Title
ROLE OF RNA BRANCHING AND ITS RELATION TO PACKAGING BY VIRAL CAPSID PROTEIN

Author
SINGARAM, SURENDRA WALTER

Publication Date
2016

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

Theory of RNA branching and its relation to packaging by viral capsid protein

A dissertation submitted in partial satisfaction of the requirements

for the degree of Doctor of Philosophy in Chemistry

by

Surendra Singaram

2016

ABSTRACT OF DISSERTATION

Theory of RNA branching and its relation to packaging by viral capsid protein

by

Surendra Singaram

Doctor of Philosophy in Chemistry

University of California, Los Angeles, 2016

Professor William M. Gelbart, Chair

We have developed coarse-grained models of RNA bound by capsid protein (CP) in response to recent *in vitro* studies on the self-assembly of viral RNA by capsid protein. Under typical *in vitro* self-assembly conditions and in particular for the case of many ssRNA viruses whose CP have cationic N-termini, the adsorption of CP onto the (anionic) RNA is non-specific because the CP concentration exceeds the largest dissociation constant for CP-RNA binding. Following an introductory chapter, which recounts the history and physics of the *in vitro* self-assembly experiments, Chapters 2-4 of this dissertation explore simple lattice models of single-stranded (ss) RNA in the presence of interacting bound particles.

In Chapter 2 our investigation opens with a simple model to account for the measured yield of packaged RNA (nucleocapsids) by CP. We treat the RNA as a 1D lattice, whose sites represent CP binding sites, to calculate the yield of RNA packaged as a function of

the CP:RNA mol ratio. The measured yield of *in vitro* assembled nucleocapsids agrees exceptionally well with our simple 1D model.

In Chapters 3 and 4 we extend our 1D model to 2D to better account for the branched nature of RNA. Our theoretical work provides the statistical-thermodynamic grounding for understanding the *in vitro* "competition" experiments. In these experiments two RNAs compete for a limited amount of CP. The exchange of CP between the RNAs was found to be reversible at pH 7 and, separately, it was found that long RNAs were able to strip CPs from shorter ones. Our Monte Carlo simulations demonstrate that, for a given RNA mass, the sequence with the highest affinity for protein is the one with the most compact secondary structure arising from self-complementarity; similarly, a long RNA steals protein from an equal mass of shorter ones because of the energetic preference of forming one large cluster of CPs over forming two smaller clusters.

Finally, in chapter 5 we turn our attention away from the self-assembly experiments to focus on branched polymers. We show there that the 3D size of a branched polymer with N monomers can be directly calculated from a sequence of $N-2$ integers, known as the Prüfer sequence. The calculation was performed numerically and shown to be far more efficient (memory-wise) than typical calculations of the 3D size.

This dissertation of Surendra Singaram is approved by

Thomas G. Mason

Margot Elizabeth Quinlan

Robijn F. Bruinsma

William M. Gelbart, Committee Chair

University of California, Los Angeles

2016

Dedications

To my friends and family.

Contents

Acknowledgements

None of this work would have materialized if it were not for the "tough-love" of my friend and mentor Prof. Avinoam Ben-Shaul. In fact, the entirety of my dissertation was done under his watchful eye. For his active role in my development – both professionally and socially – I ask the reader to pay special recognition to him.

I also thank Professors William M. Gelbart and Charles M. Knobler for their warm friendship, for their advice (not limited to the appropriate use of *italics*), and for their introducing me to the art of beautiful science.

Chapter 2 is a version of "Simple Statistical Mechanical Theory for the Yield of RNA Packaging by CCMV Capsid Protein" by Surendra W. Singaram, William M. Gelbart, and Avinoam Ben-Shaul.

Chapter 3, Chapter 4, and Appendix C are together a version of Surendra W. Singaram, Rees F. Garmann, Charles M. Knobler, William M. Gelbart. "Role of RNA Branchedness in the Competition for Viral Capsid Proteins". *Journal of Physical Chemistry B*. 2015. 119 (44). 13991-14002. DOI: 10.1021/acs.jpcb.5b06445

Chapter 5 is a version of Surendra W. Singaram, Ajaykumar Gopal, and Avinoam Ben-Shaul. "A Prüfer-Sequence Based Algorithm for Calculating the Size of Ideal Randomly Branched Polymers". *Journal of Physical Chemistry B*. 2016.DOI:10.1021/acs.jpcb.6b02258

Vita

Santa Cruz High School, Santa Cruz, 2004

Undergraduate Research with Prof. Richmond Sarpong, UC Berkeley, 2006-2007

Undergraduate Research with Prof. Ronald Cohen, UC Berkeley, 2007-2008

Leo Brewer Award, 2007

B.S. Chemistry, Minor Mathematics, University of California at Berkeley, 2008

Research Scientist with Dr. Jonathan Williams, Max Planck Institute for Air Chemistry, Germany, 2009

Visiting Graduate Researcher with Prof. Avinoam Ben-Shaul, Hebrew University of Jerusalem, Israel, 2012-2016

Publications

1. Borodavka, A.; Singaram, S.W.; Stockley, P.G.; Gelbart, W.M.; Ben-Shaul, A.; Tuma, R. Sizes of Long RNA Molecules in Dilute Solution Are Determined by the Branching Patterns of Their Secondary Structures. 2016. *Submitted to Biophys. J.*.

2. Singaram, S.W.; Gopal, A.; Ben-Shaul, A. Prüfer-Sequence Based Algorithm for Calculating the Size of Ideal Randomly Branched Polymers. *J. Phys. Chem. B.* 2016. DOI: 10.1021/acs.jpcb.6b02258 .

3. Singaram, S.W.; Garmann, R.F.; Knobler, C.M.; Gelbart, W.M. Role of RNA Branchedness in Competition for Viral Capsid Proteins. *J. Phys. Chem. B.* 2015. 119 (44), 13991-14002. DOI:10.1021/acs.jpcb.5b06445

4. Comas-Garcia, M.; Garmann, R.F.; Singaram S.W.; Ben-Shaul, A.; Knobler, C.M.; Gelbart, W.M. Characterization of viral capsid protein self-assembly around short single-stranded RNA. *J. Phys. Chem. B.* 2014. 118 (27), 7510-7519.

5. Rollins, A.W.; Fry, J.L.; Hunter, J.F.; Kroll, J.H.; Worsnop, D.R.; Singaram, S.W.; Cohen, R.C. Elemental analysis of aerosol organic nitrates with electron ionization high-resolution mass spectrometry. *Atm. Meas. Tech.*. 2010. 3, 301-310. DOI: 10.5194/amt-3-301-2010.

6. Bunnelle, E.M.; Smith, C.R.; Lee, S.K.; Singaram, S.W.; Rhodes, A.J.; Sarpong, R. Pt-catalyzed cyclization/migration of propargylic alcohols for the synthesis of 3($2H$)-

furanones, pyrrolones, indolizines, and indolizinones. *Tetrahedron.* 2008. 64, 7008-7014.

7. Motamed, M.; Bunnelle, E.M.; Singaram, S.W.; Sarpong, R. Pt(II)-Catalyzed Synthesis of 1,2-Dihydropyridines from Aziridinyl Propargylic Esters. *Org. Lett.*. 2007. 9(11), 2167-2170. DOI: 10.1021/ol070658i

Chapter 1

Introduction

Viruses have a long history of tormenting living systems. They do so by hijacking the host cell's machinery for their own purposes, which manifests (in the best-case scenario) as a runny nose. This sinister activity went unchecked for thousands of years, because it was only until recently that we began to understand what a virus is.

"The architecture of a virus is beautifully simple", as a good friend of mine once said. Indeed, many single-stranded RNA plant viruses have just two structural components. The first is the genetic material of the virus, which is encoded in the form of (thousands-of-nucleotides-long) RNA. The second is a container (called the capsid) made of many identical copies of protein (called capsid protein); the capsid serves to protect the viral RNA cargo while the virus is between hosts. The combination of the protein container together with its viral RNA cargo is called a virus. RNA-genome viruses are the most numerous, but life is plagued by a large variety of DNA viruses as well.

One of the remarkable characteristics of single-stranded (ss) RNA viruses is that many of them can self-assemble *in vitro* from purified RNA and capsid protein (CP) components. This type of self-assembly –if it occurred on the macroscopic scale– would be like dumping the raw materials for a house on a plot of the land and watching the house build itself!!

In 1955, the reconstitution of tobacco mosaic virus from purified components by Fraenkel-

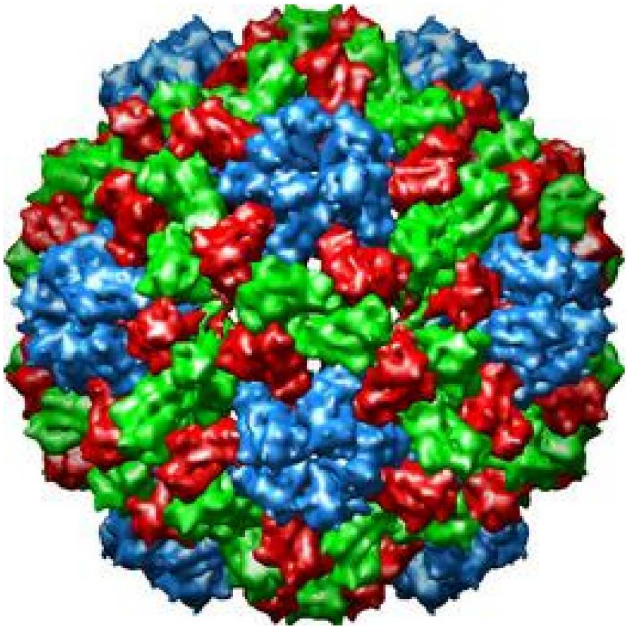

Figure 1.1: The reconstruction of the exterior capsid of CCMV at 2.7 Å resolution from X-ray diffraction [Speir 2006]. The capsid is made of 180 chemically identical CPs, arranged in T=3 symmetry corresponding to the three distinct geometric conformations the CP (colored, in blue, red, and green) in the capsid.

Conrat and Williams [Fraenkel-Conrat 1955] became the first example of making a virus from "scratch". In 1967, a second example was provided by Bancroft and Hiebert [Bancroft 1967] when they reconstituted the first *spherical* virus, cowpea chlorotic mottle virus (CCMV). Cowpea chlorotic mottle virus (CCMV) is a tri-partite ssRNA virus, each of its three RNA molecules is packaged separately into a capsid that is 28 nm in diameter (see Fig. 1.1). The capsid is comprised of 180 copies of the 190-residue capsid protein arranged with $T = 3$ Caspar-Klug symmetry [Caspar 1962], where the triangulation number T is the number of different geometrical conformations the CPs adopt in the capsid.

Over the last four decades several pivotal studies on CCMV have helped unravel the basic properties of the capsid protein. First, Adolph and Butler [Adolph 1977] established through sedimentation studies that CPs of disassembled CCMV exist as dimers in solution. Johnson and Speir later confirmed that the dimer (hereafter referred to as CP_2) is stabilized

by a strong non-covalent "molecular clamp" between the amino- and carboxy-terminal arms of each CP [Johnson 1997]. The CP_2 is the fundamental assembling unit of the capsid.

Lateral interactions between CP_2 arise from a combination of hydrophobic, electrostatic, and divalent metal-mediated interactions. The attractive force is pH dependent, decreasing with increasing pH [Johnson 2005, Garmann 2014b]. The pH dependence is thought to come from the electrostatic repulsion of deprotonated acidic residues, Glu81 in particular. At neutral pH, the CP_2s interact weakly because the residue is deprotonated and negatively charged, but if the solution is acidified the CP_2s interact strongly because the acidic residues become neutralized. Johnson et. al. found the CP_2-CP_2 interaction energy in an acidified solution (pH = 4.75) to be 3.7 kcal/mol (i.e. $\approx$ 6 kT at 300 K) [Johnson 2005]. To date, there have been no direct measurements of the CP_2-CP_2 interaction in the neutral pH solution. *The lateral interaction energy between CP_2s is controlled by pH.*

The other important interaction governing virus assembly is of course the CP_2-RNA interaction. In contrast to the interaction between a pair of CP_2s, the CP_2-RNA interaction is independent of the pH and is controlled by the ionic strength. The CP_2 interacts with RNA through the basic N-terminal arms hanging off each monomer. Residues 1-26 contribute +10 charges that interact with the negatively charged backbone of the RNA [Johnson 1997]. Furthermore, CP and RNA co-sediment as a complex at low ionic strength (0.1 M), but migrate as separate species at high (1 M) ionic strength. *CP_2-RNA interactions are controlled by ionic strength and are effectively pH-independent.*

A recent *in-vitro* experimental study of the assembly of nucleocapsids formed from CCMV CP and (the 3200 nt-long) RNA 1 of the Brome Mosaic Virus (BMV) has quantified the yield of packaged RNA as a function of the CP_2:RNA ratio [Cadena-Nava 2012]. The ratio of CP_2 to RNA needs to be as high as 160 to package all of the RNA in solution, hence this ratio has been dubbed the "magic ratio". Recall, each CP_2 has two flexible N-terminal tails containing 20 basic residues, so that at this ratio the N-terminal cationic charges total 3200, matching the 3200 phosphate (anionic) charges of the RNA backbone. In the following

chapter we present a simple statistical mechanical theory that accounts for the measured yield. Using a 1D lattice as a model of RNA whose sites represent CP_2 binding sites, we calculate the yield of RNA packaged as a function of the CP_2:RNA mol ratio.

Immediately following the discovery of the "magic ratio", competition experiments were devised pitting two RNAs of different length against each other in a solution containing enough CP_2 to package *one* but not both RNAs [Comas-Garcia 2012]. In a typical competition experiment, BMV RNA 1 (B1) serves as the defending champion, while shorter RNAs serve as the challengers. Challengers whose lengths are less than 2 kb are unable to compete at all with B1 (i.e. only B1 is packaged). While challengers with length greater than or equal to 2 kb can compete, they encapsidate less efficiently. Inspired by these beautiful experiments, we developed (Chapter 3) a Monte Carlo (MC) algorithm to generate an ensemble of (2D) branched polymers in the presence of interacting bound particles. We then used the MC algorithm to develop a simple phenomenological theory for understanding why B1 out-competed the short RNAs (Chapter 4). Furthermore, we found that a highly-branched polymer had a competitive advantage over a less branched polymer equal in length.

In Chapter 5, we turn our attention away from virus assembly to focus on branched polymers. One family of branched polymers in the study was chosen to have the same branching statistics as the secondary structures of random RNA sequences. More explicitly, we showed how one can use the Prüfer sequence [Pruefer 1918] representation of a branched polymer to calculate its 3D size assuming it behaves as an ideal polymer. This marks the first time the 3D size was calculated directly from a polymer's intrinsic sequence representation. The calculation was performed numerically and was shown to be far more efficient (memory-wise) than traditional calculations of the 3D size.

The author hopes you having an enjoyable time reading his dissertation...

Chapter 2

Yield of BMV RNA Packaging by CCMV CP

2.1 Experimental background

Nucleocapsids are containers made of protein (called the capsid) carrying genetic information, usually in the form of RNA. In this work we focus on a special class of nucleocapsids whose capsids are comprised of the capsid protein of the cowpea chlorotic mottle virus (CCMV). Recall, purified CCMV capsid protein is present as a dimer in solution (henceforth referred to as CP_2). The nucleocapsids can be readily prepared *in-vitro* following a two-step, pH-dependent, assembly reaction [Garmann 2014b].

In the first step, purified RNA and CP_2 are mixed together in a pH 7 buffer of low ionic strength (0.1M), resulting in the non-specific binding of CP_2 to RNA. CP_2 interacts primarily electrostatically with the (anionic) RNA through a pair of flexible, highly basic (cationic) N-terminal tails. The pH is then lowered to $\approx$ 5 whereupon nuclease-resistant capsids are formed.

The gel electrophoretic mobility assay (GEMA) of assembly reactions involving RNA 1 ($\approx$ 3200 nts) of the Brome Mosaic Virus (BMV) and increasing amounts of CCMV CP_2 is

shown in Fig. 2.1. The RNA is visualized by staining with ethidium bromide (EtBr). EtBr is a fluorescent tag that intercalates between base pairs, so the signal intensity reflects the concentration of RNA. The left-most lane contains pure BMV RNA1 and the rightmost lane is purified wild-type CCMV. Notice, the RNA runs faster than the capsid and that, in general, adding CP_2 to RNA results in complexes that run more slowly than pure RNA.

When the CP_2:RNA ratio reaches ≈ 70 another slower-moving band appears, this band corresponds to the nucleocapsid. Note, the two-step assembly reaction is not cooperative in the sense that protein bound to unpackaged RNA is not redistributed to package more RNA. In other words, the hallmark of a cooperative assembly reaction is a fixed "free RNA" band that becomes progressively fainter as more CP_2 is added, as opposed to its position shifting continuously (see Fig. 2.1). CP_2s distributed over the RNA, together with the fact that nucleocapsids begin to form when the CP_2:RNA ratio is less than 90 (i.e. the number of CP_2 in the wild type T=3 capsid), suggests that the yield of nucleocapsids could simply correspond to the fraction of RNAs bound with a sufficient number m^* (> 90) of CP_2.

The CP_2-CP_2 interactions are short-ranged and their attractive strength is pH dependent, and increasing with decreasing solution pH. Transfer of CP_2 from one RNA to another likely occurs in the neutral pH solution where the CP_2-CP_2 interactions are weaker. Further, in this work we will assume dilute solution conditions, allowing us to neglect associations between RNA-protein complexes.

Our assembly mechanism, then, treats each RNA-protein complex independently: an RNA is packaged if it binds at least m^* CP_2s. For the wild-type CP_2 whose two N-termini contain 20 basic residues, the CP_2:RNA ratio needs to be as high as 160 to package all the RNA [Cadena-Nava 2012]. These 160 CP_2s involve a total of ≈ 3200 N-terminal cationic residues, thereby matching the total phosphate (anionic) charge of the RNA. In general, an RNA with N nucleotides will have M CP_2 binding sites, where $M = \dfrac{N}{20}$.

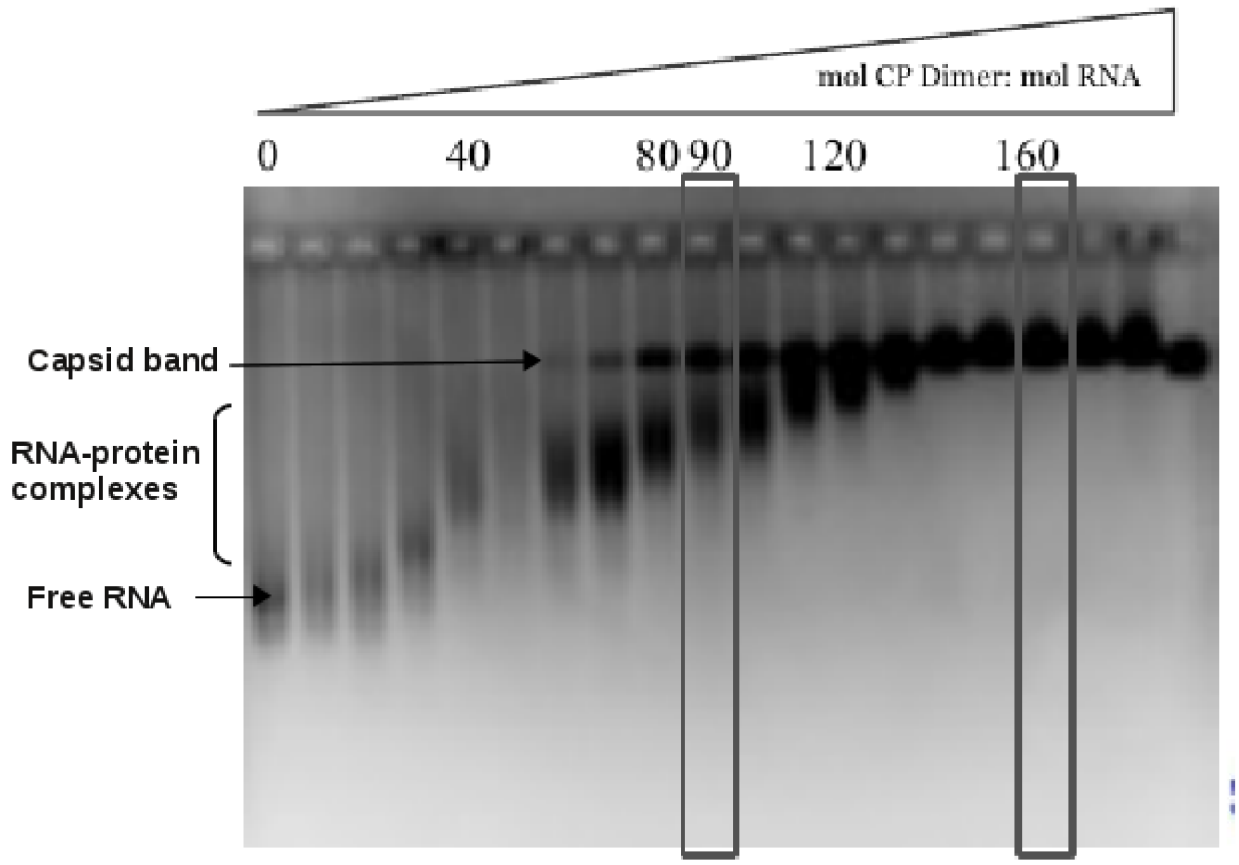

Figure 2.1: GEMA run at pH = 4.5 of assembly reactions with BMV RNA1 ($\approx$ 3200 nt) and CCMV CP_2. Sample is loaded at the top of the gel and migrates towards the bottom. The leftmost lane contains pure BMV RNA1 and the rightmost lane is purified wild-type CCMV. Other lanes contain the same concentration of BMV RNA 1 and varying concentrations of CP_2. From left to right, the CP_2:RNA mol ratio increases from 10 to 180. Capsids begin to appear around CP_2: RNA $\approx$ 70. When the mol ratio is 160, all the RNA is packaged. The bands between the free RNA and capsid band correspond to RNA-protein complexes. GEMA reproduced courtesy of Dr. Rees Garmann and Dr. Maurcio Comas-Garcia.

