (an arrow marks the $3'$ end), bases as dots, and base pairs as thin lines joining complementary bases. This structure can be decomposed into canonical loop types [34, 40]: hairpin loops (a stretch of unpaired bases closed by one base pair; yellow), stacked base pairs (two consecutive base pairs with no unpaired bases between them; blue), an interior loop (two base pairs separated by unpaired bases on both sides of the loop; purple), a bulge loop (two base pairs separated by unpaired bases on only one side of the loop; orange), a multiloop (three or more base pairs; green), and an exterior loop (the loop containing the two ends of the strand; gray). (b) An equivalent polymer graph representation, with the strand depicted as a directed thick circular arc, bases as dots, base pairs as straight lines joining complementary bases, and loops colored as in (a). (c) A sample pseudoknot with base pairs $i{\cdot}j$ and $d{\cdot}e$ (with $i < d$) that fail to satisfy the nesting property $i < d < e < j$, yielding crossing lines in the corresponding polymer graph (d).

general, each sequence is compatible with multiple secondary structures. Figure 3.1a depicts a secondary structure in which some bases are paired and others are unpaired, and illustrates the decomposition of this secondary structure into different *loop* types. Each secondary structure is compatible with an ensemble of *tertiary structures* corresponding to the three-dimensional atomic coordinates of the strand. Remarkably, empirical potential functions based on secondary structure alone [34–36] have great utility for studying the properties of natural and engineered RNA and DNA structures [1, 36–39]. For a given sequence, the free energy of secondary structure s is estimated as the sum of the empirically determined free energies[1] of the constituent loops [34–36]

$$G(s) = \sum_{\text{loop}\in s} G(\text{loop}), \tag{3.1}$$

each defined with respect to the free energy of the unpaired reference state.

The loop-based secondary structure models have enabled the development of efficient dynamic programming algorithms for characterizing the equilibrium properties of a DNA or RNA molecule. For algorithmic purposes, it is convenient to represent a secondary structure as a *polymer graph*, with the strand drawn along the circumference of a circle and base pairs depicted as a straight lines joining complementary bases (Figure 3.1b). The class of secondary structures that are considered in dynamic programs is usually defined to

[1] These energies are reported as standard state free energies G° corresponding to 1 mol/liter NaCl and 37°C.

exclude *pseudoknots* (Figure 3.1c), which correspond to polymer graphs with crossing lines (Figure 3.1d).

The exclusion of pseudoknots is founded on both modeling and algorithmic considerations. Energy models for pseudoknots are difficult to formulate due to the increased significance of geometric issues and tertiary interactions. Furthermore, if the ensemble of allowed secondary structures is augmented to include all possible pseudoknots, determination of the minimum free energy (MFE) structure can be NP-hard [41, 42]. Consideration of restricted classes of pseudoknots enables the specification of polynomial-time MFE determination [42, 43] and partition function [12, 38] algorithms. Although pseudoknots exist in nature [44] and have been incorporated in synthetic DNA systems [45, 46], many natural and synthetic structures of interest do not include pseudoknots [5, 6, 36], and we will not consider them here.

The dynamic programs enable calculation of the minimum free energy secondary structure [40, 47, 48] and the partition function [8] over the ensemble of unpseudoknotted secondary structures Ω, the size of which grows exponentially with strand length N. They require $\mathcal{O}(N^4)$ time and $\mathcal{O}(N^2)$ storage to do so, though the time complexity may be improved to $\mathcal{O}(N^3)$ [12, 49]. For a description of the algorithms, see the references in this paragraph.

The partition function for a single strand,

$$Q = \sum_{s \in \Omega} e^{-G(s)/kT},$$

can be used to calculate the equilibrium probability of a given secondary structure s by

$$p(s) = \frac{1}{Q} e^{-G(s)/kT}, \tag{3.2}$$

and therefore has profound implications for the development of rigorous sequence design methods [16]. Adaptations of the partition function algorithm allow the calculation of other important equilibrium properties, including the probability of any base pair [8], thermodynamically representative samplings of secondary structures in the ensemble Ω [50], and the average number of incorrectly paired bases relative to a design target [16]. These tools are useful in practice for the analysis and design of functional nucleic acid systems [1, 50–53].

3.3 Multistranded secondary structure energy models and algorithms

Our goal is to simply extend the single-stranded algorithms to collections of multiple strands. We first need to identify what constitutes a particle and a compound for nucleic acid systems. Since a strand is the largest irreducible unit in a system of interacting strands, it obviously fits the definition of a "particle" as defined in Chapter 2. As before, strands of different types are distinguishable from each other. Two strands of differing sequences are obviously of different types. Strands with the same sequence are of different types if one has a

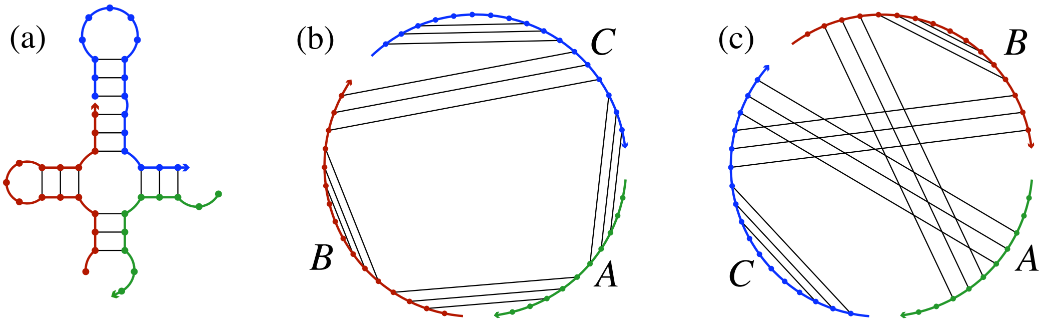


Figure 3.2: Multistranded secondary structure model. (a) A connected unpseudoknotted secondary structure for a complex of three distinct strands with sequences A, B, and C. The set of distinct circular permutations is $\Pi = \{ABC, ACB\}$. (b) Polymer graph representation of the secondary structure with no crossing lines corresponding to $\pi = ABC$. Loop classifications are the same as for the single-stranded case (Figure 3.1a and b). (c) Alternative polymer graph with crossing lines corresponding to $\pi = ACB$.

fluorescent label, for example, or other distinguishing feature. With this definition of a particle in mind, we proceed to define the multistranded energy model and use it to define a "compound."

To each of L interacting strands, which may be of different types, we assign each a unique identifier in $\{1, \ldots, L\}$. As for the single-stranded case, the secondary structure of multiple interacting strands is defined by a list of base pairs, where here each base is specified by a strand identifier and a position on that strand. For example, $i_n \cdot j_m$ denotes base i of strand n pairing with base j of strand m.[2]

A polymer graph for a secondary structure can be constructed by ordering the strands and drawing them in succession from 5′ to 3′ around the circumference of a circle with a *nick* between each strand and straight lines connecting paired bases (Figure 3.2). The distinct ways to order the L labeled strands on a circle correspond to the set $\overline{\Pi}$ of *circular permutations* containing $(L-1)!$ *orderings* or strand identifiers (e.g., $\overline{\Pi} = \{123, 132\}$ for a complex of three strands). The set $\Pi \subseteq \overline{\Pi}$ is a maximal subset of distinct circular permutations with respect to strand type. Consequently, Π may contain fewer than $(L-1)!$ members (e.g., $\Pi = \{AAB\}$ for a complex of three strands, two of type A and one of type B). *Cyclic permutations* of strand identifiers for a particular ordering $\pi \in \Pi$ change the strand identifiers such that their relative ordering on the circle is preserved. For example, for $\pi = AAA$, the orderings of strand identifiers 123, 231, and 312 correspond to the same circular permutation, as do 132, 321, and 213.

For a given secondary structure, if every circular permutation $\pi \in \Pi$ corresponds to a polymer graph with crossing lines, then the secondary structure is *pseudoknotted*. A polymer graph with no crossing lines can be decomposed into *loops* as for the single-stranded case, and all loops containing one nick are *exterior loops*. A secondary structure is *connected* if no loop contains more than one nick (i.e., no subset of the strands is free of the others), in which case the L strands constitute an *ordered complex*. The *Representation Theorem* (the proof is shown in Appendix B) states that for every unpseudoknotted connected secondary structure s, there is exactly one circular permutation $\pi \in \overline{\Pi}$ that yields a polymer graph with no crossing lines. It follows

[2] The expressions $i_n \cdot j_m$ and $j_m \cdot i_n$ denote the same base pair. For convenience, we adopt the convention that the bases in a pair are ordered first by strand identifier and then by position on the strand.

that since $\Pi \subseteq \overline{\Pi}$, there is exactly one $\pi \in \Pi$ for which a secondary structure s has no crossing lines. The result of this theorem is that each ordered complex is independent from all others and therefore constitutes a compound, as defined in Chapter 2.

For a secondary structure s for which the strands in the polymer graph are labeled with identifiers, the free energy $\overline{G}(s)$, is the sum of the free energies of the constituent loops plus a strand association penalty G^{assoc} [54] applied $L-1$ times for a complex of L strands:

$$\overline{G}(s) = (L-1)\, G^{\text{assoc}} + \sum_{\text{loop} \in s} G(\text{loop}). \tag{3.3}$$

As evident by this equation, all cyclic permutations of strand identifiers on a polymer graph with no crossing lines are equivalent, having identical loop decompositions and identical free energies. This fact defines the relevant permutation group $\mathcal{G}_j$ of ordered complex j as the cyclic group of maximal size, $C_{v(\pi)}$. For example, if $\pi = AAAA$, the possible cyclic groups are of size 1, 2, and 4 (with the former two being subgroups of the latter), so $v(\pi) = 4$. In this case, the partition of the set of strand identifier arrangements under the action of the cyclic group C_4 is described by the white boxes of Figure 2.2. As another example, consider the circular permutation $\pi = ABAB$, where identifiers 1 and 2 label A strands and 3 and 4 label B strands, the cyclic groups are of size 1 and 2, and the partition of the set of identifier arrangements under C_2 is $\{1324, 2413\}$ and $\{1423, 2314\}$.

3.4 Application of the methods of Chapter 2 to nucleic acid systems

We now have everything in place to apply the methods of Chapter 2 to systems of interacting nucleic acids. The problem is as follows: given total concentrations $x^0 \in \mathbb{R}_{>0}^{|\Psi^0|}$ of the strand types in the set Ψ^0, what are the equilibrium concentrations $x \in \mathbb{R}_{>0}^{|\Psi|}$ of the resulting ordered complexes?

3.4.1 Definition of the set Ψ

The first step is to identify the ordered complexes to consider in the calculation, i.e., to determine the members of the set Ψ.[3] We may choose to include only ordered complexes that are expected to be physically significant, e.g., choose a maximum cutoff size $L_{\max}$, as described in Section 2.4.5. In this case, the set Π_{A_k} must be specified by finding all circular permutations for stoichiometry A_k for all allowed stoichiometries, which is generally difficult for large complexes with differing strand types. The size of Π_{A_k} is equal to the number of distinct fixed necklaces[4] containing A_{ik} beads of the i-th type for a total of $L_k = \sum_{i \in \Psi^0} A_{ik}$ beads.

[3] Note that this definition of Ψ is consistent with that of Chapter 2 in that it is the set of all compounds, which for the system of interacting nucleic acid strands is the set of all *ordered* complexes. When we collapse the problem of concentration determination to include only supercompounds, we get a set Ψ^{super} that is equal to the set Ψ in [17].

[4] Two *fixed* necklaces are indistinguishable if they can be transformed to each other by rotation (cyclic permutation), as opposed to *free* necklaces, which are indistinguishable from each other if they can by transformed by rotation or flipping (dihedral permutation).

Application of the Pólya enumeration theorem gives [26]

$$|\Pi_{A_k}| = \frac{1}{L_k} \sum_{d|\Delta} \phi(d) \, \frac{(L_k/d)!}{\prod_{i=1}^{|\Psi^0|}(A_{ik}/d)!},$$

where Δ is the greatest common divisor of the nonzero entries in A_k, $\phi(d)$ is Euler's totient function, and $\sum_{d|\Delta}$ denotes the sum over all the positive divisors of Δ. While $|\Pi_{A_k}|$ is known, listing the elements of Π_{A_k} is nontrivial, and we have yet to develop an efficient algorithm for doing so.

Using a brute-force algorithm, we can list the contents of the sets Π_{A_k}, from which the set Ψ is determined, as the sets Π_{A_k} constitute a partition of Ψ.

3.4.2 Ordered complex partition functions

The partition function of a particular cyclic permutation of an ordered complex in which all strands are labeled with unique identifiers is

$$\overline{Q}_j = \sum_{s \in \overline{\Omega}} e^{-\overline{G}(s)/kT},$$

where $\overline{\Omega}$ is the set of all connected unpseudoknotted secondary structures and $\overline{G}(s)$ is given by (3.3). We can compute it by extending the single-stranded dynamic programming technique to multiple strands. If strand l has length N_l, then the algorithm operates on a single concatenated strand of length $N \equiv \sum_{l=1}^{L} N_l$. The details of this procedure are outlined in [17]. Like the single-stranded algorithm, the recursions require $\mathcal{O}(N^4)$ time and $\mathcal{O}(N^2)$ space in their most transparent form, but the time complexity can be reduced to $\mathcal{O}(N^3)$ using standard methods [12, 49]. By (2.7), $Q_j(s) = \overline{Q}_j(s)/|\mathcal{G}_j|$, where $\mathcal{G}_j = C_{v(\pi)}$. Thus, $Q_j = \overline{Q}_j/v(\pi)$.

This is the same result derived by Dirks in [17]. Indeed, the development in Section 2.3.1 is a more general statement of that result. The fundamental difference is that we take the group $\mathcal{G}_j$ acting on the set Y_j, whereas Dirks takes it to act on a secondary structure s. In Dirks's treatment, corrections to the free energy of a secondary structure due to symmetry and those due to algorithmic overcounting are separated in this specific example. This is conceptually pleasing in the context of algorithm development and evaluation of base pair probabilities (see Section 3.5 and comments in [17]), but does not expose the essential feature that gives rise to (2.3) and (2.10)—that there is a fixed number of states in a system with all particles being distinguishable corresponding to a given state where particles of like type are indistinguishable—though it is a consequence.

The total number of partition function evaluations for the example with size cutoff $L_{\max}$ is [26]

$$|\Psi| = \sum_{L=1}^{L_{\max}} \sum_{d|L} \phi(L/d) \, \frac{|\Psi^0|^d}{L} = \sum_{L=1}^{L_{\max}} \sum_{l=1}^{L} \frac{|\Psi^0|^{\gcd(l,L)}}{L},$$

which is the number of distinct fixed necklaces with size $1 \leq L \leq L_{\max}$ that can be made from $|\Psi^0|$ types of beads. Thus, the time complexity for calculating the partition functions for all ordered complexes in Ψ is $\mathcal{O}(|\Psi^0|^{L_{\max}} N_{\max}^3 / L_{\max})$, where $N_{\max}$ is the largest number of bases in an ordered complex.

3.4.3 Diluteness of nucleic acid solutions

The techniques we have developed are only valid for dilute solutions, so we need to find the concentration regime for which a nucleic acid solution is dilute. Polymer solutions are dilute at concentrations below the overlap concentration ρ^*, which corresponds to the concentration at which the polymers in solution start to penetrate each other and therefore interactions among them cannot be ignored. It is defined by [55]

$$\rho^* \, \frac{4}{3} \pi R_g^3 \approx 1,$$

where ρ^* is a per-volume number density and R_g is the radius of gyration of the polymer chains.

In general, the radius of gyration for an ordered complex is very sequence dependent. De Gennes [56] estimated R_g for a poly-(AT) ssDNA strand neglecting excluded volume interactions, and Müller [57] estimated it for a random sequence with excluded volume interactions. In both cases, the estimated R_g is similar to those of typical branched polymers, which are slightly smaller than those for unbranched polymers [58]. Thus, in order to underestimate ρ^*, we consider only linear polymers of length $\mathcal{N}$, where $\mathcal{N} = N/2$ for a purely double-stranded ordered complex and $\mathcal{N} = N$ for a purely single-stranded ordered complex.

We choose two different models to estimate R_g, one which explicitly takes into account rigidity and one that considers excluded volume. First, we take the polymers to be phantom wormlike chains (WLC). In this case, the radius of gyration is [59]

$$R_g^{\mathrm{WLC}} = \left(\frac{\xi_p}{3} \mathcal{N} b_0 - \xi_p^2 + 2 \, \frac{\xi_p^3}{\mathcal{N} b_0} - 2 \, \frac{\xi_p^4}{(\mathcal{N} b_0)^2} \left(1 - e^{-\mathcal{N} b_0 / \xi_p} \right) \right)^{\frac{1}{2}},$$

where b_0 is the stack height and ξ_p is the persistence length. For the second model, we consider a complex to be a self-avoiding random walk (SAW) with step length equal to the Kuhn statistical length, equal to twice the persistence length [55]. Here, the radius of gyration is

$$R_g^{SAW} = 0.406 \left(\frac{\mathcal{N} b_0}{2 \xi_p} \right)^{0.588} (2 \xi_p),$$

where the prefactor and exponent come from renormalization group calculations of SAWs in 3D [55].

Using DNA as an example, the stack height and persistence length for ssDNA are 0.63 nm and 1.5 nm, respectively [60], and those for dsDNA are 0.34 nm and 150 nm [61]. Figure 3.3 shows the overlap concentration for solutions of single-stranded and double-stranded DNA for each of the two models containing only ordered complexes with N bases. We see that solutions with micromolar concentrations of ordered

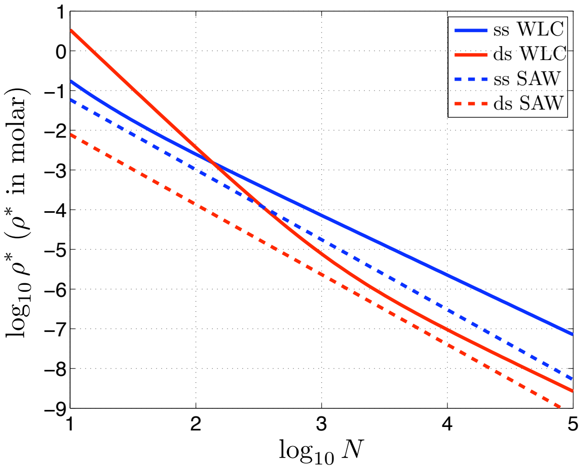


Figure 3.3: The overlap concentration ρ^* for DNA solutions versus the number of bases in the strands. The blue curves are for purely unpaired strands and the red for purely double-stranded ordered complexes. The broken curves are for the SAW model and the solid curves are for the WLC model.

complexes with $N \lesssim 1000$ can safely considered to be dilute. These are typical experimental conditions for oligonucleotides in solution.

3.4.4 Concentration and distribution determination

With partition functions in hand, we have only to apply the techniques of Section 2.4.2 to get the equilibrium concentrations of all the complexes in Ψ for a dilute solution. We first construct the set Ψ^{super}, and we call each member a *complex*. Complex k consists of all ordered complexes with stoichiometry A_k, i.e, the members of the set Π_{A_k}. For the case where the set Ψ contains all ordered complexes up to size $L_{\max}$, the total number of complexes is given by (2.25). We then apply the techniques of Section 2.4.3 to get the equilibrium concentrations of the complexes, x_Π. Application of (2.24) recovers the concentrations of ordered complexes x from x_Π.

Recall that the time complexity of the concentration determination is $\mathcal{O}(|\Psi^{\text{super}}|\,|\Psi^0|^2)$. Considering that we typically have $N_{\max} \gg |\Psi| \geq |\Psi^{\text{super}}| \geq |\Psi^0|)$, we see that concentration determination is much, much faster than computing the partition functions.

As a final note, the equilibrium distributions of populations of ordered complexes in a small box with M_s solvent molecules and m_i^0 strands of type $i \;\forall\; i \in \Psi^0$ may be calculated using (2.11) through (2.15). The difficult step in this calculation is constructing the set Λ. Pseudocode for an efficient method for doing so is in Appendix C.

3.4.5 Free energy landscapes of interacting nucleic acid strands

While we have chiefly exploited the mathematical properties of convexity, it is also interesting to consider its physical significance. The equilibrium and kinetic properties of a nucleic acid system are determined by the features of the underlying free energy landscape [62, 63]. A free energy landscape based on nucleic acid secondary structure may be represented as a graph with each vertex corresponding to a different state of the system (i.e, the pairing status of every base in the system) and each edge corresponding to an elementary step between states (e.g., formation, breakage, or shifting of a single base pair [10]). States that are likely at equilibrium are represented by deep basins in the landscape, and the rate of conversion between two different states is dependent on the nature of the basins and passes that separate them. No underlying convexity is evident in this discrete free energy landscape. Now suppose we coarse-grain the state space so that each state corresponds to a different set of complexes (with the fine-grained base-pairing information captured by the ensemble and partition function of each complex). In the thermodynamic limit of large species populations, the free energy landscape may be treated as continuous in the complex concentrations, in which case it becomes strictly convex.

3.5 Base-pairing observables

We have already shown how to calculate important experimental observables in the form of equilibrium population distributions for small systems and equilibrium concentrations for large systems. Here, we describe the calculation of more detailed base-pairing information about the ensemble of states in a given system.

For an ordered complex of L strands labeled with unique identifiers, the equilibrium probability of each intrastrand and interstrand base pair can be calculated by backtracking through the partition function algorithm, applying a particular algorithmic transformation at each step (see [8, 17, 38] for details). We denote the equilibrium probability of base pair $i_n \cdot j_m$ in ordered complex k (where the strands are distinguishable) as $p_k(i_n \cdot j_m)$, which is the output of the backtracking recursions.

If an ordered complex contains some indistinguishable strands, when we examine the probabilities of individual base pairs, new distinguishability issues arise. For example, consider an ordered complex involving two indistinguishable copies of strand A (with identifiers 1 and 2) and one copy of strand B (with identifier 3). Base pairs $i_1 \cdot j_3$ and $i_2 \cdot j_3$ are indistinguishable since strands 1 and 2 are both of type A. Likewise, without the global structural context, we cannot distinguish between the inter- and intrastrand base pairs $i_1 \cdot j_2$ and $i_1 \cdot j_1$.

We now develop a quantity analogous to base pair probabilities that appropriately treats the indistinguishability of strands in a complex. Let Θ be the set of strand types in the ordered complex and $\{\theta\}$ be the set of all strand identifiers corresponding to strands of type $\theta \in \Theta$ (hence $L = \sum_{\theta \in \Theta} |\{\theta\}|$).

We define the expected number of base pairs between base i on strands of type $A \in \Theta$ and base j on strands of type $B \in \Theta$ in ordered complex k to be $E_k(i_{\{A\}} \cdot j_{\{B\}}) \in [0, \min(|\{A\}|, |\{B\}|)]$. Then,

$$E_k(i_{\{A\}} \cdot j_{\{B\}}) = \sum_{l_A \in \{A\}} \sum_{l_B \in \{B\}} p_k(i_{l_A} \cdot j_{l_B})$$

represents a sum over the contributions of each type of distinct base pair, where each term $p_k(i_{l_A} \cdot j_{l_B})$ is an output of a backtracking recursion. This result can be used to calculate experimental observables for a dilute solution of complexes at equilibrium. For a mixture of strands at equilibrium, the expected concentration of base pairs between base i of strands of type A and base j of strands of type B is

$$\langle x(i_A \cdot j_B) \rangle = \sum_{k \in \Psi} E_k(i_{\{A\}} \cdot j_{\{B\}}) \, \langle x_k \rangle \, .$$

For experimental studies, it is usually more convenient to measure the expected fraction of A strands or B strands that form this base pair: $f_A(i_A \cdot j_B) = \langle x(i_A \cdot j_B) \rangle / x_A^0$ and $f_B(i_A \cdot j_B) = \langle x(i_A \cdot j_B) \rangle / x_B^0$, respectively. Similarly, the expected concentration $\langle x(i_A) \rangle$ of strand species $A \in \Psi^0$ with base i paired to any other base is

$$\langle x(i_A) \rangle = \sum_{B \in \Psi^0} \sum_{j=1}^{N_B} \langle x(i_A \cdot j_B) \rangle \, ,$$

and the expected fraction of A strands that have base i paired is $f_A(i_A) = \langle x(i_A) \rangle / x_A^0$.

3.6 Stochastic simulation of nucleic acid hybridization kinetics

Stochastic simulation of nucleic acid hybridization kinetics [10] employ Gillespie-type algorithms [64, 65]. In these simulations, the free energy landscape is typically coarsened such that each point on the landscape corresponds to a given secondary structure, with each structure having a set of neighbors differing by the breakage or formation of a single base pair. A trajectory in the simulation is generated by choosing a starting secondary structure (often a completely unpaired state, as would be seen, e.g., in T-jump experiments) and successively moving to a neighboring structure. The probability of moving from structure i to a structure j in the set of neighbors is proportional to k_{ji}. The choice of k_{ji} should be as close to the physical conditions as possible, but in practice is somewhat arbitrary [10, 66]. Regardless of how they are chosen, the rate constants must satisfy detailed balance [64], or

$$\frac{k_{ji}}{k_{ij}} = \exp\left\{ -\frac{G(j) - G(i)}{kT} \right\}, \tag{3.4}$$

where $G(i)$ and $G(j)$ are the free energy of structures i and j, respectively, given by (3.1) for the single-stranded case. Satisfaction of detailed balance ensures that long-time averages of a trajectory (or long-time averages of many trajectories) give the equilibrium distribution of secondary structures, as given by (3.2).

To date, these simulations have been limited to single-stranded folding. Joseph Schaeffer and Erik Winfree are developing algorithms for stochastic simulation of multiple interacting strands in a small box [67]. Unlike the single-stranded simulation, where the entire system contains only one strand and therefore a state is defined by the single secondary structure, a state in the multistranded system is defined by all secondary structures present in the box. Furthermore, the free energy of the system requires corrections for the indistinguishability of like strands. This system corresponds to “system 3” of Section 2.3.3 since every strand in the simulation is distinguishable from any other because they occupy unique addresses in the computer’s memory. The distinguishability correction in this case is given by (2.10). Because $n'(m)$ is independent of which secondary structures are present, the same distinguishability correction is applied to each ordered complex in the simulation.

It is useful to redefine the reference state for the stochastic simulation such that

$$Q'_{\text{box}} \equiv Q'_{\text{ref}} \sum_{m \in \Lambda} Z'(m, M_s, T) = Q_{\text{box}}, \tag{3.5}$$

where Q_{box} is given by (2.12) with $Q_{\text{ref}} = \prod\limits_{i \in \Psi^0} (m_i^0!/M_s^{m_i^0})$. Enforcing (3.5) gives

$$Q'_{\text{ref}} = M_s^{-M^0}.$$

For convenience, we can adjust the energy model to include Q'_{ref}. If $G(s)$ is the free energy associated with secondary structure s for ordered complex j, we define an adjusted free energy

$$G^\dagger(s) = G'(s) + (L_j - 1)kT \, \log M_s,$$

where $G'(s)$ is the free energy of secondary structure s if all strands are distinguishable. With this newly defined energy, the alternative partition function for population m is then

$$\begin{aligned} Z^\dagger(m, M_s, T) &= \prod_{j \in \Psi} \frac{1}{m_j!} \left(\sum_{s \in \Omega_j} \exp\left\{ -(L_j - 1)\log M_s - \frac{G'(s)}{kT} \right\} \right)^{m_j} \\ &= \prod_{j \in \Psi} \frac{M_s^{m_j(1-L_j)} \, (Q'_j)^{m_j}}{m_j!} = M_s^{-\sum_{j \in \Psi} m_j L_j} \, Z'(m, M_s, T) = M_s^{-M^0} \, Z'(m, M_s, T) \end{aligned}$$

since $\sum_{j \in \Psi} m_j L_j = M^0$. Thus,

$$Q_{\text{box}} = \sum_{m \in \Lambda} Z^\dagger(m, M_s, T).$$

With this adjusted energy model, the free energy of the box for a given point in the stochastic simulation is simply

$$\sum_{s \in \text{box}} G^\dagger(s). \tag{3.6}$$

Because we only adjusted the reference state, use of (3.6) for the free energy of a state in the multistranded stochastic simulations gives the correct equilibrium distribution corresponding to the physical case where strands of like sequence are indistinguishable. Thus, the analogue of (3.4) for the multistranded case where the transition is from state a of the box to state b is

$$\frac{k_{ba}}{k_{ab}} = \exp\left\{-\frac{\sum_{s\in b} G(s) - \sum_{s\in a} G(s)}{kT}\right\}.$$

Chapter 4

NUPACK: a web-based tool for automated analysis of nucleic acids

The algorithms described in Chapter 3 provide powerful new tools for analysis of interacting nucleic acids for a wide range of applications in biology and biotechnology. To date, web-based packages such as mfold [68] and ViennaRNA [69] have seen wide use worldwide for folding single nucleic acid strands. Taking this as evidence that the scientific community working with nucleic acids has a high demand for web-based analysis applications, we developed NUPACK (Nucleic Acid Package), a web-based software package for thermodynamic analysis at the level of nucleic acid secondary structure for multiple interacting strands (without pseudoknots) [17] or a single strand (with or without a class of pseudoknots) [12, 13]. The structure of NUPACK is such that additional modules, such as sequence design and kinetics, may be conveniently added, both to the compute engine and to the web application. NUPACK's URL is `http://www.nupack.org/`.