2.2 Theory

Using a 1D lattice as a model of RNA whose sites represent CP_2 binding sites, we calculate the yield of RNA packaged as a function of the CP_2:RNA mol ratio. Recall that an RNA with N nucleotides will have M CP_2 binding sites, where $M = \frac{N}{20}$. Since we are interested in the packaging of long RNA and are primarily motivated by the experimental results presented earlier, we consider packaging a 3200-nt RNA, which we model as a 1D lattice with 160 binding sites (each site corresponds to 20 nts). We calculate the yield of RNA packaged as a function of the CP_2:RNA mol ratio and of the strength of CP_2-CP_2 nearest-neighbor (attractive) potential energy ϵ ($\epsilon < 0$). Note, we assume the CP_2s interact strongly with the RNA, so we neglect the concentration of unbound CP_2 in each of the models we consider.

2.2.1 RNA as a 1D lattice

Random Adsorption

We first consider the limiting case of random adsorption of the CP_2s (i.e. $\epsilon = 0$). In our system we allow for an ensemble of n RNAs, each modeled as a 1D lattice with M sites, with a total of P bound particles (representing CP_2s). A site has probability σ of being occupied with $\sigma = \frac{P}{nM}$, the ratio of the total concentrations of CP_2 and RNA binding sites. The fraction of occupied sites, σ, is related to the CP_2:RNA mol ratio (x) by:

$$x = M\sigma \tag{2.1}$$

and the probability that an RNA is bound with exactly j CP_2s is [1]:

$$p(j;\sigma) = \frac{M!}{(M-j)!\,j!}\sigma^j(1-\sigma)^{M-j}. \tag{2.2}$$

In Fig. 2.2 we show how the distribution of bound particles depends on σ for a lattice with

[1] Appendix A justifies our use of the binomial distribution

160 sites. Note the "stoichiometric ratio" occurs when $x = 90$, or $\sigma \approx 0.6$. The distributions broaden with decreasing coverage, indicating there is still an appreciable fraction of RNA with more than the average share of CP_2 for lower values of σ.

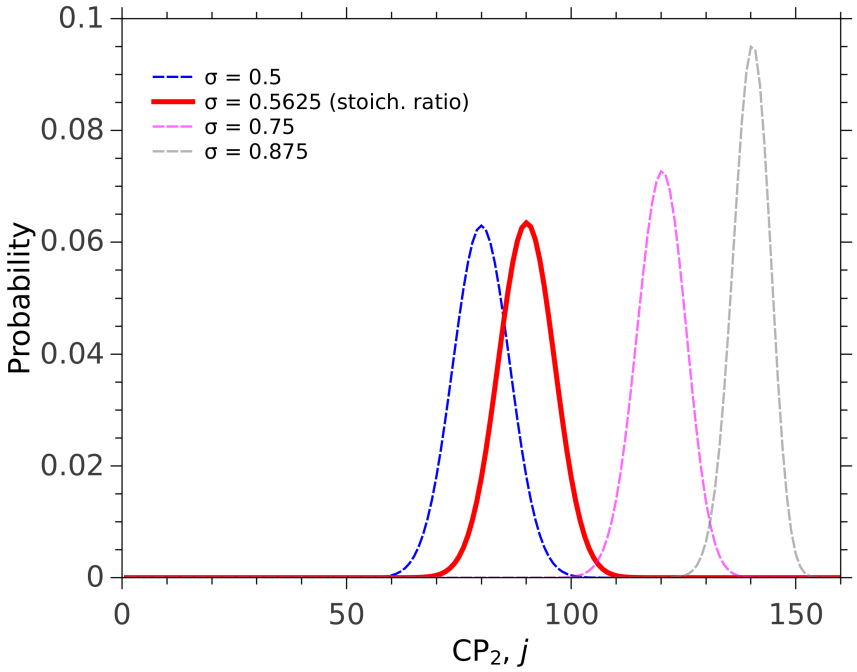

Figure 2.2: CP_2 distributions, for each of several values of σ, with RNA modeled as a 1D lattice with $M = 160$ sites (our simple model of 3200 nt RNA) and $\epsilon = 0$. The probability $p(j;\sigma)$ of finding j CP_2s on an RNA molecule is given by (2.2). **Blue** - $\sigma = 0.5$, RNA is half-covered with 80 CP_2 on average. **Red** - $\sigma = 0.5625$, the "stochiometric ratio" corresponding to 90 CP_2, the number of CP_2 in the T=3 capsid. **Pink** - $\sigma = 0.75$; 120 CP_2 per RNA. **Gray** - $\sigma = 0.875$; 140 CP_2 per RNA.

Our simple hypothesis, which we will constantly revisit, is that an RNA molecule with at least 90 CP_2s will be packaged upon pH lowering. In the case of random adsorption, the yield of packaged RNA, $f(x)$, is simply the fraction of RNAs with at least 90 bound particles when $\epsilon = 0$:

$$f(x) = \sum_{j=90}^{160} p(j;\sigma). \tag{2.3}$$

Since we are primarily interested in understanding the packaging of BMV RNA 1, we use σ (see Eq. (2.1)) with $M = 160$. In Fig. 2.3 we compare the yield from random adsorption (see Eq. (2.3)) to the yield of nucleocapsids as measured in the *in-vitro* assembly reactions such as those in Fig. 2.1. The Random Adsorption Model agrees well with the experimental measurements for the upper and lower third of CP_2:RNA ratios (i.e. $x < 60$ and $x > 120$). However, random adsorption significantly deviates from the experimental curve (see Fig. 2.2) for the intermediate values of x, underestimating the yield for $x < 90$ and overestimating it for $x > 90$.

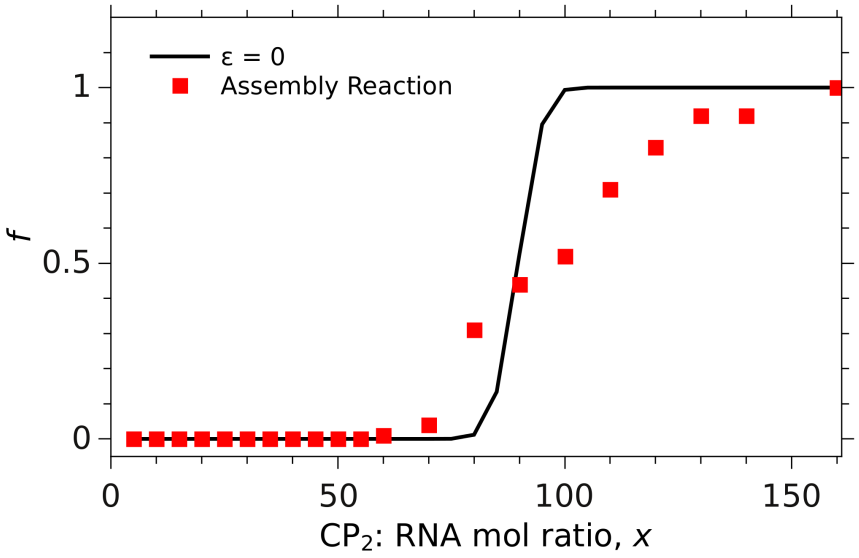

Figure 2.3: Yield (f) of nucleocapsids from the Random Adsorption Model (**black**), for a 1D lattice with 160 sites, and from the assembly reactions (**red**). The yield of an assembly reaction is derived, experimentally, from the fraction of RNA in the capsid band (see Fig. 2.1). The theoretical yield is taken to be the fraction of RNAs with at least 90 dimers.

CP_2-CP_2 interactions

To better account for the physics of packaging, we introduce CP_2-CP_2 interactions in our 1D model through the nearest-neighbor potential energy ϵ . After all, the CP_2s do interact with each other even in the neutral pH solution (albeit weakly). Our criterion for packaging remains unchanged so that a lattice must have at least 90 bound particles to be considered "packaged". The yield of nucleocapsids is simple to calculate once we know the probability distribution of CP_2 over the lattices.

The probability of finding j particles bound to an RNA lattice is proportional to the sum of Boltzmann weights of the states corresponding to j proteins on the lattice. We allow j to fluctuate, so that, on average, $\langle j \rangle$ sites are occupied. The probability of finding j particles on the lattice is proportional to $Q(j, M, \epsilon, T) \exp\left[\frac{\mu j}{kT}\right]$, where μ is the chemical potential and $Q(j, M, \epsilon, T)$ is the partition function for j proteins distributed on M sites[2]. Summing over all possible values of j yields the grand canonical partition function $\Xi(M, T, \mu)$:

$$\Xi(M, T, \mu) = \sum_{j=0}^{M} Q(j, M, \epsilon, T) \exp\left[\frac{\mu j}{kT}\right]. \tag{2.4}$$

The probability $p(j; \mu, \epsilon)$ of finding j proteins on a lattice with M sites is

$$p(j; \mu, \epsilon) = \frac{Q(j, M, \epsilon, T) \exp\left[\frac{\mu j}{kT}\right]}{\Xi(M, T, \mu)}. \tag{2.5}$$

Finally, μ is determined by the following constraint:

$$\langle j \rangle = \sum_{l=0}^{M} p(l; \mu, \epsilon) l. \tag{2.6}$$

Shown in Fig. 2.4 is the probability distribution as a function of the nearest-neighbor interaction energy ϵ, for $M = 160$ and $\langle j \rangle = 90$. Note that the distribution becomes wider

[2]The closed-form of $Q(j, M, \epsilon, T)$ is derived in Appendix B

and more skewed as the interaction energy becomes more negative (i.e. more attractive). For example, when the interaction energy is $\epsilon = -10$ kT, there is a larger fraction of RNAs close to saturation than in random adsorption ($\epsilon = 0$ kT).

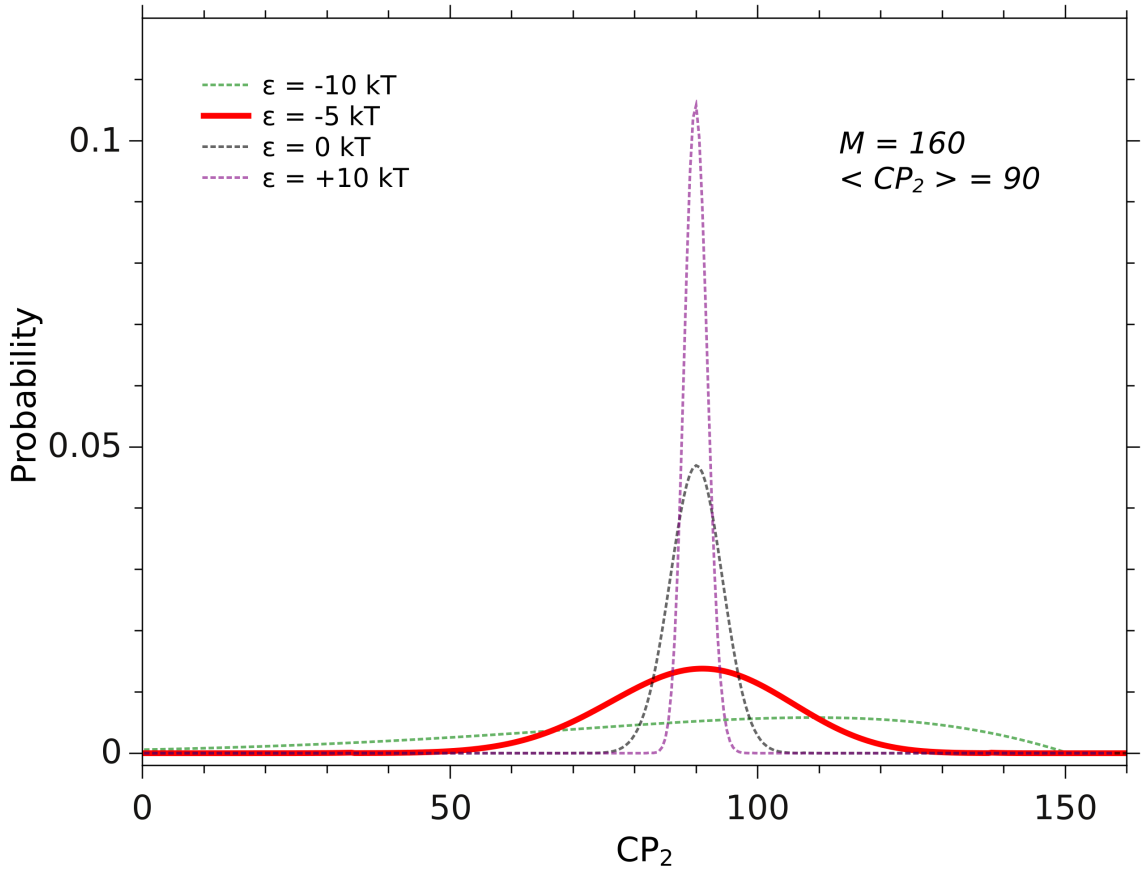

Figure 2.4: CP_2 distribution as a function of ϵ on a lattice with $M = 160$ and $\langle CP_2 \rangle = 90$. **Green**- $\epsilon = -10$ kT ;**Red**- $\epsilon = -5$ kT; **Black**- $\epsilon = 0$ kT (random adsorption); **Purple**- $\epsilon = +10$ kT.

The yield of packaged RNA is calculated by substituting Eq. 2.5 into Eq. 2.3, using ϵ = -5 or -10 kT, to investigate the effect of attractive CP_2-CP_2 interactions on packaging RNA. As in random adsorption, our criterion for packaging is that at least 90 CP_2s be bound to the lattice. As the interaction energy becomes more attractive, the yield (f) increases for $x < 90$ (see Fig. 2.5), which follows directly from the shape of the underlying distribution. When

$x < 90$, a significant fraction of lattices have at least 90 bound particles, resulting in a higher yield of nucleocapsids than for random adsoprtion (e.g. compare $\epsilon = -10$ kT to $\epsilon = 0$ kT in Fig. 2.5). This is, of course, due to the "skewness" of the underlying probability distribution when $\epsilon = -10$ kT as described above. Note, when $x > 90$ the yield of nucleocapsids is *lower* in the presence of attractive CP_2-CP_2 interactions. In this capitalist regime, RNAs that would otherwise be packaged give up their particles so that other RNAs can be saturated.

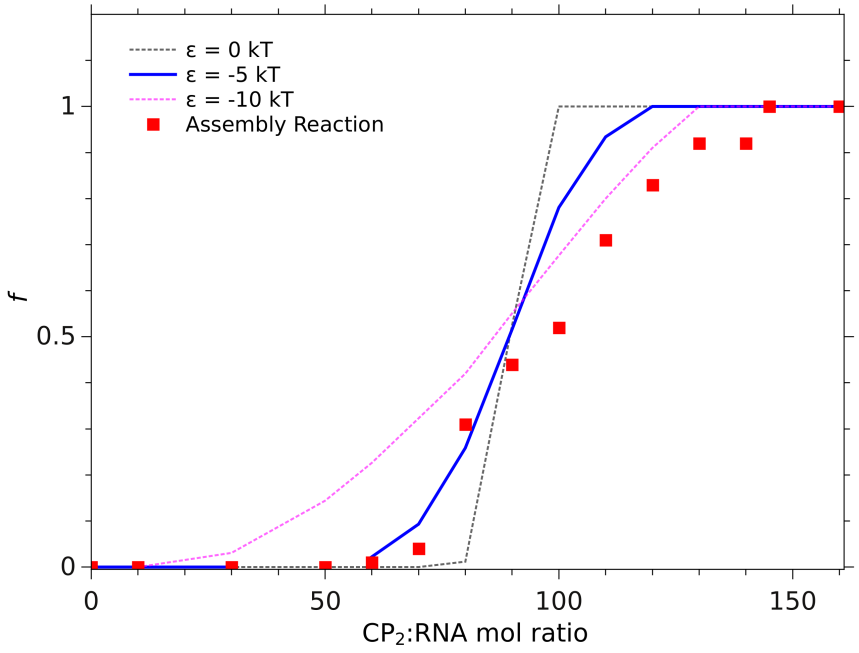

Figure 2.5: Yield of nucleocapsids as a function of the CP_2:RNA ratio, for each of several different values of the nearest-neighbor energy ϵ. **Pink**- $\epsilon = -10$ kT; **Blue**- $\epsilon = -5$ kT; **Black**- $\epsilon = 0$ kT; **Red**- Assembly Reaction (Expt.).

Introducing the nearest-neighbor potential energy improves our model. Our theoretical yields are in better agreement with the assembly reactions, though we still see significant deviations from the experimental values. When ϵ = -10 kT, for example, the model accounts well for experimental yields for high CP_2:RNA ratios, but does not do as well for lower ratios.

Critical CP_2 cluster

In this section, we calculate the yield of packaged RNA as a function of the CP_2:RNA ratio (x) by introducing an additional condition for encapsidation: the size of a nearest-neighbor string of bound protein dimers must be at least m^* (the minimum cluster size). The theoretical yield $f(x)$ is formulated as a convolution of the probability of there being at least 90 CP_2 on an RNA (for a given value of x) and the probability of finding a nearest-neighbor string of length at least m^*. To aid in the calculation of the yield, we employed a simple Metropolis Monte Carlo [Metropolis 1953] algorithm to equilibrate a system of a thousand 1D lattices, each with 160 sites, for a range of x values. We allow for CP_2-CP_2 interactions through the nearest-neighbor potential energy ϵ. The measured yield (f) from the assembly experiments is accounted for by $m^* = 30$ and $\epsilon = -5$ kT, see Fig. 2.6.

2.3 Conclusion

In summary, we have shown the measured yield of *in-vitro* assembled nucleocapsids can be accounted for using our simple 1D lattice model of the RNA. The yield was calculated for a variety of CP_2:RNA ratios, CP_2-CP_2 interactions, and minimum cluster sizes (m^*). The best agreement with the experimentally measured yield was found by imposing two necessary and sufficient conditions for packaging: at least 90 particles must be bound to the lattice and the size of a nearest-neighbor string of particles must be at least m^*. The measured yield is accounted for by a m^* value of 30 and a nearest-neighbor dimer attraction energy, ϵ, of -5 kT. This simple model can be generalized to include several other important physical aspects, for example: the 2D branched (versus 1D) nature of RNA, the heterogeneity of protein affinity of different RNA subsequences, and novel features associated with packaging of multiple RNAs.

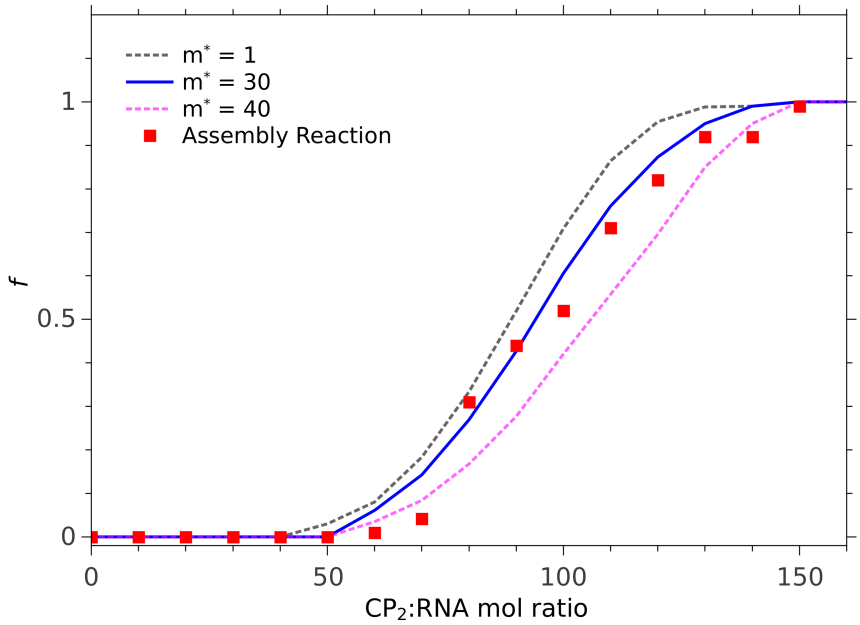

Figure 2.6: Yield (f) of nucleocapsids as a function of CP_2:RNA ratio and m^* (the minimum cluster size) with $\epsilon = -5$ kT. We impose two conditions for packaging: the lattice must have at least 90 CP_2s as well as a string of at least m^* bound particles. Monte Carlo simulations on an ensemble (1000) of 1D lattices, each with 160 sites, were used to compute f. The yields from the assembly reactions (**red**) are accounted for when the cluster size is bewteen $m^* = 30$ (**blue**) and $m^* = 40$ (**pink**).

Chapter 3

Configurational-Bias Monte Carlo of RNA bound with CP

3.1 Generating linear polymers in the presence of interacting bound particles

3.1.1 Introduction

We consider a linear polymer with excluded volume, whose monomers represent particle binding sites, as a model for RNA bound with protein. The monomers are assumed to be the same size as the binding particle, so that neighboring sites can be occupied. Since protein-protein interactions are attractive, we would expect a linear polymer dressed with protein to be more compact than a linear polymer of the same length in the absence of protein. Our aim is to study how the size of the linear polymer depends on how much protein is bound to it.

This particular investigation is, for ease of computation, carried out by placing the linear polymer on a square lattice such that (i) it does not cross itself and (ii) pairs of occupied sites separated by a distance equal to the lattice spacing have potential energy ϵ ($\epsilon < 0$). Fig. 3.1

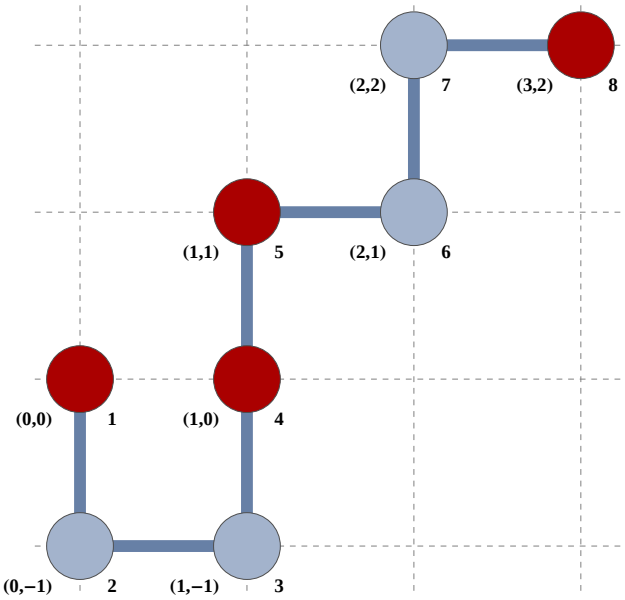

Figure 3.1: Example conformation of a linear polymer dressed with protein. Red dots correspond to occupied sites and blue dots correspond to unoccupied sites. The connectivity of the chain is denoted by the thick black line.

shows a possible configuration of a linear polymer bound by particles. The red and blue dots in the figure correspond to occupied and unoccupied sites, respectively. The connectivity of the chain is shown by the thickend black line (e.g. site 1 is connected to site 2, etc.). Sites 1 and 4 – even though they lie distant from one another along the chain contour – interact with potential energy ϵ, because they are occupied sites separated by a distance equal to the lattice spacing. Sites 4 and 5 also interact with energy ϵ, but this interaction will not affect the size of the polymer. In order to study how the polymer's size is coupled to the number and interaction energy of the particles bound to it, we need to generate a thermal ensemble of configurations of the polymer when it is dressed with a certain number of particles that attract each other with a certain energy ϵ. To do so, we use the Rosenbluth method [Frenkel 1996] [Tzlil 2005] described below.

3.1.2 Rosenbluth method

Our goal is to generate a thermal ensemble of polymer configurations of length L (equivalently, the number of protein-binding sites), where each configuration has B particles bound to it. We describe how to generate a single configuration and this method is then repeated to generate an ensemble of configurations. We generate a polymer configuration by adding one site (either unoccupied or occupied with a particle) at a time to a growing polymer. After all L sites have been added and the polymer is grown, the fraction of occupied sites f is $\frac{B}{L}$. Let i, where i runs from 1 to L, be the index of the site which we add to the growing polymer. When $i-1$ sites have been added so that B_{i-1} sites are occupied, we use

$$f_i = \frac{B - B_{i-1}}{L - i + 1} \tag{3.1}$$

to determine whether the next site (the ith site) is occupied. Note that f_i is the fraction of occupied sites among the sites to be added. Finally, let $\alpha = \exp\frac{-\epsilon}{kT} > 1$ be the Boltzmann factor, where ϵ is the attractive potential energy (i.e. $\epsilon < 0$). The Boltzmann constant and temperature are denoted by k and T, respectively.

The following procedure can be illustrated by generating the polymer shown in Fig. 3.1 by adding one site at a time. This construction highlights the underlying principles of the Rosenbluth method. The polymer has eight binding sites, four bound particles, and $\alpha = \exp 3$ (i.e. $\epsilon = -3\ kT$). We start the simulation at the origin (0,0) and decide with probability $f_1 = \frac{P}{L}$ whether to add an occupied or unoccupied site. More explicitly, we decide if this site is occupied by picking at random a real number r between 0 and 1 and comparing its value to f_1: if r is less than f_1 the first site is occupied, otherwise it is unoccupied. In the case of our example polymer with $L = 8$ and $P = 4$, we have $f_1 = \frac{1}{2}$, so that the first site is occupied with probability

$$P_{1,o}^{(0,0)} = \frac{1}{2}. \tag{3.2}$$

The subscripts (i, o) on the left-hand side of Eq. (3.2) denote the site index ($i = 1$) and state

("o" for occupied) of the site, while the superscript denotes the site's position on the lattice. Here we found $r < \frac{1}{2}$, i.e., the first site is occupied. How do we add the second site?

The second site can be in any of the four positions: (0,-1), (0,1), (-1,0), or (1,0), and in either of two states (i.e., there are 8 outcomes for the second site). Since the first site is occupied, we decide whether the next site is occupied or unoccupied with probability proportional to $f_2\alpha$, with $f_2 = \frac{B-1}{L-1}$, or to $1 - f_2$, respectively. The Boltzmann factor α is introduced because if an occupied site is added it will be next to the already occupied $i = 1$ site. In general, we introduce a multiplicative Boltzmann factor α (> 1) for every nearest-neighbor pair of occupied sites associated with the new site (should it be chosen to be occupied), so that occupied sites are more likely to be placed near other occupied sites. Summing over these weights for the possible outcomes generates the local partition function for site 2:

$$w_2 = 4(1 - f_2) + 4f_2\alpha. \tag{3.3}$$

Here the factor of four comes from the four possible places we can add an occupied or an unoccupied site. As expected, there are eight terms in Eq. (3.3). The probability that position (0,-1) – or any of the other three positions – is occupied is $\frac{f_2\alpha}{w_2}$, and the probability that it is unoccupied is $\frac{1-f_2}{w_2}$. Returning to our polymer where (using Eq. (3.1)) $f_2 = \frac{3}{7}$, we decided (see Fig. 3.1) that the next site is unoccupied and in position (0,-1) with probability

$$P_{2,u}^{(0,-1)} = \frac{1-f_2}{w_2}, \tag{3.4}$$

where $f_2 = \frac{3}{7}$. The subscript "u" means the the second site is unoccupied. Operationally, we choose a particular outcome by first partitioning the interval $[0, 1]$ into non-overlapping sub-intervals proportional in length to the probability of each outcome. Next, we pick at random a real number r between 0 and 1 and select the outcome associated with the sub-interval containing r. Now let's add the third site.

This time the new (third) site has only three possible positions: (0,-2), (1,-1), or (-1,-

1) since the position (0,0) is unavailable. If an occupied site is added at any one of these positions it would not form a nearest-neighbor pair of occupied sites because the other occupied site is at position (0,0). The local partition function for the third site is

$$w_3 = 3(1 - f_3) + 3f_3, \tag{3.5}$$

where $f_3 = \frac{3}{6}$. We decided (see Fig. 3.1) that the third site is unoccupied and in position (1,-1) with probability

$$P_{3,u}^{(1,-1)} = \frac{1 - f_3}{w_3} \tag{3.6}$$

following the same notation as before. Now let's add the fourth site.

The fourth site also has three possible positions: (1,0), (1,-2), and (2,-1). If an occupied site is added at the position (1,0) it would form a nearest-neighbor pair of occupied sites (with site 1), but not if it is added at the other two positions. The local partition function for the fourth site, then, is:

$$w_4 = f_4\alpha + 2f_4 + 3(1 - f_4), \tag{3.7}$$

where $f_4 = \frac{3}{5}$. The first term in Eq. (3.7) corresponds to placing the occupied site at position (1,0), forming a nearest-neighbor pair of occupied sites, so this outcome is increased by the Boltzmann factor α. The middle term corresponds to adding the occupied site at the other two positions. The factor of two arises because both positions have the same energy. Finally, the last term corresponds to adding an unoccupied site. Again there is a factor of three because there are three possibles places to add the unoccupied site. The probability that we decided site 4 is occupied at position (1,0) is

$$P_{4,o}^{(1,0)} = \frac{f_4\alpha}{w_4}, \tag{3.8}$$

where the "o" means the state of the site is occupied.

Continuing in this way, we will eventually add all eight sites to generate the polymer

shown in Fig. 3.1. The probability of generating the polymer in that configuration (call it a) is equal to the probability of sequentially adding each site at the corresponding position and state. The probability of generating configuration a, $P_R(a)$, is the product of probabilities (3.2), (3.4), (3.6), and (3.8) together with probabilities $P_{5,o}^{(1,1)}, P_{6,u}^{(2,1)}, P_{7,u}^{(2,2)}, P_{8,o}^{(3,2)}$:

$$\begin{aligned} P_R(a) = & \quad P_{1,o}^{(0,0)} P_{2,u}^{(0,-1)} P_{3,u}^{(1,-1)} P_{4,o}^{(1,0)} P_{5,o}^{(1,1)} P_{6,u}^{(2,1)} P_{7,u}^{(2,2)} P_{8,o}^{(3,2)} \\ = & \quad \frac{f_1(1-f_2)(1-f_3)f_4\alpha f_5\alpha(1-f_6)(1-f_7)f_8}{w_1w_2w_3w_4w_5w_6w_7w_8} \end{aligned} \tag{3.9}$$

where the w_is have been previously defined and $w_1 = 1$.

The probability in Eq. (3.9) is the probability of generating configuration a, which *is not equal to the Boltzmann probability*. The denominator is called the Rosenbluth factor for configuration a and is denoted $W(a)$. Since all eight sites are eventually added, the product of fractions in the numerator $(f_1, (1-f_2), ...)$ is the same for all configurations; we denote this product of fractions by c. Then the Rosenbluth probability takes the general form

$$P_R(a) = \frac{c\alpha^l}{W(a)}, \tag{3.10}$$

where α^l is Boltzmann factor for the configuration with l nearest-neighbor pairs of occupied sites. The Boltzmann factor (unnormalized) for the configuration is α^2, so that the Boltzmann *probability* for the configuration, $P_B(a)$, is:

$$P_B(a) = \frac{\alpha^2}{Z} = \frac{P_R(a)W(a)}{\sum_{j=1} P_R(j)W(j)}, \tag{3.11}$$

where Z is the partition function for the polymer (8 sites with 4 bound proteins). The summation in the denominator is over all distinct configurations generated by the Rosenbluth method. Each term in the sum is proportional to the Boltzmann factor for the configuration, as can be seen from Eq. (3.10). The second equality in Eq. (3.11) holds because the product

of fractions in the numerator of Eq. (3.9) is the same for all configurations. In order to compute the average radius of gyration $\langle R_g \rangle$, for example, one would evaluate:

$$\langle R_g \rangle = \frac{\sum\limits_{j=1} R_g(j) P_R(j) W(j)}{\sum\limits_{j=1} P_R(j) W(j)} \tag{3.12}$$

where the summation is over all *distinct* configurations generated by the Rosenbluth method. Equivalently, one could also evaluate:

$$\langle R_g \rangle = \frac{\sum\limits_{m=1} R_g(m) W(m)}{\sum\limits_{m=1} W(m)} \tag{3.13}$$

where the summation is over all configurations generated by the Rosenbluth method.

3.1.3 Size of linear polymer in the presence of interacting bound particles

In this section we show how the size (i.e. the radius of gyration) of a linear polymer depends on the number of particles bound to it. More explicitly, we compute the radius of gyration at three coverage fractions ($\frac{B}{L}$): 0, 0.5, and 1 over a range of lengths (5 to 500 monomeric units); the particle interaction energy is $\epsilon = -3$ kT, consistent with estimates of the CP_2-CP_2 interaction energy [Johnson 2005]. In Fig. 3.2 we show a few snapshots from our simulations for a linear polymer with 50 monomers at the three coverage fractions. In the absence of bound particles (i.e., $\frac{B}{L}$) the configurational statistics of a linear polymer with excluded volume is the same as a self-avoiding random walk (SAW), when one identifies the polymer length L with the number of steps in the SAW. The radius of gyration (R_g) of a SAW on a 2D latices scales as $L^{\frac{3}{4}}$ [Li 1995], with which our simulations agree (see Fig. 3.3). When the coverage fraction is 1, the R_g varies as $L^{\frac{1}{2}}$ (see Fig. 3.3) – consistent with the scaling behavior of a collapsed polymer in two dimensions (i.e. a 2D condensed phase). The

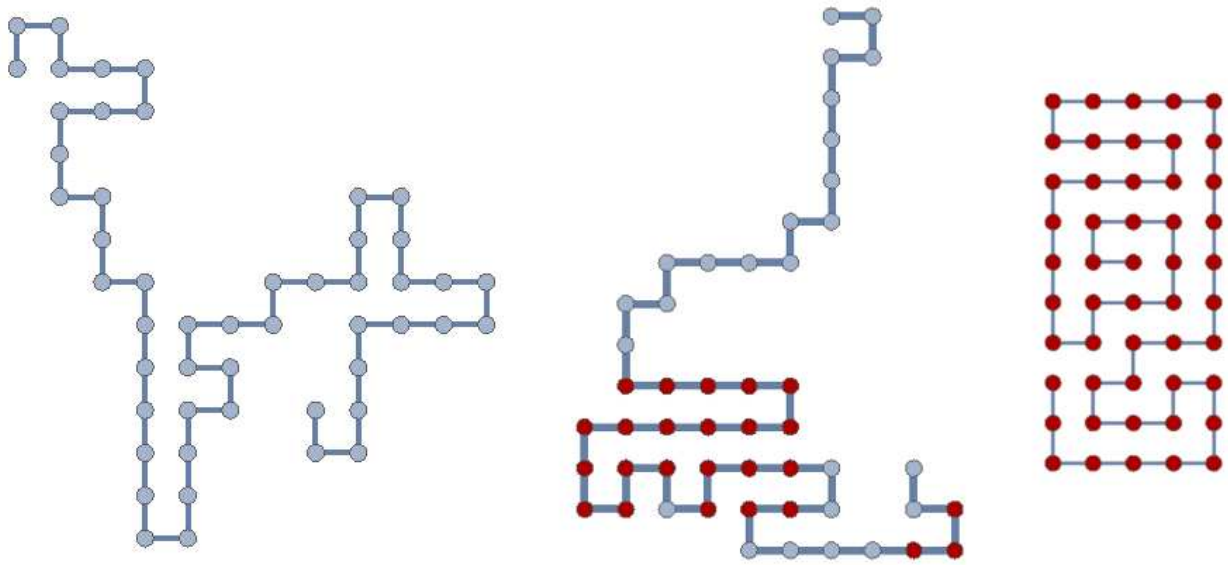

Figure 3.2: Example configurations of a linear polymer with 50 monomers dressed with particles at different coverage fractions. Bound particles are shown in red and blue vertices correspond to unoccupied sites. Coverage fractions of configurations from left to right: 0, 0.5 and 1; $\epsilon = -3$ kT.

R_g varies as $L^{\frac{2}{3}}$ when the coverage fraction is 0.5 (see Fig. 3.3). The fact that we observe scaling at all is interesting. At intermediate coverage fractions, the polymer of length L may behave as a block copolymer with B units and $L - B$ units whose respective size scale as a collapsed polymer and a polymer absent of particles. Perhaps this perspective can explain the scaling behavior of linear polymers at intermediate coverage fractions when the attraction is strong enough to cause the particles to form a condensed phase on the linear polymer.

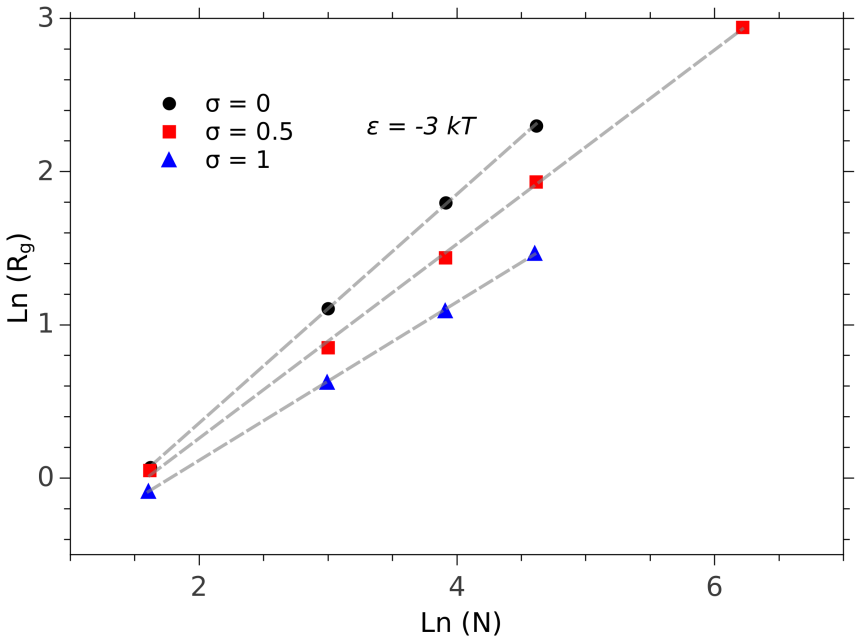

Figure 3.3: Values of $\ln R_g$ versus $\ln N$ to show scaling behavior of 2D linear polymer with interacting bound particles. The slope of the curve corresponds to the scaling exponent ν (see text). Black curve – coverage fraction of 0 ($\nu = 0.75$), purple – coverage fraction of 0.5 ($\nu = 0.66$), orange – coverage fraction of 1 ($\nu = 0.5$).

3.2 Generating branched polymers in the presence of interacting bound particles

3.2.1 Introduction: RNA as a tree graph

Single-stranded RNA folds on itself to form a branched structure composed of duplexes and loops of single-stranded sequences. This branched structure can be represented as a tree graph by mapping each bulge, hairpin loop, or internal loop to a vertex and every stem to an edge [Gan 2004]. (A tree graph is a collection of points with each point connected to at least one other point (vertex), and with no closed path of connecting edges.) Recall that each CP dimer interacts strongly and non-specifically with 20 nucleotides. Curiously, every edge-vertex pair represents about 20 nucleotides on average, so this suggests that we associate every vertex with a CP dimer binding site. Furthermore, since the duplexes are rigid and the loops are semiflexible this branched tree can explore many, non-self-intersecting (the excluded volume effect) spatial configurations. Taken all together, we represent single-stranded RNA as a branched tree with exluded volume, whose vertices represent particle binding sites, as a model for RNA bound with protein. As before with the linear polymer model, we will need to generate a thermal ensemble of configurations of the branched tree dressed with interacting bound particles. In the next section, we describe how to generate the ensemble by adapting the previous method for linear polymers to branched polymers.

3.2.2 Rosenbluth method

We will use the same notation as in 3.1.2. Our goal is to generate an ensemble of conformations of a polymer with an arbitrary branching pattern comprised of L vertices with M CP bound to its backbone, with $f = \frac{M}{L}$ denoting the overall fraction of occupied vertices. Using the branched polymer in Fig. 3.4 as an illustrative example, we first randomly choose one of its vertices (no matter of what order), e.g., vertex 1, place it at site $(x, y) = (0, 0)$ of the

2D square lattice, and with probability $f_1 = f = \frac{M}{L}$ populate this site with one CP. Next, in order to begin generating ("growing") an L-vertex, M-protein chain conformation, we randomly pick one of its bonded vertices, which in this example must be vertex 2, and note that there are now 8 possible states for it, corresponding to placing this vertex in any of the 4 vacant neighboring sites, in each case with or without a bound CP. The choice illustrated in 3.4, where vertex 2 is at $(x, y) = (0, 1)$ with a bound CP, is sampled with probability $\frac{\alpha f_2}{w_2}$ where $f_2 = \frac{(M-1)}{(L-1)}$, and $\alpha = \exp\frac{-\epsilon}{kT}$. The sum $w_2 = 4f_2\alpha + 4(1 - f_2)$ is conveniently referred to as the local partition function of vertex 2. We similarly proceed to the 3-fold coordinated vertex 3, which is further connected to two additional bonds leading to vertices 4 and 5. Again we randomly connect one of those to the partially formed tree; note, however, that if vertex 4 is chosen first then both vertices 3 and 4 are still not "saturated" and the next vertex (vertex 5, 6, or 7) may be connected to either of them; the choice is arbitrary. This procedure continues until all bonds are satisfied.