4.1 NUPACK compute engine code base

The NUPACK compute engine is written entirely in the C programming language and consists of various utilities for calculation of partition functions, pair probabilities, minimum free energy structures, and equilibrium concentration determination. The current compute engine is version 2.0, which is almost entirely different from earlier versions. Existing code for partition function, pair probability, and MFE structure calculations for single strands, including a subset of physically relevant pseudoknots was edited, reorganized, and integrated with new code for calculations involving multiple interacting strands.

These utilities are bundled into convenient executables which are run through the command line. The source code can be downloaded from `nupack.org` and easily compiled. The NUPACK User Guide, also available from `nupack.org` gives complete descriptions of the capabilities of the NUPACK compute engine. A portion of its contents appears in Appendix D, and gives an indication of the capabilities of the compute engine.

The code base is organized such that shared files, such as energy parameters, and utility code (such as loop energy calculations) can be easily incorporated into new entries in the code base. This allows facile incorporation of new modules into NUPACK by present and future developers, with design and kinetics modules already in development.

4.2 NUPACK web application

The NUPACK web application launched on January 30, 2007, coinciding with online publication of [17]. The purpose of the web application is to enable rapid analysis of interacting strands without the need for scripting or direct interaction with the compute engine executables. Such a tool is essential for enabling a diverse group of biologists and biotechnologists easy access to the computational tools developed in Chapter 3 and elsewhere in the Pierce Lab (and in the Winfree Lab for future releases). In this chapter, I present the essential features of the web application, found at `nupack.org`, and comment on the reasoning behind its structure.

4.2.1 General navigational objectives

The central objective in designing the structure of the web application is to ask the user only for the most relevant information about the nucleic acid sequences of interest and present the results in a clean, concise, and unambiguous way. The input of sequences and concentrations is facile and not cluttered by computational details of less interest to most users, such as particulars of energy parameters. After a calculation is completed, the more essential information (such a base pair probabilities) is presented before less informative information (such as MFE structures and free energies of ordered complexes). By carefully constructing the interface, the user is naturally directed through the website to the features he or she desires without having to read excess text or other instructions. Nevertheless, help pages are available for every page, again providing exactly the necessary level of detail. These basic design objectives are pervasive throughout the website and are the motivation for all of the workings of the NUPACK web application described in this chapter.

At present, NUPACK allows for thermodynamic analysis of interacting strands. As with the code base, the web application is structured to optimize integration of new modules. A design module (built by Joseph Zadeh), for example, will launch in Spring 2007. The user can easily navigate from module to module as they are added.

4.2.2 Presentation of calculated results

Before proceeding to a demonstration of NUPACK's capabilities through a sample calculation (promised in Section 3.1), we describe the way in which data from the calculations are packaged and presented. All graphics are of publication quality and are downloadable from the website in scalable vector graphics (SVG) format matching the W3C standards. This enables editing with any of several available vector graphics editing software packages.

First, if the user wishes to compute a melt curve, we plot the fraction of unpaired bases in solution versus temperature. The fraction of unpaired bases is given by

$$f_{\text{unpaired}} = 1 - \frac{1}{x_A^0 + x_B^0 + \cdots} \left(x_A^0 \sum_{i=1}^{N_A} f_A(i_A) + x_B^0 \sum_{i=1}^{N_B} f_B(i_B) + \cdots \right),$$

where the ellipses indicate summing over the elements of Ψ^0. We plot the fraction of bases that are unpaired because this is proportional to absorbance at 260 nm, since ssDNA has greater absorbance than does dsDNA at that wavelength [70]. An example melt curve is shown in Figure 4.7 in Section 4.2.3.

The user often wishes to view MFE secondary structure of a given ordered complex. Naturally, we wish to represent the structure in an intuitive way in two dimensions. The primary difficulty arises in automatically rendering the structure such that no lines in the drawing overlap. The problem of automatically generating non-overlapping secondary structure drawings has long been studied [71–73]. While we are currently developing sophisticated drawing algorithms, at present, we do not address this problem and apply a simple algorithm that produces aesthetically pleasing structures, but may have overlap. In the cases that do overlap and for pseudoknots, the secondary structure is represented as a polymer graph.

The secondary structure drawing algorithm produces output similar to Figures 3.1a and 3.2a, with dots representing bases, thick lines representing the backbone with arrowheads at the $3'$ end, and thin lines connecting paired bases. It creates "ladders" to represent helices and circles to represent the loops of types described in Figure 3.1 (except for stacked-base loops, which are included in the helical ladders). The circles and ladders are then pieced together. The sequence is represented using the coloring scheme commonly used in sequencing gels (A = green, T/U = red, G = black, and C = blue), with each dot representing a base being of a given color. Using this technique avoids the clutter of using Roman letters, minimally expressing the sequence and structure information. Figure 4.1 shows some sample drawings of MFE structures generated by the NUPACK web application.

In addition to MFE depiction, the more physically relevant base pair probabilities for a given ordered complex k are presented in a matrix of probabilities, where entry i_n, j_m has a square whose area and color scale with $p_k(i_n \cdot j_m)$. The matrix is augmented by a column at right where the ith entry is the probability that base i is unpaired at equilibrium. Note that these are pair probabilities for a given connected ordered complex, meaning that they represent the probability that base i of strand n and base j of strand m are paired at equilibrium *given that they are in connected ordered complex k*. Therefore, there will be substantial $p_k(i_n \cdot j_m)$ even for high temperatures, since the ordered complex is defined to be connected. Figure 4.2 shows a sample probability matrix output by NUPACK for the synthetic walker in [38], the MFE of which is shown in Figure 4.1a.

A comprehensive picture of the base pairing throughout the entire solution is given by an ensemble pair fraction plot, where entry i_A, j_B has a square whose area and color scales with $f_A(i_A \cdot j_B)$. Note that

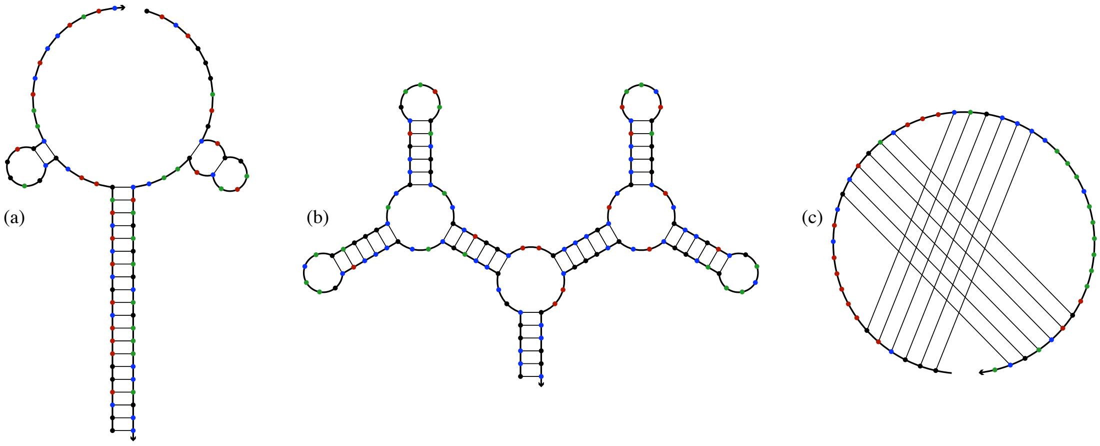


Figure 4.1: Secondary structure drawings generated by the NUPACK web application. All temperatures are 37°C. (a) MFE secondary structure drawing of the walker, comprised of two strands (W1 and W2), from [38]. (b) MFE secondary structure of one of the large RNA multiloop designs of [16]. (c) Pseudoknotted MFE secondary structure of human telomerase, described in [13].

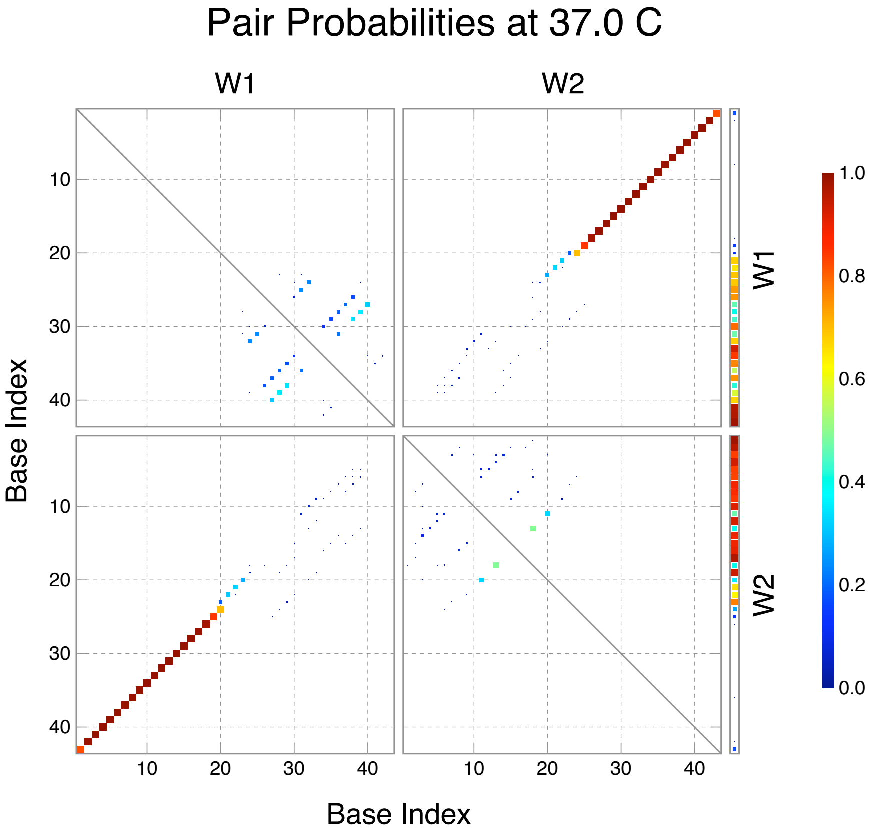


Figure 4.2: Pair probability plot for the walker from [38] at 37°C. The MFE structure is shown in Figure 4.1a. The diagonal line indicates that this pairing matrix is necessarily symmetric.

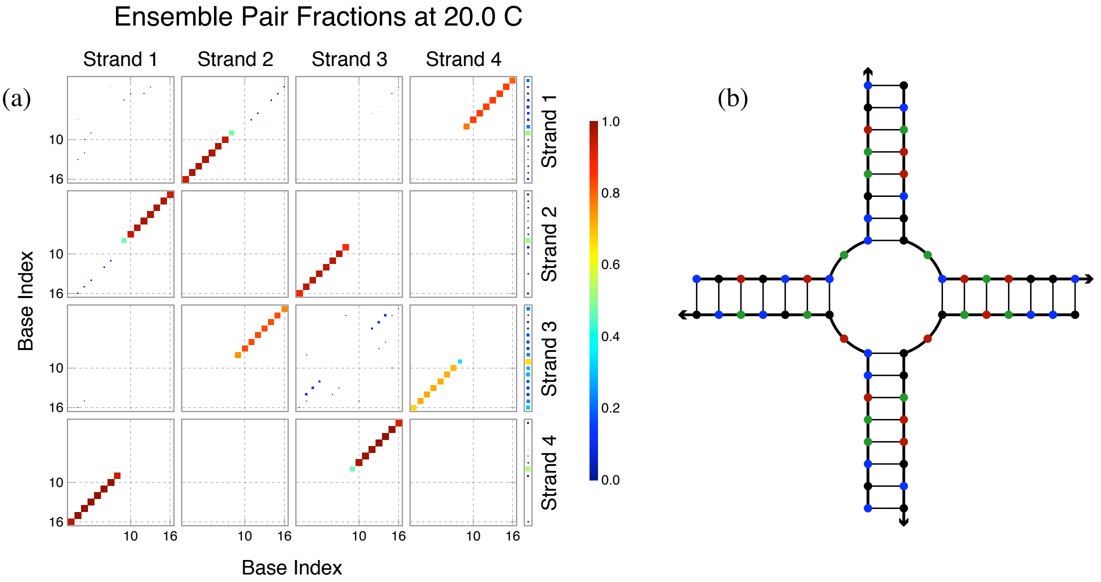


Figure 4.3: (a) Ensemble pair fraction plot of a solution containing the sequences for the immobile cruciform junction of [74] at 20°C. The concentrations of strand 1 and strand 2 are both 6.25 μM. The concentration of strand 3 is 7.25 μM and that of strand 4 is 5.25 μM. There is no diagonal line in the matrix because it is asymmetric. (b) The MFE structure of the dominant ordered complex for reference.

because x_A^0 need not equal x_B^0, the pair fraction matrix can be asymmetric. Figure 4.3 shows an ensemble pair fraction plot for a solution containing unequal concentrations of strands that tend to form a four-armed cruciform junction [74].

4.2.3 An example calculation using the NUPACK web application

To illustrate the capabilities of NUPACK and to demonstrate the utility of the techniques developed in Chapter 3, we present an example calculation. In the course of the calculation, we show screen shots directly from the NUPACK web application as rendered with Firefox 2.0, accessed March 23, 2007. The content of each page is self explanatory, but the content of the associated help pages are given in Appendix E for reference. These help pages also serve to give a detailed description of the contents of the web application.

For the sample calculation, we consider three strands, A, B, and C, in a dilute solution. The sequences and initial concentrations of the strands are shown in the screen shot of the input page (Figure 4.5). The sequences are designed such that they tend to form the ordered complex depicted in Figure 4.4a. In the absence of strand B, the ordered complex in Figure 4.4b readily forms at low enough temperatures. Thus, for $[\mathrm{A}]_0 = [\mathrm{C}]_0 = 250\ \mathrm{nm} > [\mathrm{B}]_0 = 50\ \mathrm{nM}$, we expect both ordered complexes A-B-C and A-C to be present at low temperatures.

The calculations were completed on the NUPACK webserver in approximately 14 minutes with an additional 3 minutes for graphics rendering. Figure 4.6 shows a screen shot of the main results page, which gives information about the entire solution. The melt profile (shown as a thumbnail on the results page in Figure

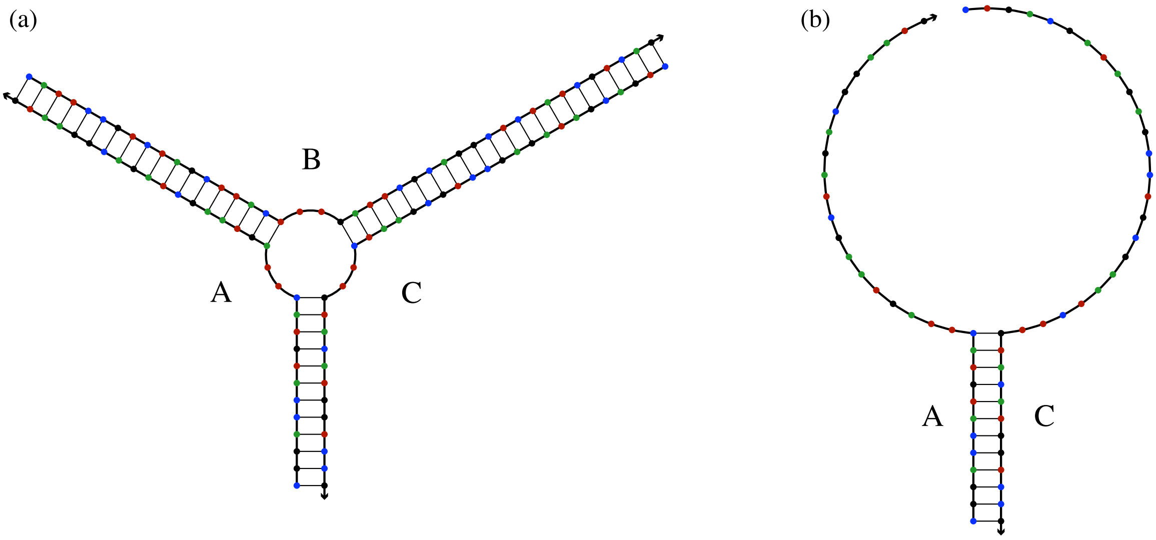


Figure 4.4: (a) Low-temperature MFE secondary structure for ordered complex A-B-C. (b) MFE secondary structure for ordered complex A-C. Strand IDs are labeled with the appropriate letters.

4.6) is shown in Figure 4.7, demonstrating two dominant melting transitions. The screen shot in Figure 4.6 is at $T = 39°\text{C}$, indicative of the state of the system before the first melt transition. The histogram of equilibrium concentrations indicates that roughly stoichiometric quantities of ordered complexes A-C and A-B-C are present at low temperatures.

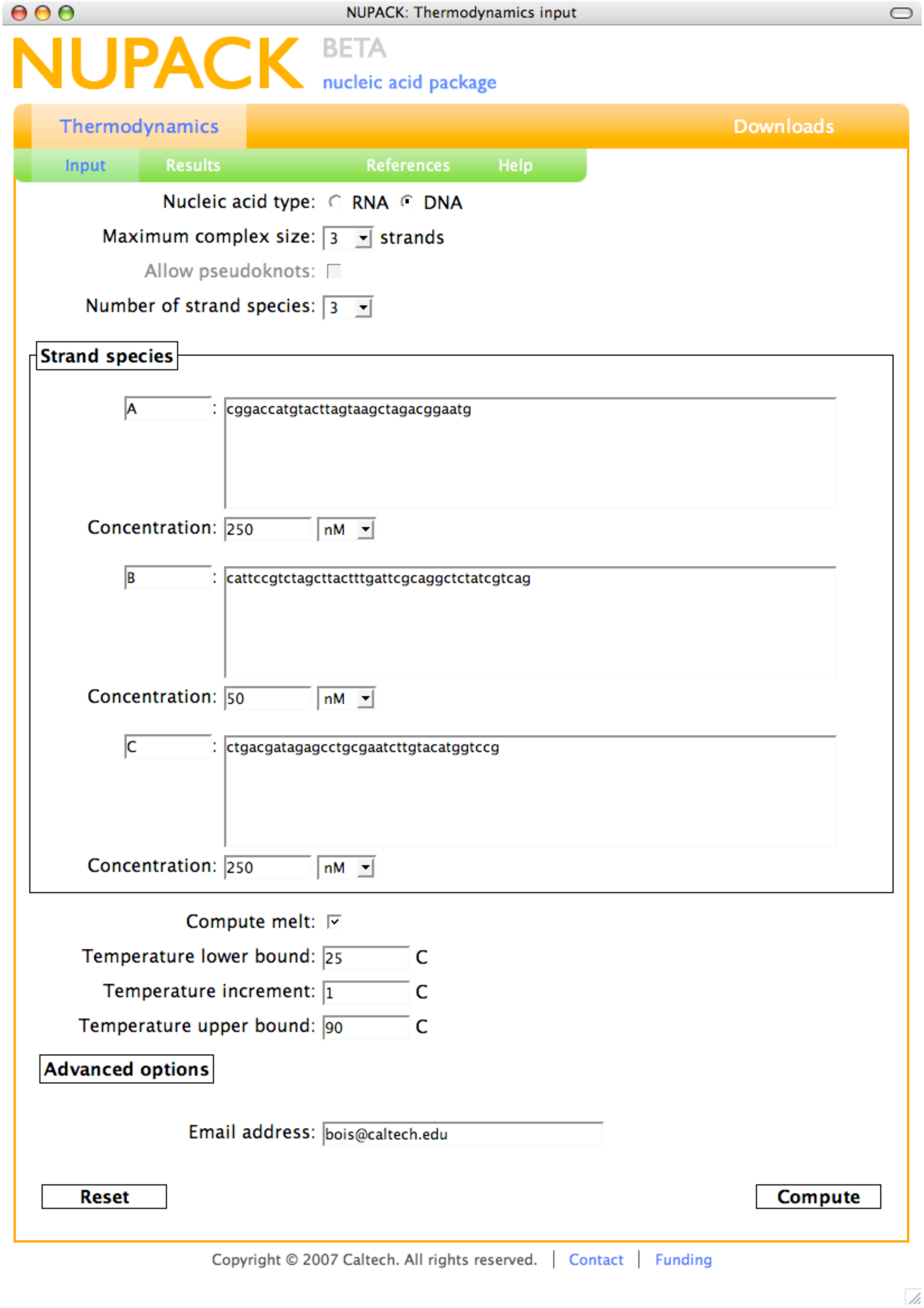
Use of the temperature slide bar allows the user to investigate which ordered complexes melt at which temperatures. The histogram of concentrations on the results page for $T = 64°\text{C}$ (Figure 4.9), indicative of the state of the system in the plateau between the two dominant melt transitions, shows that the A-B-C ordered complex is still intact while the A-C ordered complex is largely melted. The ensemble pair fraction plots (Figure 4.8) indicate that the B strands remain tightly bound to the A and C strands at this temperature.

It is more difficult to discern whether all helices are intact in the A-B-C ordered complex at $T = 64°\text{C}$. To investigate details on a given ordered complex, the user may click on its histogram bar on the results page and be brought to a details page for that specified ordered complex. Figure 4.10 shows the details page for the A-B-C complex, demonstrating that all helices are intact at $64°C$.

Careful analysis shows that the A strand separates from the A-B-C ordered complex at $T = 75°\text{C}$ (screen shot in Figure 4.11 and ensemble pair fractions in Figure 4.12), while the helix between strands B and C is still intact, accounting for the brief change in concavity in the melt profile (and a third melt transition) at that temperature. The final melt transition occurs near $T = 78°\text{C}$, corresponding to the melting of the B-C helix.

4.2.4 Comparison with experimental results

To test the accuracy of the predicted melt profile generated by NUPACK, absorbance experiments were performed on the example system. Figure 4.13 shows the calculated melt profile and an experimentally



Nucleic acid type: ○ RNA ● DNA

Maximum complex size: 3 strands

Allow pseudoknots: □

Number of strand species: 3

Strand species

A : cggaccatgtacttagtaagctagacggaatg

Concentration: 250 nM

B : cattccgtctagcttactttgattcgcaggctctatcgtcag

Concentration: 50 nM

C : ctgacgatagagcctgcgaatcttgtacatggtccg

Concentration: 250 nM

Compute melt: ☑

Temperature lower bound: 25 C

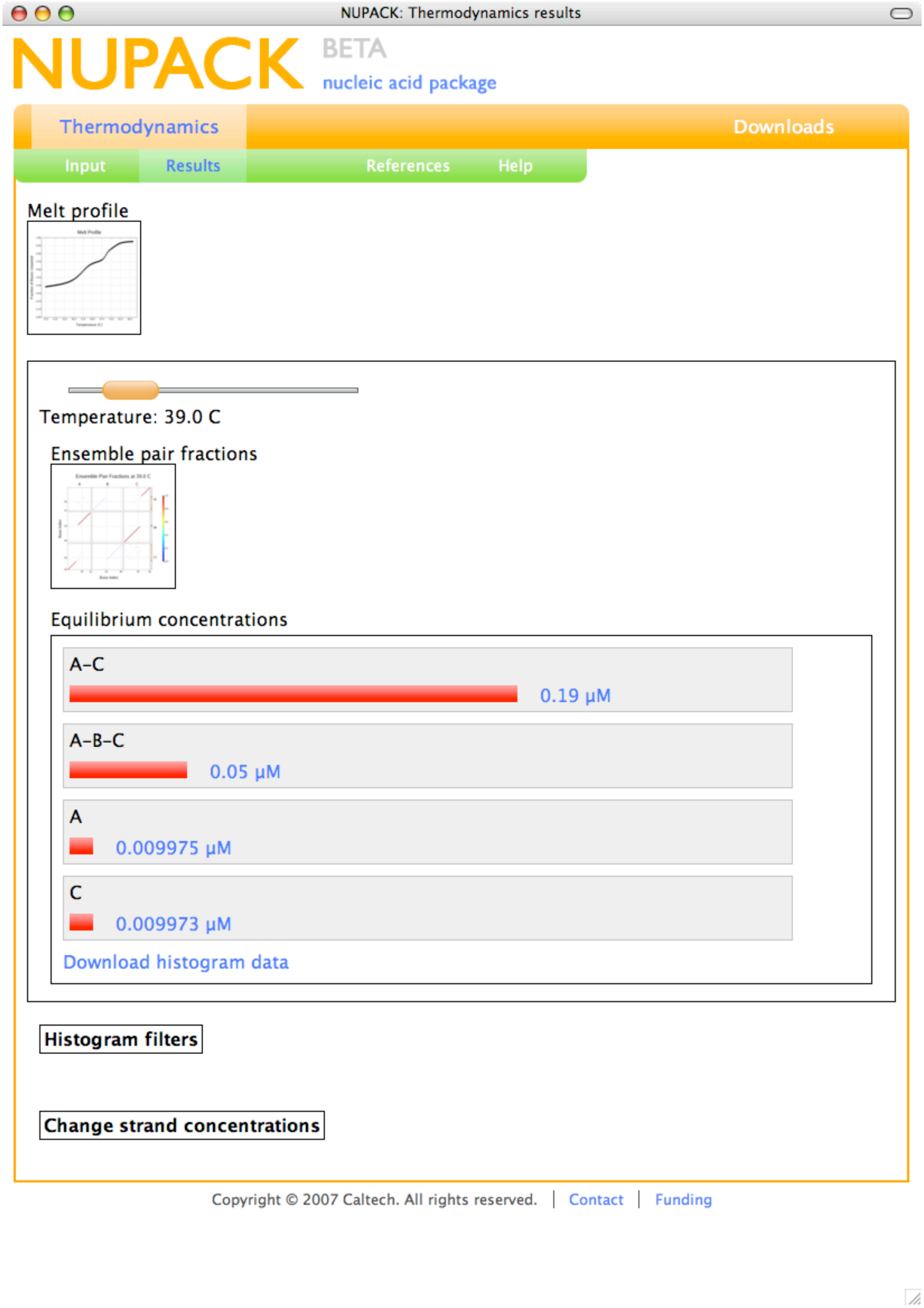
Temperature increment: 1 C

Temperature upper bound: 90 C

Email address: bois@caltech.edu

Copyright © 2007 Caltech. All rights reserved.

Figure 4.5: Screen shot of the input page for the example calculation.



Melt profile

Temperature: 39.0 C

Ensemble pair fractions

Equilibrium concentrations

A–C
0.19 µM

A–B–C
0.05 µM

A
0.009975 µM

C
0.009973 µM

Download histogram data

Copyright © 2007 Caltech. All rights reserved.

Figure 4.6: Screen shot of the main results page at $T = 39°\mathrm{C}$ for the example calculation. The thumbnail images of the melt profile and ensemble pair fraction plot expand when clicked. Larger versions of these plots, taken directly from the NUPACK SVG output, are shown in Figures 4.7 and 4.8, respectively.

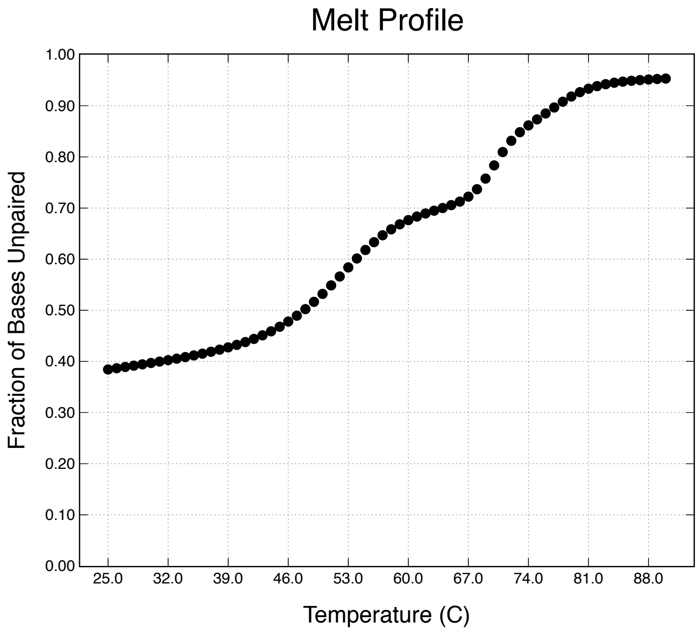


Figure 4.7: The melt profile, given in thumbnail size in the results page in Figure 4.6, for the example system.