In more general terms, after placing $i - 1$ vertices of an arbitrary tree graph in positions $\vec{r}_1, \vec{r}_2, ..., \vec{r}_{i-1}$ there is always at least one unsaturated vertex (and when $i-1 = L-1$ there is only one such vertex). We add the next vertex by randomly choosing one of the unsaturated vertices, say vertex k whose lattice position is $\vec{r}_k$, and randomly pick one of its 4 neighboring sites $\vec{r}_k \pm \vec{e}_x$ or $\vec{r}_k \pm \vec{e}_y$, say $\vec{r}_k + \vec{e}_x$ ($\vec{e}_x$ is a unit vector along the x axis, etc.). Let $p_i = (\vec{r}_k + \vec{u})$ denote the joint probability of placing vertex i at $\vec{r}_k + \vec{u}$ (where $\vec{u} = \pm\vec{e}_x, \pm\vec{e}_y$) and finding it occupied by a CP, and let $q_i(\vec{r}_k + \vec{u})$ denote the joint probability of choosing the same site but leaving it empty. These probabilities are given by

$$p_i(\vec{r}_k + \vec{u}) = \frac{\theta(\vec{r}_k + \vec{u}) f_i \alpha^{n(\vec{r}_k+\vec{u})}}{w_i} \tag{3.14}$$

and

$$q_i(\vec{r}_k + \vec{u}) = \frac{\theta(\vec{r}_k + \vec{u})(1 - f)}{w_i}. \tag{3.15}$$

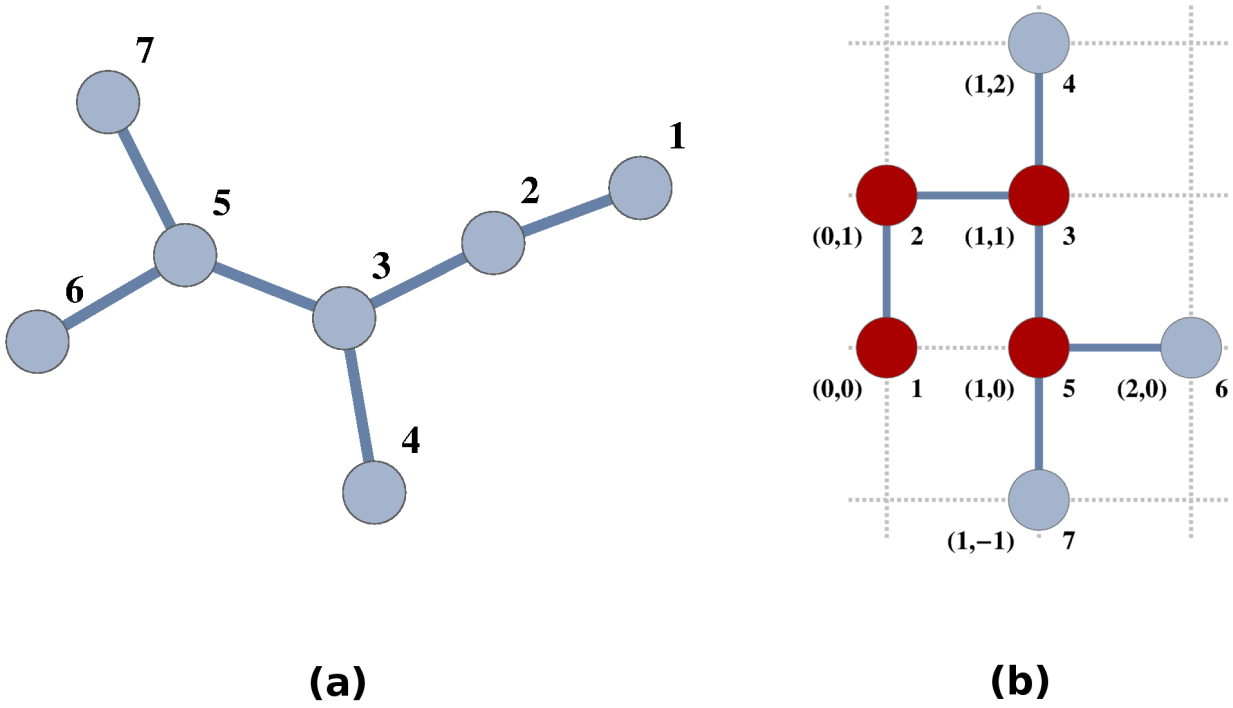

Figure 3.4: (a) A tree graph corresponding to the secondary structure of a small RNA molecule, with edges and vertices representing base-pair (bp) duplex stems and single-stranded (ss) loops, respectively. (b) A tertiary configuration of this tree graph, now with bound CP (red circles), on a 2D square lattice. The specified (x,y) coordinates define a particular tertiary configuration. With ϵ (< 0) the attraction energy between nearest-neighbor CP pairs, the energy of this particular configuration is 4ϵ.

Here $f_i = \dfrac{(M - M_{i-1})}{(L - i + 1)}$, with M_{i-1} denoting the number of CP already bound to the partially generated tree of $i - 1$ vertices and $\theta(\vec{r}_k + \vec{u}) = 1$ or 0 depending on whether site $\vec{r}_k + \vec{u}$ is available for accommodating vertex i or is already taken by a previous vertex, respectively. Let $n(\vec{r}_k + \vec{u})$ denote the number of CP occupying nearest-neighbor sites around $\vec{r}_k + \vec{u}$, the prospective site of vertex i. If the chosen site is already occupied we keep searching for a vacant one; once found we set $\vec{r}_i = \vec{r}_k + \vec{u}$ and continue to place vertex $i + 1$. The local

partition function associated with vertex i is

$$w_i = \sum_{\vec{u}} \theta(\vec{r}_k + \vec{u})[f_i \alpha^{n(\vec{r}_k+\vec{u})} + (1 - f_i)]. \tag{3.16}$$

Continuing to vertices $i+1, ..., L$ we end up generating a CP-dressed tree graph configuration Ψ specified by the spatial coordinates of the L vertices, $\vec{r}_1, ..., \vec{r}_L$, and their respective CP occupancies. The total CP-CP interaction energy in this conformation is $E(\Psi) = n(\Psi)\epsilon$, where $n(\Psi) = \dfrac{\sum\limits_i n^i_\Psi(\vec{r}_i)}{2}$ is the total number of nearest-neighbor CP pairs. We disregard here the binding energy of the CP to the polymer backbone because in our simulation of the competition experiments the number of bound CP is fixed – only their distribution over the competing polymers is changing. The overall (often-called) Rosenbluth probability of generating an arbitrary conformation is

$$P_R(\Psi) = \frac{\alpha^{n(\Psi)} \prod\limits_{i=1}^{L} \eta_i}{W(\Psi)} = \frac{1}{\binom{L}{M}} \frac{\exp \frac{-E(\Psi)}{kT}}{W(\Psi)}, \tag{3.17}$$

with $\eta_i = f_i$ or $(1 - f_i)$ denoting the probabilities of finding vertex i populated by a CP or remaining vacant, respectively. The product of these functions does not depend on the order of binding the CP to M of the L vertices, and its numerical value is the inverse of the number of such sequences, i.e., $\dfrac{1}{\binom{L}{M}}$. Note, however, that this (inverse) binomial factor does not imply that the M CP are randomly distributed over the L vertices; their distribution is dictated by Eq.s (3.14) and (3.15). The denominator in Eq. (3.17), $W(\Psi)$, is the generalized Rosenbluth factor, defined [Frenkel 1996][Rosenbluth 1955] by

$$W(\Psi) = \prod_{i}^{L} w_i(\Psi) \tag{3.18}$$

with w_i given by Eq. (3.16).

Repeated applications of the procedure above yields the ensemble of conformations of the

CP-dressed polymer, $\{\Psi\}$, which we use to calculate the averages χ of relevant properties of interest, such as the radius of gyration, R_g, or the CP-CP interaction energy, E. Note, however, that the conformations are sampled according to their Rosenbluth Monte Carlo (RMC) probabilities in Eq. (3.17), rather than in proportion to their Boltzmann weights in the canonical ensemble $\exp[\frac{-E(\Psi)}{kT}] = \binom{L}{M} W(\Psi) P_R(\Psi)$. The proper thermodynamic average of a property χ is thus given by

$$\langle \chi \rangle = \frac{\sum\limits_{\Psi} \chi(\Psi) W(\Psi)}{\sum\limits_{\Psi} W(\Psi)} \tag{3.19}$$

where the sum runs only over the sub-ensemble of conformations sampled by the RMC algorithm. Finally we note that the denominators in the last equation are proportional to the partition function of the system, which in the RMC ensemble of conformations is given by

$$Q(L, M, T) = \binom{L}{M} \sum_{\Psi} W(\Psi; L, M, T). \tag{3.20}$$

The numerical value of Q is proportional to the number of configurations sampled. However, this number is irrelevant to our simulations of the competition experiments (i.e., it cancels out) because we are only interested in ratios of partition functions corresponding to polymers with the same L and M.

3.3 Future work

3.3.1 Prune-enriched Rosenbluth method (PERM)

One of the main drawbacks of the Rosenbluth method is that the distribution of $W(\Psi)$ is so wide that a few, rare chains dominate the sample. More explicitly, the variance of the average of any property χ, computed using Eq. (3.19), is

$$\mathrm{Var}(\langle\chi\rangle) = \mathrm{Var}(\chi)\frac{\sum_{\Psi} W^2(\Psi)}{(\sum_{\Psi} W(\Psi))^2} = \mathrm{Var}(\chi)\frac{\langle W^2\rangle_R}{M\,\langle W\rangle_R^2}, \tag{3.21}$$

where $\langle ...\rangle_R$ denotes using $P_R(\Psi)$ to average over the M configurations generated using the Rosenbluth method. Thus the error of any average property computed using the Rosenbluth method increases with the width of the $W(\Psi)$ distribution.

The main idea behind the Prune-enriched Rosenbluth method (PERM) is to shrink the width of the $W(\Psi)$ distribution by eliminating $W(\Psi)$s that are too large or too small – thereby attenuating the error in the average without introducing any new biases [Grassberger 1997, Grassberger 2002]. This is implemented by generating a configuration Ψ using the method above (see section 3.2), and whenever the partial weight $W_n(\Psi) = \prod_{i=1}^{n} w_i(\Psi)$ $(n < L)$ exceeds some upper threshold value $W_n^>$ we *enrich* our sample by introducing a copy of configuration Ψ, while simultaneously reducing the value of $W_n(\Psi)$ by a factor of two. We similarly define a lower threshold value $W_n^<$ and *prune* (i.e. eliminate) every second configuration if $W_n(\Psi) < W_n^<$; at the same time we double the partial weight of the other configuration. In its simplest form, one chooses the limits of $W_n^>$ and $W_n^<$ in advance. More sophisticated implementations are done by modulating $W_n^>$ and $W_n^<$ "on the fly", and the best strategy in PERM is largely determined by optimizing the choice in $W_n^>$ and $W_n^<$. As discussed elsewhere [Grassberger 1997, Grassberger 2002], a good place to start is with the original Rosenbluth method by initially setting $W_n^< = 0$ and setting $W_n^>$ to a very large number, preventing pruning and enriching. Once the first chain of full length is generated, one switches to $W_n^> = c^> Z_n$ and $W_n^< = c^< Z_n$, where $Z_n = \frac{1}{m}\sum_{\Psi_{i=1}}^{\Psi_m} W_n(\Psi_i)$ is the "partition" sum of m configurations function and $c^>, c^<$ are constants of order unity.

Chapter 4

Theory of protein binding competition experiments

4.1 Introduction

In this chapter we argue that the competition between different RNAs for CP is determined by the differing extent to which they bind CP. For a given RNA mass, we propose that the sequence with the highest affinity for protein is the one with the most compact secondary structure arising from self-complementarity; similarly, a long RNA steals protein from an equal mass of a shorter one. In both cases it is the lateral attraction between bound proteins that determines the relative CP affinities of the RNA templates, even though the individual binding sites are identical. We demonstrate this with Monte Carlo simulations, generalizing the Rosenbluth method for excluded-volume polymers to include branching of the polymers and their reversible binding by protein.

4.2 Experimental background

Nucleocapsids are prepared in vitro via a two-step assembly reaction. In the first step, RNA and capsid protein (CP) are mixed together in a pH 7 solution of low ionic strength, whereby

virtually all the CPs are bound to RNA. At this stage, CPs weakly interact with each other in the following sense: though CP-CP interactions exist, they aren't strong enough to form a nucleocapsid. In the second step, the pH is lowered to 5 resulting in RNAase-resistant nucleocapsids.

CPs exist as dimers (CP_2) and these dimers are the basic assembly unit of the capsid. Each CP_2 contains 20 positive charges which bind, non-specifically, to about 20 nucleotides. All the RNA ends up encapsidated, after the pH lowering step, when the total charge of the N-terminal tails of the CP_2s is as large as the (opposite) charge on the RNA. The competition experiments [Comas-Garcia 2012] are designed to exploit this unique CP_2:RNA mol ratio so that CP_2 must "choose" which RNA to package.

In a competition experiment, two RNAs of different length are pitted against each other in a solution containing enough CP_2 to package *one* but not both RNAs. In the canonical competition experiment, Brome Mosaic Virus (BMV) RNA 1 (B1) serves as the defending champion, while shorter RNAs serve as the challengers. B1 is 3.2 kb in length; challengers whose lengths are less than 2 kb are not even packaged (i.e. they do not compete at all!). Challengers longer than or equal to 2 kb are able to compete, but are encapsidated less efficiently. Furthermore, regardless of the order in which the champion and the challenger are mixed at neutral pH, the result is the same: B1 wins. For example, suppose a pH 7 solution containing one of the challengers and CP_2 is allowed to incubate for 24 hr before B1 added. After the pH is lowered, the result is identical to an acidified solution initially containing all three components –B1 always wins to the same extent. We show the results of this "reversibility" experiment in Fig. 4.1. From chapter 2, we understand that a lone sharp band in the Gel Electrophoretic Mobility Assay (GEMA) from an assembly experiment implies (i) a nucleocapsid and (ii) the CP_2:RNA ratio was close to the magic ratio. This means that over the course of the experiment B1 must have stolen a substantial amount of protein from its competitor.

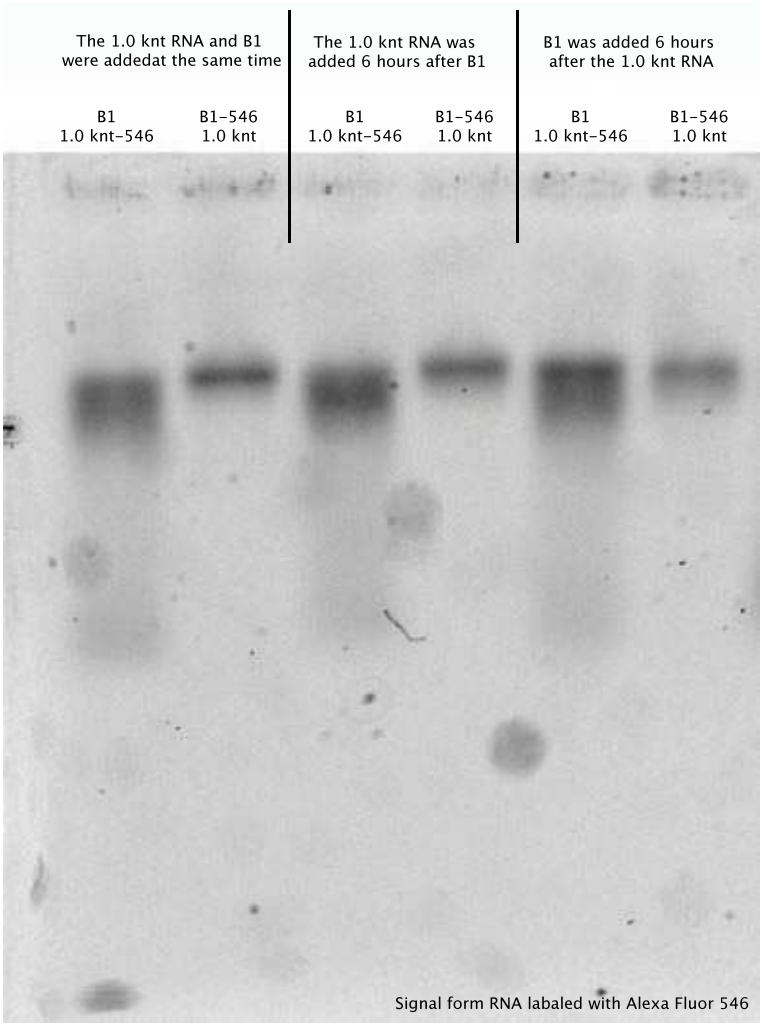

Figure 4.1: B1 is completely packaged regardless of the order which it is added to the pH 7 solution, as shown in the EMAs of the competition experiment between B1 and 1 kb RNA. The RNA was visualized using AlexaFluor 546 fluorescently labeled Uracil; on average, there was one flourescent label per RNA molecule. The result of the competition experiment when both RNAs were added at the same time, when the 1 kb RNA was added 6 hrs after B1, and when B1 was added 6 hrs after the kb RNA, are shown in the left-most pair, middle pair, and right-most pair of lanes. Each pair of lanes involved the same competitors and the same order of (B1) addition, but with the flourescent label on one or the other of the two RNAs. Reproduced courtesy of Dr. Mauricio Comas-Garcia.

4.3 Theory

To simulate the competition for binding of CP between long vs. short RNAs, or branched vs. linear RNAs, or compact vs. extended branched RNA molecules, we use the simplest model that captures the essential qualitative aspects of this phenomenon. A basic premise of the model is that CP binding does not affect the secondary structure of the RNA molecule. On the other hand, attractions between proteins bound to nearest-neighbor sites will be a dominant factor in determining the tertiary structure of the RNA, i.e., its configuration in 3D space. As in several previous studies[Gan 2004, Fang 2011, Yoffe 2008] the secondary structures of the ssRNA molecules will be mapped onto their tree graph representations, whereby base-pair (bp) duplexes are treated as rigid edges (all of the same length) and the single-stranded (ss) loops (the tree vertices) connecting them are regarded as flexible joints. The basic unit in the branched polymer is a duplex-stem (edge) and its attendant ss-loop (vertex). For computational reasons the largest tree graphs considered in this work are composed of 50 stem-loop pairs, corresponding to RNA chains of about 1000 nt in which typically about 60% of the nt are paired in duplexes whose average length is about 5 bp.

When we compete RNA molecules of different length against each other we attribute to them the same branchedness, i.e., the same relative numbers of 1-fold vertices (hairpin-loops), 2-fold vertices (connecting only two duplexes), and of three- and higher-order branch points. In this way we can focus on the effect of different numbers of binding sites on the ability of a molecule to compete for capsid proteins. In the same vein, when we compete equal-length RNA molecules, we either attribute to them different distributions of vertex orders (e.g., as in the case of a branched vs. linear RNA) or we keep the vertex-order distributions the same but scramble the vertices so that the molecules are more extended or more compact.

Finally, the tertiary structures of the tree graphs will be represented by embedding them with different configurations on a two-dimensional (2D) square lattice. While motivated by computational simplicity, this limitation to 2D structures is not unreasonable considering that the RNA backbone of viral capsids serves as the template for the nucleation of a 2D

(albeit curved) protein shell protecting the genomic material. The use of a square lattice, where the maximal vertex order is 4, is not a severe restriction in view of the fact that fifth- or higher- order vertices in RNA secondary structures are very rare [Gopal 2014]. Accordingly, in translating the original RNA sequences to tree graphs we have counted all loops of order five or larger as 4th-order vertices. We note also that 4 is also the number of contacts per dimer in the 180-CP capsids of CCMV.

Recall, each of the CP-dimer building blocks is attracted non-specifically to the negatively charged RNA genome through the two cationic N-terminal arms of its constituent monomers, totaling 20 positive charges. This number, 20, is also the average number of nt negative charges per stem-loop pair because, on average, the RNA duplexes consist of 5bp (hence 10 nt) and ss-loops typically contain 10nt. Furthermore, the physical size ("footprint") of a stem-loop pair is also comparable to that of a CP-dimer. Thus, at full coverage ("saturation") the number of CP dimers bound to an RNA molecule equals its number of stem-loop pairs: each vertex-edge pair in our tree-graph lattice model is a potential site for the reversible binding of one CP dimer (hereafter simply CP), as illustrated in Fig. 4.2.

Our model assumes attractive CP-CP interactions between CP pairs occupying nearest-neighbor (NN) sites, whether bonded by a stem (see, e.g., vertices 1 and 2 in Fig. 4.2) or not (e.g., vertices 1 and 4). On energetic grounds these attractive interactions obviously favor compact conformations of the CP-dressed branched polymer, which on the other hand are generally disfavored entropically. In our Monte Carlo (MC) simulations of the CP-dressed polymers we allow for conformational changes of the branched polymer, as well as for rearrangements of the reversibly-bound CP on the branched tree backbone, enabling the structure to reach thermodynamic equilibrium. Only self-avoiding polymer conformations are allowed, thus respecting excluded-volume interaction.

As discussed above, we explicitly allow for changes in the *tertiary* structure of the tree graph representations of the RNA, which arise due to the lateral interactions between the bound particles. Note that the connectivity of the tree graph does not change, even as it is

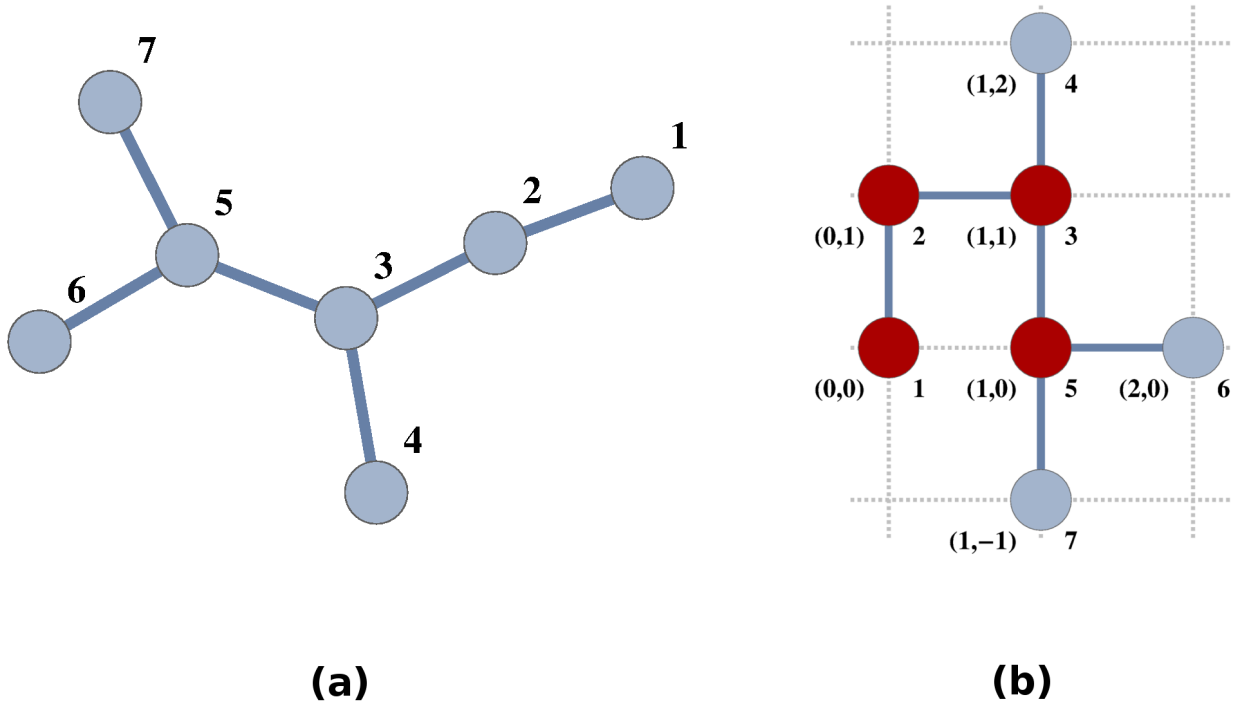

Figure 4.2: (a) A tree graph corresponding to the secondary structure of a small RNA molecule, with edges and vertices representing base-pair (bp) duplex stems and single-stranded (ss) loops, respectively. (b) A tertiary configuration of this tree graph, now with bound CP (red circles), on a 2D square lattice. The specified (x,y) coordinates define a particular tertiary configuration. With ϵ (< 0) the attraction energy between nearest-neighbor CP pairs, the energy of this particular configuration is 4ϵ

configured differently on the lattice in which it is embedded (see Fig. 4.2). Recalling how the tree-graph connectivity is determined directly from the naked RNA secondary structure, we are effectively neglecting changes in secondary structure due to interaction of the RNA with its capsid protein. A recent study of satellite tobacco mosaic [Zeng 2012] virus, for example, suggests significant differences between the secondary structure of its naked RNA and that of the genome in its mature nucleocapsid. In the present work, on the other hand, where we treat the initial binding of capsid protein by RNA – instead of the formation of a complete viral nucleocapsid – we proceed with the simplifying assumption that the dominant effect of

capsid protein is to impose tertiary organization on the RNA secondary structure.