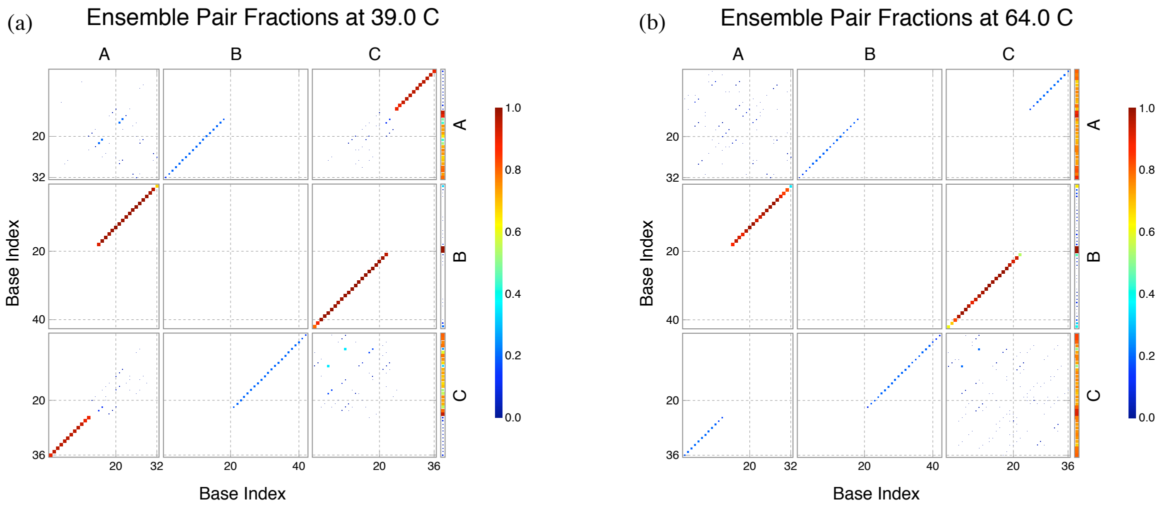
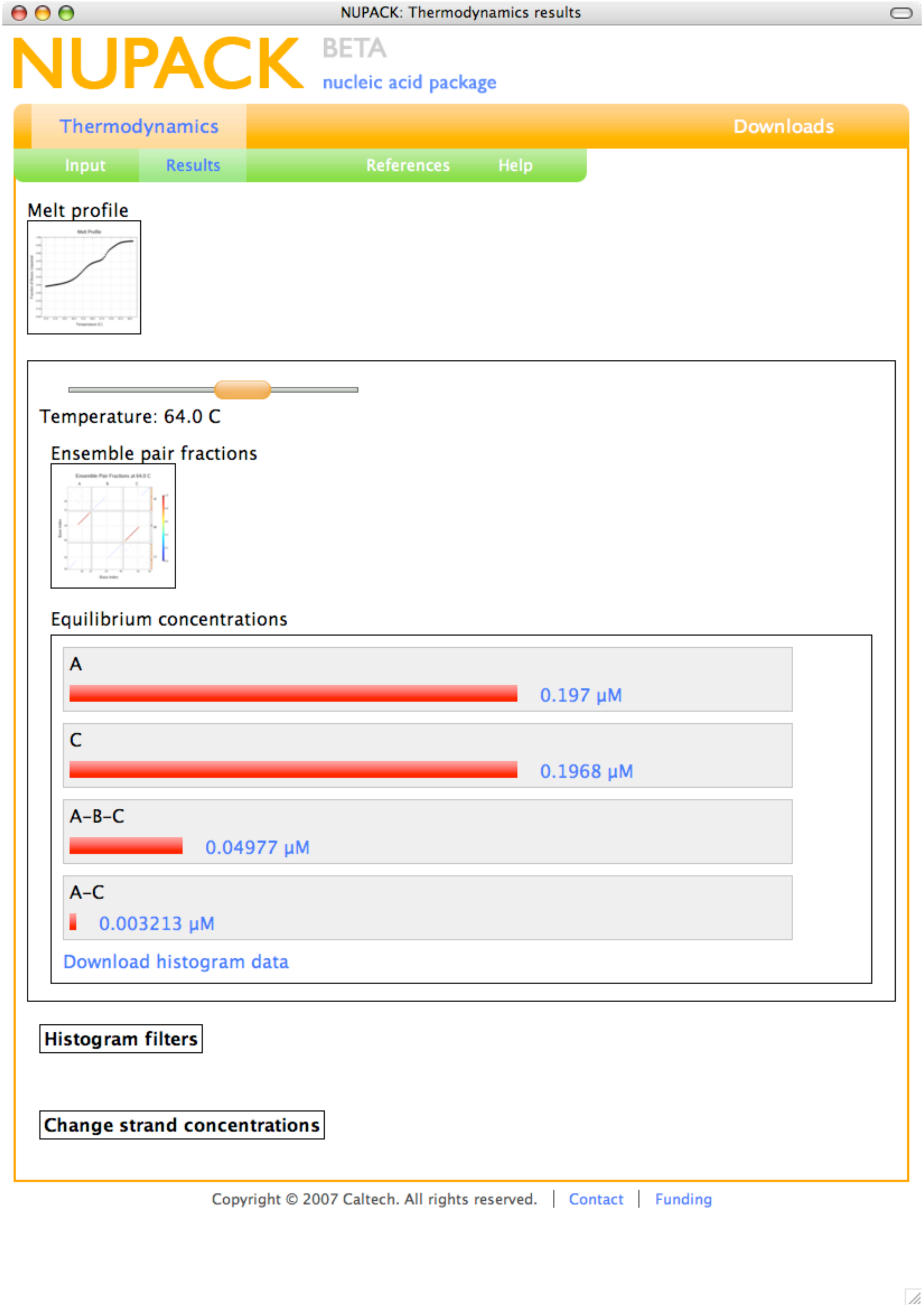


Figure 4.8: (a) Ensemble pair fractions plot for the example system for $T = 39°\mathrm{C}$. (b) Ensemble pair fractions plot for $T = 64°\mathrm{C}$.



Melt profile

Temperature: 64.0 C

Ensemble pair fractions

Equilibrium concentrations

A 0.197 μM

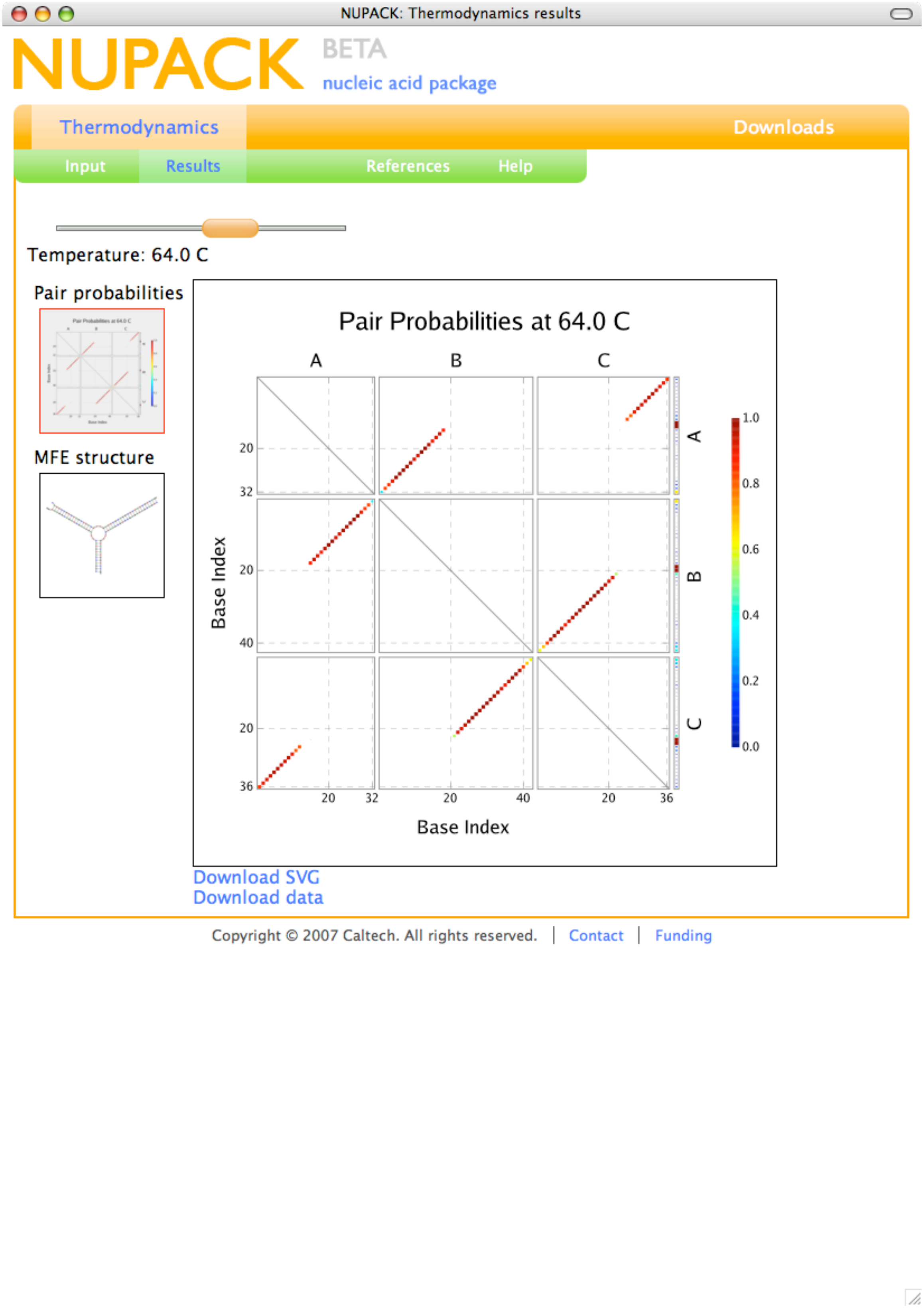
C 0.1968 μM

A–B–C 0.04977 μM

A–C 0.003213 μM

Copyright © 2007 Caltech. All rights reserved. | Contact | Funding

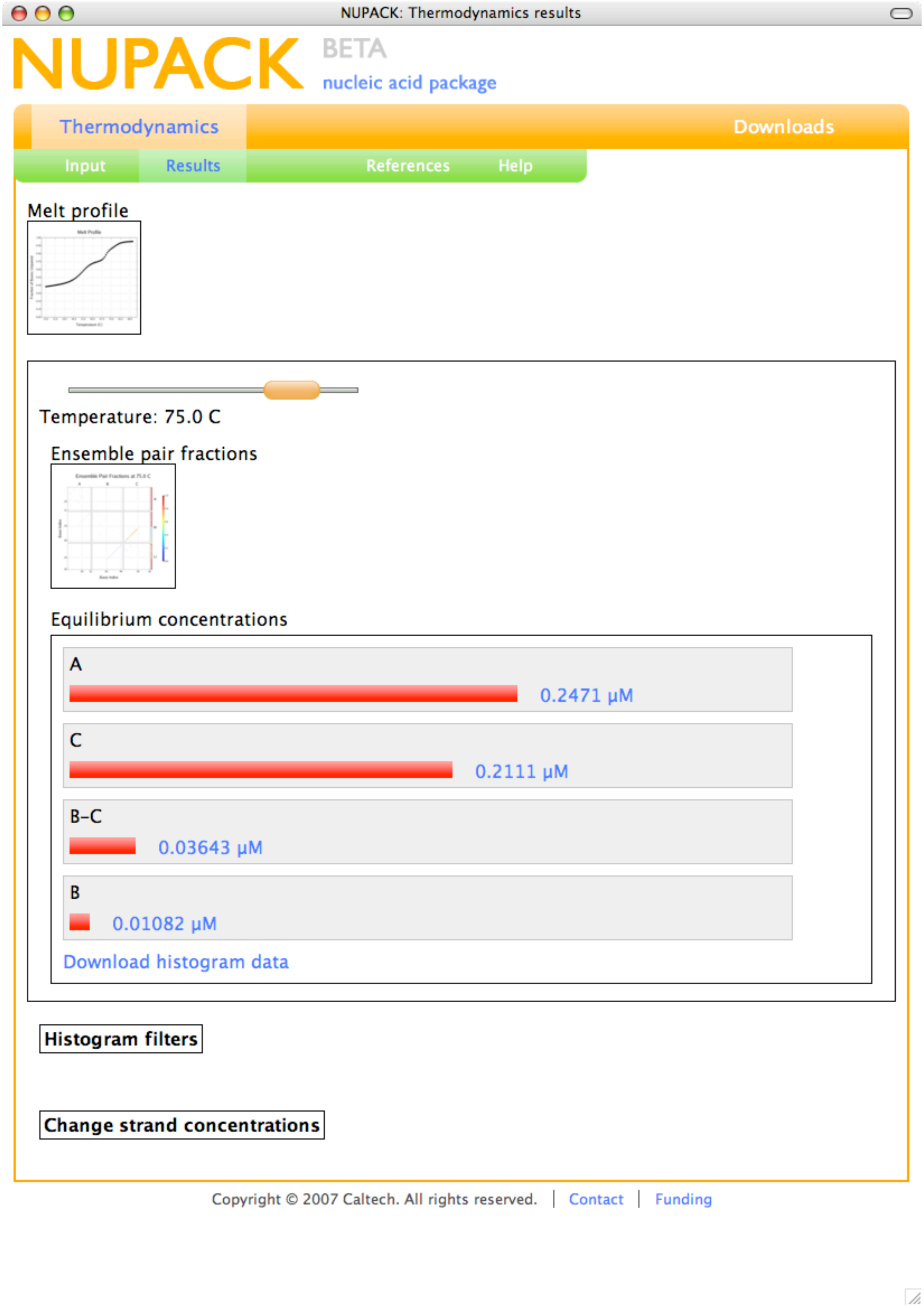
Figure 4.9: Screen shot of the results page at $T = 64°\mathrm{C}$.



Temperature: 64.0 C

Copyright © 2007 Caltech. All rights reserved.

Figure 4.10: Screen shot of the details page for ordered complex A-B-C at $T = 64°\mathrm{C}$.



NUPACK: Thermodynamics results

Melt profile

Temperature: 75.0 C

Ensemble pair fractions

Equilibrium concentrations

A 0.2471 μM

C 0.2111 μM

B–C 0.03643 μM

B 0.01082 μM

Download histogram data

Histogram filters

Change strand concentrations

Copyright © 2007 Caltech. All rights reserved. | Contact | Funding

Figure 4.11: Screen shot of the results page at $T = 75°\mathrm{C}$. An enlarged version of the ensemble pair fractions plot is given in Figure 4.12.

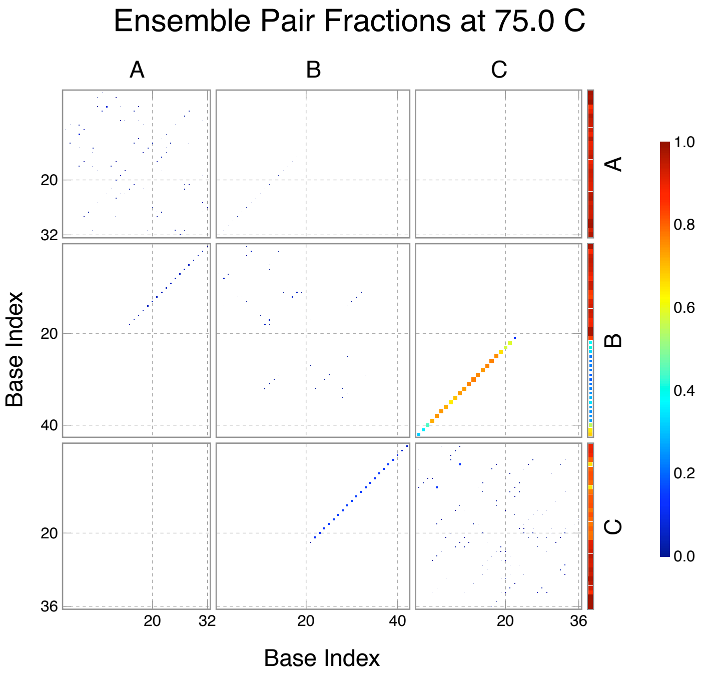


Figure 4.12: Ensemble pair fractions plot for the example system at $T = 75°\mathrm{C}$.

determined absorbance curve (scaled to match the tails of the calculated profile). The two dominant melt transitions are clearly seen, and the minor melt transition may be responsible for the flattening of the absorbance curve around $78°\mathrm{C}$. The temperatures predicted for the melt transitions are in good agreement with the experimental values.

This test case has demonstrated the utility of the methods developed in the previous chapters and their application through the NUPACK web application in analyzing solutions of interacting nucleic acids. The demands on the user are minimal and access to the essential data is facile. NUPACK has already greatly accelerated the work in the Pierce Lab for nucleic acid-based technology research [75], and will hopefully grow to do so for users worldwide.

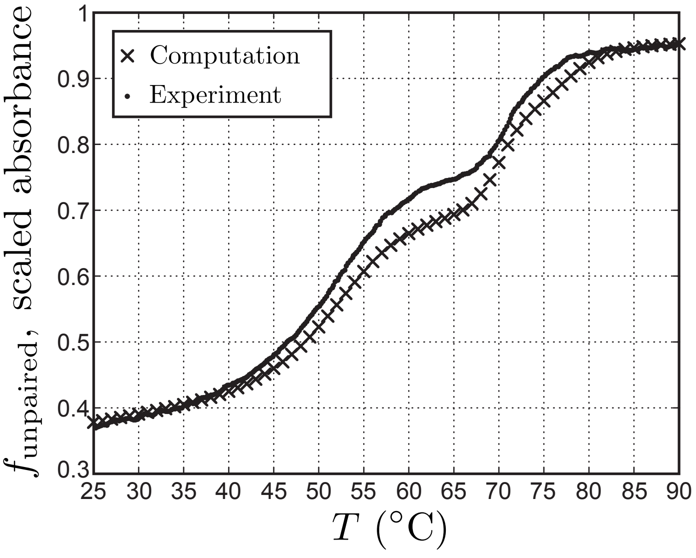


Figure 4.13: Comparison of predicted and experimental melt profiles for the example system. Scaled experimental absorbance measurements (at 260 nm) are shown as dots and the predicted melt profile (reproduced from Figure 4.7) as ×s. See Appendix G.1 for experimental methods.

Chapter 5

Thermodynamics of hybridization chain reaction

5.1 Introduction

With the tools for analysis of dilute solutions of interacting strands in place, we now use and expand them to study a designed DNA-based system called hybridization chain reaction (HCR) [1].

The HCR system consists of three strand species in a dilute solution, hairpin 1 (H1), hairpin 2 (H2), and initiator (I) and is described in Figure 5.1 and its caption. Because the H1 and H2 strands do not interact in the absence of I, the system has an "off state," and thus serves as an amplification of the presence of I, producing nicked double helix polymers from each single I strand. The gel in Figure 5.2 shows the length distribution of HCR polymers for various initiator concentrations, providing an experimental demonstration the amplification capabilities of the system.

In the analysis in this chapter, the sequences studied are those in [1]. Several other HCR sequences have been analyzed, and the results are all qualitatively similar, so we present analysis of the published system except where noted.

5.2 HCR products and their free energies

5.2.1 Definition of HCR products

We see experimentally that H1 and H2 do not interact in the absence of I (lane 2 of the gel in Figure 5.2), but their sequences are such that polymer formation is possible in principle, even in the absence of initiator. Furthermore, a polymer consisting of alternating H1 and H2 strands has sticky ends that may pair to form a ring. As such, there are several ways in which H1 and H2 may interact to form what we generally call "HCR products" (described in Figure 5.3), all of which should be carefully considered in our study. We therefore define the set Ψ to include all HCR products and the I, H1, and H2 monomers.

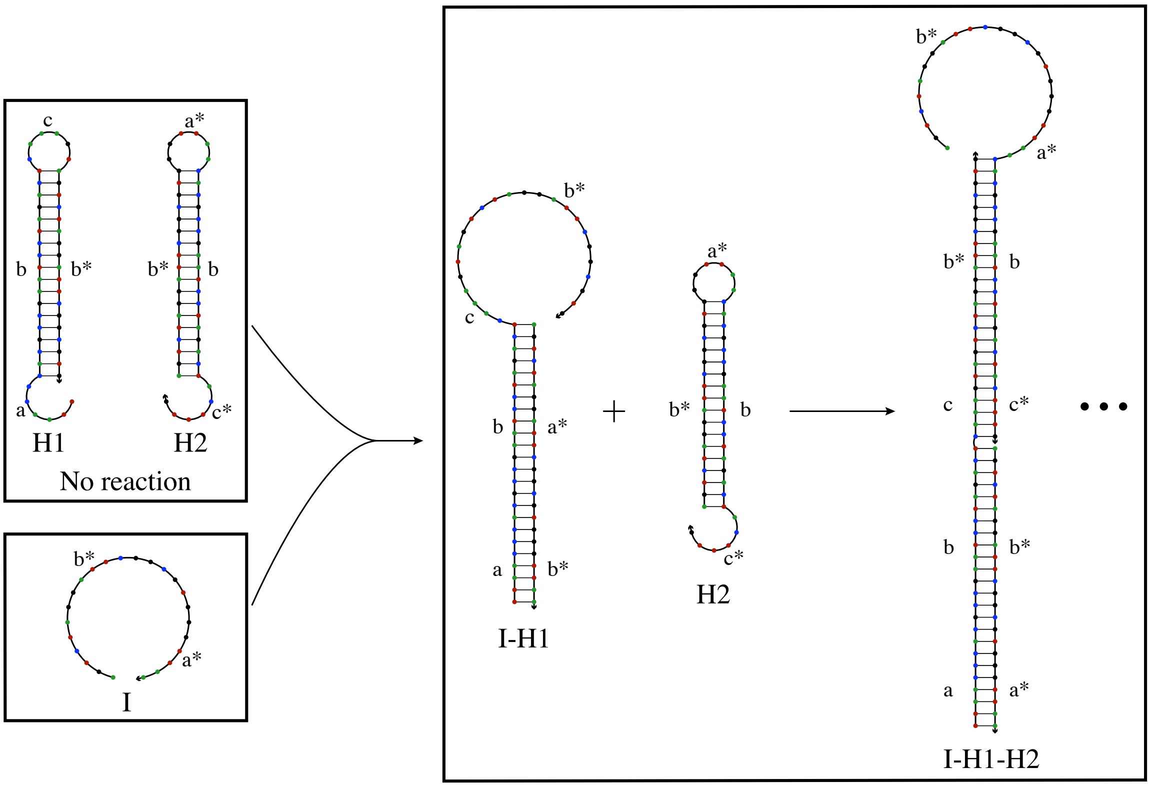


Figure 5.1: Schematic of HCR function. The sequences shown are from [1] and are also shown in Table 5.1. (Recall the NUPACK sequencing color scheme ATGC.) Subsequences are marked with a letter and their complements with asterisks. In absence of I, H1 and H2 do not interact. When I is introduced, the a* segment of I binds with the a sticky end of H1. The stem of H1 is opened by strand displacement giving ordered complex I-H1. The resulting sticky end c b* acts on H2 as I did to H1. After binding of H2, the resulting sticky end is again a*b*, the same sequence as I, which binds to H1. Thus, an HCR product consists of alternating H1 and H2 monomer units.

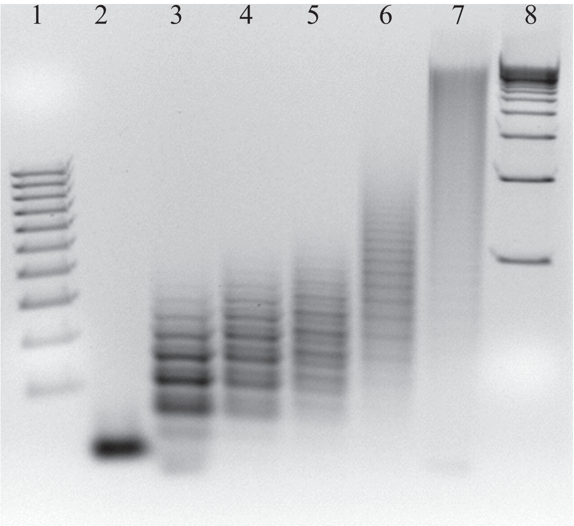


Figure 5.2: Agarose gel with ethidium bromide staining demonstrating HCR polymer lengths. The initial concentration of H1 and H2 were 1 μM for all lanes. Lanes 2-7 have varying I concentrations of (0.00, 10.00, 3.20, 1.00, 0.32, and 0.10 μM). Lanes 1 and 8 are DNA markers with 100 bp and 500 bp increments, respectively. This is a higher-resolution version of the figure in [1] with modifications for labeling. See Appendix G.2 for experimental methods.

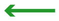
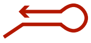
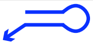
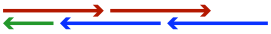
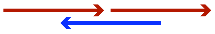
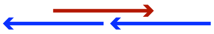
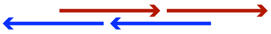
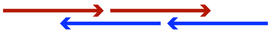
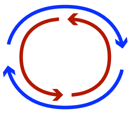
x_{I}	G_{I}	initiator (I)	
x_{H1}	G_{H1}	hairpin 1 (H1)	
x_{H2}	G_{H2}	hairpin 2 (H2)	
x_n	G_n	initiated polymer	
x_n^1	G_n^1	n odd, ending in H1	
x_n^2	G_n^2	n odd, ending in H2	
x_n^3	G_n^3	n even, ending in $3'$	
x_n^5	G_n^5	n even, ending in $5'$	
x_n^{r}	G_n^{r}	n even, ring	

Figure 5.3: The six types of HCR polymers (column 4) and associated description (column 3) and variables describing concentration (column 1) and free energy (column 2), given for species j by $G_j = -kT\ \log Q_j$. In the drawings in column 4, arrowheads indicate the $3'$ end and adjacent parallel lines indicate paired regions with other lines indicating unpaired regions. The numerical superscripts will not pose a problem because these variables are never raised to a power.

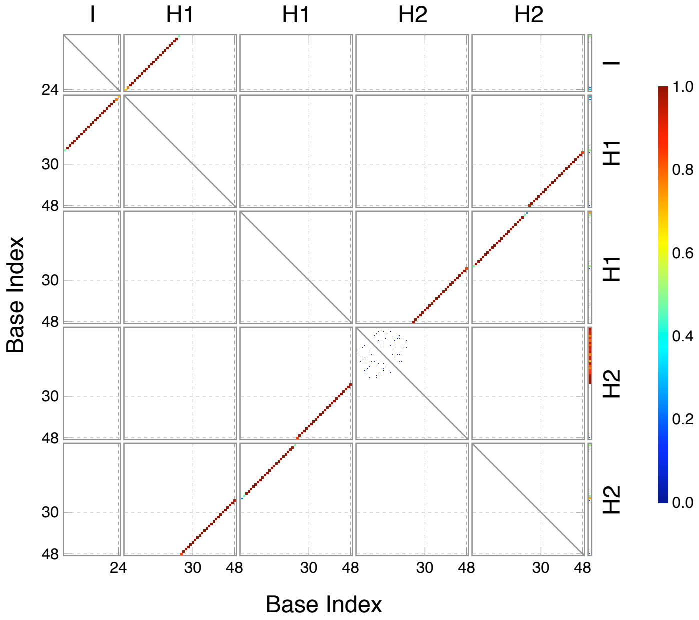


Figure 5.4: The pair probability matrix (generated by NUPACK) for an initiated HCR polymer with $n = 4$ at $T = 23°\mathrm{C}$. Aside from some low-probability spurious base pairing in the sticky end, the designed secondary structure dominates.

To describe the HCR products, we define an alternating HCR polymer product or ring of length n to consist of $\lceil n/2 \rceil$ H1 monomers and $\lfloor n/2 \rfloor$ H2 monomers (or $\lfloor n/2 \rfloor$ H1 and $\lceil n/2 \rceil$ H2 for n odd with uninitiated polymers ending in H2, see Figure 5.3) and possibly an initiator. For initiated polymers, $n \geq 1$ and for uninitiated polymers and rings, $n \geq 2$. The strand ordering corresponding to an HCR polymer begins with I if initiated and continues H1-H1$\cdots$H1-H2-H2$\cdots$H2.