Our statistical thermodynamic simulations of CP-RNA binding patterns and competition experiments include the following cases:

(1) One long RNA molecule, represented by a 50-vertex tree graph, competing for 50 CP particles against 2 shorter, 25-vertex, molecules. The long and the short tree graphs associated with these molecules each have the same vertex-order distribution, corresponding (as explained below in section 4.5.1) to that of random RNA sequences with uniform nt composition (i.e., equal numbers of A, U, G, and C).

(2) Two 50-vertex molecules sharing 50 CP and –separately– two 25-vertex molecules sharing 25 CP. Here the idea is to investigate the possibility of unequal binding of CP by identical branched polymers. For these studies we again use the branching pattern associated with random sequences and uniform nt composition.

(3) A compact 50-vertex polymer competing with an extended 50-vertex polymer for 50 CP. The vertex-order distributions of the compact and extended trees are identical, but their radii of gyration are markedly different. The procedure for generating these compact and extended trees, involving the repositioning of branch points within a tree graph, is outlined in section 4.5.3.

(4) The same compact, 50-vertex, branched polymer as in point 3, competing for 50 CP against a 50-vertex linear polymer.

4.4 Thermodynamic/Kinetic Monte Carlo simulations

We have used two complementary procedures to simulate the competition experiments, hereafter referred to as *thermodynamic* and *kinetic* simulations, respectively. In both approaches, in analogy to a previous extension [Tzlil 2005] of the Rosenbluth Monte Carlo (RMC) [Rosenbluth 1955, Frenkel 1996] algorithm, we first generate representative ensembles of low free energy configurations of the relevant CP-dressed polymer – e.g., 1000 con-

formations of the 50-vertex tree, dressed with M CP. For each of these ensembles (i.e., for each value of M between 0 and 50) we calculate properties of the dressed polymer, such as its radius of gyration, as well as its partition function and thus any desired thermodynamic function. In the thermodynamic simulations, we use these partition functions to evaluate the probability of any division of the $M = M_1 + M_2$ proteins between the competing polymers (e.g., the linear vs. branched polymers), thus obtaining the equilibrium distribution of bound proteins and the winner of the competition, if any. The kinetic simulations employ a variant of the Metropolis algorithm [Frenkel 1996, Metropolis 1953] whereby CP are exchanged between the competing polymers, allowing for concomitant changes in the spatial conformations of both polymers. The configurations of the competing polymers are sampled from the ensembles of configurations generated by our RMC procedure. To satisfy the principle of detailed balance the exchange probabilities in these simulations are weighted according to the joint partition functions of the competing structures before and after the exchange, rather than simply their Boltzmann weights. Below we outline our RMC procedure for generating CP-dressed polymers. Their use in the thermodynamic and kinetic approaches is described in sections 4.4.1 and 4.4.2, respectively.

4.4.1 Thermodynamic simulations

Consider a system of two polymers P_1 and P_2 of equal length, L , competing for the binding of M capsid proteins, $(M < L)$. At equilibrium, the probability of finding M_1 proteins on P_1 and $M_2 = M - M_1$ on P_2 is

$$P(M_1; M) = \frac{Q_1(M_1)Q_2(M - M_1)}{Q_{tot}(M)}, \tag{4.1}$$

with $Q_{tot}(M) = \sum_{M_1=0}^{M} Q_1(M_1)Q_2(M - M_1)$ denoting the total partition function of the system in equilibrium. The Helmholtz free energy of the system is thus $A_{eq}(M, L) = -kT \ln Q_{tot}(M, L) \approx -kT \ln[Q_1(M_1^*)Q_2(M_2^*)]$, where the second near-equality is based on the

maximum term approximation, with $M_1^*, M_2^* = M - M_1^*$ denoting the most probable distribution, namely, the maximal $P(M_1; M)$, reflecting the most probable division of the CP among the two proteins. For very large values of M the most probable populations, M_1^* and M_2^*, are practically identical to the average equilibrium values $\langle M_1 \rangle = M_1^{eq} = \sum_{M_1=0}^{M} M_1 P(M_1; M)$ and $\langle M_2 \rangle = M_2^{eq} = \sum_{M_1=0}^{M} (M - M_1) P(M_1; M)$, respectively. For the finite, yet large, values of M in our simulations this is a very good approximation.

We start the competition experiments with an arbitrary initial state where the M proteins are divided between the two species –M_1^i on P_1 and $M_2^i = M - M_1^i$ on P_2 – and then let the system relax to the equilibrium value M_1^*, M_2^*. In our thermodynamic simulations we first generate RMC ensembles of (generally about 1000) polymer-CP conformations, for all $L \geq M_1 \geq 0$ and $L \geq M_2 \geq 0$. Using Eq. (3.20) we calculate the partition functions $Q_1(M_1)$ and $Q_2(M_2)$, and hence the free energy difference between any two states, i and f, corresponding to different divisions of the M proteins between P_1 and P_2:

$$\Delta A = A_f - A_i = -kT \ln \frac{P(M_1^f; M)}{P(M_1^i; M)}. \tag{4.2}$$

In particular, the free energy change from an arbitrary initial state to the equilibrium one is

$$\Delta A = A_{eq} - A = kT \ln P(M_1^i; M) \approx -kT \ln \frac{P(M_1^*; M)}{P(M_1^i; M)}. \tag{4.3}$$

Using Eq. (3.20) we also calculate the average energies of the two competing species, $\langle E_1(M_1) \rangle$, $\langle E_2(M_2 \rangle$ and hence,

$$\Delta E = [\left\langle E_1(M_1^f) \right\rangle + \left\langle E_2(M_2^f) \right\rangle] - [\langle E_1(M_1^i) \rangle + \langle E_2(M_2^i) \rangle]. \tag{4.4}$$

ΔS can now be derived from $T\Delta S = \Delta E - \Delta A$. Finally, as a measure of the compactness of any tree of interest, naked or CP-dressed, we calculate its average radius of gyration using Eq. (3.19) with $\chi(\Psi) = R_g(\Psi)$ and

$$R_g^2(\Psi) = (\frac{1}{2L^2}) \sum_{i,j} [r_i(\Psi) - r_j(\Psi)]^2, \tag{4.5}$$

with $r(\Psi)$ denoting the position of vertex i in configuration Ψ.

The same simulation procedure can be applied to model the exchange of CP between polymers of different length. In our numerical calculations we have only considered the case where a polymer of length L competes for M CP ($0 \leq M \leq L$) with two polymers of length $\frac{L}{2}$, again generating large ensembles of CP-dressed configurations of both species. Using M_1 to denote the number of proteins on the large polymer, and M_2 and M_3 on the two short polymers, the generalization of Eq. (4.1) to this particular case is

$$P(M_1; M) = \frac{Q_1(M_1) \sum_{M_2=0}^{M-M_1} Q_2(M_2) Q_3(M - M_1 - M_2)}{Q_{tot}(M)}, \tag{4.6}$$

with $Q_{tot}(M)$ the normalizing factor. All relevant thermodynamic and structural properties can be derived in analogy to the previous case.

4.4.2 Kinetic simulations

In this mode of simulation we again begin with an arbitrary initial state with M_1 CP bound to P_1 and M_2 to P_2, both of length L, but now let the system reach equilibrium through CP exchange between the two polymers, coupled with simultaneous conformational changes of the polymers. This procedure resembles the familiar MC simulations involving particle (i.e., CP) exchange except that the probabilities for moving from one state to another are not governed by their relative Boltzmann weights according to the Metropolis criterion. Instead, detailed balance is ensured and equilibrium is reached through attempted moves of CP from one polymer to another, with both the initial and final configurations randomly chosen from the ensembles that we have already generated using the Rosenbluth MC algorithm described in 3.2.

Explicitly, each step begins with a random choice of one of the M CP particles and an attempt to move it from one tree to the other. If this particle happens to be on P_1 we randomly select one conformation from the previously generated ensemble of conformations of P_1 with M_1 bound CP, and one conformation of P_2 with M_2 bound CP, and attempt a move ending up with another (randomly sampled) conformation of P_1 from the ensemble of conformations of P_1 with $M_1 - 1$ CP and a (randomly sampled) conformation of P_2 from its ensemble of conformations with $M_2 + 1$ CP. Recalling that CP-polymer configurations in the RMC ensembles were not randomly sampled, but rather according to their partition functions, the move is accepted or rejected according to a Metropolis criterion using the partition functions rather than the Boltzmann weight of the individual configurations involved. In other words, since the probability of any state of the two-polymer system with a given distribution of the CP among them is given by (4.1), the forward and backward rates of transferring one CP from P_1 to P_2 are related by the modified detailed-balance ratio

$$\frac{\overrightarrow{k}(M_1 \to M_1 - 1)}{\overleftarrow{k}(M_1 - 1 \to M_1)} = \frac{Q_1(M_1 - 1)Q_2(M - M_1 + 1)}{Q_1(M_1)Q_2(M - M_1)}. \tag{4.7}$$

A trial move for transferring a particle from P_1 to P_2 in our kinetic simulations is thus accepted with probability

$$acc(M_1 \to M_1 - 1) = \min[1, \frac{Q_1(M_1 - 1)Q_2(M - M_1 + 1)}{Q_1(M_1)Q_2(M - M_1)}]. \tag{4.8}$$

4.5 Results

To explain the viral assembly competition experiments, the first set of results presented in this section aims to understand why long RNAs steal capsid proteins from shorter RNAs. To this end, in section 4.5.1, we consider the competition between a 50-vertex tree and two 25-vertex trees. In section 4.5.2 this computer experiment is contrasted with one where two identical branched RNAs compete with each other. In section 4.5.3 we consider the

competition between two trees with the same vertex-order distributions but with markedly different radii of gyration, i.e., one is significantly more compact than the other. In section 4.5.4, to further examine the role of branchedness in the competition experiments, we treat the competition between a branched tree and a linear tree of the same length. Finally, in section 4.5.5 a few comments will be made regarding the kinetics of CP redistribution.

Unless specifically stated otherwise, we use a CP-CP interaction energy of $\epsilon = -3$ kT in all simulations, consistent with estimates of the attractive energy between CCMV dimers [Johnson 2005]. For all the competition experiments analyzed below we have carried out both thermodynamic simulations, as well as kinetic – particle exchange – simulations. The kinetic simulations were sampled every 1000 MC particle exchange steps and generally involved at least 600,000 steps in total, well beyond the time scale needed for the system to reach equilibrium. (While we cannot relate the MC step times to real-time events, our kinetic simulations nevertheless shed light on the relative time scales of the various processes involved.) Temporal evolution profiles of CP populations will thus be shown for shorter time scales. The thermodynamic simulations are computationally intensive, and are based on ensembles of $\sim$ 1000 RMC configurations for each of the tree graphs involved in the competition simulations. From the thermodynamic simulations we present the probability distributions of CP among the competing trees, all showing excellent agreement with the kinetic population profiles. Detailed thermodynamic analyses of energies, free energies and entropies of the competing trees are included in the Supporting Information (see App. C).

4.5.1 A long tree vs. two short trees

In this simulation an $L = 50$ tree (see center of Fig. 4.3) competes for 50 CP with two half-size $L = 25$ trees (depicted on the left and right). The large and small trees are randomly branched polymers, both derived using the same vertex order distribution, as previously determined from analyses of many secondary structures of long random RNA sequences [Gopal 2014]. A pictorial summary of this simulation is presented in Fig. 4.3,

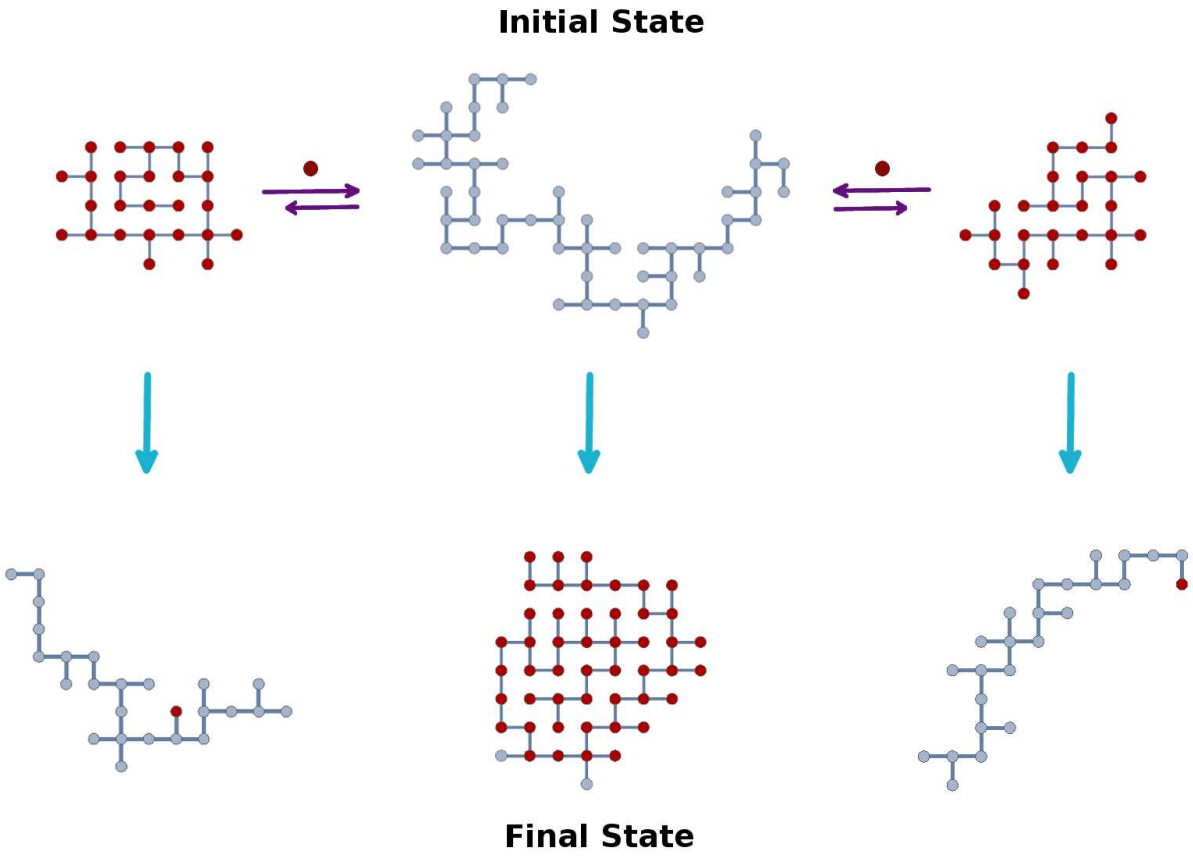

Figure 4.3: Snapshots from simulation of an experiment where initially all CP are bound to two small trees. Once equilibrium is reached nearly all CP are bound to the large tree.

showing snapshots from: the initial state of the system, where all CP are bound to the two small trees while the large tree is devoid of CP; and the final state where nearly all CP are on P_1 while the two P_2 trees are essentially stripped of all their CP. More precisely, as shown in Table 4.1 and illustrated in Fig. 4.3, when equilibrium is reached nearly all CP – $M_1 = \langle M_1 \rangle \pm \delta M_1 = 48 \pm 2$ out of 50 – end up on the large tree. The temporal changes in the distribution of the CP between the long and short trees is shown in Fig. 4.4, revealing the rapid establishment of the equilibrium distribution and the range of fluctuations around the average CP populations.

The insert in Fig. 4.4, a plot of $P(M_1; M)$ (see eq. (4.6) with $M = 50$), shows the

Comp. Exp. P_1 *vs.* P_2	**Initial →** **Equilibrium** **CP on:** P_1 & P_2	Δ A (kT)	Δ E (kT)	Δ S (k)	**Initial →** **Equilibrium** R_g P_1 & P_2
50 vs. 2 x 25	0 & 2 x 25 → 48 & 1 x 1 (50)	-29	-27	2	5.0 & 2.1 → 3.3 & 3.0
Compact vs. Extended	0 & 50 → 49 & 1 (50)	-11	2	13	3.2 & 3.2 → 3.0 & 5.2
Branched vs. Linear	0 & 50 → 48 & 2 (50)	-6	11	17	3.2 & 3.0 → 3.0 & 6.2

Table 4.1: Structure, energetics, and CP populations in competition experiments. P_1 and P_2 denote the trees competing for CP. The second column reports the number of CP on the competing trees in the initial (M_1^i , M_2^i , in the first row) and the final equilibrium states ($M_1^{eq} = \langle M_1 \rangle$, $M_2^{eq} = \langle M_2 \rangle$, in the second row). Indicated in parenthesis are the most propable CP populations on P_1 at equilibrium, M_1^* (50 in all cases). The third through fifth columns report the changes in free energy, energy, and entropy between the initial and the final equilibrium states. The sixth column tabulates the radius of gyration for each polymer initially and at equilibrium.

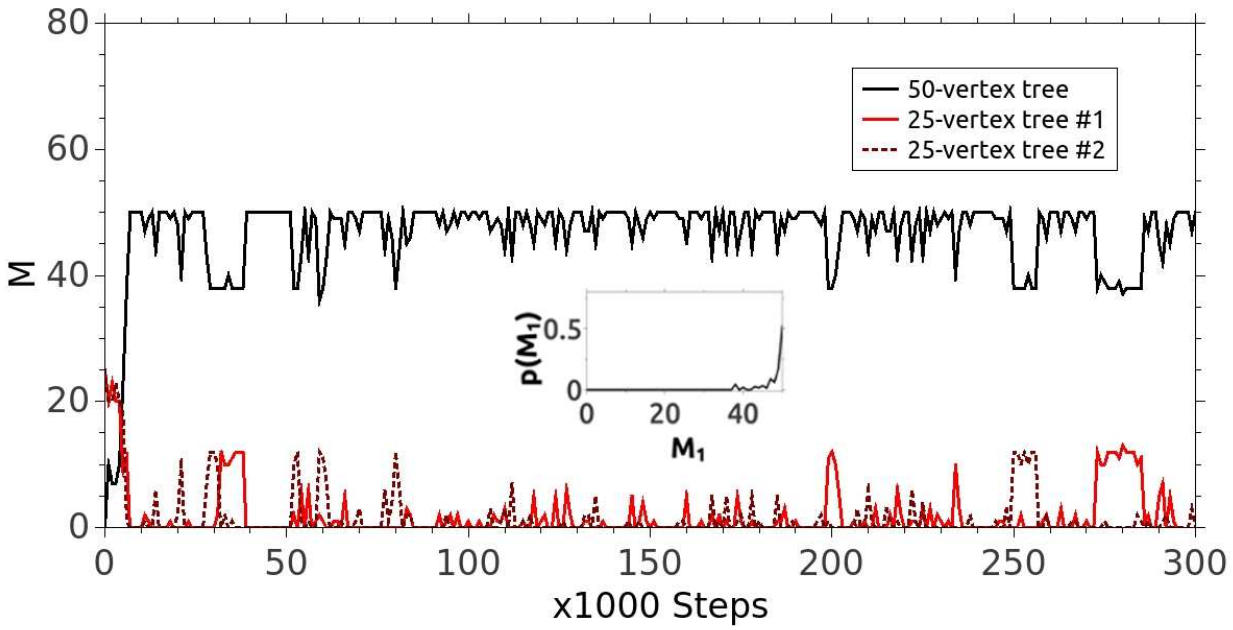

Figure 4.4: Temporal evolution of the CP populations on the large (black trace) and the two small (red and brown) trees, as obtained by the kinetic (particle exchange) simulations. The CP populations were sampled every 1000 steps. The inset shows the probability distribution of CP between the two trees, as derived from the thermodynamic simulations.

distribution of the number of particles (M_1) bound on the large tree, after the competition between the one 50-vertex and two 25-vertex trees has reached equilibrium. Note that virtually all of the 50 particles are bound to the large tree, even though they all started on the two small trees. This effect is also seen, although somewhat less strongly, when the lateral interaction energy between bound particles is weaker than the -3 kT value used here. More explicitly, values of this energy have been estimated [Zlotnick 2005] for CCMV CP at each of three pHs (4.75, 5.0, and 5.25), and a linear extrapolation of them to neutral pH gives a value of -1.67 kT. Using this for ϵ in our simulations, we find a $P(M_1; M)$ distribution that is qualitatively the same as that shown in the Fig. 4.4 insert for $\epsilon = -3kT$ – see Fig. C.6 in the Supporting Information, i.e., the long molecule takes significantly more than its share of the binding particles. We have used the larger value of ϵ to emphasize more clearly the effect

of polymer length on the competition for binding particles. Similarly, we use this value for reporting below the results of competitions between compact and extended branched trees, and between compact/branched and linear trees, where (see Fig.s C.7 and C.8) the smaller value of ϵ again gives qualitatively the same results for the $P(M_1; M)$ particle distributions, i.e., the compact/branched tree binds significantly more than its share of the particles.