We can use NUPACK to compute the partition function and base pair probabilities for an HCR polymer when n is not too large. We can therefore estimate the dominance of designed HCR polymers versus spurious ordered complexes at equilibrium. In a calculation run on the HCR systems at $23°\mathrm{C}$ that considered all ordered complexes up to size 9 (but not rings) with $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = [\mathrm{I}]_0 = 1\ \mu\mathrm{M}$, the HCR polymers and the monomers accounted for 85.30% of the H1 in solution.[1] Due to the fact that HCR products dominate the equilibrium mixture, we limit the set Ψ to only include HCR products, I, H1, and H2.

We can also use the base pair probabilities output by NUPACK to check for spurious base pairing within each HCR polymer. Figure 5.4 shows the pair probability matrix for an initiated polymer with $n = 4$ at $T = 23°\mathrm{C}$. The designed HCR secondary structure dominates, with a small amount of secondary structure on the free end of the polymer. This is indicative of the fact that the free energy of a given HCR polymer is dominated by the designed secondary structure, a typical trait of designed HCR systems for all n (plots not shown).

[1] Better designs can have much higher percentages, e.g., 99.95% for the "HCR system 2" sequences in Table 5.1.

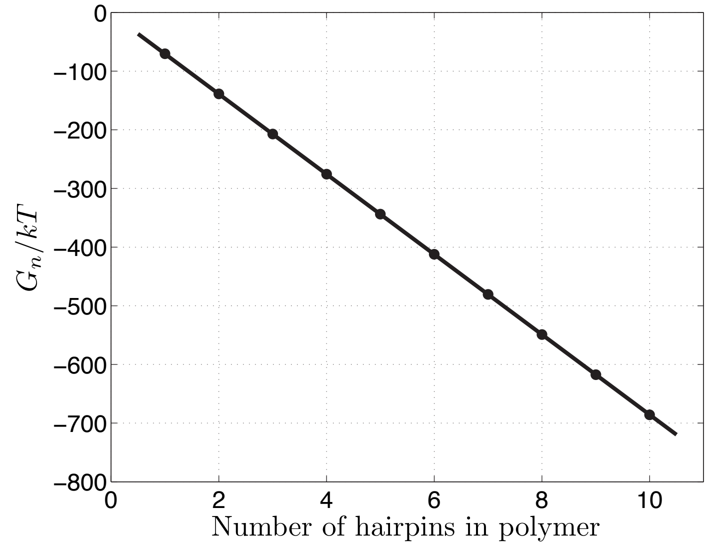


Figure 5.5: The free energy of initiated HCR polymer products at $T = 23°\mathrm{C}$ as a function of n. The incremental change in free energy for addition of a monomer is independent of n.

5.2.2 Free energy of HCR products

As n gets large, computation of the partition function (and therefore G_n) for an HCR polymer becomes intractable. We therefore seek estimates for the free energy of long HCR polymers based on free energies we can calculate for short polymers and monomers. In particular, if the free energy of monomer addition is independent of n, we can write the free energy of an HCR polymer as the sum of the free energy changes from addition of monomers.

Figure 5.5 shows the free energy for the ordered complexes corresponding to initiated HCR polymers for $1 \leq n \leq 10$ at $T = 23°\mathrm{C}$. The free energies fall on a line with the difference between a polymer of length n and one of length $n-1$ being $G_n - G_{n-1} = -40.22$ kcal/mol. The free energy change for addition of a monomer is not dependent on whether or not the polymer contains initiator (data not shown). While not very pronounced on this system, there is a sequence-dependent difference in free energy change for adding an H1 monomer versus an H2 monomer. Here, the difference in free energy between polymers of length n and $n-1$ for n odd is -40.21 kcal/mol and -40.24 kcal/mol for n even, with both these values unchanging with n.

Given that the change in free energy for adding monomers is independent of n, we can define free energy changes for all HCR steps, defined in Figure 5.6. We can then write the free energies of all HCR products in terms of these values:

$$G_n = G_{\mathrm{I}} + \Delta G_{\mathrm{I}} + \left\lfloor \frac{n+1}{2} \right\rfloor G_{\mathrm{H1}} + \left\lfloor \frac{n}{2} \right\rfloor G_{\mathrm{H2}} + \left\lfloor \frac{n-1}{2} \right\rfloor \Delta G_1 + \left\lfloor \frac{n}{2} \right\rfloor \Delta G_2 \tag{5.1a}$$

$$G_n^1 = \Delta G_{12}^5 + \frac{n+1}{2} G_{\mathrm{H1}} + \frac{n+1}{2} G_{\mathrm{H2}} + \frac{n-1}{2} \Delta G_1 + \frac{n-3}{2} \Delta G_2, \;\; n \text{ odd}, \;\; \geq 3 \tag{5.1b}$$

$$G_n^2 = \Delta G_{12}^3 + \frac{n-1}{2} G_{\mathrm{H1}} + \frac{n+1}{2} G_{\mathrm{H2}} + \frac{n-3}{2} \Delta G_1 + \frac{n-1}{2} \Delta G_2, \;\; n \text{ odd}, \;\; \geq 3 \tag{5.1c}$$

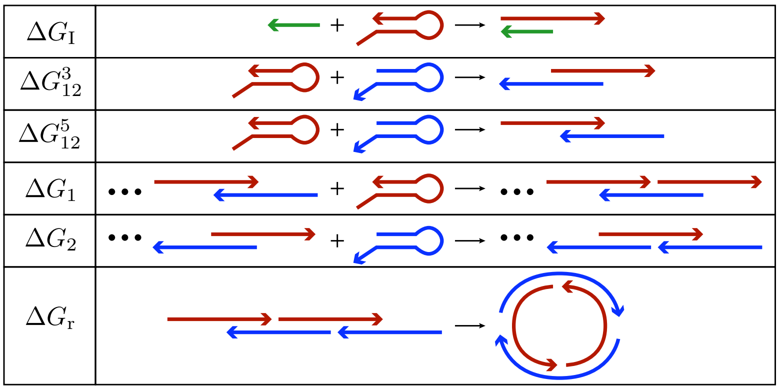


Figure 5.6: The free energy changes for the various steps in HCR processes. The processes are, from top to bottom, initiation of a polymer, two hairpins binding with the $3'$ ends unpaired, two hairpins binding with the $5'$ end unpaired, addition of an H1 monomer to the $5'$ end of an H2 strand at the end of a polymer, addition of an H2 monomer to the $3'$ end of a an H1 strand at the end of a polymer, and ring closure. The free energy change is equal to the specific free energy of the right-hand side of the arrow minus that of the left-hand side (e.g., $\Delta G_{\rm I} = G_1 - G_{\rm H1} - G_{\rm I}$). All are independent of n, except ΔG_r. The color coding of strands is consistent with that in Figure 5.3.

$$G^3_n = \Delta G^3_{12} + \frac{n}{2}\,G_{\rm H1} + \frac{n}{2}\,G_{\rm H2} + \frac{n-2}{2}\,\Delta G_1 + \frac{n-2}{2}\,\Delta G_2, \quad n \text{ even}, \quad \geq 2 \tag{5.1d}$$

$$G^5_n = \Delta G^5_{12} + \frac{n}{2}\,G_{\rm H1} + \frac{n}{2}\,G_{\rm H2} + \frac{n-2}{2}\,\Delta G_1 + \frac{n-2}{2}\,\Delta G_2, \quad n \text{ even}, \quad \geq 2 \tag{5.1e}$$

$$G^{\rm r}_n = G^5_n + \Delta G_{\rm r}(n), \quad n \text{ even}, \quad \geq 2. \tag{5.1f}$$

We can compute $G_{\rm I}$, $G_{\rm H1}$, $G_{\rm H2}$, and G_1 using NUPACK. We then know $\Delta G_{\rm I} = G_1 - G_{\rm H1} - G_{\rm I}$. We can compute G_n for several values of n (up to only $n = 5$ is usually sufficient) and perform linear regressions (as in Figure 5.5) to get ΔG_1 and ΔG_2.

Since the two products depicted in the H1 + H2 binding in Figure 5.6 belong to the same ordered complex H1-H2, we cannot directly use NUPACK to compute their free energies, and therefore need to approximate them. One strategy is to compute the free energy for the ordered complex H1-H2 ($G_{\rm H1H2} = -kT\,\log Q_{\rm H1H2}$), as well as the free energies for the dominant secondary structure for each of the two products ($G^3_{\rm H1H2}(s_{\rm MFE})$ and $G^5_{\rm H1H2}(s_{\rm MFE})$) and use the approximation

$$Q_{\rm H1H2} \approx Q^3_{\rm H1H2} + Q^5_{\rm H1H2}$$

$$\text{and}\quad \frac{Q^3_{\rm H1H2}}{Q^5_{\rm H1H2}} \approx \frac{\exp\{-G^3_{\rm H1H2}(s_{\rm MFE})/kT\}}{\exp\{-G^5_{\rm H1H2}(s_{\rm MFE})/kT\}}$$

to get $\Delta G^3_{12} = -kT\,\log Q^3_{\rm H1H2} - G_{\rm H1} - G_{\rm H2}$ and $\Delta G^5_{12} = -kT\,\log Q^5_{\rm H1H2} - G_{\rm H1} - G_{\rm H2}$. Using this approximation, we get $\Delta G^3_{12} = -1.50$ kcal/mol and $\Delta G^5_{12} = -3.34$ kcal/mol at $T = 23°$C.

We cannot use NUPACK to calculate the free energy of ring formation because it consists of a pseudoknot (all polymer graphs representing the ringed secondary structures result in crossing lines). The free energy can be estimated by calculating the free energy of the formation of the additional base pairs ($\Delta G_{\mathrm{r}}^{\mathrm{bp}}$) upon ring formation and correcting with a ring closure penalty. The ring closure probability is described by the Jacobson-Stockmayer J factor, which is the ratio of equilibrium constant for ring closure to that of bimolecular association [76]. Thus, the ring closure penalty is $-kT\ \log J(n)$, and

$$\Delta G_r(n) = \Delta G_{\mathrm{r}}^{\mathrm{bp}} - kT\ \log J(n). \tag{5.2}$$

Calculation of the J factor is difficult for these rings because the strand breaks preclude the use of the theory developed for linking number-dependent ring-closure probability [59]. There is, however, a nonzero twist modulus, at least locally, to the double-stranded regions of the ring due to stacking interactions, so a direct application of wormlike chain ring closure probability [76] also introduces error. Both theories share the common result that the J factor goes to zero rapidly as n falls below one persistence length (≈ 150 bp for B-DNA [61]) and as $n^{-3/2}$ beyond about four persistence lengths [59]. For our analysis, we overestimate the ring closure penalty by approximating the J factor as the lower bound of a J factor for a twistable wormlike chain with torsional energy similar to its bending energy. For any of the appropriate models for ring closure, the maximal J factor is slightly greater than 10^{-2} and occurs between one and two persistence lengths on a peak that spans about two persistence lengths [59]. For HCR systems, this corresponds to n between 6 and 10, assuming the nicked HCR product has the same persistence length of B-DNA. In this regime, the ring closure penalty is typically less than $5kT$ (3 kcal/mol at $T = 23°\mathrm{C}$). By comparison, the free energy for base pair formation at the free ends of the uninitiated polymer to form a ring is -40.45 kcal/mol at $T = 23°\mathrm{C}$. Thus, the details of the approximation for loop closure are not important, as the free energy of base pairing dominates, as will be evident in our subsequent analyses.

With the estimates for the ΔGs and (5.1), we now have enough information to completely characterize the thermodynamics of an HCR system. We did so by only computing a handful of partition functions for ordered complexes with relatively few bases. In quoting these free energies, however, it is important to note that the loop-based free energy model does not take into account any tertiary structure outside of that implicitly included in the empirical loop parameters. Therefore, the calculated free energies neglect the contributions due to the length of the polymer and associated chain entropy or bending rigidity. However, the base-pairing energies are tens of kT per HCR monomer, and are expected to dominate the free energy.

5.3 Properties of the HCR free energy landscape

We have already seen in Section 3.4.5 that coarsening the free energy landscape of a nucleic acid solution such that the state variables are the ordered complex concentrations results in a continuous, strictly convex landscape in the thermodynamic limit. Because we do not consider the intermediate steps of going from one state to another (e.g., from having two separate H1 and H2 hairpins to having an H1-H2 ordered complex), we have no knowledge of the rates of these processes with this level of coarsening. In fact, for any degree of coarsening, we only know the relative free energies of one point on the landscape to another and no other information.

As such, we must carefully consider what portions of the landscape are kinetically accessible from a given initial condition. With this in mind, we proceed to investigate the properties of various parts of the landscape, beginning with the (unique) global free energy minimum.

5.3.1 Free energy of a dilute solution on HCR strands

Using the results of Section 5.2.2, we may write the free energy of a solution containing HCR strands. We define the set Ψ to include the monomers H1, H2, and I, and all HCR products described in Figure 5.3 up to a maximal n, called $n_{\max}$.

Using (2.17), the per-solvent free energy for a solution with HCR strands (in units of kT) is

$$\begin{aligned} g(x,T) = & g_{\mathrm{ref}} + x_{\mathrm{I}}\left(\log x_{\mathrm{I}} + \frac{G_{\mathrm{I}}}{kT} - 1\right) + x_{\mathrm{H1}}\left(\log x_{\mathrm{H1}} + \frac{G_{\mathrm{H1}}}{kT} - 1\right) + x_{\mathrm{H2}}\left(\log x_{\mathrm{H2}} + \frac{G_{\mathrm{H2}}}{kT} - 1\right) \\ & + \sum_{n=1}^{n_{\max}} x_n \left(\log x_n + \frac{G_n}{kT} - 1\right) + \sum_{n \text{ even}, \geq 2}^{n_{\max}} \left[x_n^5\left(\log x_n^5 + \frac{G_n^5}{kT} - 1\right) + x_n^3\left(\log x_n^3 + \frac{G_n^3}{kT} - 1\right)\right] \\ & + \sum_{n \text{ odd}, \geq 3}^{n_{\max}} \left[x_n^1\left(\log x_n^1 + \frac{G_n^1}{kT} - 1\right) + x_n^2\left(\log x_n^2 + \frac{G_n^2}{kT} - 1\right)\right] \\ & + \sum_{n \text{ even}, \geq 2}^{n_{\max}} x_n^{\mathrm{r}}\left(\log x_n^{\mathrm{r}} + \frac{G_n^{\mathrm{r}}}{kT} - 1\right), \end{aligned} \tag{5.3}$$

where the expressions in (5.1) and (5.2) are used for the free energies of the HCR products. The expressions of the constraints on x, equivalent to (2.18b), are

$$x_{\mathrm{I}}^0 = x_{\mathrm{I}} + \sum_{n=1}^{n_{\max}} x_n \tag{5.4a}$$

$$x_{\mathrm{H1}}^0 = x_{\mathrm{H1}} + \sum_{n=1}^{n_{\max}} \left\lfloor \frac{n+1}{2} \right\rfloor x_n + \sum_{n \text{ odd}, \geq 3}^{n_{\max}} \left(\frac{n+1}{2} x_n^1 + \frac{n-1}{2} x_n^2\right) + \sum_{n \text{ even}, \geq 2}^{n_{\max}} \frac{n}{2}\left(x_n^3 + x_n^5 + x_n^{\mathrm{r}}\right) \tag{5.4b}$$

$$x_{\mathrm{H2}}^0 = x_{\mathrm{H2}} + \sum_{n=1}^{n_{\max}} \left\lfloor \frac{n}{2} \right\rfloor x_n \sum_{n \text{ odd}, \geq 3}^{n_{\max}} \left(\frac{n-1}{2} x_n^1 + \frac{n+1}{2} x_n^2\right) + \sum_{n \text{ even}, \geq 2}^{n_{\max}} \frac{n}{2}\left(x_n^3 + x_n^5 + x_n^{\mathrm{r}}\right). \tag{5.4c}$$

The choice of $n_{\max}$ is somewhat arbitrary. In a strategy similar to that described in Section 2.4.5, $n_{\max}$ should be chosen such that the number-weighted concentration (nx_n) of HCR polymer products are close

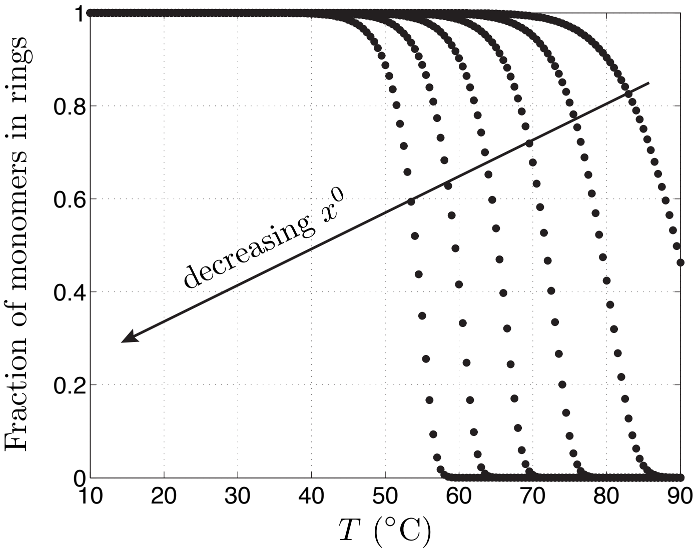


Figure 5.7: Melt curves for ring formation in the HCR system. The y-axis is the fraction of H1 and H2 monomers that are contained in ring products. The concentrations of H1, H2, and I (described by the vector x^0) are equal, varying from 10^{-5} to 10^{-10} M going from right to left on the plot.

to zero as n gets close to $n_{\max}$, making sure $n_{\max}$ is above any nucleation barrier for polymerization. In practice, $n_{\max}$ can be up to about 10^6 and the concentration solver will still converge in a matter of minutes (on a 3.06 GHz Intel® Xeon™ processor with 4 GB of RAM).

5.3.2 Minimal free energy state of the HCR system

After having computed all the necessary HCR product free energies, we directly apply the techniques of Section 2.4.3 to compute the equilibrium concentrations of HCR products. This corresponds to the unique global free energy minimum of the HCR free energy landscape, where it is coarsened such that the state variables are the HCR product concentrations. At moderate concentrations and low temperatures, e.g, $[\mathrm{I}]_0 = [\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$ and $T = 23°\mathrm{C}$, rings completely dominate the equilibrium products. Specifically, less than $\approx 10^{-5}\%$ of H1 strands are in ordered complexes other than the $n = 6$ ring under these conditions. Figure 5.7 shows the melt profile for the HCR system over a range of hairpin concentrations. Over all temperatures and concentrations, the dominant species are rings with $n = 6$ and monomers I, H1, and H2. The initiator remains unbound, having no impact on the equilibrium state.

5.3.3 The kinetic basis for the absence of rings

While rings dominate at equilibrium, they are not seen experimentally in the HCR system, as they are kinetically inaccessible. Because the experimental system is constructed by mixing solutions, each containing only strands of type I, H1, or H2, the starting point on the free energy landscape for the experimental systems is $x_{\mathrm{I}} = x_{\mathrm{I}}^0$, $x_{\mathrm{H1}} = x_{\mathrm{H1}}^0$, and $x_{\mathrm{H2}} = x_{\mathrm{H2}}^0$, with the concentrations of all other species being zero. Therefore,

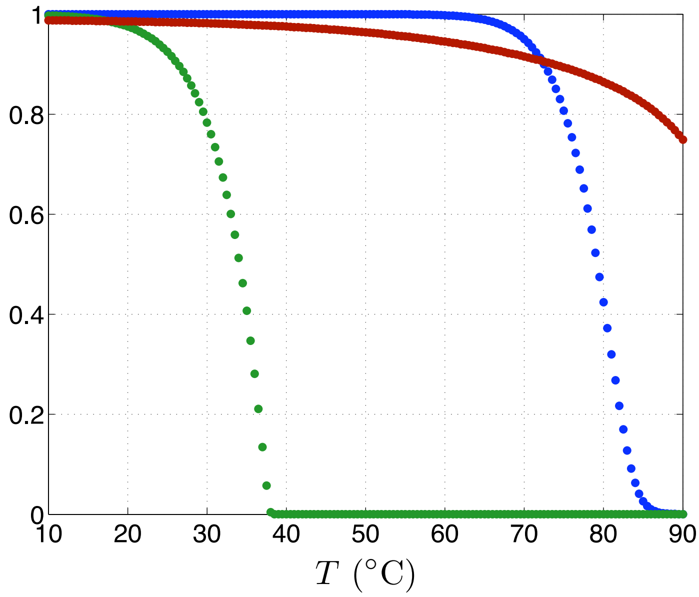


Figure 5.8: Equilibrium melt curves for the HCR system with $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$ in absence of initiator. Red curve: the fraction of bases in the stems of H1 and H2 that are paired. Blue curve: Fraction of H1 and H2 monomers that are in rings. Green curve: Fraction of H1 and H2 monomers that are in uninitiated HCR polymers if rings are not allowed.

the first step to form any HCR product must be either H1 binding with H2 or I binding with H1, followed by subsequent alternating binding of H2 and H1 monomers. If the first step is I binding H1 (which we will see is much more likely), I is bound to the sticky end on the H1 end of the polymer, thereby precluding it from binding with the sticky end of H2 on the other end of the polymer. In order for a ring to form, the initiator must unbind before ring closure can proceed. As there are 24 base pairs between I and H1, initiator unbinding is very unlikely.

Given that it is much more kinetically favorable for rings to form from uninitiated polymers, the first step in ring formation is binding of H1 and H2. If the stems of the H1 and H2 hairpins are intact, initiating HCR polymer growth with H1-H2 binding is kinetically hindered because there is no toehold from which to initiate base pairing. The sticky ends at the end of the stems (the $5'$ end of H1 and $3'$ end of H2) are complementary to the loop of the other hairpin, and therefore base pairing must initiate via loop invasions. For loops with only 6 bases, this process is sterically hindered.

Figure 5.8 shows calculated melt curves for ring formation, uninitiated polymer formation, and hairpin stem melting for the HCR system. For any temperature at which ring or polymer formation is favorable, the stems of the hairpins are intact, thereby kinetically hindering the nucleation of polymerization.

To probe the possibility of ring formation in a system where hybridization between H1 and H2 is not so hindered, we consider an HCR system where the length of the loops in the hairpins is 12 bases, rather than 6 (Figure 5.9). Figure 5.10 shows the results of a calculation analogous to that of Figure 5.8. For the modified system, the stems are more open for temperatures at which ring formation is favorable, so rings are

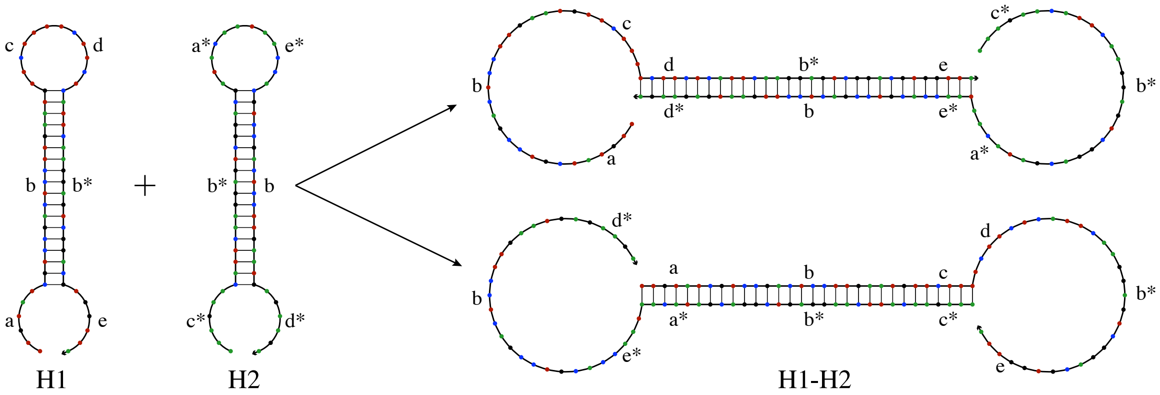


Figure 5.9: The modified HCR system to study ring formation. As in Figure 5.1, the complement of a subsequence is marked with an asterisk and the base coloring scheme is that of NUPACK. The two dominant HCR products for uninitiated binding of H1 and H2 are shown. Any of the sticky ends in these products may initiate hybridization with another monomer thereby propagating HCR.

more likely to form. Figure 5.11 shows verification of ring formation. The modified H1 and H2 were added to a solution at 1 μM concentration and slowly annealed from 95°C to 37°C. The result is shows in lane 1. After annealing, λ exonuclease was added to the solution, which sequentially digests double-stranded DNA starting from the end of a strand [77] (lane 2). As seen on the gel, the lower band of HCR products remains, indicating that there was no end immediately available for digestion by the exonuclease. By contrast, the DNA marker ladder is digested (lanes 3 and 4). Thus, the lower band corresponds to a ring. We have shown that, if less kinetically hindered, rings will form (from experiment) and are the thermodynamic minimum (from calculations). In the standard HCR system, however, they are kinetically inaccessible.

5.3.4 Characterizing kinetic traps in the HCR free energy landscape

In the previous section, we saw that H1-H2 binding is kinetically hindered because the hairpin stems are intact at temperatures where HCR polymers may form. Besides being kinetically hindered, H1-H2 binding is thermodynamically unfavorable, in contrast to I-H1 binding. For the standard HCR system, when one H1 strand and one H2 strand are separate in solution, they have a total 36 base pairs (18 in each stem). By contrast, the H1-H2 complex has only 24. The "break even point" in terms of number of base pairs (ignoring loop energy penalties and concentration effects) for uninitiated HCR polymer products is $n = 4$, when the HCR polymer and the lone hairpins each have 72 base pairs. Thus, uninitialized polymerization has a significant barrier.

While a qualitative discussion of the pathways to form HCR products was given in the previous section, the level of coarsening of the free energy landscape used to determine the equilibrium does not consider any of the details necessary to describe the sequential nature of HCR product production. It is therefore useful to have a picture of the connectivity of the free energy landscape given the sequential nature of the formation of HCR products. To construct a representative landscape, we consider a box containing a single

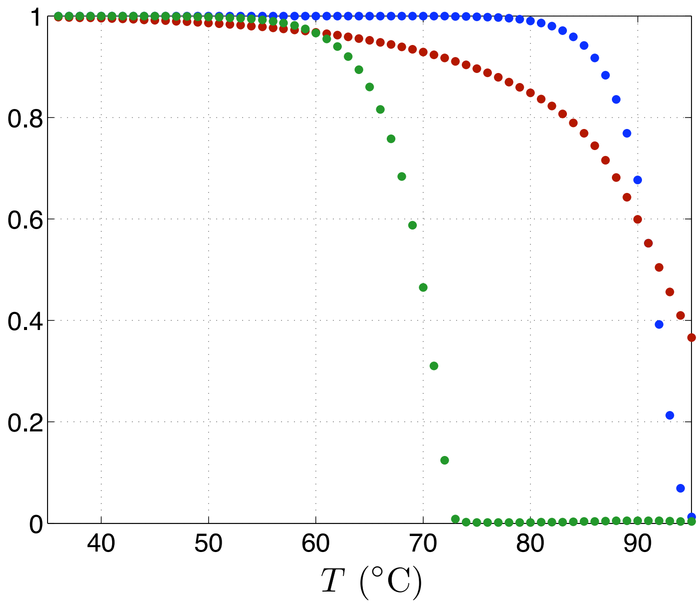


Figure 5.10: Equilibrium melt curves for the modified HCR system with longer loops with $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$ in absence of initiator. Red curve: the fraction of bases in the stems of H1 and H2 that are paired. Blue curve: Fraction of H1 and H2 monomers that are in rings. Green curve: Fraction of H1 and H2 monomers that are in uninitiated HCR polymers if rings are not allowed.

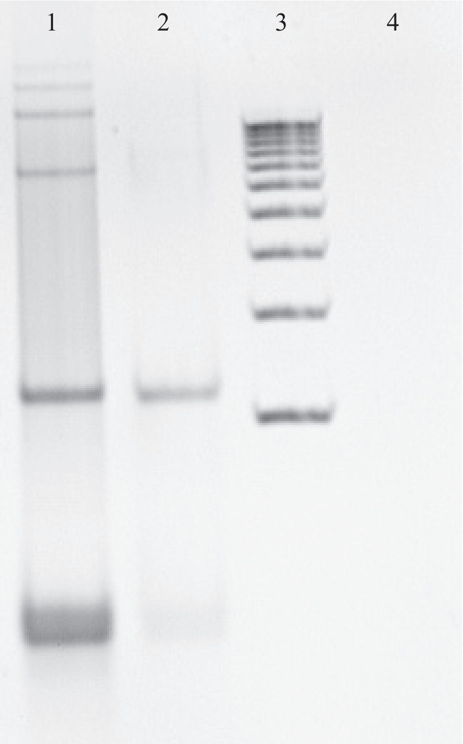


Figure 5.11: Ethidium bromide labeled acrylamide gel demonstrating the existence of rings in the modified HCR system. Lane 1: modified H1 and H2 at 1 μM. Lane 2: H1 and H2 with exonuclease. Lane 3: 100 bp DNA markers. Lane 4: 100 bp DNA markers with exonuclease. See Appendix G.3 for experimental methods.

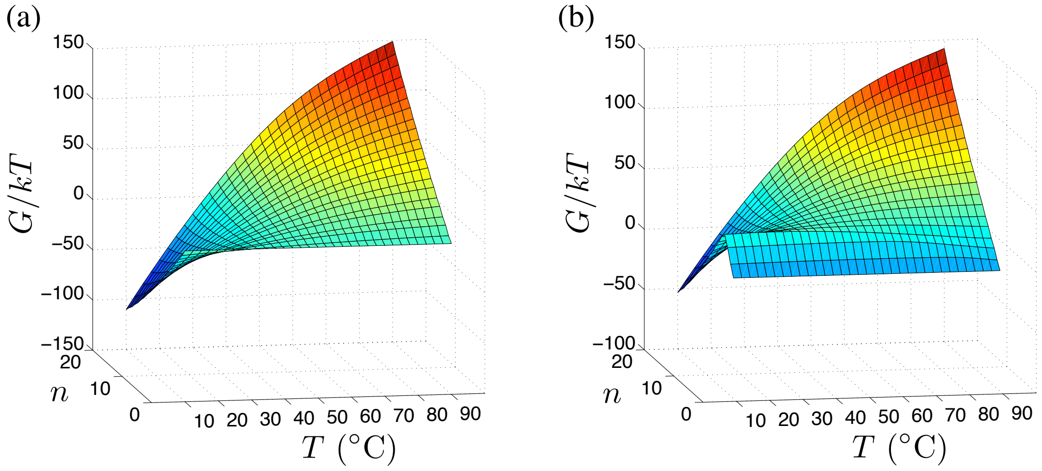


Figure 5.12: (a) A representative free energy landscape for growth of initiated HCR polymer products. (b) The corresponding representative landscape for growth of uninitiated HCR polymer products.

initiator and 20 hairpins, 10 each of H1 and H2. The volume of the box (and therefore M_s) is such that $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$ and $[\mathrm{I}]_0 = 0.1\ \mu\mathrm{M}$. We grow a single HCR polymer product in the box from the available strands with population m of the box being $m = (m_{\mathrm{I}}, m_{\mathrm{H1}}, m_{\mathrm{H2}}, m_n)^T$ with

$$m_{\mathrm{I}} = \begin{cases} 1 & n = 0 \\ 0 & n \geq 1 \end{cases}, \quad m_{\mathrm{H1}} = 10 - \left\lceil \frac{n}{2} \right\rceil, \quad m_{\mathrm{H2}} = 10 - \left\lfloor \frac{n}{2} \right\rfloor, \quad \text{and} \quad m_n = \begin{cases} 0 & n = 1 \\ 1 & n \geq 1 \end{cases}.$$

As the polymer grows, the free energy of the box is calculated as $G_{\mathrm{box}} = -kT \log Z(m, M_s, T)$, where $Z(m, M_s, T)$ is given by (2.11). We can construct a similar landscape for a box that is absent of initiator. Figure 5.12 shows these landscapes constructed as a function of n and T. For temperatures low enough such that polymer growth is favorable, each step results in a decrease in the free energy of the system for initiated polymers. Conversely, the first three steps (up to $n = 4$) of polymer formation for the uninitiated case result in increases in free energy, indicating a barrier for uninitiated polymer formation. Since uninitiated growth is necessary for ring formation, it, too, experiences this barrier. Polymer growth is unfavorable at high temperatures whether or not initiator is present.

Figure 5.13 shows the landscape construction for the modified HCR system used in ring detection. The extra bases in the hairpin loop serve to lower the barrier for uninitiated polymer growth, with the barrier being surmounted after $n = 2$, as opposed to $n = 4$ for the standard system. In addition to more open stems at temperatures where polymers can form, the low barrier for uninitiated polymer nucleation of the modified system promotes ring formation, as was verified in the gel in Figure 5.11.

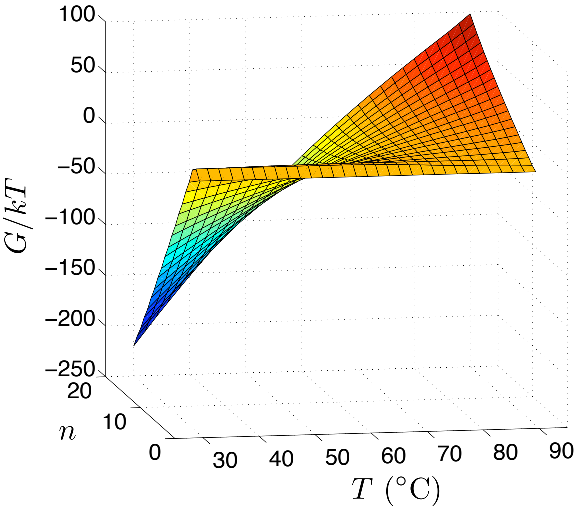


Figure 5.13: A representative free energy landscape for growth of uninitiated HCR polymers for the modified system of Section 5.3.3.

5.3.5 Reusability of the HCR hairpins

The previous discussion established that the hairpins do not polymerize in the absence of initiator, regardless of temperature. These results suggest that the HCR hairpins are reusable if they can be separated from the initiator. To probe this effect, an HCR system was constructed out of DNA H1 and H2 with RNA initiator (see Figure 5.14). After HCR polymers formed (lane 2), the initiator was digested by adding RNase H, which is an enzyme that digests the RNA in a DNA-RNA hybrid but does not degrade DNA or unpaired RNA [78]. The resulting HCR polymers lengthen (lane 3) due to end-joining and the shift in equilibrium toward longer polymers as a result of the absence of initiator. Ring formation is precluded because the HCR polymers are already long, thereby making the ring-closure penalty too great to enable ring-closure pairing of the chain ends. The system is then annealed (lane 4), serving to both melt the HCR products and to thermally destroy the RNase H, resulting in recharged H1 and H2 monomers in solution. Reintroduction of an RNA initiator again triggers HCR (lane 5), demonstrating the reusability of the monomers.

The discovery of the reusability of the hairpins was only made after the calculations in Figure 5.8 revealed the high temperature stability of the HCR hairpins. This is an example where a computational study led to the discovery of a novel property of the system and directed experiments.

5.4 Determination of HCR equilibrium for $n_{\max} \to \infty$

In Section 5.3.2, we calculated the equilibrium state of the HCR system by specifying a maximum polymer length $n_{\max}$ and using the techniques of Section 2.4.3 to determine the equilibrium concentrations given initial concentrations of $x^0 = (x_{\mathrm{I}}, x_{\mathrm{H1}}, x_{\mathrm{H2}})^T$ for each temperature we wished to consider. The time complexity of the calculation is $\mathcal{O}(n_{\max})$, thereby scaling linearly with the length of HCR polymers we consider.

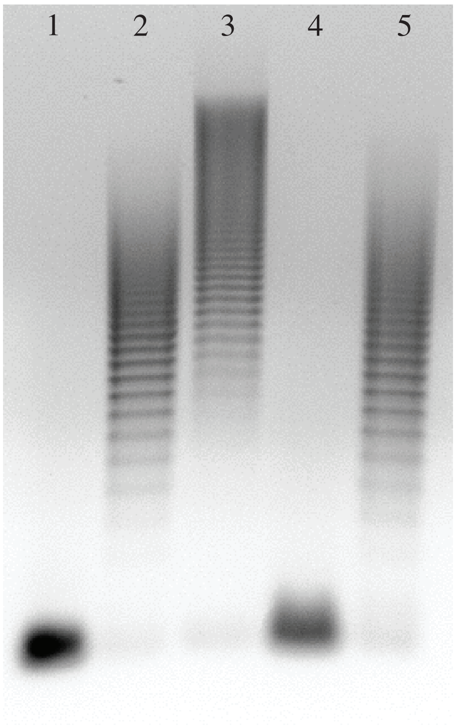


Figure 5.14: A gel from the experiment testing HCR hairpin reusability. Lane 1: Hairpins alone in solution. Lane 2: Addition of 0.2× RNA initiator. Lane 3: Addition of RNase H. Lane 4: Heat to 90°C for 90 seconds and then move to 37°C incubator. Lane 5: Re addition of RNA initiator. See Appendix G.4 for experimental methods.

This strategy is effective, and indeed produces essentially the same results when $n_{\max}$ is large as the treatment for which $n_{\max} \to \infty$ (described in the following pages), but does not immediately answer some basic questions about the HCR system we can explore by continuing further in our theoretical development by taking $n_{\max} \to \infty$ before resorting to numerical solvers. Because uninitiated polymers and rings are kinetically inaccessible and to avoid clutter in the resulting expressions, we henceforth consider only initiated polymers.

5.4.1 Lagrange dual function

We follow a development similar to that in Section 2.4 and begin by defining the Lagrange multipliers $\lambda = (\lambda_{\mathrm{I}}, \lambda_{\mathrm{H1}}, \lambda_{\mathrm{H2}})^T$ to enforce the conservation of mass constraints (5.4). The Lagrangian is

$$\begin{aligned}\mathcal{L}(x,\lambda) = g(x,T) &+ \lambda_{\mathrm{I}} \left(x_{\mathrm{I}}^0 - x_{\mathrm{I}} - \sum_{n=1}^{n_{\max}} x_n \right) \\ &+ \lambda_{\mathrm{H1}} \left(x_{\mathrm{H1}}^0 - x_{\mathrm{H1}} - \sum_{n=1}^{n_{\max}} \left\lfloor \frac{n+1}{2} \right\rfloor x_n \right) \\ &+ \lambda_{\mathrm{H2}} \left(x_{\mathrm{H2}}^0 - x_{\mathrm{H2}} - \sum_{n=1}^{n_{\max}} \left\lfloor \frac{n}{2} \right\rfloor x_n \right).\end{aligned}$$

The first KKT condition (2.20a) gives that the optimal x and λ satisfy

$$x_{\mathrm{I}} = \exp\left\{-\frac{G_{\mathrm{I}}}{kT} + \lambda_{\mathrm{I}}\right\} \tag{5.5a}$$

$$x_{\mathrm{H1}} = \exp\left\{-\frac{G_{\mathrm{H1}}}{kT} + \lambda_{\mathrm{H1}}\right\} \tag{5.5b}$$

$$x_{\mathrm{H2}} = \exp\left\{-\frac{G_{\mathrm{H2}}}{kT} + \lambda_{\mathrm{H2}}\right\} \tag{5.5c}$$

$$x_n = \exp\left\{-\frac{G_n}{kT} + \lambda_{\mathrm{I}} + \left\lfloor\frac{n+1}{2}\right\rfloor \lambda_{\mathrm{H1}} + \left\lfloor\frac{n}{2}\right\rfloor \lambda_{\mathrm{H2}}\right\}. \tag{5.5d}$$

The Lagrange dual function is

$$h(\lambda) = \inf_x \mathcal{L}(x,\lambda) = g_{\mathrm{ref}} + \lambda^T x^0 - x_{\mathrm{I}} - x_{\mathrm{H1}} - x_{\mathrm{H2}} - \sum_{n=1}^{n_{\max}} x_n \tag{5.6}$$

where the dependencies of the concentrations on λ are given by (5.5). Substitution of (5.1) into (5.5d) gives

$$x_n = \begin{cases} K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}\, \xi^{\frac{n-1}{2}} & n \text{ odd} \\ K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}\, \xi^{\frac{n-2}{2}} & n \text{ even} \end{cases} \tag{5.7}$$

where the equilibrium constants are defined by

$$K_{\mathrm{I}} \equiv \exp\{-\Delta G_{\mathrm{I}}/kT\},$$

$$K_1 \equiv \exp\{-\Delta G_1/kT\},$$

$$\text{and } K_2 \equiv \exp\{-\Delta G_2/kT\},$$

and

$$\xi \equiv K_1 x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}. \tag{5.8}$$

Substitution of (5.7) and (5.8) into (5.6) and taking $n_{\max} \to \infty$ gives $h(\lambda) = -\infty$ for $\xi \geq 1$ and

$$h(\lambda) = g_{\mathrm{ref}} + \lambda^T x^0 - x_{\mathrm{I}} - x_{\mathrm{H1}} - x_{\mathrm{H2}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{1 + K_2 x_{\mathrm{H2}}}{1-\xi} \tag{5.9}$$

for $\xi < 1$.

The dual function is strictly concave because for $\xi < 1$, it is equal to the sum of strictly concave functions (the strict concavity was proved in Lemma 2.2), and taking $h(\lambda) = -\infty$ for $\xi \geq 1$ is equivalent to an extended-value extension of the function, making it concave by definition [28]. Similarly, $\lim_{n_{\max}\to\infty} g(x,T)$ (with $g(x,T)$ given by (5.3)) is a strictly convex function of x. The constraints (5.4) are linear and the corresponding stoichiometry matrix (with an infinite number of columns) has full row rank (as shown in Section 2.4.3), so strong duality holds. Thus, taking $n_{\max} \to \infty$ did not affect our ability to solve a low-dimensional unconstrained dual problem to find the equilibrium concentrations.

5.4.2 HCR equilibrium solver

Although the task of finding the equilibrium concentrations of HCR products is still an unconstrained convex programming problem, we cannot directly apply the techniques of Section 2.4.3 because the governing equations are now different, e.g., the matrix A no longer appears. We therefore develop a new solver for HCR equilibria.

The gradient of the dual function, $\nabla h(\lambda)$, defined for $\xi < 1$, is given by

$$\frac{\partial h}{\partial \lambda_{\mathrm{I}}} = x_{\mathrm{I}}^0 - x_{\mathrm{I}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{1 + K_2 x_{\mathrm{H2}}}{1 - \xi} \tag{5.10a}$$

$$\frac{\partial h}{\partial \lambda_{\mathrm{H1}}} = x_{\mathrm{H1}}^0 - x_{\mathrm{H1}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{1 + K_2 x_{\mathrm{H2}}}{(1 - \xi)^2} \tag{5.10b}$$

$$\frac{\partial h}{\partial \lambda_{\mathrm{H2}}} = x_{\mathrm{H2}}^0 - x_{\mathrm{H2}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{\xi + K_2 x_{\mathrm{H2}}}{(1 - \xi)^2}. \tag{5.10c}$$

The Hessian, $\nabla^2 h(\lambda)$, which is real symmetric negative definite for $\xi < 1$ because $h(\lambda)$ is strictly concave, is given in Appendix F.

Given the gradient and Hessian, we may find the unique global maximum (satisfying $\nabla h(\lambda) = 0$) of the dual function using a trust region method. As in Section 2.4.3, we find $\min_\lambda(-h(\lambda))$ and use a dogleg step with the symmetric positive-definiteness of $\nabla^2(-h(\lambda))$ enabling use of Cholesky decomposition for Newton matrix inversion. Note that computation of the Hessian for the custom HCR solver is much faster than the case with finite $n_{\max}$. As we have previously shown, the optimal λ automatically satisfies the second KKT condition (2.20b) and uniquely determines the equilibrium concentrations, x by (5.5). After solving for the equilibrium concentrations, we can comprehensively compute thermodynamic properties of the HCR polymers products, which is the goal of the rest of this chapter.

5.5 HCR equilibrium properties

Growth of HCR polymers is akin to the process called living polymerization (also called reversible or equilibrium polymerization) because there is no termination step in the polymerization process [79–81]. Extensive theoretical and experimental work has been done to characterize equilibrium properties of living polymerization [81]. In particular, the mean field treatment of Dudowicz, Freed, and Douglas [82], discusses many of the general thermodynamic properties of initiated living polymers with a single monomer type. The results of Section 5.5.1 and some of Section 5.5.2 agree qualitatively with those of [82], but the fact that HCR products

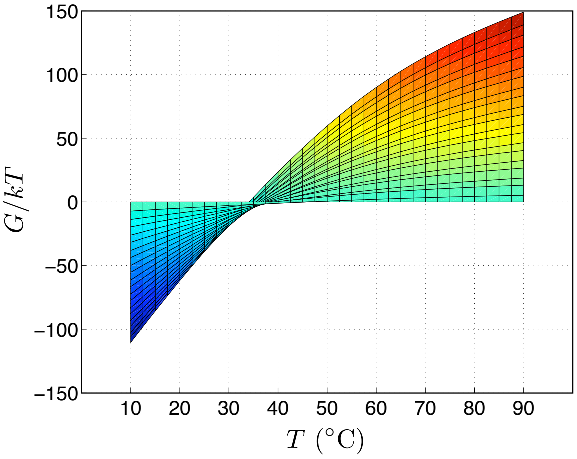


Figure 5.15: The free energy landscape of Figure 5.12a viewed along the n-axis.

contain two monomer types, only one of which may bond to initiator, gives rise to some striking differences (Section 5.5.3).

Unfortunately, solution of $\nabla h(\lambda) = 0$ is not tractable analytically, as it is for the case of a single monomer species [82], but ability to rapidly solve for equilibrium concentrations as described in Section 5.4.2 enables generation of meaningful plots. After solving for λ, the concentrations of each polymer of length n is given by (5.7), though their explicit concentrations are often not necessary because relevant properties of the system, such as the fraction of monomers in HCR polymers, average polymer length, etc., can be written in terms of x_{I}, x_{H1}, x_{H2}, ξ, and the equilibrium constants. In some special cases, we can get meaningful analytical results that describe essential features of the HCR system. In this section, we use both analytical developments and results from the solver to characterize the equilibrium behavior of the HCR system.

5.5.1 Temperature and hairpin concentration dependence of HCR

We have already investigated the temperature dependence of HCR in Section 5.3, with emphasis on effects of annealing on formation of uninitiated HCR polymer products. For initiated polymers, Figure 5.12a shows that formation of polymers becomes unfavorable at high temperatures. Indeed, viewing the landscape of Figure 5.12a along the n-axis (Figure 5.15) reveals that HCR polymers do not form above a temperature of about 37°C. The same observation applies for the uninitiated case (not shown because visualization is difficult on paper).

To quantify the temperature dependence, we plot the fraction of H1 and H2 in polymers versus temperature. The polymerization temperature, T_p, is then the temperature at which the inflection point in this curve occurs [82]. Figure 5.16 shows melt curves for the HCR system for varying ratios of initiator to H1 concentrations. The polymerization temperature is about 37°C, agreeing with that deduced from the free energy landscape, and changes little with $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$, but the melting transition becomes more broad. For constant $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ with x_{H1}^0 varying, the width of the melting transition stays constant, but the melting temperature

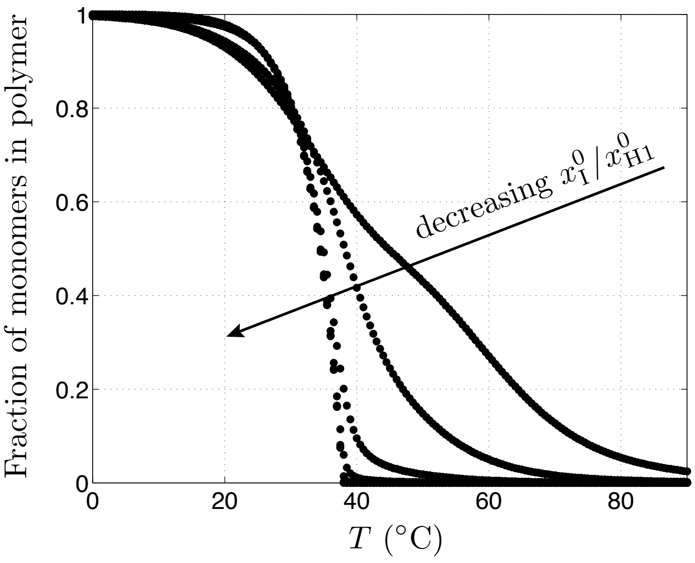


Figure 5.16: Melt profiles for the HCR system with $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$. Going from the right curve to the left, $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ varies from 10 to 10^{-5}, decreasing by a factor of 10 between each.

varies (Figure 5.17). The polymerization temperature grows nearly linearly with the logarithm of the hairpin concentration (Figure 5.18).

Polymerization has strong dependence both on temperature and hairpin concentration. As we found T_p for a given hairpin concentration, we can similarly find a critical monomer concentration $[\mathrm{H1}]_0^*$ for a given temperature. We therefore generate curves analogous to those in Figure 5.17 with monomer concentration varying on the x-axis (Figure 5.19). The critical H1 concentration is given by the inflection points in the curves. Figure 5.20 shows that the logarithm of $[\mathrm{H1}]_0^*$ varies close to linearly with temperature.

5.5.2 Polymer length distributions

After running the HCR solver to get λ and therefore x_{I}, x_{H1}, and x_{H2}, the probability distribution for polymer lengths can be easily calculated from (5.7). Let $p(n)$ be the equilibrium probability of a given ordered complex in a dilute solution of HCR components being an HCR polymer of length n and $p(\mathrm{H1})$ and $p(\mathrm{H2})$ be the probability of it being a lone H1 and H2 strand, respectively. Then,

$$p(\mathrm{H1}) = \frac{x_{\mathrm{H1}}}{X},$$

$$p(\mathrm{H2}) = \frac{x_{\mathrm{H2}}}{X},$$

$$p(n) = \begin{cases} \dfrac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}\, \xi^{\frac{n-1}{2}}}{X} = \dfrac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}}{\xi^{\frac{1}{2}}\, X}\, e^{-n/\zeta} & n \text{ odd} \\ \dfrac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}\, \xi^{\frac{n-2}{2}}}{X} = \dfrac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}}{\xi\, X}\, e^{-n/\zeta} & n \text{ even,} \end{cases} \tag{5.11}$$

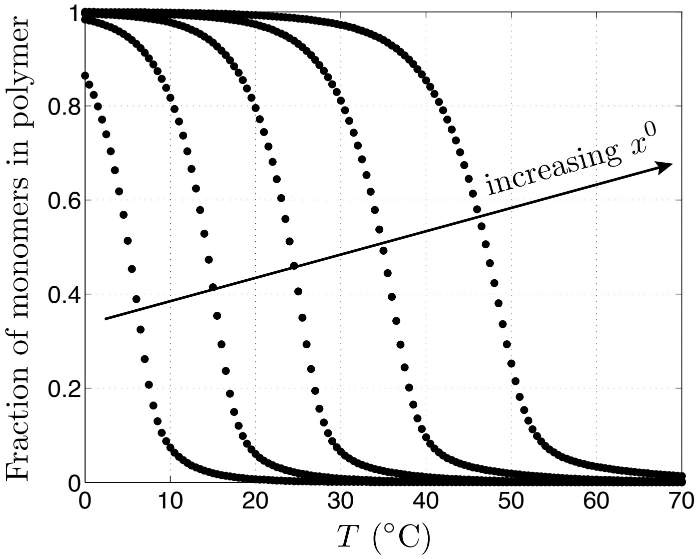


Figure 5.17: Melt profiles for the HCR system with $x^0_{\mathrm{H1}} = x^0_{\mathrm{H2}}$ and $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}} = 0.1$. Going from the left curve to the right, $[\mathrm{H1}]_0$ varies from 10^{-3} to 10 μM, increasing by a factor of 10 between each.

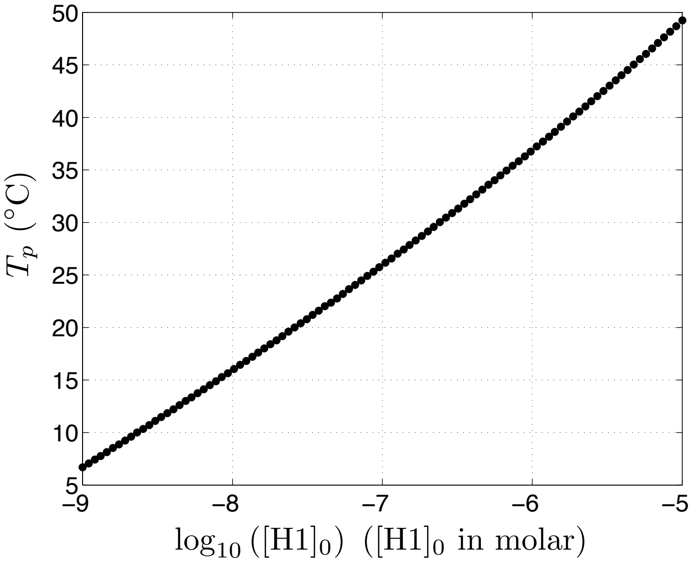


Figure 5.18: Polymerization temperature T_p versus H1 concentration for $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}} = 0.1$.

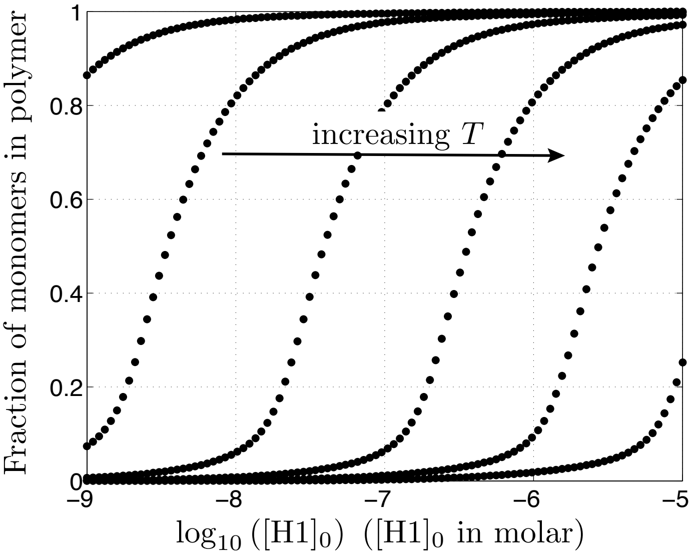


Figure 5.19: Fraction of monomers in HCR polymers for $x^0_{\mathrm{H1}} = x^0_{\mathrm{H2}}$ and $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}} = 0.1$ versus H1 concentration. Going from left to right, the temperature increases in 10-degree increments from $T = 0$ to $T = 50°$C.

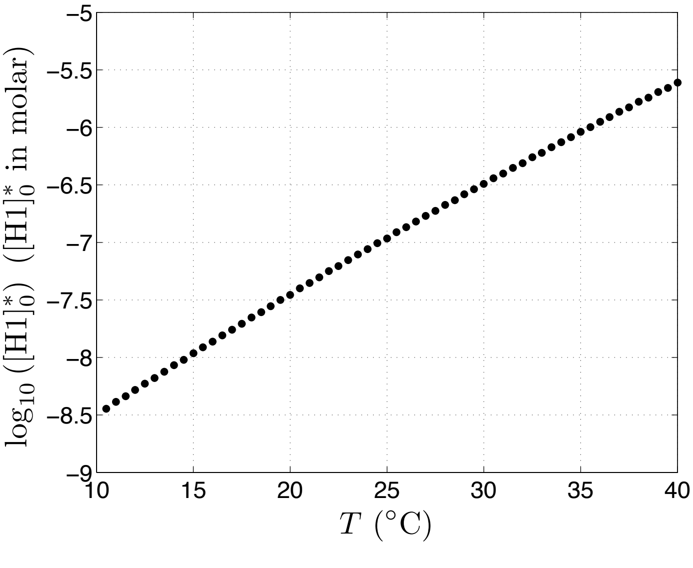


Figure 5.20: The critical H1 concentration $[\mathrm{H1}]^*_0$ versus temperature for $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}} = 0.1$.