The "asymmetry" of the sharing of particles by large and small trees, presenting equal numbers of equal-affinity binding sites, also depends on the ratio of the total number of particles to the total number of binding sites, i.e., on the overall "coverage". For example, the simulations described above (and presented throughout the rest of this work for competition between different kinds of trees) refer to experiments for which the number of capsid proteins is only sufficient to saturate each one or the other of two competitor RNA species, thus corresponding to an overall coverage of 1/2. Accordingly, the $P(M_1; M)$ distribution in the inset of Fig. 4.4 shows the results for the number of particles on the 50-vertex tree when a total of 50 particles is available to the large and small trees. This is the coverage that gives maximum asymmetry of particle sharing: clearly, as the coverage approaches 0 or 1 the asymmetry disappears. But at intermediate values of 1/4 and 3/4, for example, the asymmetry is still quite strong, as shown in Fig. C.9, where their P(M1;M) distributions are compared with that for 1/2. Zandi and van der Schoot [Zandi 2009] have emphasized the role of CP:RNA stoichiometry in a related but different context– namely, the crossover from smaller to larger capsid size as the overall CP:RNA molar ratio increases (see their Fig. 3). In their situation it is the preferred capsid curvature (size) that determines the scenarios for stoichiometry dependence, whereas in our case it is the length and branching of the RNA template.

Much as in an Ostwald ripening process, the primary driving force for the transition of nearly all CP from the two small polymers to a single large polymer is the energetic preference of forming one large nearly compact 2D island of CP particles rather than two smaller ones. Indeed, as reported in 4.1 for this particular competition "experiment", the "stealing" of

proteins by the large polymer is driven predominantly by energy, with only minor entropy changes. More explicitly, while the relatively high-energy perimeters of both the large and small CP islands are quite ramified, the overall "coastline" of the single large patch is smaller than the combined coastlines of the two smaller CP islands on the small trees. There is, however, one important difference between the present case and familiar ripening processes, such as the coagulation of liquid droplets in 3D [Voorhees 1992] or of adsorbate islands on 2D surfaces [Zinke-Allmang 1992]. Namely, the preferred aggregation of the CP on the larger tree is coupled to a simultaneous change in structure of the embedding substrates, i.e., their host RNA molecules.

In Table 4.1 we also note a small entropic contribution to the process illustrated in Fig. 4.3. One (minor) contribution to this entropy change is the balance of the conformational entropy loss experienced by the large tree upon its collapse following CP binding, and the concomitant entropy gain by the two small trees. (Note that conformational entropy changes can be significant when polymers of very different branching patterns compete with each other, as will shall see in sections 4.5.3 and 4.5.4). Another small entropic contribution is due to the translational ("mixing") entropy of the CP remaining on the small trees and of the vacancies in the large tree.

The establishment of binding equilibria is the result of multiple CP binding and unbinding processes. We note also that the migration of the CP from the two small trees to the large one is reminiscent of a 2D condensation transition. Indeed, were the CP adsorbed on an infinite 2D square lattice, the lateral attraction between neighbors, $\epsilon = -3$ kT, would be strong enough to drive a first-order 2D condensation transition, leading to the coexistence of a dense phase and a dilute gas phase of CP. In our case the condensed phase resides on the large tree where the CP can form a large island with minimal edge energy, with the few CP on the small trees constituting the dilute phase. We should nevertheless remember that, unlike a simple 2D lattice, the branched polymers serving as the substrates of the two phases are finite, highly ramified, and conformationally flexible objects.

4.5.2 Two identical trees

We have also studied the sharing of CP between a pair of identical small trees and between a pair of identical large trees. To this end we have carried out competition simulations involving two 25-vertex trees sharing 25 CP, and –separately– two 50-vertex trees sharing 50 CP. In the latter case, the analogy to a 2D condensation transition is even more compelling than in the previous simulations. The two 50-vertex tree system undergoes rapid phase separation with the majority of CP residing on one tree, enjoying low energy, while the remaining CP form an entropy-rich dilute gas phase on the other tree. As shown in Fig. 5, starting with half of the CP population in each of the two 50-vertex trees, symmetry breaking soon takes place with most CP (about 90%) settling on one tree, coexisting with a dilute CP population on the other tree. Although the 50-vertex tree is of finite size and (even in its most compact form) does not provide a perfect square lattice, the CP-CP interaction energy of $\epsilon = -3$ kT –which is substantially stronger than the critical interaction energy for the condensation transition of a 2D lattice gas on a square lattice ($\epsilon = -1.76$ kT) –is strong enough to drive the phase separation of the bound CP, despite the imperfect lattice provided by the underlying tree. We also note that once symmetry is broken, it stays that way on the time scale of our simulations.

The competition between the two shorter, 25-vertex, trees reveals a qualitatively different behavior. Here the imperfection of the underlying lattices spanned by these short trees, and hence the smaller (average) number of nearest neighbor CP-CP contacts, does not suffice to induce a kinetically-stable phase separation. What we see instead is a frequent swinging of the majority of CP between the two trees, with roughly 70% of the CP on one tree and the rest on the other. This behavior is clearly consistent with the rapid migration of CP from the two short trees to the long one in the competition between the 50-vertex tree and the two 25-vertex trees. The equilibrium distributions for the competition experiments involving two identical trees, as well as simulation snapshots of initial and final CP and tree conformations, are shown in the Supporting Information (see App. C).

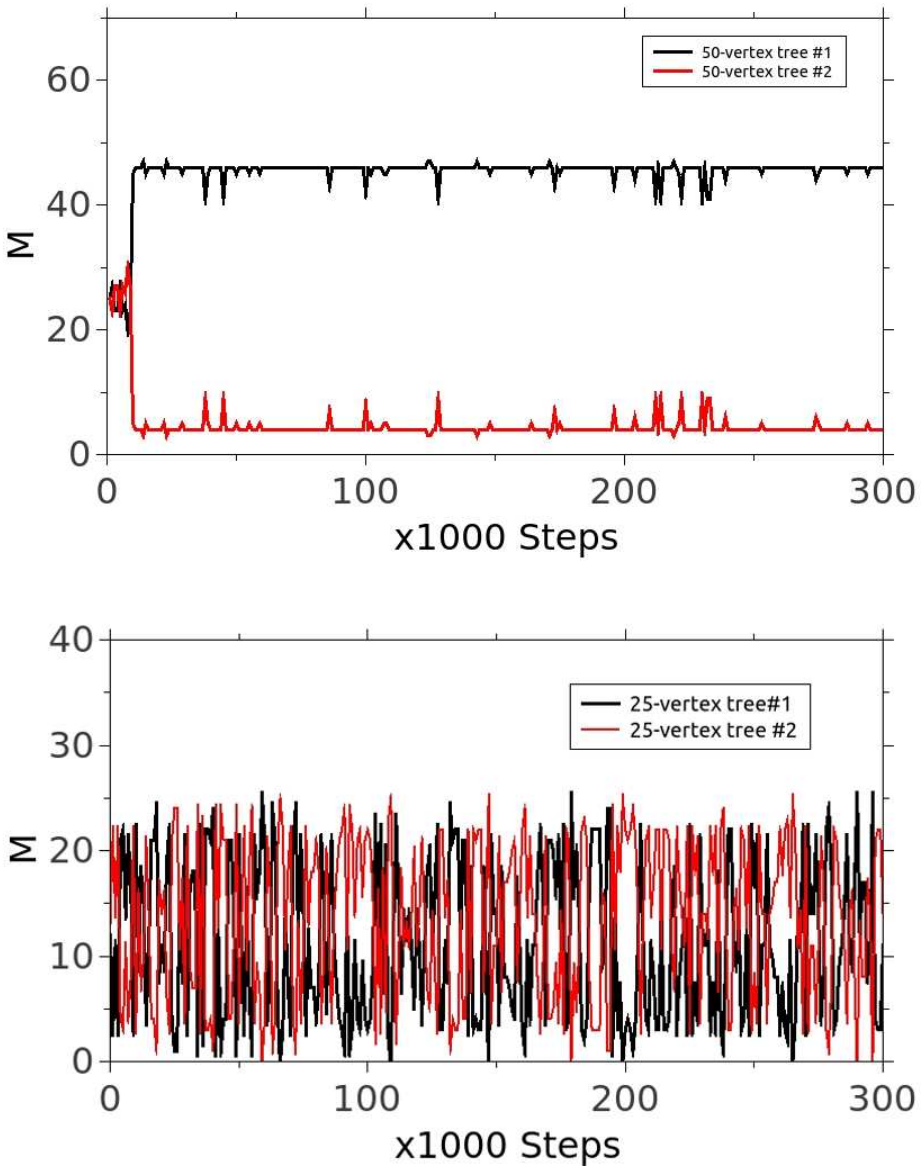

Figure 4.5: Time dependence of the CP population exchange between two identical trees. Populations were sampled every 1000 steps. Top: Starting with the equal sharing of 50 CP among two 50-vertex trees, phase separation occurs rapidly, with the majority of CP condensing on one tree coexisting with a dilute CP population on the other. Bottom: Phase separation also takes place in the two 25-vertex-tree system, but owing to the small system size the CP population swings often between the two trees.

4.5.3 Compact tree vs. extended tree

In this section we consider the competition for CP between two branched polymers having the same number of vertices and the same distribution of vertex orders, yet markedly different radii of gyration. Qualitatively, as can be seen in Fig. 4.6, the high-order junctions of the extended polymer reside near its ends, whereas in the compact polymer they are located centrally. More specifically, as in the previous subsections, the vertex-order distributions of both the extended and the compact trees considered are those of tree graphs corresponding to random RNA sequences. However, the compact and extended trees are those with smallest and largest possible Rg , respectively, among the trees with this vertex-order distribution. To derive their structure we have first numerically labeled all vertices of an arbitrarily-chosen 50-vertex tree having the given vertex-order distribution, and have assigned to it its unique Prüfer sequence [Pruefer 1918]. A key property of Prüfer sequences and their one-to-one unique mappings onto branched tree graphs is that any permutation ("Prüfer shuffling") of the sequence – while generating a new tree graph connectivity – leaves the vertex-order distribution invariant. Accordingly, we have repeatedly shuffled the starting sequence and for each outcome regenerated the corresponding tree graph and calculated its R_g using Eq.s (3.19) and (4.5). The compact and extended trees used in our simulation represent the trees with smallest and largest radius of gyration obtained in this way.

As indicated in Table 4.1, the transfer of CP from the extended to the compact tree is driven by the conformational entropy gain of the extended tree, as could be expected in view of its similarity to a linear polymer. From Fig. 4.7 we note a short-lived metastable state (in the time range of $\sim 10,000 - 30,000$ steps) on the way to a stable equilibrium (at $\sim 35,000$ steps) where nearly all CP remain bound to the compact tree, except for small occasional fluctuations.

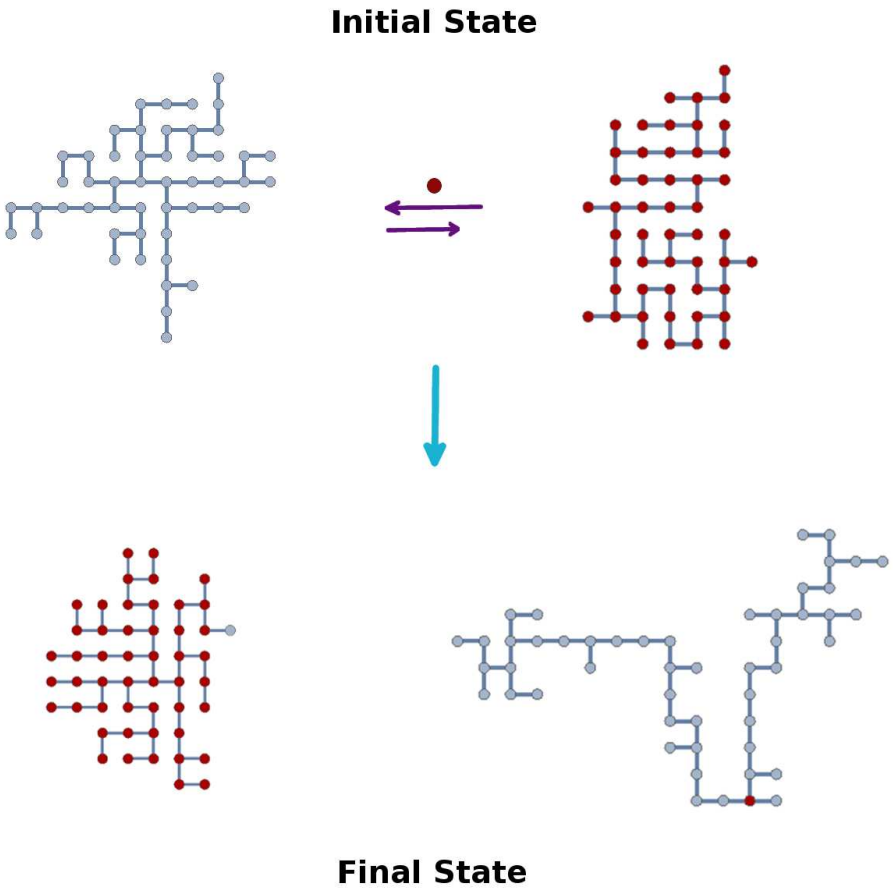

Figure 4.6: Snapshots from the initial and final equilibrium states obtained in the simulation of the competition between the compact (left) and extended (right) trees. Practically the entire CP population is eventually found on the collapsed compact tree.

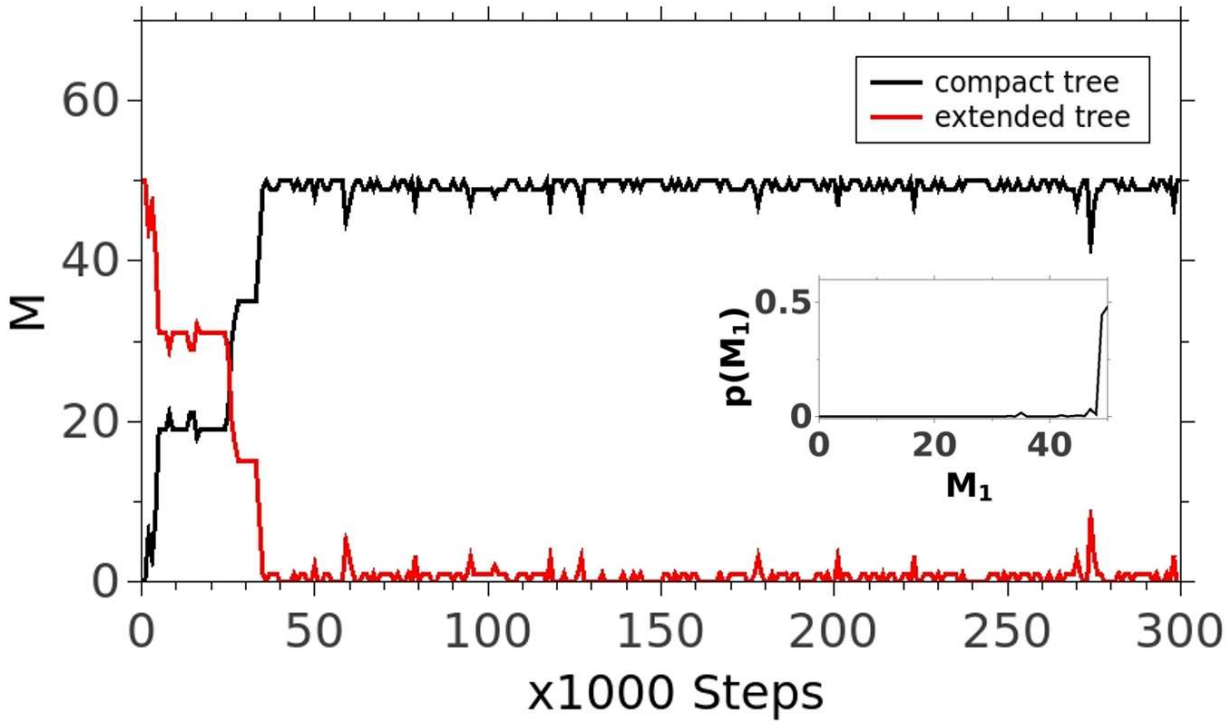

Figure 4.7: Temporal evolution of the CP sharing between the compact (black trace) and the extended (red) tree. The inset shows the probability distribution of CP between the two trees, as derived from the thermodynamic simulations.

4.5.4 Branched tree vs. linear tree

In this section we simulate the competition for CP between a linear and a branched polymer of the same length. The main purpose of these simulations is to accentuate the thermodynamic consequences of RNA branchedness with regard to protein binding. The linear polymer may be regarded as representing a non-folding RNA, e.g., polyU, for which no secondary structure arises. The branched tree used in the present simulations is identical to the compact tree of the previous section.

The simulations mimic an experiment where L-length branched polymers are added to a solution containing an equal mass of linear polymers that are already saturated with strongly bound CP; no free CP are present in solution. Snapshots from the initial and final states of this simulation, illustrating the outcome of the competition between the two 50-vertex polymers, are shown in Fig. 4.8. The -3 kT lateral attractions among the bound CP are strong enough to prompt perfect compaction of the linear polymer in the initial state. However, once the branched polymer is added, the linear polymer gives up essentially all of its CP, despite the energy penalty associated with this move due to the ramified edges of the collapsed branched polymer in comparison to those of the linear polymer. The compensation of this unfavorable energetic contribution is the substantial gain in conformational entropy of the linear polymer, as noted in Table 4.1. Stated differently, the loss of conformational entropy of the branched polymer upon collapsing to its most compact form is smaller than the corresponding entropy gain by the linear polymer upon giving up its bound proteins.

The time evolution of the CP populations on the linear vs. branched trees is shown in Fig. 4.9, revealing that, except for occasional sizeable fluctuations, once equilibrium is reached nearly all CP are bound to the branched tree (see inset).

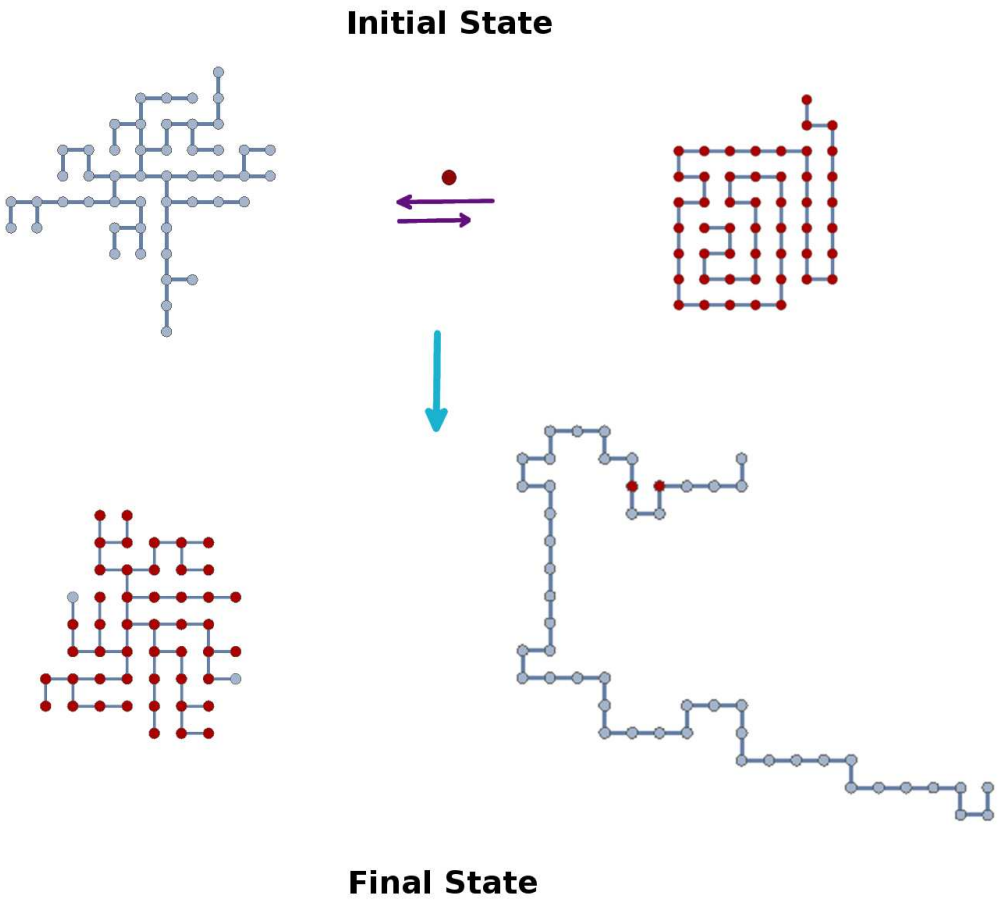

Figure 4.8: Snapshots from the initial and final equilibrium states obtained in the simulation of the competition between the branched (left) and linear (right) trees. Practically all the CP population is eventually found on the collapsed compact tree. The linear tree enjoys a substantial gain of conformational entropy, overcoming the unfavorable loss of CP-CP energy upon the migration of CP to the branched tree.