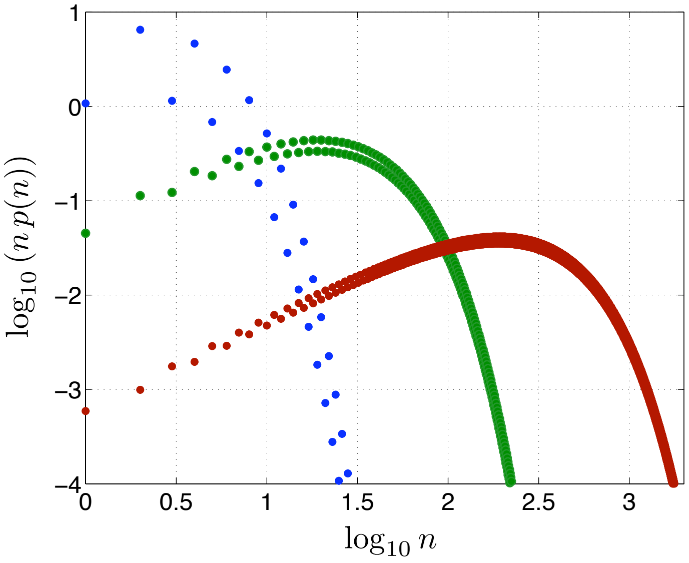


Figure 5.21: Number-weighted polymer length distributions for the HCR system for $T = 23°\mathrm{C}$ and $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$ and $[\mathrm{I}]_0/[\mathrm{H1}]_0 = 1,\ 0.1,\ 0.01$ for the blue, green, and red curves, respectively. Both scales are logarithmic. The equilibrium concentrations of H1 and H2 are not included on the plot.

with X being the total concentration of HCR products and H1 and H2,

$$X = x_{\mathrm{H1}} + x_{\mathrm{H2}} + \sum_{n=1}^{\infty} x_n = x_{\mathrm{H1}} + x_{\mathrm{H2}} + K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \, \frac{1 + K_2 x_{\mathrm{H2}}}{1 - \xi},$$

and $\zeta \equiv -2/\log \xi$. We see that polymers of even length and of odd length each have different exponential distributions, as evident in Figure 5.21, which shows the number-weighted distributions ($np(n)$) for various initiator concentrations. (The even/odd disparity is discussed in Section 5.5.3.) We also see that the distribution gets flatter with a maximum at a higher n as the initiator concentration decreases. As evident by (5.11), the shortest polymers have the greatest probability since $\xi < 1$.

Using (5.11), we can derive the moments of the distribution. The mth moment is

$$\begin{aligned} \langle n^m \rangle &= \frac{1}{X}\left(x_{\mathrm{H1}} + x_{\mathrm{H2}} + \sum_{n=1}^{\infty} n^m \, x_n \right) \\ &= \frac{1}{X}\left[x_{\mathrm{H1}} + x_{\mathrm{H2}} + 2^m \, K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \left(\Phi(\xi, -m, 1/2) + K_2 x_{\mathrm{H2}} \, \Phi(\xi, -m, 1) \right) \right], \end{aligned}$$

where $\Phi(z, s, \alpha)$ is the Lerch transcendent, which has the property that

$$\Phi(z, s-1, \alpha) = \alpha \Phi(z, s, \alpha) + z \, \frac{d}{dz} \, \Phi(z, s, \alpha).$$

Given that $\Phi(\xi, 0, 1/2) = \Phi(\xi, 0, 1) = (1 - \xi)^{-1}$, we can compute all the moments, successively computing

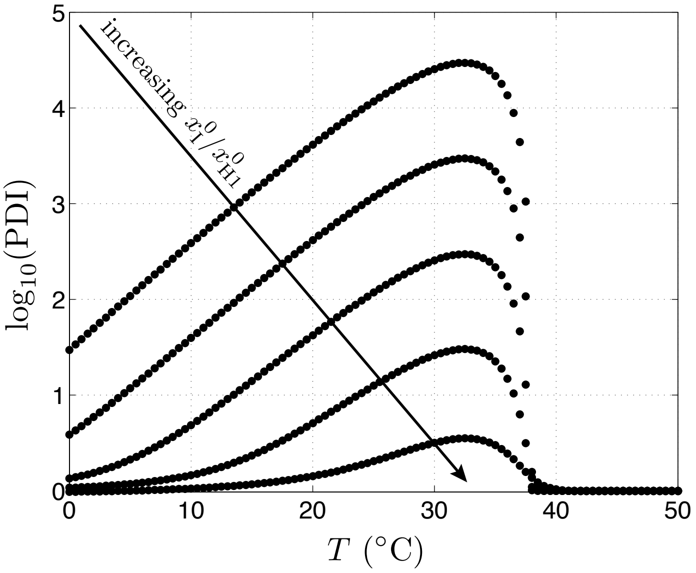


Figure 5.22: The polydispersity index (PDI) versus temperature for $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$. The value of $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ goes from 10^{-5} (top curve) to 0.1, increasing by a factor of 10 between each curve.

the mth from the $m-1$st. In particular,

$$\langle n\rangle = \frac{1}{X}\left(x_{\mathrm{H1}} + x_{\mathrm{H2}} + K_{\mathrm{I}}x_{\mathrm{I}}x_{\mathrm{H1}}\,\frac{1+\xi+2K_2x_{\mathrm{H2}}}{(1-\xi)^2}\right) \tag{5.12}$$

$$\text{and}\quad \left\langle n^2\right\rangle = \frac{1}{X}\left(x_{\mathrm{H1}} + x_{\mathrm{H2}} + K_{\mathrm{I}}x_{\mathrm{I}}x_{\mathrm{H1}}\,\frac{1+6\xi+\xi^2+4K_2x_{\mathrm{H2}}(1+\xi)}{2(1-\xi)^3}\right),$$

from which the equilibrium polydispersity index (PDI), $\left\langle n^2\right\rangle / \langle n\rangle^2$, may be calculated.

In addition to the mean polymer length $\langle n\rangle$, which includes three species for $n=1$ (H1, H2, and I-H1), we also define ℓ to be the average polymer length where the lone H1 and H2 monomers are excluded.

$$\ell = \frac{1+\xi+2K_2x_{\mathrm{H2}}}{(1+K_2x_{\mathrm{H2}})(1-\xi)}. \tag{5.13}$$

Figure 5.22 shows the PDI versus temperature for various $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ with $x_{\mathrm{H1}}^0 = x_{\mathrm{H2}}^0$. Of course, the PDI is close to unity above T_p because the solution consists of mainly lone monomers. The PDI peaks at a temperature just less than the polymerization temperature ($T_p = 37$°C) and then decays toward unity as temperature decreases, meaning that the polymer length distribution gets increasingly sharp at lower temperature. In the low temperature limit, where the entropic effects are minimized, we expect all monomers to be in monodisperse polymers and all initiators to be consumed. Therefore, at low temperatures, ℓ is independent of temperature and given by

$$\ell \approx \langle n\rangle \approx \frac{2x_{\mathrm{H1}}^0}{x_{\mathrm{I}}^0}, \tag{5.14}$$

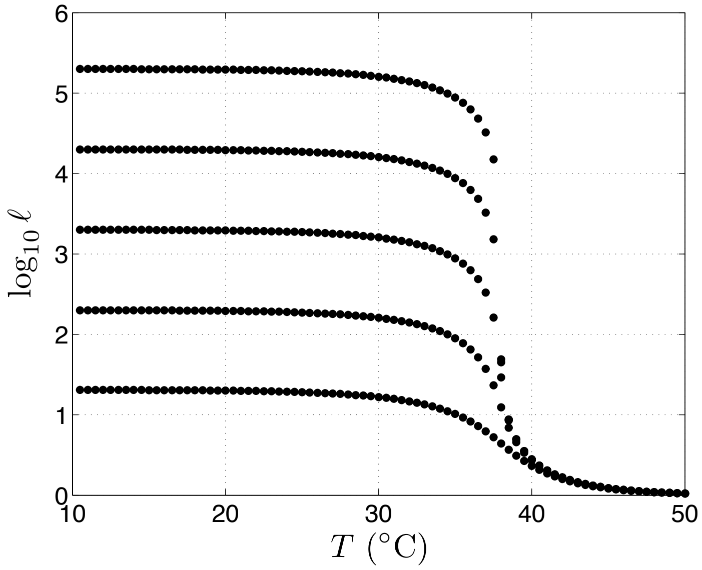


Figure 5.23: Average polymer length ℓ (neglecting lone monomers) versus temperature for $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$. The value of $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ goes from 10^{-5} (top curve) to 0.1, increasing by a factor of 10 between each curve.

where $2x_{\mathrm{H1}}^0$ is the total monomer concentration. Indeed, Figure 5.23 shows that ℓ comes to a plateau at low temperatures. Interestingly, the transition to long polymers occurs only over a few degrees, just below the melting temperature. Figure 5.24 shows that the calculated low-temperature values of ℓ corresponding to the plateaus in Figure 5.23 fall on the line given by (5.14).

5.5.3 Even-odd asymmetry in polymer lengths

As stated in Section 5.5.2, polymers of even length and odd length have different distributions. From (5.11), we see that

$$\frac{x_n}{x_{n-1}} = \begin{cases} K_1 x_{\mathrm{H1}} & n \text{ odd}, \;\; \geq 3 \\ K_2 x_{\mathrm{H2}} & n \text{ even}, \;\; \geq 2 \end{cases} \tag{5.15}$$

and we typically have $x_{\mathrm{H1}} \neq x_{\mathrm{H2}}$, which results in more even or odd polymers at equilibrium, depending on what values x_{H1} and x_{H2} take. For $x_{\mathrm{H1}}^0 = x_{\mathrm{H2}}^0$, $x_{\mathrm{H2}} > x_{\mathrm{H1}}$, since all polymers are initiated and only H1 can bind with the initiator. This is expressed mathematically by referring to (5.10b) and (5.10c) and noting that solution of $\nabla h(\lambda) = 0$ yields

$$\frac{x_{\mathrm{H1}}(1-\xi)^2 + K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}(1 + K_2 x_{\mathrm{H2}})}{x_{\mathrm{H2}}(1-\xi)^2 + K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}(\xi + K_2 x_{\mathrm{H2}})} = 1. \tag{5.16}$$

Because $\xi < 1$, $x_{\mathrm{H2}} > x_{\mathrm{H1}}$.

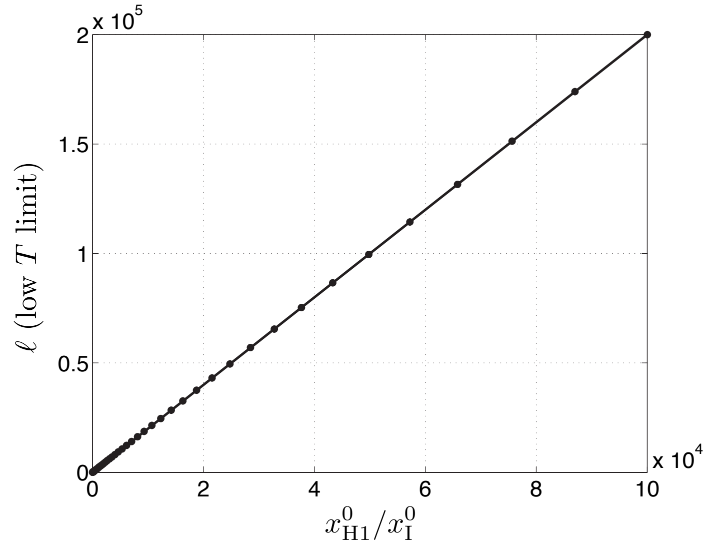


Figure 5.24: The low-temperature average polymer length ℓ (neglecting lone monomers) versus $x^0_{\mathrm{H1}}/x^0_{\mathrm{I}}$ for $[\mathrm{H1}]_0 = [\mathrm{H2}]_0 = 1\ \mu\mathrm{M}$. The line is given by (5.14).

The total fraction of H1 and H2 monomers that are in polymers with n even or odd, respectively, is

$$\phi^{\mathrm{even}} = \frac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} K_2 x_{\mathrm{H2}}}{X(1-\xi)}, \tag{5.17a}$$

$$\text{and } \phi^{\mathrm{odd}} = \frac{K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}}}{X(1-\xi)}. \tag{5.17b}$$

Similarly, the fraction of polymers with n even or odd, respectively, is

$$\phi^{\mathrm{even}}_{\ell} = \frac{K_2 x_{\mathrm{H2}}}{1 + K_2 x_{\mathrm{H2}}}, \tag{5.18a}$$

$$\text{and } \phi^{\mathrm{odd}}_{\ell} = \frac{1}{1 + K_2 x_{\mathrm{H2}}}, \tag{5.18b}$$

where the subscript ℓ refers to the fact that the lone H1 and H2 monomers are not considered, as in (5.13). Figure 5.25 shows $\phi^{\mathrm{even}}_{\ell}$ for various ratios of hairpin concentrations, corresponding to a range of $\pm 10\%$ pipette errors that one might expect experimentally. As the ratio of initiator to hairpin concentrations decreases, there can be a qualitative shift in even versus odd dominance of the resulting HCR products at equilibrium. While not in quantitative agreement, the gel in Figure 5.2 shows a change in even dominance, evident in the lower of the double-bands being darker for larger $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}}$ and the higher being darker for smaller $x^0_{\mathrm{I}}/x^0_{\mathrm{H1}}$. The even-odd asymmetry in HCR products went unnoticed until the calculations in this section were completed, another example of theory and computation complementing experiment.

The preference for polymers ending in either H1 or H2 is a result of the inherent asymmetry in the system arising from the fact that only H1 may bind to the initiator. If both H1 and H2 could bind to the initiator, the

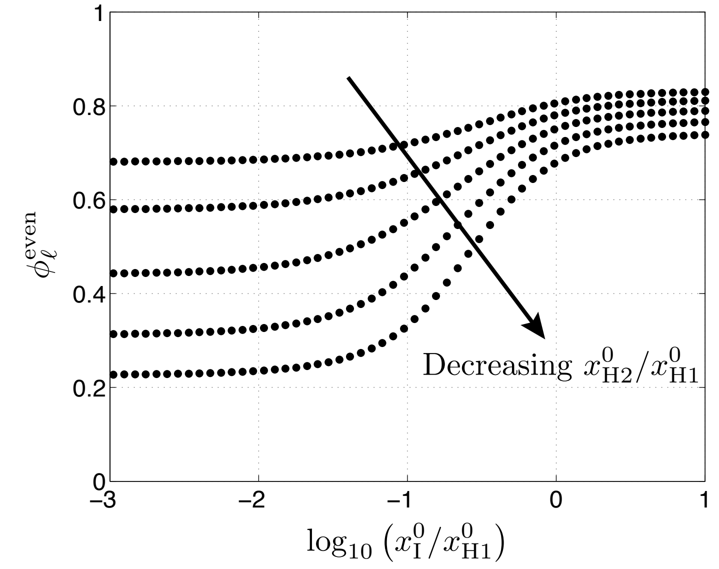


Figure 5.25: ϕ_ℓ^{even} versus $x_{\text{I}}^0/x_{\text{H1}}^0$ for various $x_{\text{H2}}^0/x_{\text{H1}}^0$ at $[\text{H1}]_0 = 1$ μM and $T = 23$°C. The bottom curve has $x_{\text{H2}}^0/x_{\text{H1}}^0 = 0.9$ and the top has $x_{\text{H2}}^0/x_{\text{H1}}^0 = 1.1$, with 0.05 increments between each curve. The middle curve, for which $x_{\text{H2}}^0/x_{\text{H1}}^0 = 1$, has a limiting value of $\phi_\ell^{\text{even}} < 0.5$ as $x_{\text{I}}^0/x_{\text{H1}}^0$ gets small because $K_1 \neq K_2$ for the real system.

asymmetry no longer exists, and the preference for even or odd polymer lengths vanishes for $K_1 = K_2$. The resulting system behaves as a standard initiated living polymer where each H1/H2 pair is a "monomer."

5.5.4 The limit of long HCR polymer lengths

We see from (5.12) and (5.13) that as the average polymer length gets long, which is the case when $x_{\text{I}}^0/x_{\text{H1}}^0$ gets small (see (5.14) and Figures 5.23 and 5.24), ξ approaches unity. The difference between x_{H1} and x_{H2} approaches zero by (5.16) and the discrepancy between even and odd polymers is then diminished if $K_1 \approx K_2$ (see (5.15)). In this limit, $x_{\text{H1}} \approx x_{\text{H2}} \lesssim K_1^{-1} \approx K_2^{-1}$, and the probability distribution of polymer lengths (not considering the individual monomers in the distribution) is given by (*cf.* (5.11))

$$p_\ell(n) = \left(1 - \sqrt{\xi}\right)\left(\sqrt{\xi}\right)^{n-1}.$$

This matches the distribution of uninitiated living polymerization in which there is only one type of monomer [83] (and the evenness and oddness of n is not relevant). Here, $\sqrt{\xi} \approx K_1 x_{\text{H1}} \approx K_2 x_{\text{H2}}$ is the equilibrium probability that a given monomer is alone in solution (i.e., not in a polymer), which helps clarify the physical meaning of ξ.

HCR System 1 (published in [1])	
I	AGTCTAGGATTCGGCGTGGGTTAA
H1	TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG
H2	AGTCTAGGATTCGGCGTGGGTTAACACGCCGAATCCTAGACTACTTTG

HCR System 2 (unpublished)	
I	GCCGACCACTACCAGCAGAACACC
H1	GGTGTTCTGCTGGTAGTGGTCGGCCCAGACGCCGACCACTACCAGCAG
H2	GCCGACCACTACCAGCAGAACACCCTGCTGGTAGTGGTCGGCGTCTGG

Table 5.1: The sequences for the two HCR systems studied. Subsequences a, b, and c are colored red, blue, and green, respectively. Underlined subsequences represent those marked with asterisks.

5.6 Sequence dependence on HCR properties

We have already seen in Section 5.3.3 that an adjustment in the general design of H1 and H2 can lead to qualitatively different behavior of the HCR system. Here, we demonstrate sequence dependence of the shape of the free energy landscape, the polymerization temperature, and the critical hairpin concentration. In addition to the published HCR system (which we will call "HCR system 1") whose sequences are shown in Figure 5.1 and reproduced in Table 5.1 for reference, we analyzed another unpublished system in use in the Pierce Lab [84]. The sequences for this system ("HCR system 2") are shown in Table 5.1. HCR system 1 has 50% GC content, while HCR system 2 has 64%. We therefore expect higher melting temperatures and a lower critical hairpin concentration for system 2.

Figure 5.26 shows the free energy landscape for HCR system 2 in the absence of initiator next to that of HCR system 1 for reference. The barrier for uninitiated HCR polymerization is lower for HCR system 2 because the energy penalty of having closed hairpin loops relative to the benefit of base pairing upon hairpin binding is smaller when there is more GC content. Furthermore, the landscape is steeper as HCR proceeds, indicative of the greater energy benefit per base pair.

Figure 5.27 shows the comparisons of T_p and $[\mathrm{H1}]_0^*$ for the two HCR systems. The polymerization temperature is higher for HCR system 2, and the critical monomer concentration is lower, as expected from the higher GC content.

5.7 Further studies

The preliminary experimental results and analytical and computational analyses presented in this chapter complement each other and have served to clarify many of the physical principles governing the behavior of the HCR system. This merits further study of this system, both experimentally and theoretically, to fully characterize its fundamental features and provide a framework on which to design new HCR-based applications. Possible studies include systematic variation of the length of the subsequences a, b, and c (which was done

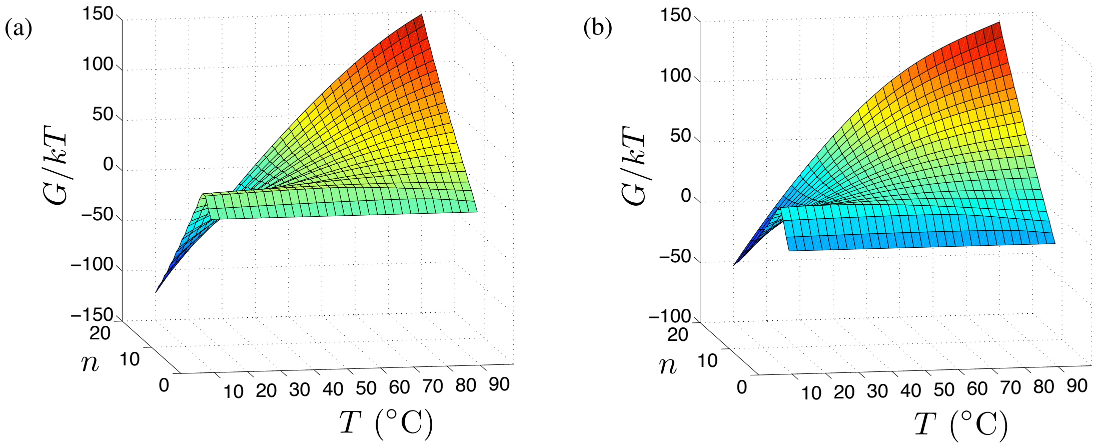


Figure 5.26: (a) Free energy landscape of HCR system 2 in the absence of initiator, generated following the prescription of Section 5.3.4. (b) Reproduction of the landscape for HCR system 1 from Figure 5.12a for comparison.

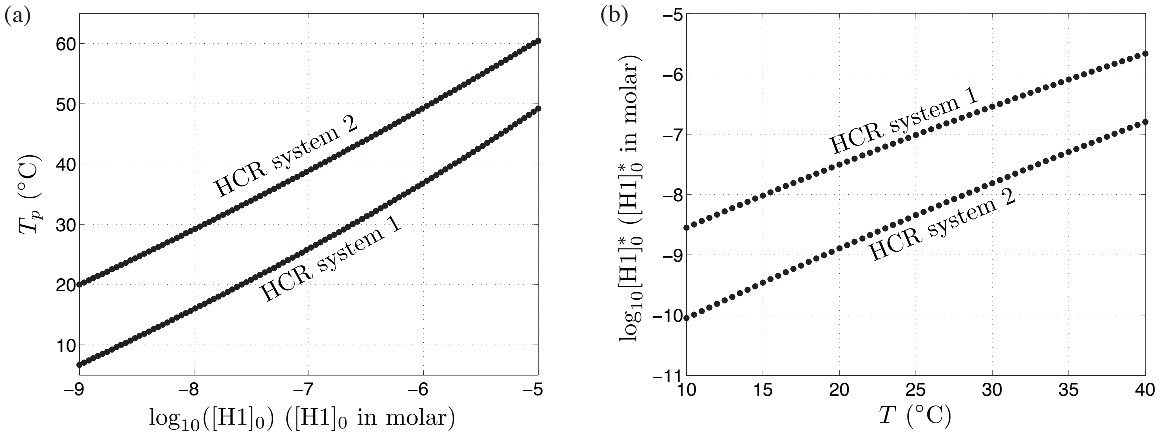


Figure 5.27: (a) The polymerization temperature T_p versus $[\mathrm{H1}]_0$ for HCR system 1 (bottom curve) and for HCR system 2 (top curve). (b) The critical monomer concentration $[\mathrm{H1}]_0^*$ versus temperature for HCR system 1 (top curve) and for HCR system 2 (bottom curve). All curves have $[\mathrm{H1}]_0 = [\mathrm{H2}]_0$ and $x_\mathrm{I}^0/x_\mathrm{H1}^0 = 0.1$.

in the ring study of Section 5.3.3) and of their respective GC content. With this expertise, the HCR analysis tools can aide sequence design of new systems.

The analysis in this chapter contained commentary on kinetic processes, but the quantitative results all deal with the thermodynamics of HCR systems. The kinetics of HCR chain growth could provide a rich array of phenomena to explore, albeit requiring more sophisticated experimental techniques. Indeed, the growth of living polymers tends to proceed in phases that are qualitatively different from the equilibrium state. Typically, the polymer length distribution is first a sharp Poisson distribution before more slowly relaxing to the broad exponential distribution like those in (5.11) [85]. In fact, low observed polydispersity at short timescales is often an indicator of a living (i.e., unterminated) polymerization process [79].

Kinetic effects may also be the cause that, for low initiator concentrations, experimental results based on gel electrophoresis show that essentially none of the H1 and H2 monomers are converted to HCR polymers [86]. Furthermore, the onset of this phenomenon occurs abruptly as $x_{\mathrm{I}}^0/x_{\mathrm{H1}}^0$ is decreased. The thermodynamics-based calculations of this chapter do not predict this phenomenon, as thermodynamic melt properties are independent of initiator concentration (see, e.g., Figure 5.16). Kinetic studies, both theoretical and experimental, are needed to explain this phenomenon and to complement the thermodynamic analysis presented in this chapter.

Chapter 6

Summary and outlook

The theory and algorithms presented in this thesis provide new tools for analyzing the equilibrium properties of interacting nucleic acid strands in synthetic and biological systems. For an ordered complex of an arbitrary number of strands, it is now possible to calculate the partition function over all unpseudoknotted connected secondary structures. The approach rigorously treats distinguishability effects that arise in the multi-stranded setting and provides a basis for the analysis of dilute solutions containing multiple ordered complexes at equilibrium.

If the number of strands and complexes is small, the partition function can be used to calculate the equilibrium population distribution of each complex species, useful in interpreting long-time results from stochastic kinetic simulations of small numbers of molecules. Alternatively, if the number of strands is large, the equilibrium concentration of each ordered complex species can be determined by minimizing the free energy of the system subject to conservation of mass. This is a (high-dimensional) strictly convex programming problem; strong duality and the special form of the KKT conditions imply that we may instead solve the (low-dimensional) unconstrained strictly concave dual problem, leading to an efficient solution framework with uniqueness and global convergence guarantees. Partition function information can then be used to calculate base-pairing expectations for individual complexes or for the system as a whole.

Software for performing all of these calculations is available for download (for research purposes) at `nupack.org`. The NUPACK web application provides researchers worldwide a user-friendly interface for using the software. The results are clearly presented in downloadable publication-quality graphics which provide a concise, yet thorough, presentation of the most relevant properties.

The analysis tools of NUPACK have been extended to study the HCR system. Because the free energy change for addition of a monomer to an HCR polymer is independent of polymer length, we can calculate the free energy for an HCR product of arbitrary length, which allows thermodynamic characterization of the system including polymers of all lengths. The thermodynamics of HCR are similar to those of living polymers, with the striking result that polymers of even or odd length are favored, depending on the conditions—shown computationally and also seen experimentally. Computational study of the HCR system also reveals that the hairpins are reusable, subsequently demonstrated experimentally.

The theoretical and computational study of the HCR system illustrates the utility of the analysis tools described in this thesis in interpreting and directing experiments. NUPACK makes this process easy, and it has already become a staple tool among experimentalists and theorists in the Pierce Lab for design and construction of nucleic acid-based devices. As its development continues and it expands to include improved thermodynamics (such as inclusion of two-stranded pseudoknots), design, and kinetics, it will continue to be a powerful tool for biologists and biotechnologists working with nucleic acids.

Bibliography

[1] R. M. Dirks, N. A. Pierce, Triggered amplification by hybridization chain reaction, *Proc. Natl. Acad. Sci. USA* **101**, 15275-15278 (2004).

[2] J. Watson, et al., *Molecular Biology of the Gene* (Benjamin Cummings, San Francisco, 2004), 5th ed.

[3] B. Alberts, et al., *Molecular Biology of the Cell* (Garland Science, New York, 2002), 4th ed.

[4] N. C. Seeman, P. S. Lukeman, Nucleic acid nanostructures: bottom-up control of geometry on the nanoscale, *Reports Prog. Phys.* **68**, 237-270 (2005).

[5] N. C. Seeman, From genes to machines: DNA nanomechanical devices, *Trends Biochem. Sci.* **30**, 119-125 (2005).

[6] F. C. Simmel, W. U. Dittmer, DNA nanodevices, *Small* **1**, 284-299 (2005).

[7] J. Bath, A. J. Turberfield, DNA nanomachines, *Nature Nanotechnology* **2**, 275–284 (2007).

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

[9] M. Zuker, Calculating nucleic acid secondary structure, *Curr. Opin. Struc. Biol.* **10**, 303–310 (2000).

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

[11] H. Isambert, E. D. Siggia, Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribosome, *Proc. Natl. Acad. Sci. USA* **97**, 6515–6520 (2000).

[12] R. M. Dirks, N. A. Pierce, A partition function algorithm for nucleic acid secondary structure including pseudoknots, *J. Comput. Chem.* **24**, 1664–1677 (2003).

[13] R. M. Dirks, N. A. Pierce, An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots, *J. Comput. Chem.* **25**, 1295–1304 (2004).

[14] L. Good, Diverse antisense mechanisms and applications, *Cell. Mol. Life Sci.* **60**, 823-824 (2003).

[15] T. S. Bayer, C. D. Smolke, Programmable ligand-controlled riboregulators of eukaryotic gene expression, *Nat. Biotechnol.* **23**, 337-343 (2005).

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

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

[18] T. L. Hill, *An Introduction to Statistical Thermodynamics* (Dover, New York, 1986).

[19] D. Chandler, *Introduction to Modern Statistical Mechanics* (Oxford University Press, New York, 1987).

[20] D. A. McQuarrie, *Statistical Mechanics* (University Science Books, Sausalito, CA, 2000).

[21] E. T. Jaynes, The Gibbs Paradox, *Maximum Entropy and Bayesian Methods*, C. R. Smith, G. J. Erickson, P. O. Neudorfer, eds. (Kluwer Academic Publishers, Dordrecht, Holland, 1992), pp. 1–22.

[22] R. H. Swendsen, Statistical mechanics of classical systems with distinguishable particles, *J. Stat. Phys.* **107**, 1143–1166 (2002).

[23] J. F. Nagle, Regarding the entropy of distinguishable particles, *J. Stat. Phys.* **117**, 1047–1062 (2004).

[24] L. D. Landau, E. M. Lifshitz, *Statistical Physics Part 1* (Butterworth-Heinemann, New York, 1980), 3rd ed.

[25] J. A. Gallian, *Contemporary Abstract Algebra* (Houghton Mifflin, Boston, 2002), 5th ed.

[26] C. A. Charalambides, *Enumerative Combinatorics* (Chapman & Hall/CRC, Boca Raton, FL, 2002).

[27] W. R. Smith, R. W. Missen, *Chemical Reaction Equilibrium Analysis: Theory and Algorithms* (John Wiley & Sons, New York, 1982).

[28] S. Boyd, L. Vandenberghe, *Convex Optimization* (Cambridge University Press, Cambridge, 2004).

[29] J.-B. Hiriart-Urruty, C. Lemaréchal, *Convex Analysis and Minimization Algorithms I* (Springer, New York, 1993).

[30] J. Nocedal, S. J. Wright, *Numerical Optimization* (Springer, New York, 1999).

[31] G. A. Shultz, R. B. Schnabel, R. H. Byrd, A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties, *SIAM J. Numer. Anal.* **22**, 47-67 (1985).

[32] L. N. Trefethen, I. David Bau, *Numerical Linear Algebra* (SIAM, Philadelphia, 1997).

[33] L. Lovász, *Combinatorial Problems and Exercizes* (Elsevier, Amsterdam, 1993).

[34] I. Tinoco, Jr., O. C. Uhlenbeck, M. D. Levine, Estimation of secondary structure in ribonucleic acids, *Nature* **230**, 362-367 (1971).

[35] J. SantaLucia, Jr., A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics, *Proc. Natl. Acad. Sci. USA* **95**, 1460-1465 (1998).

[36] D. H. Mathews, J. Sabina, M. Zuker, D. H. Turner, Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure, *J. Mol. Biol.* **288**, 911-940 (1999).

[37] M. N. Stojanovic, D. Stefanovic, A deoxyribozyme-based molecular automaton, *Nat. Biotechnol.* **21**, 1069-1074 (2003).

[38] J.-S. Shin, N. A. Pierce, A synthetic DNA walker for molecular transport, *J. Am. Chem. Soc.* **126**, 10834-10835 (2004).

[39] W. Shih, J. Quispe, G. Joyce, A 1.7-kilobase single-stranded DNA that folds into a nanoscale octahedron, *Nature* **427**, 618-621 (2004).

[40] M. Zuker, P. Stiegler, Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information, *Nucleic Acids Res.* **9**, 133-147 (1981).

[41] R. B. Lyngsø, C. N. S. Pedersen, RNA pseudoknot prediction in energy-based models, *J. Comput. Biol.* **7**, 409-427 (2000).

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

[43] E. Rivas, S. R. Eddy, A dynamic programming algorithm for RNA structure prediction including pseudoknots, *J. Mol. Biol.* **285**, 2053–2068 (1999).

[44] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, Pseudobase: structural information on RNA pseudoknots, *Nucleic Acids Res.* **29**, 194-195 (2001).

[45] E. Winfree, F. Liu, L. A. Wenzler, N. C. Seeman, Design and self-assembly of two-dimensional DNA crystals, *Nature* **394**, 539-544 (1998).

[46] A. J. Turberfield, et al., DNA fuel for free-running nanomachines, *Phys. Rev. Lett.* **90**, 118102 (2003).

[47] M. S. Waterman, T. F. Smith, RNA secondary structure: a complete mathematical analysis, *Math. Biosci.* **42**, 257-266 (1978).

[48] R. Nussinov, J. R. Pieczenik, J. R. Griggs, D. J. Kleitman, Algorithms for loop matchings, *SIAM J. Appl. Math.* **35**, 68-82 (1978).

[49] R. B. Lyngsø, M. Zuker, C. N. S. Pedersen, Fast evaluation of internal loops in RNA secondary structure prediction, *Bioinformatics* **15**, 440-445 (1999).

[50] Y. Ding, C. E. Lawrence, A statistical sampling algorithm for RNA secondary structure prediction, *Nucleic Acids Res.* **31**, 7280-7301 (2003).

[51] D. H. Mathews, Using an RNA secondary structure partition function to determine confidence in base pairs predicted by free energy minimization, *RNA* **10**, 1178-1190 (2004).

[52] R. Penchovsky, R. R. Breaker, Computational design and experimental validation of oligonucleotide-sensing allosteric ribozymes, *Nat. Biotechnol.* **23**, 1424-1433 (2005).

[53] V. Patzel, et al., Design of siRNAs producing unstructured guide-RNAs results in improved RNA interference efficiency, *Nat. Biotechnol.* **23**, 1440-1444 (2005).

[54] V. A. Bloomfield, D. M. Crothers, I. Tinoco, Jr., *Nucleic Acids: Structures, Properties, and Functions* (University Science Books, Sausalito, CA, 2000).

[55] M. Doi, S. F. Edwards, *The Theory of Polymer Dynamics* (Oxford University Press, New York, 1986).

[56] P.-G. de Gennes, Statistics of branching and hairpin helices for the dAT polymer, *Biopolymers* **6**, 715–729 (1968).

[57] M. Müller, Statistical physics of RNA folding, *Phys. Rev. E* **67**, 021914 (2003).

[58] B. H. Zimm, W. H. Stockmayer, The dimensions of chain molecules containing branches and rings, *J. Chem. Phys.* **17**, 1301–1314 (1949).

[59] H. Yamakawa, *Helical Wormlike Chains in Polymer Solutions* (Springer-Verlag, Berlin, 1997).

[60] M. C. Murphy, I. Rasnik, W. Cheng, T. M. Lohman, T. Ha, Probing single-stranded DNA conformational flexibility using fluorescence spectroscopy, *Biophys. J.* **86**, 2530–2537 (2004).

[61] S. B. Smith, Y. J. Cui, C. Bustamante, The overstretching of B-DNA: the elastic response of individual double-stranded and single-stranded DNA molecules, *Science* **271**, 795-799 (1996).

[62] J. N. Onuchic, Z. Luthey-Schulten, P. G. Wolynes, Theory of protein folding: the energy landscape perspective, *Annu. Rev. Phys. Chem.* **48**, 545–600 (1997).

[63] S.-J. Chen, K. A. Dill, RNA folding energy landscapes, *Proc. Natl. Acad. Sci. USA* **97**, 646–651 (2000).

[64] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, *J. Comput. Phys.* **22**, 403–434 (1976).

[65] D. T. Gillespie, Exact stochastic simulation of coupled chemical reactions, *J. Phys. Chem.* **81**, 2340–2361 (1977).

[66] W. Zhang, S.-J. Chen, RNA hairpin-folding dynamics, *Proc. Natl. Acad. Sci. USA* **99**, 1931–1936 (2002).

[67] J. M. Schaeffer, J. S. Bois, R. M. Dirks, N. A. Pierce, E. Winfree, Kinetic analysis of multi-stranded nucleic acid systems (unpublished) .

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

[69] I. L. Hofacker, Vienna RNA secondary structure server, *Nucleic Acids Res.* **31**, 3429–3431 (2003).

[70] R. Thomas, Recherches sur la denaturation des acides desoxyribonucleiques, *Biochim. Biophys. Acta* **14**, 231–240 (1954).

[71] B. A. Shapiro, J. Maizel, L. E. Lipkin, K. Currey, C. Whitney, Generating non-overlapping displays of nucleic acid secondary structure, *Nucleic Acids Res.* **12**, 75–88 (1984).

[72] R. E. Bruccoleri, G. Heinrich, An improved algorithm for nucleic acid secondary structure display, *Comput. Applic. Biosci.* **4**, 167–173 (1988).

[73] G. Muller, C. Gaspin, A. Etienne, E. Westhof, Automatic display of RNA secondary structures, *Comput. Applic. Biosci.* **9**, 551–561 (1993).

[74] N. R. Kallenbach, R.-I. Ma, N. C. Seeman, An immobile nucleic acid junction constructed from oligonucleotides, *Nature* **305**, 829–831 (1983).

[75] N. A. Pierce (2007). Personal communication.

[76] H. Jacobson, W. H. Stockmayer, Intramolecular reaction in polycondensations. 1. the theory of linear systems, *J. Chem. Phys.* **18**, 1600–1606 (1950).

[77] J. W. Little, An exonuclease induced by bacteriophage λ. II. nature of the enzymatic reaction, *J. Biol. Chem.* **242**, 679–686 (1967).

[78] I. Berkower, J. Leis, J. Hurwitz, Isolation and characterization of an endonuclease from *Escherichia coli* specific for ribonucleic acid in ribonucleic acid–deoxyribonucleic acid hybrid structures, *J. Biol. Chem.* **248**, 5914–5921 (1973).

[79] G. C. Odian, *Principles of Polymerization* (John Wiley and Sons, New York, 1991), 3rd ed.

[80] S. C. Greer, Physical chemistry of equilibrium polymerization, *J. Phys. Chem. B* **102**, 5413–5422 (1998).

[81] S. C. Greer, Reversible polymerizations and aggregations, *Annu. Rev. Phys. Chem.* **53**, 173–200 (2002).

[82] J. Dudowicz, K. F. Freed, J. F. Douglas, Lattice model of living polymerization. I. basic thermodynamic properties, *J. Chem. Phys.* **111**, 7116–7130 (1999).

[83] P. J. Flory, *Principles of Polymer Chemistry* (Cornell University Press, Ithaca, New York, 1953).

[84] J. E. Padilla (2007). Personal communication.

[85] S. S. Das, et al., Living poly(α-methylstyrene) near the polymerization line. VII. molecular weight distribution in a good solvent, *J. Chem. Phys.* **111**, 9406–9417 (1999).

[86] R. M. Dirks (2006). Personal communication.

[87] G. Strang, *Linear Algebra and its Applications* (Thompson Learning, Inc., Toronto, 1988), 3rd ed.

Appendix A

Convexity of ideal gas free energy

We prove here that the function $g(m)$ given by (2.26) and (2.27a) in Section 2.5.2 is convex.

A function is convex if its Hessian is positive semidefinite [28]. The Hessian of $g(m)$ is given by

$$\left[\nabla_m^2 g(m)\right]_{kl} = \begin{cases} \frac{1}{m_k} - \frac{1}{M} & \text{for } k = l \\ \frac{1}{M} & \text{for } k \neq l \end{cases} . \tag{A.1}$$

A matrix is positive semidefinite if all principle submatrices have nonnegative determinants [87], where a principle submatrix is formed by deleting the same number of rows as columns from the matrix. Any $n \times n$ principle submatrix of $\nabla_m^2 g(m)$, H^{sub}, has the form

$$H_{ij}^{\text{sub}} = \begin{cases} \frac{1}{m_i} - \frac{1}{M^{\text{sub}}} & \text{for } i = j \\ \frac{1}{M^{\text{sub}}} & \text{for } i \neq j \end{cases} ,$$

where $M^{\text{sub}} \leq \sum_{i=1}^{n} m_i$ with equality when $H^{\text{sub}} = \nabla_m^2 g(m)$. Here, we have relabeled the subscripts on the m_i such that they are sequential from 1 to n, i.e., the subscripts may not refer to the same compound as that in (A.1). This is inconsequential for this proof, for it is only the form of H^{sub} described above that is important. To compute the determinant, we perform the following operations (which do not affect the determinant) to get H^{sub} in upper triangular form. Starting with row n and working up to row 2, from each row i subtract row $(i-1)$. As a result, row 1 stays the same and the resulting entries for $i \geq 2$ are

$$H_{ij}^{\text{sub}} \sim \begin{cases} \frac{1}{m_i} - \frac{1}{M^{\text{sub}}} & \text{for } i = j \\ -\frac{1}{m_{i-1}} & \text{for } i = j+1 \\ 0 & \text{otherwise} \end{cases} ,$$

where $\sim$ denotes equivalence for matrices with equal determinants. Now, starting with column $n-1$ and working left to column 1, to each column j add the product of m_{j+1}/m_j and column $(j+1)$. For $i \geq 2$, this gives

$$H_{ij}^{\text{sub}} \sim \begin{cases} \frac{1}{m_i} & \text{for } i = j \\ 0 & \text{otherwise} \end{cases} .$$

The entries in row 1 are given by

$$H_{1j}^{\text{sub}} \sim \begin{cases} \frac{M^{\text{sub}} - \sum_{k=1}^{n} m_k}{m_1\, M^{\text{sub}}} & \text{for } j = 1 \\ -\frac{\sum_{k=j}^{n} m_k}{m_j\, M^{\text{sub}}} & \text{for } 2 \leq j \leq n-1 \\ -\frac{1}{M^{\text{sub}}} & \text{for } j = n \end{cases} .$$

We now have an upper triangular matrix whose determinant is the product of the diagonal entries,

$$\det(H^{\text{sub}}) = \frac{M^{\text{sub}} - \sum_{j=1}^{n} m_j}{m_1\, M^{\text{sub}}} \prod_{j=2}^{n} \frac{1}{m_j}.$$

Thus, the determinate of all principle submatrices are nonnegative with the determinant of the whole Hessian being zero. Thus, $\nabla_m^2 g(m)$ is positive semidefinite and $g(m)$ is convex. □

Appendix B

Proof of the Representation Theorem

Following is a proof of the Representation Theorem, reproduced from [17]. It was developed by Robert Dirks with an alternate derivation developed independently by Joseph Schaeffer and Erik Winfree. The theorem states that for L interacting strands, each labeled with a unique identifier, for every unpseudoknotted connected secondary structure s, there is exactly one circular permutation π that yields a polymer graph with no crossing lines.

The proof is by induction on the number of strands in the complex, L. The theorem holds for $L = 1$ since there is only one circular permutation. We now assume that the theorem holds for $L - 1$ strands and attempt to show that it holds for L strands.

Let s be an unpseudoknotted connected secondary structure of L strands so that there must exist a circular permutation π for which the polymer graph has no crossing lines. We create a connectivity graph for s in which each strand is represented by exactly one node and there is an edge between nodes if there exists at least one base pair between the corresponding strands. Since s is connected, the connectivity graph must have either a leaf node or a node that is part of a cycle whose removal will not break the connectedness of the resulting graph. Let l be some such node and let s' be the secondary structure which has had the corresponding strand removed. Then s' is a connected secondary structure of $L - 1$ strands. By supposition, the circular permutation π' of the strands in s' that corresponds to π (omitting strand l) is the only one that yields a polymer graph with no crossing lines. Hence, the only polymer graphs for structure s that have the possibility of no crossing lines are those that are obtained by inserting strand l between two strands of s' in circular permutation π'.

Now we show that the only position where strand l can be added back into the polymer graph with circular permutation π' without introducing crossing lines is the original position n corresponding to circular permutation π. Consider inserting l into π' at positions $m \neq n$. A line drawn from m to n must cross some base pair $i \cdot j$ in the polymer graph or s' would not be connected. In the original strand ordering π for structure s, there must exist a base pair $d \cdot e$ connecting strand l to another strand in the complex. This base pair $d \cdot e$ cannot cross $i \cdot j$, so both d and e are on the n side of $i \cdot j$. If we now insert strand l at a position m, crossing lines are produced because one end of $d \cdot e$ is on the m side of $i \cdot j$ and the other is on the n side. This implies that n is the only position where l can be added back to s' without introducing crossing lines. Hence, the original polymer graph corresponding to circular permutation π is the only one without crossing lines. □

Appendix C

Pseudocode for construction of the set Λ

```
// Given $\Psi^0$, $\Psi$ (or $\Psi^{\text{super}}$, which can be substituted for $\Psi$), $A$, $m^0$, find the set $\Lambda$
// $\Lambda$ is stored as a 2D array where $\Lambda_{k,j}$ is the number of ordered complex (or complex) $j$ in population $k$

Sort $A$ such that columns corresponding to single-stranded complexes are first

// Calculate maximum number of each complex type based on exhausting the limiting single strand
Set $m_j^{\Psi,\max} = 0$ for all $j$  // $m^{\Psi,\max}$ contains max possible population of multistranded complexes ($|\Psi| - |\Psi^0|$ entries)
for $j = 1, |\Psi| - |\Psi^0|$
  for $i = 1, |\Psi^0|$
    if $A_{i,|\Psi^0|+j} > 0$
      $m_j^{\Psi,\max} = \max\left(m_j^{\Psi,\max}, \lfloor m_i^0 / A_{i,|\Psi^0|+j} \rfloor\right)$
    end if
  end for
end for

// Store population with all single-stranded complexes
Set $\Lambda_{i,1} = m_i^0$ for $1 \leq i \leq |\Psi^0|$ and $\Lambda_{j,1} = 0$ for $|\Psi^0| + 1 \leq j \leq |\Psi|$

// Generate entries in $\Lambda$ that have multistranded complexes
Initialize $m_j^{\Psi} = 0$ for all $j$  // $m^{\Psi}$ contains populations of multistranded complexes ($|\Psi| - |\Psi^0|$ entries)

$k = 2$
$i = |\Psi| - |\Psi^0|$
while $i \geq 1$
  $m^{\Psi}_{|\Psi|-|\Psi^0|}$ ++

  //Check to see if the current $m^{\Psi}$ results in a negative number of single strands
  $\text{legalPopulation} = 1$ // $\text{legalPopulation} = 1$ if no species has a negative population
  for $i = 1, |\Psi^0|$
    $\text{dotProduct} = 0$
    for $j = |\Psi^0| + 1, |\Psi| + |\Psi^0|$
      $\text{dotProduct} \mathrel{+}= A_{i,j}\, m^{\Psi}_{j-|\Psi^0|}$
    end for
    if $m_i^0 - \text{dotProduct} < 0$
      $\text{legalPopulation} = 0$
      break
    end if
  end for
```

```
// Generate next population
  if legalPopulation $== 1$ and $m^{\Psi}_{|\Psi|-|\Psi^0|} \leq m^{\Psi,\max}_{|\Psi|-|\Psi^0|}$
    $i = |\Psi| - |\Psi^0|$
  else
    if $i > 1$
      $m^{\Psi}_{i-1} ++$
      Set $m^{\Psi}_j = 0$ for $i \leq j \leq |\Psi| - |\Psi^0| - 1$
      $m^{\Psi}_{|\Psi|-|\Psi^0|} = -1$
      $i --$
    else
      $i = 0$
    end if
  end if

  if $i == |\Psi| - |\Psi^0|$
    // Get single strand counts
    for $i = 1, |\Psi^0|$
      $\mathrm{dotProduct} = 0$
      for $j = |\Psi^0| + 1, |\Psi| + |\Psi^0|$
        $\mathrm{dotProduct} \mathrel{+}= A_{i,j}\, m^{\Psi}_{j-|\Psi^0|}$
      end for
      $m^{\Psi^0}_i = m^0_i - \mathrm{dotProduct}$  // $m^{\Psi^0}$ contains populations of single strands ($|\Psi^0|$ entries)
    end for
    for $i = 1, |\Psi^0|$
      $\Lambda_{k,i} = m^{\Psi^0}_i$
    end for

    // Put multistranded counts in $\Lambda$
    for $j = |\Psi^0| + 1, |\Psi|$
      $\Lambda_{k,j} = m^{\Psi}_{j-|\Psi^0|}$
    end for
    $k ++$
  end if
end while
```

Appendix D

Excerpts from the NUPACK 2.0 User Guide

Following are portions of the text from the NUPACK 2.0 User Guide, released January 29, 2007. This serves to describe the basic structure and funtionality of the NUPACK compute engine code base. Notation is consistent with [17].

D.1 Directories

After compiling, the root directory `nupack` contains a Makefile and the following sub-directories:

`src`

Source code.

`bin`

Executables.

`lib`

Static libraries defining functions for linking at compile time.

`parameters`

Parameter files for RNA and DNA free energy models.

`doc`

Documentation, including this user guide, reference papers, and example input and output files.

It is convenient to set the environment variable `NUPACKHOME`, specifying an absolute path to the root directory `nupack` (e.g., `NUPACKHOME=/usr/local/nupack` or `NUPACKHOME=/home/`*`username`*`/nupack`).

D.2 Compilation

Core routines are written in C. To compile the code, type `make` in the root `nupack` directory. This will create the executables described in this user guide.

D.3 NUPACK conventions

Following are formatting standards for all input and output in NUPACK:

- All executables except `concentrations` and `distributions` take input from an input file *`prefix`*`.in`, where *`prefix`* is a command line argument. If *`prefix`* is not specified or *`prefix`*`.in` is absent or improperly formatted, the user is prompted for input on the screen.
- All sequences are listed $5'$ to $3'$. The bases in an *ordered complex* are indexed starting with 1 at the $5'$-most base of the first strand and ending at the $3'$-most base of the last strand. For example, if an ordered complex has three strands of length 15, 20, and 13, respectively, the fifth base of the third strand has index 40.
- Valid bases are `A`, `C`, `G`, `T`, and `U`. For RNA calculations, `T` is automatically converted to `U`, and vice versa for DNA calculations.
- *Secondary structures* may be specified in dot/parentheses notation (e.g, `((...))` specifies that bases 1 and 2 are paired to bases 7 and 6, respectively, while bases 3, 4, and 5 are unpaired). Different strands are separated by a + sign (no whitespace). Four types of "parentheses" are accepted: `()`, `[]`, {}, and <>. Within a specified structure, each type of parentheses must satisfy a nesting property but different types need not be nested, allowing specification of *pseudoknotted* structures (though highly nested pseudoknots may not be specifiable with only four types of parentheses).
- Secondary structures may also be specified in *pair list* format, where each line consists of two whitespace-separated integers $[i \;\; j]$, $i < j$, specifying that base i is paired to base j. Any secondary structure, including highly-nested pseudoknots, may be specified in this way.
- Comment lines begin with a `%` symbol.
- In input files, comment lines may be interspersed with input data. However, there may be no empty lines in the input files.

The following physical considerations are universal throughout NUPACK:

- Except where noted, all energy units are kcal/mol and all concentration units are molar.

- The zero free energy reference state for all calculations is a system where all relevant strands are present with no base pairs.

- The base pairs considered in the calculations include Watson-Crick ($A \cdot U/T$ and $G \cdot C$) and wobble ($G \cdot U/T$) pairs.

- Except where noted, all calculations and reported values appropriately take into account *distinguishability corrections* and *strand association penalties*.

- All executables involving *partition function* calculations consider an ensemble $\Omega(\pi)$ containing all *connected* unpeudoknotted secondary structures for a given ordered complex. If `-pseudo` (see below) is selected, $\Omega(\pi)$ also includes the class of pseudoknots specified in Reference [3].

The following option flags are recognized by multiple NUPACK executables:

`-material` *`parameters`*

The parameter files defining the nucleic acid material (all assuming 1 M NaCl) are specified via the argument *`parameters`* which represents either a filename prefix or a shorthand identifier for an included parameter set. If the filename does not contain a relative or absolute path, then the program will look for the files first in the current directory, and then in the directory `$NUPACKHOME/parameters`. Available filename prefixes and the corresponding shorthand identifiers currently include (DNA/RNA hybrids are not allowed):

- `RNA_mfold2.3` (default; shorthand: `rna`)
  Mfold v2.3 parameter files `*.dG` and `*.dH` for RNA allowing calculations at different temperatures; includes pseudoknot parameters from Reference [3].

- `DNA_mfold2.3` (shorthand: `dna`)
  Mfold v2.3 parameter files `*.dG` and `*.dH` for DNA allowing calculations at different temperatures; there are no pseudoknot parameters.

- `RNA_mfold3.0` (shorthand: `rna37`)
  Mfold v3.0 parameter file `*.dG` for RNA for calculations at 37 °C; includes pseudoknot parameters from Reference [3].

`-dangles` *`treatment`*

The way in which dangle energies are incorporated is specified by *`treatment`*, which may have the following values:

- `none`: No dangle energies are incorporated.

- `some`: (default) A dangle energy is incorporated for each unpaired base flanking a duplex (a base flanking two duplexes contributes only the minimum of the two possible dangle energies).

- `all`: A dangle energy is incorporated for each base flanking a duplex regardless of whether it is paired.

`-T` *`temperature`*

Temperature specified in °C (default is 37).

`-multi`

Specify a calculation involving complexes of multiple interacting strands.

`-pseudo`

Include the class of physically relevant pseudoknots defined in Reference [3] in Ω. This option is currently only available for single-stranded RNA calculations. An error message is returned if `-pseudo` is specified in combination with either `-multi` or `-material dna`.

D.4 Executables

D.4.1 Calculate the partition function

Command: `pfunc [-T` *`temperature`*`] [-multi] [-pseudo] [-material` *`parameters`*`] [-dangles` *`treatment`*`] [`*`prefix`*`]`

Description: Computes the partition function, Q, for an ordered complex over the set $\Omega(\pi)$.

Input: For single-stranded calculations, the input file contains the strand sequence specified on a single line. If `-multi` is specified, the input file must contain the following entries on separate lines:

- The number of *distinct* strand species, $|\Psi^0|$.
- The sequences for each distinct strand species, each on a separate line.
- L integers from the range 1 to $|\Psi^0|$ representing the distinct circular permutation $\pi \in \Pi$ of the L strands in the ordered complex.

Note that strand species defined on different lines are treated as distinct even if they have the same sequence.

Example 1

Partition function for a single RNA strand at 37°C including pseudoknots.

Input file contents:

```
GGGCUGUUUUUCUCGCUGACUUUCAGCCCCAAACAAAAAAUGUCAGCA
```

Command: `pfunc -pseudo`
`$NUPACKHOME/doc/examples/jcc04_telomerase/jcc04_telomerase`

Example 2

Partition function for an ordered complex of four DNA strands at 23°C, two of which are *indistinguishable*.

Input file contents:

```
3
AGTCTAGGATTCGGCGTGGGTTAA
TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG
AGTCTAGGATTCGGCGTGGGTTAACACGCCGAATCCTAGACTACTTTG
1 2 2 3
```

Command:
```
pfunc -T 23 -multi -material dna
      $NUPACKHOME/doc/examples/pnas04_hcr/pnas04_hcr_basic
```

Output: Following header comments, the free energy of the ordered complex (given by $\Delta G = -kT \log Q$) is written to the screen. The value of the partition function is written immediately below.

D.4.2 Calculate base-pairing observables

Command: `pairs [-T` *`temperature`*`] [-multi] [-pseudo] [-material` *`parameters`*`] [-dangles` *`treatment`*`] [`*`prefix`*`]`

Description: Computes *pair probabilities* $p(i_n \cdot j_m; \pi)$ for the ordered complex corresponding to the specified circular permutation $\pi \in \Pi$. When `-multi` is selected, also computes the *expected number of base pairs* $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$.

Input: Same as for the executable `pfunc`.

Output: The output is written to the files:

- *`prefix`*`.ppairs`

  Contains the probability of each type of base pair in the ordered complex. The relevant quantities are $p(i_n \cdot j_m; \pi)$, the probability that base i of strand n is paired to base j of strand m in the ordered complex corresponding to distinct circular permutation π. All strands in the ordered complex are considered to be distinct; there are no distinguishability corrections. For example, the two strands labeled 2 in Example 2 are considered distinct. One might think of them as strand $2a$ and $2b$, and a given base of strand $2a$ may have different pair probabilities than the corresponding one in strand $2b$. The total number of bases in the complex is $N = \sum_{l=1}^{L} N_l$, so indexing bases from 1 to N, the pair probabilities can be stored in a symmetric $N \times N$ matrix. Augmentation by an $N + 1$st column containing the probability that each base is unpaired causes the rows to sum to unity.

  By default, the file is formatted as follows. Following header comments, the first entry is the integer N.

The remaining entries come in triplets of the form $[i\ \ j\ \ p]$, where $1 \leq i \leq N$ and $1 \leq j \leq N+1$ are base numbers and p is the probability of the corresponding pair. Values corresponding to $j = N+1$ represent the probability that base i is unpaired. To save space, only probabilities above the diagonal that exceed a fixed cutoff of 0.001 are stored. If `-pseudo` is selected, each row is augmented by two additional columns. The first is the probability that bases i and j form a nested pair and the second is the probability that bases i and j form a non-nested pair. In the case of $j = N+1$, these additional columns store the probability that bases i and j do not form a nested pair and the probability that they do not form a non-nested pair, respectively.