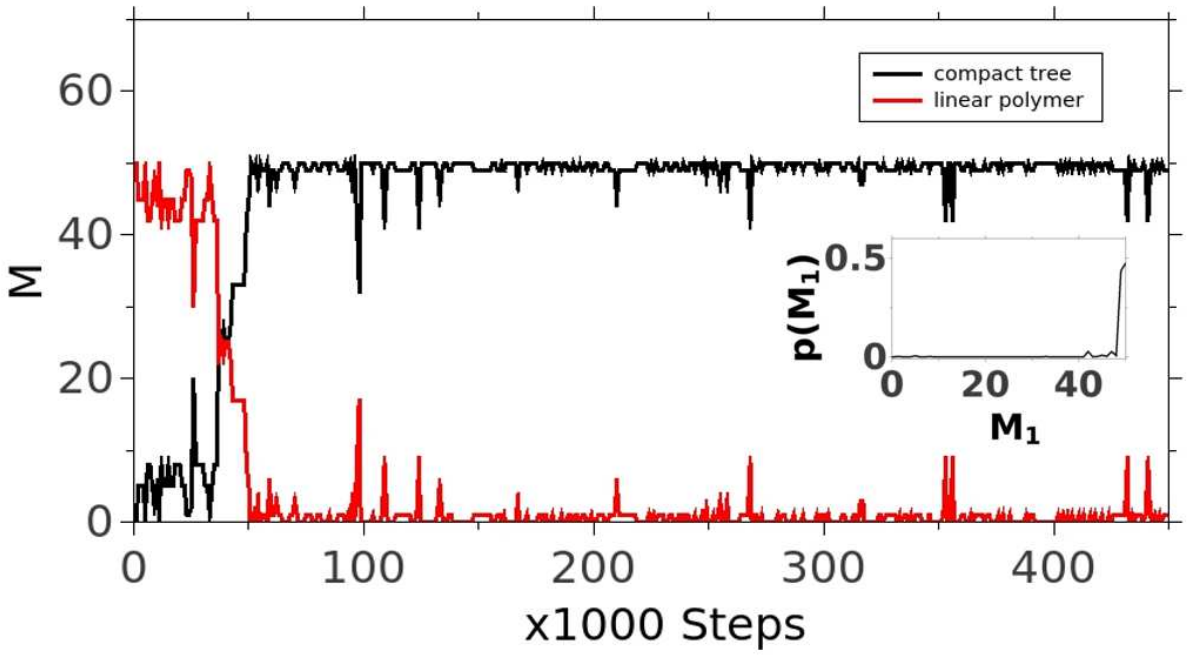

Figure 4.9: Temporal evolution of the CP populations on the branched (black trace) and the linear (red) trees, as obtained by the kinetic (particle-exchange) simulations. The inset shows the probability distribution of CP between the two trees, as derived from the thermodynamic simulations.

4.5.5 Kinetics of CP exchange between polymers

In aqueous solution there are two possible routes for CP transfer from one RNA molecule to another. The first is through evaporation-condensation events, whereby CPs desorb from one RNA, diffuse in solution, and eventually adsorb on another. The second route for CP exchange involves the formation of transient collision complexes through binary encounters of CP-dressed RNA molecules, enabling CP hopping from a binding site on one RNA to a neighboring binding site on another. As in many surface-diffusion and chemical reaction events, the unbinding and binding processes take place concertedly, resulting in lower activation energies than those implied by complete unbinding. Recent experimental results [Comas-Garcia 2014] suggest that CP exchange through RNA encounters is indeed the more likely mechanism, yet the evaporation-condensation mechanism cannot be ruled out. Notwithstanding this uncertainty, it is very reasonable to assume that the same mechanism is responsible for CP exchange in all the competition experiments modeled in our work. Recall, although we cannot relate the MC steps to real-time events, our kinetic simulations provide insight regarding the relative time scales of the various processes involved.

Starting with the competition between the initially naked 50-vertex tree and the two CP-saturated 25-vertex trees, for example, we see from Fig. 4.4 that equilibration is achieved in fewer than 10,000 MC exchange steps. Turning to Fig. 4.5 we are not surprised to observe that the equilibration time for CP sharing among the two 50-vertex trees is significantly slower than that between the two 25-vertex trees. As noted in section 4.5.2, the stability difference between the small and large trees is even more clearly exhibited by the rapid swinging of the CP population between the two small trees, as compared to the barely fluctuating populations following the symmetry breaking in the competition between the two large trees.

The time to equilibrium in the competition between the compact and extended trees, $\sim 35,000$ MC steps (Fig. 4.7), is longer than the time ($\sim 10,000$ steps) to equilibrium between two identical branched trees (Fig. 4.5). The origin of this difference is twofold.

First, the competition between the two identical 50-vertex trees starts with half of the population on each of the two trees whereas in the compact vs. extended tree competition all CP are initially adsorbed on the extended (eventually the "loser") tree. Second, when fully saturated with CP, the extended tree collapses to a more compact, and hence more stable, globule than the 50-vertex tree of sections 4.5.1 and 4.5.2. The compactness and stability of the initial structure is even more pronounced in the competition between the linear and branched trees, resulting in an even longer time to equilibrium, $\approx$ 50,000 MC steps, cf. Fig. 4.9.

4.6 Conclusion

In this work we have developed a Monte Carlo simulation approach for treating the effects of lateral interactions between proteins on their binding to branched polymers. This problem is motivated by the biologically relevant example of viral capsid protein binding to single-stranded RNA molecules, e.g., viral RNA genomes, which behave as effectively-branched polymers because of their large extent of secondary structure formation. We have focused here on the competition for binding protein by two or more polymers in order to explain the protein exchange equilibria observed in recent *in vitro* studies of competition between different RNA molecules to be packaged by capsid protein. Our model and simulations are motivated by the disordered complexes of capsid protein bound to RNA in the first, reversible, step of a two-step self-assembly of virus-like particles from RNA and CCMV CP. In particular, in this neutral-pH step the CP-CP interactions are sufficiently weak so that the complexes of bound protein are largely disordered, including only partial portions of spherical nucleocapsid (see, for example, Fig. 3b of [Garmann 2014a]). But the underlying physical situation is more general, because it provides basic insights into the ways in which lateral interactions between binding proteins determine the compaction of their polymer substrate.

To perform the simulations we have generalized the Rosenbluth Monte Carlo method of "growing" excluded-volume polymers to include arbitrary branching of the polymer (i.e., arbitrary distributions of vertex orders) as well as reversibly-bound proteins that attract each other at short distance and that –through conformational changes of the polymer– are effective at gathering in distant binding sites of the substrate. To elucidate the associated changes in energy (of the protein-protein interactions) and entropy (of the bound proteins and of the polymer conformations), we have treated protein exchange between polymers in the cases of:

(i) equal masses of long and short polymers with the same branching patterns (vertex-order distributions);

(ii) equal masses of equal-length and identical-vertex-order-distribution polymers having very different radii of gyration (because of the different placements of their branch points); and

(iii) equal masses of linear and branched polymers.

In all cases we have suppressed the contribution of "packaging signals" by insisting explicitly that all of the polymers involved are composed of identical binding sites, i.e., proteins bind to them with the same adsorption energy (affinity). As such, our work is complementary to recent theoretical studies in which specific distributions of high-affinity binding sites are shown to facilitate capsid nucleation. In this latter approach [Dykeman 2013, Dykeman 2014] polymers with neighboring high-affinity sites – in conjunction with a progressively increasing concentration of binding particles – are shown to be preferentially encapsidated over ones with uniform distributions of binding energies. This role of binding heterogeneity has also been elucidated by molecular dynamics simulations [Perlmutter 2015]; again the starting point is an explicit assignment of very different protein affinities to the different sites of the polymer.

In our work, on the other hand, all sites are associated with the same protein binding energy. Nevertheless, a distribution of effective binding energies arises from the different

possibilities for the bound proteins to interact laterally with nearest-neighbor bound proteins – not only ones that are neighbors on the polymer backbone but, more importantly, also ones that occupy distant sites on the polymer. Fig. 4.10, for example, shows typical configurations of bound particles on an equilibrated 50-vertex branched polymer at half coverage. In the equilibrium ensemble for this coverage every site has a non-zero probability of being occupied. By calculating the average number of occupied nearest-neighbors of each site we can determine the effective energy of each site. Stronger attractive energy results in CP aggregation mediating compaction of the tree they are bound to. Note in particular the localized aggregation of red and yellow particles even in the case of weaker attractive energy, as in Fig. 4.10, left. In general, aggregation will depend on the nature of the branching pattern of the polymer substrate, e.g., on the vertex-order distribution and the placement of the 3rd and higher-order branch points, and on the strength of CP-CP attraction.

Further work, both experimental and theoretical, will be important for clarifying the relative roles and importance of "packaging signals" and of protein interactions in determining the selective packaging of viral RNA genomes by their capsid protein. In the former case, local secondary/tertiary structure of the RNA enhances its affinity for protein, whereas in the latter case it is the extent and nature of branching of the larger-scale secondary/tertiary structure that determines the effective binding energies of proteins. These large-scale structures are also important in determining RNA properties – other than binding of protein – relevant to its packaging in viral capsids and virus-like particles. In particular, simulation [Perlmutter 2013] and theory [Gonca 2014] studies have shown how polymer branching enhances packagability – confinement in small volumes – by reducing the conformational entropy loss and enhancing the interaction of polymer with the capsid interior.

Note that any branched structure, such as the trees representing RNAs, can be made more compact or extended by moving its high-fold junctions closer or father away from its center. For example, Tomato bushy stunt virus (TBSV) [Wu 2013] and satellite tobacco mosaic virus (STMV)

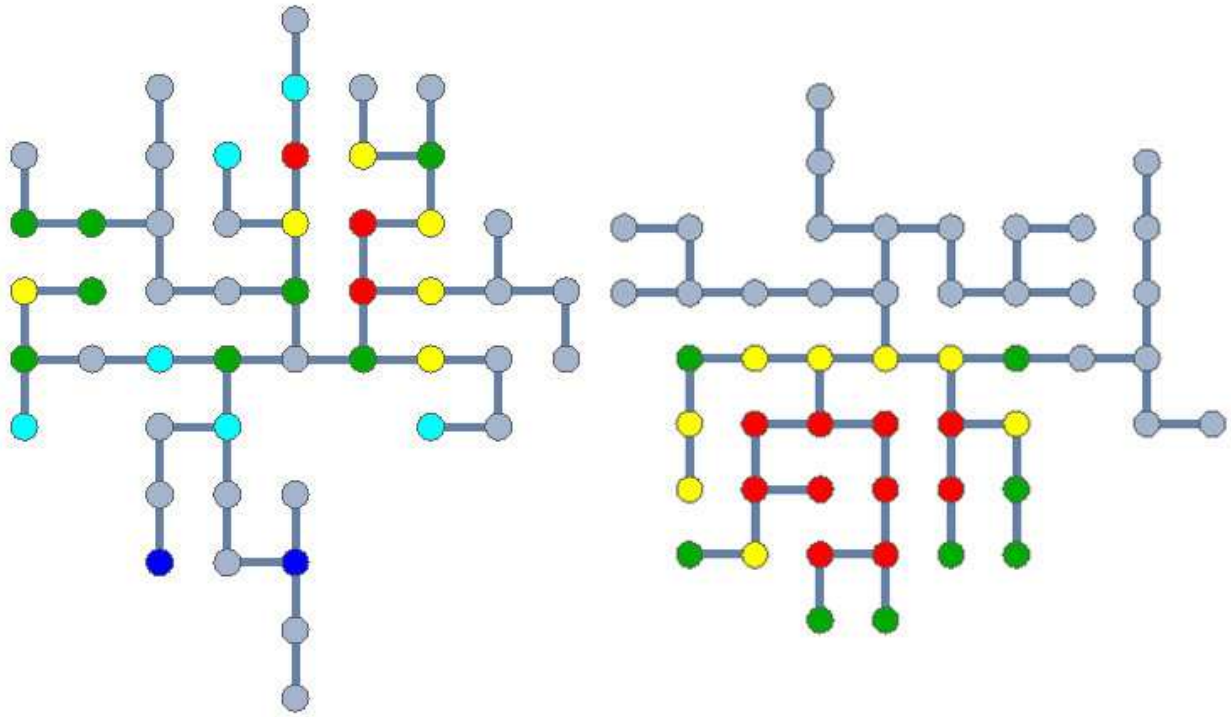

Figure 4.10: Tertiary configurations of a 50-vertex branched tree at half-coverage, with a typical distribution of bound proteins. The number of occupied nearest neighbors associated with each bound particle are color coded (red corresponding to 4, yellow to 3, and green to 2), reflecting its effective binding energy. Left: the interaction energy between CP on nearest neighbor lattice sites is relatively weak, $\epsilon = -1$ kT. Right: stronger attractive energy, $\epsilon = -3$ kT , results in aggregation of the CPs into one patch.

[Athavale 2013, Archer 2013, Garmann 2015] were found by SHAPE (selective 2'-hydroxyl acylation analyzed by primer extension) studies to have compact and extended secondary structures, respectively, for precisely this reason. More explicitly, SHAPE analyses showed that TBSV has high-fold junctions concentrated near the center of its secondary structure, while STMV has high-fold junctions near its ends. The work by [Wu 2013] suggests further that the branchedness of the secondary structure correlates with the long-range intra-genomic base-pairing interactions that are known to be important in ensuring many genome functions.

Chapter 5

Generation and Analysis of Randomly Branched Polymers Using the Prüfer Sequence Representation

The configurational statistics of branched polymers has been the theme of many physical theories, focusing generally on the scaling relationship $R_g \sim N^\nu$ between the radius of gyration of the polymer, R_g, and the number of its monomers, N. For ideal randomly branched polymers – that is, ignoring excluded volume interactions – several different elegant approaches [Zimm 1949, de Gennes 1968, Khokhlov 1994, Rubinstein 2003] revealed that in three dimensions (3D) $\nu = 1/4$, demonstrating that they are more compact than ideal linear polymers for which $\nu = 1/2$.

A branched polymer can be represented as a tree graph because of the mapping of its constituent monomers and covalent bonds to N vertices and $N-1$ edges, respectively. Any tree graph with N labelled vertices can be coded into a unique sequence of $N-2$ integers, called the *Prüfer sequence* [Pruefer 1918, Moon 1970] from which it is straightforward to recover the tree graph.

5.1 Tree graph to Prüfer sequence

In Fig. 5.1, we show how to generate the Prüfer sequence $\mathbf{p}$ of a (labelled) 12-vertex tree graph by systematically removing vertices of order 1 (hereafter, a leaf). The set $\mathbf{p}$ is initially empty; in the first step, one removes the leaf with the lowest label (vertex 7 or V7 for short) and adds the label of its neighboring vertex (V3) to $\mathbf{p}$ (see (i)). Note, V3 becomes a leaf after the first step (see (ii)). This procedure is repeated until the original tree has been reduced to two vertices (see (xi) in Fig. 5.1), resulting in the Prüfer sequence $\mathbf{p} = \{3, 5, 1, 1, 2, 5, 4, 4, 4, 6\}$.

Some of the basic properties of the tree can be easily derived from the Prüfer sequence. Denoting the index of each vertex as j (i.e. j runs from 1 to N), the degree of vertex j is simply one more than the number of times it appears in $\mathbf{p}$. For example, the degree of V1 is three since "1" appears twice in $\mathbf{p}$. The degree sequence, $\mathbf{deg}$, is $\{3, 2, 2, 4, 2, 2, 1, 1, 1, 1, 1, 1\}$ for the tree in Fig. 5.1, where the j^{th} element of $\mathbf{deg}$ is the degree of vertex j. Note that only vertices in the body of the original tree (i.e. vertices for which $\mathbf{deg}(j) \geq 2$) are elements of the Prüfer sequence.

There are many known ways to recover the tree graph from its Prüfer sequence [Moon 1970]. In the following section we present our own, alternative, method which greatly facilitates calculating graph distances directly from the Prüfer sequence.

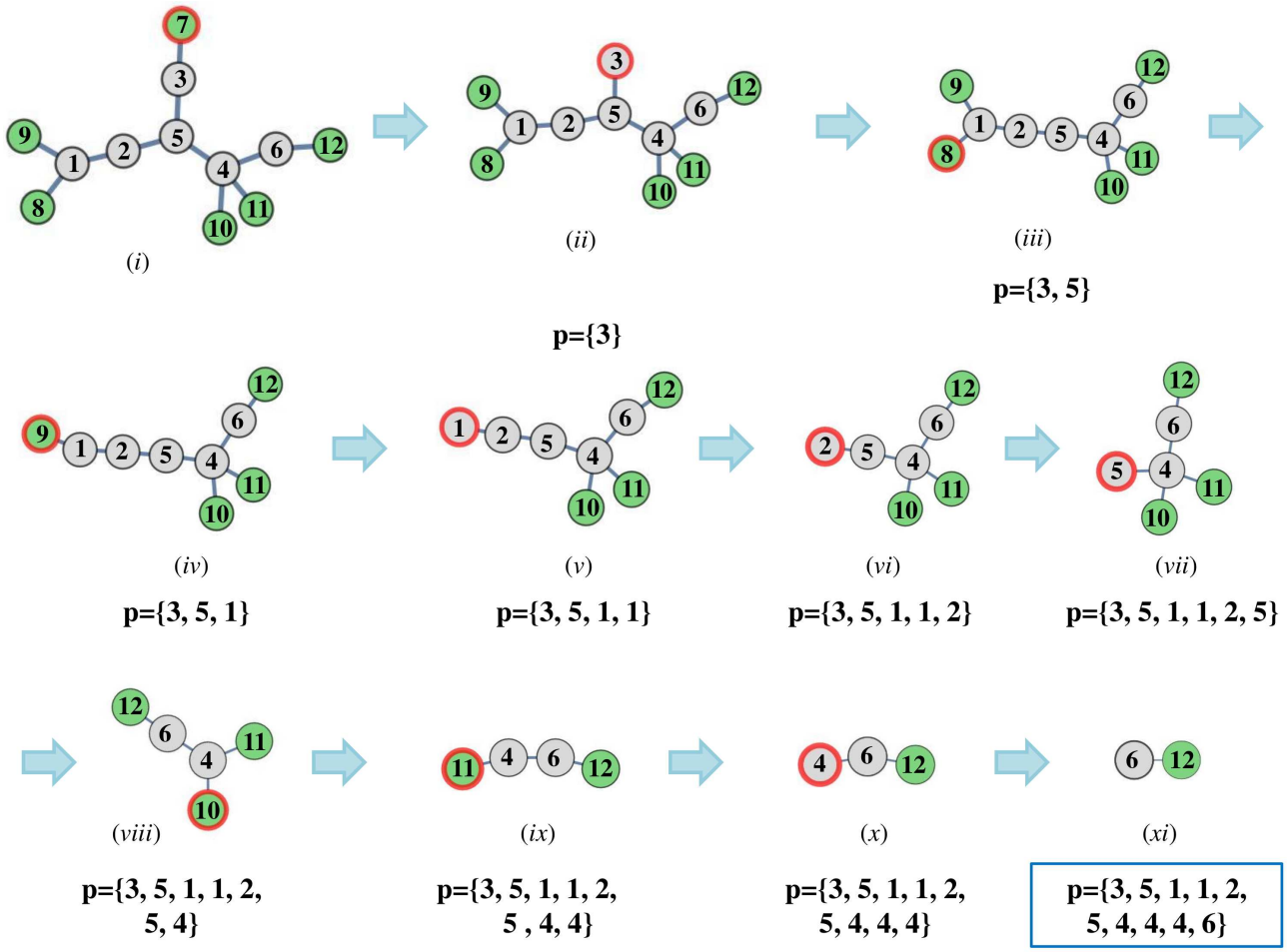

Figure 5.1: Converting a tree graph to its Prüfer sequence, **p**. Beginning with the complete tree (upper left) and proceeding to the right, the leaf with the lowest index is deleted with its edges at each step and the index of the neighboring vertex is added to **p**. This simple procedure is repeated until only two vertices of the orginal tree remain. The degree of vertex j is the jth element of the degree sequence **deg**, which can be readily calculated from **p** (see text for details).

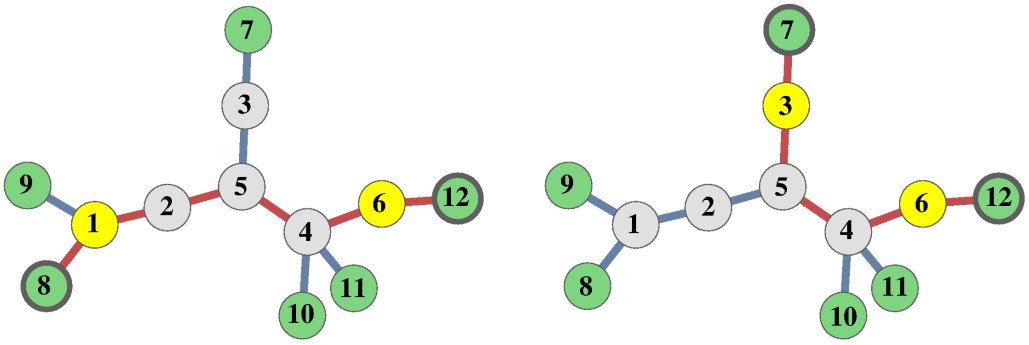

Figure 5.2: Two copies of the tree graph corresponding to the the prüfer sequence $\mathbf{p} = \{3, 5, 1, 1, 2, 5, 4, 4, 4, 6\}$ to illustrate how a linear chain of edges can be used to estimate the (3D) size of the entire tree. **a** – The graph distance between vertices 8 and 12 is measured by counting the number of edges in the path that connects the two vertices (i.e. $d(8, 12) = 6$). This distance is also the graph diameter (GD) of the tree because it is the largest distance between any two vertices. **b** – The distance between two leaves (such that one leaf is bonded to the first element of $\mathbf{p}$) can be calculated from the sequence (see Sec. 5.3.1). The 3D size of the tree is then calculated by taking the square root of the leaf-to-leaf distance.

5.2 Prüfer sequence to tree graph

One variant of our approach, which greatly facilitates the generation of tree graph ensembles and the calculation of graph metrics, is to re-label the tree such that the leaves take the highest possible values. For example, the tree in Fig. 5.2 has 12 vertices and 6 leaves; accordingly, it has been labelled so the leaves take the values 7-12. With the Prüfer sequence of the re-labelled tree ($\mathbf{p}$), one applies the recipe below (see Procedure 1) to "grow" the tree by drawing bonds between the elements of $\mathbf{p}$ when reading the sequence from right to left.