- *`prefix`*`.epairs`

  Generated when `-multi` is selected. Similar to *`prefix`*`.ppairs` except strands of the same species are considered to be indistinguishable. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$, the expected number of base i of strand species A that are paired to base j of strand species B in the ordered complex corresponding to distinct circular permutation π. The number of distinct bases in the complex is $N_{\text{distinct}} \equiv \sum_{k \in \Psi^0} N_k$, representing the total number of bases in all $|\Psi^0|$ strand species. Numbering the distinct bases from 1 to N_{distinct}, the distinct base pairs may be represented as a symmetric $N_{\text{distinct}} \times N_{\text{distinct}}$ matrix; by augmenting the matrix with an extra column that contains the expected number of base i of strand species A that are unpaired, each row sums to the number of base i of strand species A in the complex. Note that this numbering system is used even if some sequences listed in the input file are absent from the specified ordered complex.

  The file is formatted as follows. Following header comments, the first entry is the integer N_{distinct}, and the remaining entries come in triplets of the form $[i\ \ j\ \ E]$, analogously to the `.ppairs` file, except E is the expected number of the corresponding pair. To save space, only expected values above the diagonal that exceed a fixed cutoff of 0.001 are stored. Information is stored only for bases included in the specified ordered complex.

D.4.3 Find minimum free energy (MFE) secondary structure(s)

Command: `mfe [-T `*`temperature`*`] [-multi] [-pseudo] [-material `*`parameters`*`] [-dangles `*`treatment`*`] [-degenerate] [`*`prefix`*`]`

Description: Compute and store the minimum free energy and MFE secondary structure(s) in $\Omega(\pi)$. If the `-degenerate` flag is selected, all secondary structures that share the same minimum free energy are stored; otherwise only one MFE structure is stored.

Input: Same format as for the executable `pfunc`.

Output: Output is written to the file *`prefix`*`.mfe`. After header comments, each entry describes one of the possibly many degenerate MFE structures. The entries are separated by comment lines (repeated `%` signs). The first line in each entry is the number of bases in the ordered complex. The second line is the minimum free energy. The third line is the dot/parentheses depiction of the MFE structure. Subsequent lines contain the MFE structure in pair list notation.

D.4.4 Find all secondary structures within a specified free energy gap of the MFE

Command: `subopt [-T` *`temperature`*`] [-multi] [-pseudo] [-material` *`parameters`*`] [-dangles` *`treatment`*`] [`*`prefix`*`]`

Description: Similar to `mfe` except that all secondary structures in $\Omega(\pi)$ with free energies within the specified (non-negative) free energy gap of the MFE are calculated and stored. This can be very slow and the output very large if the specified gap is too large. The output is sorted by increasing free energy.

Input: Same format as for the executable `pfunc` plus one additional row containing the energy gap.

Output: Output is written to the file *`prefix`*`.subopt` with the same format as for the executable `mfe`.

D.4.5 Count the number of secondary structures in the ensemble

Command: `count [-multi] [-pseudo] [`*`prefix`*`]`

Description: Similar to `pfunc` but sets all energy parameters to zero, thereby giving a count of the number of secondary structures in $\Omega(\pi)$.

Input: Same format as for the executable `pfunc`.

Output: The number of secondary structures in Ω is written to the screen, preceded by header comments.

D.4.6 Calculate the free energy of a secondary structure

Command: `energy [-T` *`temperature`*`] [-pseudo] [-multi] [-material` *`parameters`*`] [-dangles` *`treatment`*`] [`*`prefix`*`]`

Description: Calculate the free energy of a given sequence and secondary structure.

Input: Same format as for executable `pfunc`, plus one additional row specifying the secondary structure in dot/parentheses notation. Alternatively, the structure may be represented in pair list notation.

Output: The free energy of the structure is written to the screen, preceded by header comments.

D.4.7 Calculate the equilibrium probability of a secondary structure

Command: `prob [-T` *`temperature`*`] [-pseudo] [-multi] [-material` *`parameters`*`] [-dangles` *`treatment`*`] [`*`prefix`*`]`

Description: Calculates the equilibrium probability of a given secondary structure.

Input: Same format as for the executable `energy`.

Output: The equilibrium probability of the given structure is written to the screen, preceded by header comments.

D.4.8 Calculate the identities and partition functions of all strand complexes up to a specified size

Command: `complexes [-T` *`temperature`*`] [-material` *`parameters`*`] [-ordered] [-pairs] [-mfe] [-degenerate] [-dangles` *`treatment`*`] [-timeonly] [-quiet]` *`prefix`*

Description: First calculates the identities of all distinct circular permutations $\pi \in \Pi$ of strands for all possible (unpseudoknotted) complexes up to a user-defined size $L_{\max}$ and then calculates their respective partition functions $Q(\pi)$. The partition function for a complex, Q, is obtained by summing over the partition functions of its constituent ordered complexes, $Q(\pi)$ for all $\pi \in \Pi$. Significant additional functionality can be specified via command line flags. The output of complexes can be used as the input to the executables `concentrations` and `distributions`.

Additional Options:

`-ordered`

Also store properties for ordered complexes, each corresponding to one distinct circular permutation $\pi \in \Pi$, in addition to the summed properties for complexes.

`-pairs`

Calculate base-pairing observables as for the `pairs` executable.

`-mfe`

Calculate all minimum free energy structures for each ordered complex as for the `mfe` executable. Must be used in conjunction with the `-ordered` flag. The `-degenerate` flag is only applicable in conjunction with the `-mfe` flag.

`-timeonly`

After generating all distinct circular permutations, estimate the time it would take to compute all of the partition functions. The partition function calculations are not performed, the time estimate is written to the screen, and no output files are generated.

`-quiet`

Suppress output to the screen.

Input: The input file must contain the following entries on separate lines:

- The number of distinct strand species ($|\Psi^0|$).

- Sequence for each dinstinct strand species (each on a separate line).

- Maximum complex size ($L_{\max}$).

In addition to considering all complexes up to a specified maximum number of strands $L_{\max}$, the optional file `prefix.list` can be used to manually specify ordered complexes with more than $L_{\max}$ strands. The file contains lines of the form:

- `C` $L_1\ L_2\ \ldots\ L_{|\Psi^0|}$
  The $|\Psi^0|$ integers represent the number of copies (possibly zero) of each strand species in the complex of $L = \sum_{k=1}^{|\Psi^0|} L_k > L_{\max}$ strands.

- Subsequent lines containing distinct circular permutations of L integers from the range 1 to $|\Psi^0|$ define the ordered complexes of interest for the most recently defined complex.

Example 3

For a system containing three DNA strand species at 23°C, calculate partition functions, pair probabilities, and MFE structures for all ordered complexes of up to four strands, and for additional larger complexes specified in a `.list` file.

Input file contents:

```
3
AGTCTAGGATTCGGCGTGGGTTAA
TTAACCCACGCCGAATCCTAGACTCAAAGTAGTCTAGGATTCGGCGTG
AGTCTAGGATTCGGCGTGGGTTAACACGCCGAATCCTAGACTACTTTG
4
```

List file contents:

```
C 1 2 2
1 2 2 3 3
1 2 3 2 3
C 0 3 2
2 3 2 3 2
C 1 3 2
1 2 2 2 3 3
```

Command:
```
complexes -T 23 -material dna -ordered -pairs -mfe
        $NUPACKHOME/doc/examples/pnas04_hcr/pnas04_hcr
```

Output: Unless the `-quiet` flag is selected, `complexes` reports progress to the screen. By default there is one output file:

- *prefix*.cx

  Contains the composition and free energy of each complex. The first column is an integer complex identifier, and the next $|\Psi^0| + 1$ columns are $L_1 \;\; L_2 \;\; \ldots \;\; L_{|\Psi^0|} \;\; \Delta G$.

Depending on the command line options, the following output files may also be written:

- *prefix*.ocx

  Generated if `-ordered` is selected. Contains the composition and free energy of each ordered complex. The first and second columns are integer complex and ordered complex identifiers, respectively, and the remaining $|\Psi^0| + 1$ columns are $L_1 \;\; L_2 \;\; \ldots \;\; L_{|\Psi^0|} \;\; \Delta G$ for the ordered complex.

- *prefix*.ocx-key

  Generated if `-ordered` is selected. Contains the distinct circular permutation of strands for each ordered complex. The first and second columns are integer complex and ordered complex identifiers, respectively, and the remaining L columns are integers from the range 1 to $|\Psi^0|$. Note that the value of L may be different for each complex.

- *prefix*.cx-epairs

  Generated if `-pairs` is selected. Contains the base-pairing expectation values for each type of distinct base pair in each complex. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}})$, the expected number of base i of strand species A that are paired to base j of strand species B in the complex. The file contains a list of entries each with the same format as the `.epairs` file generated by the executable `pairs`. Each entry is separated by comment lines (repeated `%` symbols). Additionally, each entry begins with a comment line containing the complex identifier *id*, expressed as "`% complex`*id*".

- *prefix*.ocx-epairs

  Generated if `-pairs` and `-ordered` are selected. Similar to `.cx-epairs` but for ordered complexes. The relevant quantities are $E(i_{\{A\}} \cdot j_{\{B\}}; \pi)$, the expected number of base i of strand species A that are paired to base j of strand species B in the ordered complex corresponding to distinct circular permutation π. The entries are separated by comment lines (repeated `%` symbols), and each entry begins with a comment line containing the complex identifier *id* and order identifier *iorder*, expressed as "`% complex`*id*`-order`*iorder*".

- *prefix*.ocx-ppairs

  Generated if `-pairs` and `-ordered` are selected. Similar to `.ocx-epairs` except that all strands in the ordered complex are assumed to be distinct. The data in each entry are the same as those in the `.ppairs` file produced by the executable `pairs`.

- `prefix.ocx-mfe`

  Generated if `-mfe` and `-ordered` are selected. Contains the minimum free energy and MFE structure(s) for each ordered complex. Each entry is formatted the same as the output for the `mfe` executable. The entries are separated by comment lines (repeated `%` symbols), and each entry begins with a comment line containing the complex identifier *id* and order identifier *iorder*, expressed as "`% complexid-orderiorder`". If the `-degenerate` flag is selected, the degenerate MFE structures for a given entry are separated by a comment line of repeated `%` symbols.

D.4.9 Calculate the equilibrium concentration of each complex in a dilute solution

Command: `concentrations [-ordered] [-pairs] [-sort method] [-quiet] prefix`

Description: Given user-defined concentrations for each strand species, calculates the equilibrium concentration of each complex species or base pair in a large dilute solution, typical of experimental conditions in a test tube. Partition function information is read in from output files generated with the executable `complexes`.

Additional Options:

`-ordered`

Performs the calculation on ordered complexes rather than complexes. The input is read from the `prefix.ocx` (output from the executable `complexes`).

`-pairs`

Compute base-pairing information for the entire solution using results from `prefix.cx-epairs` or, if `-ordered` is selected, `prefix.ocx-epairs`, as output by the executable `complexes`.

`-sort method`

The argument *method* is one of the following integers.

0: Output is listed in the same order as in the input file.

1: (default) Output is sorted by the concentration of each complex (default) or ordered complex (if `-ordered` is selected).

2: Output is sorted first by the concentration of each complex and then, if `-ordered` is selected, by the concentration of each constituent ordered complex.

3: Output is sorted first by the integer complex identifier and then, if `-ordered` is selected, by the ordered complex identifier.

4: Output is sorted first by the number of strands in each complex, then by the integers $L_1\ L_2\ \ldots\ L_{|\Psi^0|}$ defining the number of each strand type in a given complex (with L_1 having the highest precedence, followed by L_2, and so on), and finally, if `-ordered` is selected, by the integer ordered complex identifier.

`-quiet`

Suppress output to the screen.

Input: By default, input is read from the file *prefix*`.cx`. Alternatively, specifying `-ordered` causes input to be read from *prefix*`.ocx`. These files are formatted as for the output of the executable `complexes`. The temperature at which the calculation is done is read from a line in the comments of the `.cx` or `.ocx` input file that reads "`% T =` *temperature*", where *temperature* is the temperature in °C. This line is automatically included in all output files of the executable `complexes`.

The input file *prefix*`.con` specifies the total molar concentration of each of $|\Psi^0|$ strand species on a separate line. The concentration may be in scientific notation (e.g., `1e-6` for a strand species at μM concentration).

Output: Unless `-quiet` is selected, the following information is written to the screen:

- The error in conservation of mass for each strand species in molar.
- The free energy of the entire solution in kcal/L.
- The wall clock time for the calculation.

The output is written to the files:

- *prefix*`.eq`

  The content is the same as the input file (except resorted, depending on the `-sort` option) with an extra column containing the concentration of the species in molar inserted after the free energy column.

- *prefix*`.fpairs`

  Generated if `-pairs` is selected. Reports the fraction of each distinct base that is paired to each of the other distinct bases in solution. The relevant quantity is $f_A(i_A \cdot j_B)$, the expected fraction of strands of species A for which base i is paired to base j of strand species B. The number of distinct bases in the dilute solution is $N_{\rm distinct} \equiv \sum_{k=1}^{|\Psi^0|} N_k$, representing the total number of bases in all $|\Psi^0|$ strand species. Numbering the distinct bases from 1 to $N_{\rm distinct}$, the quantity $f_A(i_A \cdot j_B)$ may be stored as an (asymmetric) $N_{\rm distinct} \times N_{\rm distinct}$ matrix; by augmenting the matrix with an extra column that contains the expected fraction of base i of strand species A that are unpaired, each row sums to unity.

  The file is formatted as follows. Following header comments, the first entry is the integer $N_{\rm distinct}$. The remaining entries come in triplets of the form $[i \;\; j \;\; f]$, where $1 \leq i \leq N_{\rm distinct}$ and $1 \leq j \leq N_{\rm distinct} + 1$ are base numbers and f is the corresponding fraction from the augmented matrix. To save space, only fractions that exceed a fixed cutoff of 0.001 are stored.

D.4.10 Calculate the equilibrium expected value and distribution for each complex population for a few strands in a dilute solution

Command: `distributions [-ordered] [-maxstates` *big*`] [-writestates] [-sort` *method*`] [-quiet]` *prefix*

Description: The executable `distributions` calculates the *partition function* Q_{box} for a *box* containing a small number of strands, given user-defined *populations* for each strand species. This is used to calculate the *expected value* and *probability distribution* of the population of each species of complex (or ordered complex). Partition function information is read from output files generated with the executable `complexes`.

Additional Options:

`-ordered`

Performs the calculation on ordered complexes rather than complexes. The input is read from the *prefix*`.ocx` (output from the executable `complexes`).

`-maxstates` *big*

The maximum number of states of the box to be enumerated ($|\Lambda|$, default is `1e7`). A segmentation fault will occur if the stack size on your machine is exceeded.

`-writestates`

Write a (typically large) output file describing properties for all population states of the system.

`-sort` *method*

The argument *method* is one of the following integers.

`1`: (default) Output is sorted by the expected value of the population of each complex or ordered complex (if `-ordered` is selected).

`2`: Output is sorted first by the expected value of the population of each complex and then, if `-ordered` is selected, by the expected value of the population of each constituent ordered complex.

`3`: Output is sorted first by the integer complex identifier and then, if `-ordered` is selected, by the ordered complex identifier.

`4`: Output is sorted first by the number of strands in each complex, then by the integers $L_1\ L_2\ \ldots\ L_{|\Psi^0|}$ defining the number of each strand type in a given complex (with L_1 having the highest precedence, followed by L_2, and so on), and finally, if `-ordered` is selected, by the integer ordered complex identifier.

`-quiet`

Suppress output to the screen.

Input: Same as for the executable `concentrations`, except the file *prefix*`.con` file is replaced by *prefix*`.count`, which contains the integer population of each of $|\Psi^0|$ strand species on a separate line. The last line of the file contains the volume of the box in liters. This may be entered in scientific notation (e.g., `1.4e-18`).

Output: Unless the `-quiet` flag is selected, the following information is written to the screen:

- The number of states of the box.
- The free energy of the entire box in units of kT and in units of kcal.
- The wall clock time for the calculation.

The output is written to the files:

- *prefix*`.dist`

  The content is the same as the input file (with rows sorted according to `-sort`) with extra columns after the free energy column. The first extra column (for complex or ordered complex j) is the expected value of the population $\langle m_j \rangle$. Subsequent columns are $[p_j(0)\ \ p_j(1)\ \ \ldots\ \ p_j(\max(m^0))]$. These represent the probability that complex (or ordered complex) j has population 0, 1, . . . , $\max(m^0)$, at equilibrium.

- *prefix*`.states`

  Generated when `-writestates` is selected. Each row corresponds to a population vector, m, for the box. The first column is the probability that the population vector occurs at equilibrium. By default, the remaining entries come in pairs: an integer complex identifier and a nonzero population. If `-ordered` is selected, the remaining entries come in triples: integer complex and ordered complex identifiers and then a nonzero population. This pattern continues for all complexes (or ordered complexes) with nonzero populations.

Appendix E

Help pages for the NUPACK web application

E.1 Thermodynamics Input

Nucleic acid type: Select RNA (default) or DNA for strand type. DNA/RNA hybrids are not allowed.

Maximum complex: All complex species up to this size will be considered in the calculation (maximum of 10, default 1). If pseudoknots are allowed, the maximum complex size is 1.

Allow pseudoknots: Include a class of pseudoknots in the calculations. If checked, the maximum complex size is 1 and the nucleic acid type is RNA.

Number of strand species: Select the number of distinct strand species from the pull-down menu (maximum of 10, default 1). If the same sequence is entered twice, it is considered to represent two distinct strand species.

Strand species: For each strand species, enter the name (default "Strand #") in the left field and the sequence in the right field. Ts or Us are acceptable for both RNA and DNA and will be appropriately converted.

Concentration (if applicable): Enter the initial concentration of the strand species. Use the pull-down menu at right to specify units (default μM).

Compute melt: Perform the calculation over a range of temperatures.

Temperature: Enter the temperature (in deg. C). If "Compute melt" is selected, the temperature minimum, increment, and maximum are requested.

Advanced options: Click "Advanced options" to specify additional ordered complexes and select an energy parameter set and dangle treatment.

- **Specify additional ordered complexes (if applicable):** Enter additional ordered complexes to be included in the calculation (augmenting those automatically specified based on the maximum complex size). Each line of space-separated integers decribes one ordered complex. An example entry for an ordered complex containing one strand each of the first and third strand species and two of the second would be “1 2 3 2”.
- **RNA/DNA energy parameters:** For RNA, there are two parameter sets, mfold 2.3 (default) and mfold 3.0 (valid only at 37 deg. C). For DNA, the only option is mfold 2.3.
- **Dangle treatment:**
  - **None:** no dangle energies are considered.
  - **Some (default):** a dangle energy is incorporated for each unpaired base flanking a duplex (a base flanking two duplexes contributes only the minimum of the two possible dangle energies).
  - **All:** a dangle energy is incorporated for each base flanking a duplex regardless of whether it is paired.

Email address: Enter your email address for notification of job completion (required for long jobs).

Reset: Reset all input fields.

Compute time: This order-of-magnitude estimate for the calculation time is updated as the job is specified. Very long calculations are not allowed on the NUPACK server, but source code may be downloaded from the “Downloads” section and used for research purposes on your local machine.

Compute: Start the calculation.

E.2 Results

Melt profile: Click the thumbnail for a larger version. The x-axis is temperature and the y-axis is the fraction of all bases present in the solution that are unpaired.

Download SVG: Download a scalable vector graphics version of the plot. The file is fully editable with most vector graphics editors.

Download data: Download a text file containing the data from which the plot was generated.

Temperature slider: Use the slide bar to adjust the temperature for the displayed output.

Ensemble pair fraction plot: Click the thumbnail for a larger version. The area and color of each dot at row i and column j scales with the equilibrium fraction of base i that is paired with base j in solution.

With this convention, the plot can be asymmetric. The area and color of each dot in the column at right scales with the equilibrium probability that the corresponding base is unpaired. Fractions below 0.001 are not depicted.

Equilibrium concentrations: The histogram describes the equilibrium concentration of multiple ordered complex species (selected based on the histogram filters below). Click on a histogram bar for details about the corresponding ordered complex.

Download histogram data: Click to download a text file containing the data from which the histogram was generated. This file has the concentrations and free energies each ordered complex in the calculation (included those filtered out of the histogram).

Histogram filters: Click to display choices for histogram appearance. All specified filters are active simultaneously.

- **Contains strand species:** Display only ordered complexes containing a specified strand species. If left blank (default), all ordered complexes are considered.
- **Fraction of max concentration:** Specifies a lower concentration threshold as a fraction of the concentration of the most abundant ordered complex (default 0.01).
- **Min concentration:** Specify a lower concentration threshold in convenient units (blank by default).
- **Max bars:** The maximum number of ordered complexes to display in the histogram (default is 5, no limit if blank).
- **Redisplay:** Click to redraw the histogram with current filters.

Change strand concentrations: Click to adjust strand species concentrations and recompute ensemble results. This is typically fast because it does not require recalculation of any partition functions.

E.3 Details

Temperature slider: Use the slide bar to adjust the temperature for the output displayed in the pair probability plot and MFE structure.

Pair probability plot: Click the thumbnail for a larger version of the pair probability plot, which describes the base-pairing probabilities for a given ordered complex. By definition, these data are independent of concentration and all other ordered complexes in solution. The area and color of each dot scales with the equilibrium probability of the corresponding base pair. With this convention, the plot is symmetric, with the upper and lower triangles separated by a diagonal line. The area and color of each dot in the column at right scales with the equilibrium probability that the corresponding base is unpaired. For this depiction, all strands are treated as distinct, and probabilities below 0.001 are not shown.

MFE structure plot: Click the thumbnail for a larger version of the minimum free energy structure plot. If possible, the secondary structure is drawn with "ladders" representing paired regions and arcs representing unpaired regions. If such a drawing results in overlapping segments or the structure is pseudoknotted, the structure is depicted as a polymer graph with backbones drawn as circlular arcs and base pairs as straight lines. In either representation, the 3-prime end of each strand is marked with an arrowhead and the bases are represented by color-coded circles following standard sequencing practice: A = green, T/U = red, G = black, C = blue.

Download SVG: Download a scalable vector graphics version of the plot. The file is fully editable with most vector graphics editors.

Download data: Download a text file containing the data from which the plot was generated.

Appendix F

Hessian of the per-solvent free energy of the HCR system

The Hessian of the Lagrange dual function describing solutions of initiated HCR polymers is given by $\nabla^2_\lambda h(\lambda)$ where $h(\lambda)$ is given by (5.9). The entries in the Hessian are

$$\begin{aligned}
[\nabla^2 h]_{11} &= -x_{\mathrm{I}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{1 + K_2 x_{\mathrm{H2}}}{1-\xi} \\
[\nabla^2 h]_{12} = [\nabla^2 h]_{21} &= -K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{1 + K_2 x_{\mathrm{H2}}}{(1-\xi)^2} \\
[\nabla^2 h]_{13} = [\nabla^2 h]_{31} &= -K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{\xi + K_2 x_{\mathrm{H2}}}{(1-\xi)^2} \\
[\nabla^2 h]_{22} &= -x_{\mathrm{H1}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{(1+\xi)(1 + K_2 x_{\mathrm{H2}})}{(1-\xi)^3} \\
[\nabla^2 h]_{23} = [\nabla^2 h]_{32} &= -K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{2\xi + (1+\xi) K_2 x_{\mathrm{H2}}}{(1-\xi)^3} \\
[\nabla^2 h]_{33} &= -x_{\mathrm{H2}} - K_{\mathrm{I}} x_{\mathrm{I}} x_{\mathrm{H1}} \frac{(1+\xi)(\xi + K_2 x_{\mathrm{H2}})}{(1-\xi)^3},
\end{aligned}$$

where ξ is given by (5.8). The dependence of x_{I}, x_{H1}, and x_{H2} on λ is given by (5.5).

Appendix G

Experimental methods

This appendix contains the methods used to generate the experimental data presented in this thesis. All data were acquired by Robert Dirks. The buffers used in the experiments are described below.

- Sodium buffer: 50 mM Na_2HPO_4, 0.5 M NaCl, pH 6.8 at 25°C.
- SB buffer: 10 mM NaOH, pH adjusted to 8.5 with boric acid.
- TBE buffer: 90 mM Tris, 89 mM boric acid, 2.0 mM EDTA, pH 8.0 at 25°C.
- RNase H buffer: 50 mM Tris-HCl, 75 mM KCl, 3 mM $MgCl_2$, 10 mM dithiothreitol, pH 8.3 at 25°C.
- λ exonuclease buffer: 67 mM Glycine-KOH, 2.5 mM $MgCl_2$, 50 μg/ml BSA, pH 9.4 at 25°C.

All water used in the experiments was ultrapure (resistance of at least 18 MΩ). All nucleic acid strands were synthesized and purified by Integrated DNA Technologies (Coralville, IA). The sequences for all strands are given in the main body of the thesis.

G.1 Methods for Figure 4.13

The reaction mixture was prepared by mixing 200 μl each of a 2 μM solution of strand A, a 0.4 μM solution of strand B, and a 2 μM solution of strand C along with 800 μl of 2× sodium buffer and 200 μl of H_2O. The sample was degassed for five min. Starting at 15°C, the sample was heated at 0.5°C per minute to 90°C. The sample was then cooled from 90°C to 15°C, also at 0.5°C per minute. Absorbance at 260 nm was measured for 3.5 sec every 0.5°C. The data were smoothed using a Gaussian kernel with a standard deviation of 1°C. If a is the vector of n absorbance measurements corresponding to the vector of temperatures T, the smoothed value of the absorbance for entry i is

$$\frac{\sum_{j=1}^{n} a_j \exp\left\{-(T_j - T_i)^2/2\sigma^2\right\}}{\sum_{j=1}^{n} \exp\left\{-(T_j - T_i)^2/2\sigma^2\right\}},$$

with $\sigma = 1°\text{C}$. These smoothed data were then scaled such that the tails of the absorbance curve matched those of the calculated fraction of unpaired bases.

G.2 Methods for Figure 5.2

The procedure is as published in [1]. Stock solutions of I, H1, H2 were diluted in sodium buffer to three times their final concentrations (see caption of Figure 5.2). The samples were heated to 95°C for 2 min and then allowed to cool to room temperature for 1 h. 9 μl each of I, H2, and H1, in that order, were combined (27 μl total reaction volume). The reaction was incubated at room temperature for 24 h before running 24 μl of each product on a 1% aragrose gel containing 0.5 μg of ethidium bromide per ml of gel volume in 1× SB buffer. The gel was run at 150 V for 60 min and visualized with UV light.

G.3 Methods for Figure 5.11

Samples 1 and 2 were prepared by adding 2.3 μl each of 6 μM solutions of modified H1 and modified H2 to 1.4 μl of 10× λ exonuclease buffer. An additional 8.1 and 7.6 μl of H_2O were added to samples 1 and 2, respectively. Samples 1 and 2 were heated to 95°C and cooled to 37°C at a rate of 1°C every 30 min. Samples 3 and 4 were prepared by adding 3.5 μl of 100 bp DNA marker ladder solution (500 μg/ml, New England Biolabs, Ipswich, MA) to 1.4 μl of 10× λ exonuclease buffer and 9.1 μl (sample 3) and 8.6 μl (sample 4) of H_2O. Samples 3 and 4 were held at 37°C for 15 min. 1 μl of λ exonuclease at a concentration of 5000 units/ml (New England Biolabs) was added to each of samples 2 and 4. 0.5 μl of H_2O was added to each of samples 1 and 3. The samples were vortexed, spun down, and then incubated at 37°C for 1 h. The samples were run at 150 V for 40 min on a 10% native polyacrylamide gel made with 1× TBE buffer, stained for 30 min in a solution containing 0.5 μg of ethidium bromide per ml of gel volume, and viewed under UV light.

G.4 Methods for Figure 5.14

Each sample had 2.5 μl of 10× RNase H buffer, 16 μl of H_2O, and 2.5 μl each of 10 μM H1 and H2 solutions. All five samples were held at 90°C for 90 sec and allowed to cool to 37°C for 1 h. 0.5 μl of a 10 μM solution of RNA initiator was added to each of samples 2, 3, 4, and 5. The samples were incubated at 37°C for 1 h. 1 μl of RNase H at a concentration of 5000 units/ml (New England Biolabs) was added to each of samples 3, 4, and 5. These samples were vortexed, spun down, and incubated at 37°C for 1 h. Samples 4 and 5 were then held at 90°C for 30 min and then cooled to 37°C for 1 h. 0.5 μl of a 10 μM solution of RNA initiator was added to sample 5, which was then incubated at 37°C for 1 h. The samples were run at 150 V for 70 min on a 1% aragrose containing 0.5 μg of ethidium bromide per ml of gel volume prepared by using a 1× SB buffer and visualized with UV light.