Procedure 1 (Prüfer sequence to tree graph) *Let k index the elements of $\mathbf{p}$ from 1 to $N-2$ when reading the sequence from right to left. First, a bond is drawn between the first element ($\mathbf{p}_1$) and the leaf with the highest label (VN). Then, proceeding one element at a*

time from $\mathbf{p}_1$ *to* $\mathbf{p}_{N-2}$*, a bond is drawn between elements* $\mathbf{p}_k$ *and* $\mathbf{p}_{k+1}$*, unless* $\mathbf{p}_{k+1}$ *is already part of the tree, in which case a bond is drawn between* $\mathbf{p}_k$ *and the leaf with the highest label (of the remaining leaves). Finally, the tree is completed with a bond drawn between* $\mathbf{p}_{N-2}$ *and the leaf with the lowest label.*

As an example in the application of Procedure 1, we'll show how to generate the tree in Fig. 5.2 from its Prüfer sequence

$$\mathbf{p} = \{3, 5, 1, 1, 2, 5, 4, 4, 4, 6\}, \tag{5.1}$$

using the variable k as it appears in Procedure 1 (i.e. the rightmost and leftmost elements are indexed as $k = 1$ and $k = N - 2$, respectively). The basic idea is to start from the first element of $\mathbf{p}$ ($k = 1$, when reading from right to left) and draw bonds stringing elements together as they appear in the sequence according to Procedure 1 – one either draws a bond to a leaf or to the next element in the sequence. More explicitly, one begins with the first element of $\mathbf{p}$ (e.g. V6) and draws a bond to the leaf with the highest label (V12). Recall, leaves are added in descending order (see (i) in Fig. 5.3). A bond is then drawn between successive elements in the sequence (e.g. V4 and V6, see (ii) in Fig. 5.3) because V4 is not yet part of the tree. One continues drawing bonds between successive elements until one must draw a bond to a vertex that is already part of the growing tree e.g., a bond connecting V4 to itself is not drawn (see (iii)) even though the 4s are neighboring elements in $\mathbf{p}$ ($k = 2, 3$). Instead of drawing a bond to such vertices (and thereby creating a cycle!), a bond is drawn to the highest remaining leaf (i.e. to V11; see Procedure 1). The next element ($k = 4$) is another instance of V4 and is connected to a leaf for the same reason (see (iv) in Fig. 5.3). A new bond is then drawn between V4 and V5, because V5 ($k = 5$) is not yet part of the tree. Continuing to draw bonds this way eventually reproduces the original tree (see (xi) in Fig. 5.3).

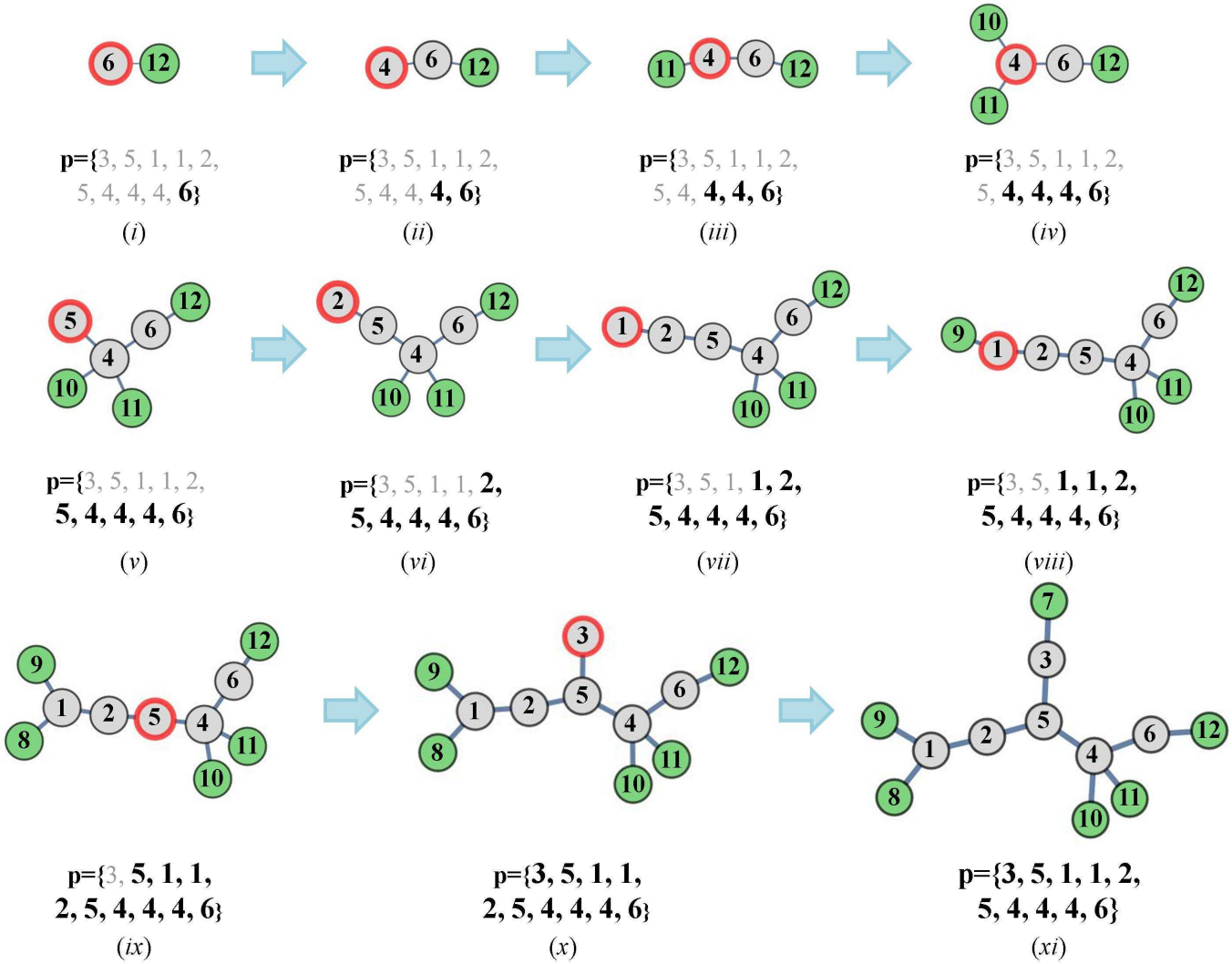

Figure 5.3: Our method to recover the tree graph from its Prüfer sequence. Leaves of the original tree are colored green, vertices in the body of the tree are in gray. The red ring highlights the "reactive" vertices of the growing tree. Whenever a bodily vertex is added to the tree its label is boldfaced in the sequence. See the text for the detailed description.

5.3 Calculating graph distances from the Prüfer sequence

The one-to-one relationship between a Prüfer sequence ($\mathbf{p}$) and its corresponding tree graph, as described in the previous sections, implies that all the properties of the tree (N, $\{f_d\}$, graph diameter, etc.) are encoded within the sequence. For example, the number of vertices N can be easily computed from the sequence, because the number of elements in $\mathbf{p}$ is equal to $N-2$. And $f_d = \dfrac{n_d}{N}$, the distribution of vertex degrees, with n_d the number of vertices with degree d, follows directly from $\mathbf{p}$ because the degree of vertex j is simply one more than the number of times j appears in $\mathbf{p}$. Note that all vertices that do not appear in $\mathbf{p}$ are of degree 1 (i.e. all the leaves). The graph diameter (GD), on the other hand, cannot be immediately calculated "upon inspection" of $\mathbf{p}$. The difficulty in computing the GD (and other graph metrics) directly from the sequence is due to the non-trivial encryption of the tree's connectivity upon generation of the Prüfer sequence. In this section, we show how our alternative method to recover the tree from $\mathbf{p}$ can be used to calculate a variety of graph metrics, including the GD.

The graph distance between two vertices is the number of edges in the path that connects them (see Fig. 5.2). As discussed in more detail later, the graph diameter (GD) of the tree is (to an excellent approximation) proportional to the largest distance between two leaves such that one of the leaves is bonded to the first element of $\mathbf{p}$. Recall that the variable k is used to index the elements of $\mathbf{p}$ from right to left, so that the rightmost and leftmost elements are indexed as $k=1$ and $k=N-2$, respectively. The distance $d(\mathbf{p}[1], v)$ between the first element of $\mathbf{p}$ and any vertex v in the body of the tree (i.e. v is not a leaf) is calculated directly from $\mathbf{p}$ as follows.

Procedure 2 (Calculation of the graph distance) *For any given tree, consider an arbitrary vertex v is not a leaf. Let k_v be the lowest index of the vertex v in $\mathbf{p}$. Move rightwards element-by-element from k_v to $k=1$, increasing the distance by one unit with each step. If*

an element is encountered that corresponds to a vertex v' which also appears in $\mathbf{p}$ *with a lower index, skip to it (i.e. to* $k_{v'}$*) and continue rightwards, increasing the distance by one unit with every element encountered until the first element is reached.*

Let us now elaborate on Procedure 2 with an example calculation as we did with our discussion of the first procedure. Since the vertex v may occur several times in $\mathbf{p}$, one starts from the *lowest index* of v in $\mathbf{p}$. For instance, suppose we wanted to calculate the distance between V5 and V6 directly from $\mathbf{p}$ (see Eq. (5.1)). Since there are two occurrences of V5 in $\mathbf{p}$ ($k = 5$ and $k = 9$), we start at the lower index of the two (i.e. $k = k_{V5} = 5$). Beginning with the lowest index of v ensures the connectivity of the graph can be "read off" from the sequence. That is, if we start at k_v we can always find a bond in the tree graph between v and its neighbor to the right in $\mathbf{p}$. Note, a bond exists between V5 and its "neighbor to the right" V4, but the same is not true for the other occurrence of V5 in $\mathbf{p}$ (i.e. a bond does not exist between V5 and V1, see Fig. 5.2), reflecting the importance of starting with the lowest index of v.

Next, we progress rightwards element-by-element, to the first element of $\mathbf{p}$, increasing the distance by one unit with each step because neighboring elements are connected by a bond. If we run into an element v which also appears in $\mathbf{p}$ with a lower index, we skip to its k_v and continue rightwards, increasing the distance by one unit with every element encountered until we have reached the first element. In our example, we would start from V5 ($k = 5$), the next element to the right ($k = 4$) is V4. After increasing the distance by one unit, we see V4 appears with indices ($k = 2, 3, 4$), so we skip to $k = k_{V4} = 2$. Finally, the next element is ($k = 1$) V6, the first element in $\mathbf{p}$, so we increase the distance by one unit and stop. The total distance from V5 to V6, as read from the prüfer sequence, is thus 2.

5.3.1 The distance between two leaves

In this section we use Procedure 2 to calculate the distance $d(p[1], t)$ between the first element of $\mathbf{p}$ and another *stump* (i.e. a vertex t which is bonded to a leaf). Clearly, $d(p[1], t) + 2$

is the distance associated with the leaves bonded to vertices $p[1]$ and t. Let us denote this "leaf-to-leaf" distance as $L1LD(t)$ and let $\mathbf{t}$ denote the set of stumps, such that the number of occurrences of a stump t is equal to the number of leaves associated with it (e.g. a stump attached to two leaves occurs twice in $\mathbf{t}$).

In order to calculate the $L1LD(t)$ we need to use Procedure 2 to calculate the distance between $\mathbf{p}[1]$ and vertices in $\mathbf{t}$. The elements of $\mathbf{t}$ are straightforward to identify in $\mathbf{p}$. The first and last elements of $\mathbf{p}$ are elements of $\mathbf{t}$, as are any elements t (of $mathbfp$) that are immediately to the right of an element that appears in $\mathbf{p}$ with a lower index. For example, if $\mathbf{p} = \{3, 5, 1, 1, 2, 5, 4, 4, 4, 6\}$ then $\mathbf{t} = \{3, 6, 1, 1, 4, 4\}$ because

- V6 and V3 are in $\mathbf{t}$ –they are the first and last elements of $\mathbf{p}$.
- V1 occurs twice in $\mathbf{t}$ –V1 ($k = 8$) appears immediately before vertex 5 ($k = 9$), which appears earlier ($k_{V5} = 5$). V1 also appears immediately before itself (see position $k = 7$).
- V4 occurs twice in $\mathbf{t}$ –V4 appears immediately before itself twice (see positions $k = 2, 3$).

As an example, let us calculate $L1LD(3)$ –the distance between the leaves bonded to V3 and V6 (i.e. $d(7, 12)$, see Fig. 5.2b). Starting at the only occurrence of V3 in $\mathbf{p}$ ($k = 10$), the next element over is V5 ($k = 9$); the distance d is increased by one unit in accordance with Procedure 2 (i.e. $d = 1$). Next, note that V5 appears with a lower index ($k_{V5} = 5$) so we skip to the lower occurrence. Its neighbor is V4 ($k = 4$) and the distance is increased by one unit (i.e. $d = 2$). After skipping to $k = k_{V4} = 2$, we increase the distance by one unit and stop (i.e $d(\mathbf{p}[1], 3) = d = 3$) because the next element over ($k = 1$) is the first element of $\mathbf{p}$. Finally, the leaf-to-leaf distance is $L1LD(3) = d(\mathbf{p}[1], 3) + 2 = 5$.

The maximum $L1LD(t)$, denoted simply as the $ML1LD$, can be used to estimate the GD. One can also estimate the average leaf-to-leaf distance by averaging over the $L1LD(t)$. More explicitly, the average leaf-to-leaf distance, denoted by $\overline{LLD}$ is calculated as follows:

$$\overline{LLD} = \frac{1}{|\mathbf{t}|} \sum_{t \in \mathbf{t}} L1LD(t). \tag{5.2}$$

Hence, it is possible to calculate metrics (e.g. the $ML1LD$ and the $\overline{LLD}$) directly from the Prüfer sequence. In the next section, we will show how to generate an ensemble of ideal randomly-branched polymers (all with the same degree distribution) by simply permuting the elements of $\mathbf{p}$, from which the average 3D size of the branched polymer can be easily derived. Note, since our focus is on *ideal* polymers –those for which we have ignored excluded volume interactions– the 3D size of the branched polymer can be estimated from the size of the linear polymer whose contour length is equal to the GD, $ML1LD$, or the $\overline{LLD}$ (see Fig. 5.2).

5.4 Prüfer shuffling and the generation of Randomly Branched Polymer (RBP) ensembles

Shuffling a Prüfer sequence $\mathbf{p}$, i.e., permuting any number of pairs of elements in the ordered sequence (see Fig. 5.4), results in a new Prüfer sequence whose tree graph necessarily has the same number of vertices and bonds and the same distribution of vertex degrees. In this sense, the two trees appear to be comparably branched. Nevertheless, they can have different connectivity and size – their radii of gyration (as computed by the Kramers formula [Rubinstein 2003]), for example, can differ significantly.

The power of the Prüfer description of branched structures is that one can efficiently generate randomly-branched-polymer (RBP) ensembles. More explicitly, carrying out a sufficiently strong shuffling of a Prüfer sequence allows generation of an ensemble of randomly-branched configurations corresponding to an arbitrary vertex size (N) and a particular distribution of branching orders ($\{p_\nu\}$).

The mean square radius of gyration, $\langle R_g^2 \rangle$, representing the average of R_g^2 over all possible

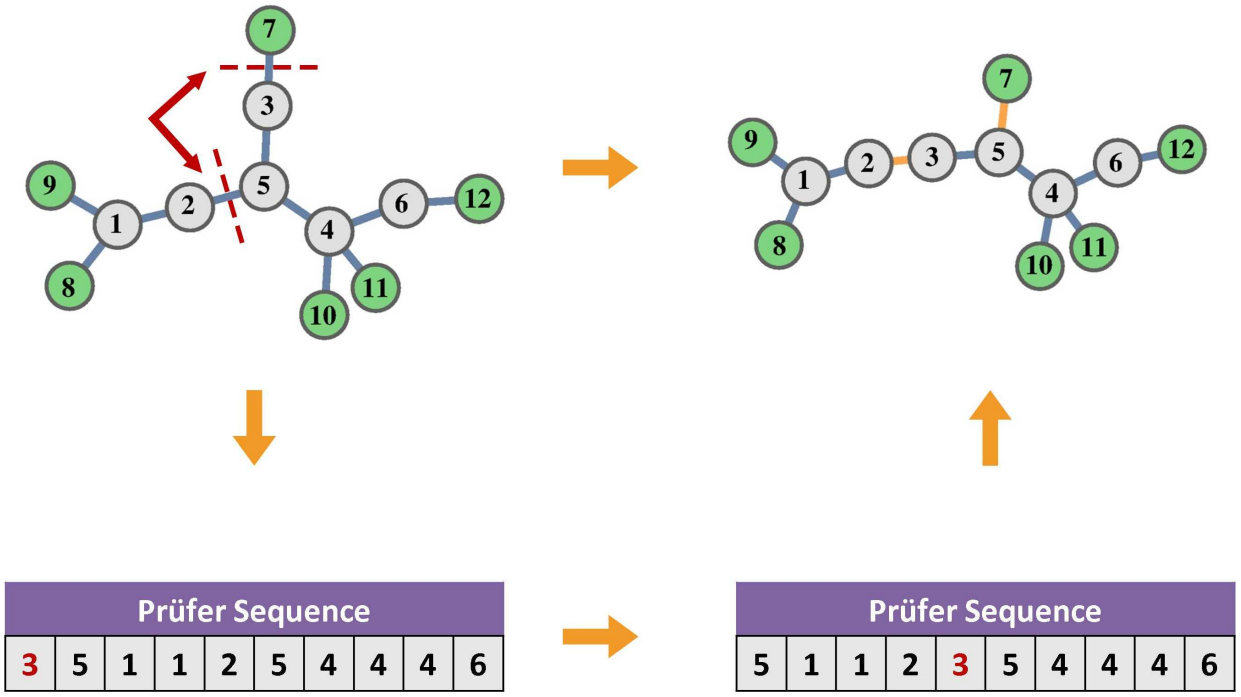

Figure 5.4: Two trees with identical vertex degree distributions are related by permuting their corresponding Prüfer sequences. To transform the tree on the left to one on the right, the bonds between vertices 2,5 and 3,7 must be broken (dashed red lines) and new bonds between vertices 2,3 and 5,7 must be made (shown in orange). Alternatively, the same transformation can be effected by moving the 3 at the left end ($k = 10$) of the Prüfer sequence to the position (shown in red) between the 2 ($k = 6$) and 5 ($k = 5$).

conformations of an ideal polymer, whether branched or linear, can be calculated using the Kramers theorem [Rubinstein 2003],

$$\langle R_g^2 \rangle = \frac{b^2}{N^2} \sum_{m=1}^{N} N_1(m)[N - N_1(m)] \tag{5.3}$$

where N is the number of monomers comprising the polymer and b is the monomer-monomer bond length. The summation extends over all possible divisions of the polymer into two parts, containing $N_1(m)$ and $N - N_1(m)$ monomers, respectively. For the special case of an ideal linear polymer Eq. (5.3) yields the well known relationship $R_g \sim N^{1/2}$, identical to

the power-law dependence of the average end-to-end distance $\langle R \rangle$. (For notational brevity we replace $\sqrt{\langle R_g^2 \rangle}$ by R_g). For ideal randomly-branched polymers, on the other hand, Eq. (5.3) yields the familiar theoretical scaling law, $R_g \sim N^{1/4}$. The average end-to-end distance of an ideal randomly-branched polymer is equivalent to our average leaf-to-leaf distance, $R_{LL} \equiv \sqrt{\langle \overline{LLD} \rangle}$ and should thus also scale as $N^{1/4}$ [Khokhlov 1994]. This behavior will be confirmed by numerical calculation in the next section, using our Prüfer algorithm. A related quantity showing the same power-law behavior is $R_{GD} \equiv \sqrt{GD}$, the square root of the graph diameter.

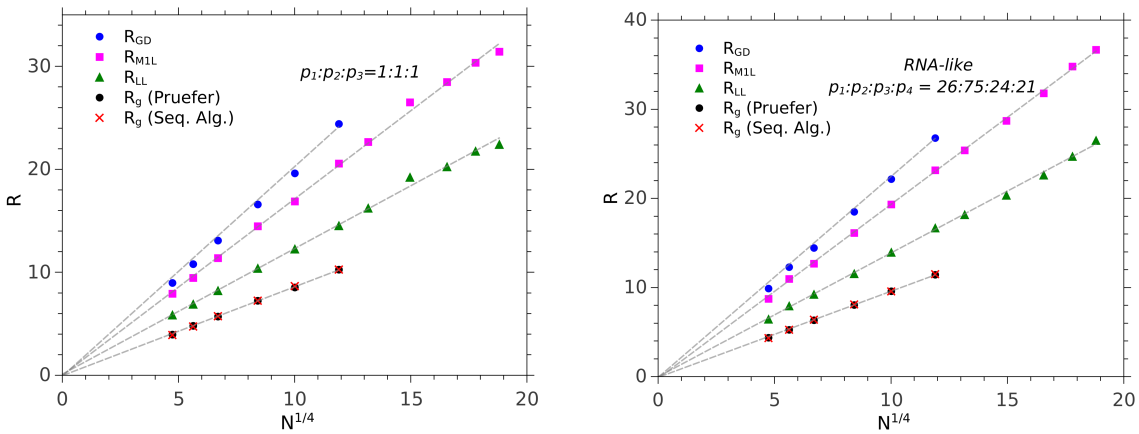

Figure 5.5: The average 3D size of randomly branched polymers as a function of (the fourth root of) N with vertex degree distributions **(a)** $p_1 = p_2 = p_3 = 1/3$, **(b)** "RNA-like" polymers, see text for details **(c)** $p_1 = 1/6; p_2 = 4/6; p_3 = 1/6$, **(d)** $p_1 = p_3 = 1/2$; $p_2 = 0$. Due to insufficient memory, calculations involving the adjacency matrix were terminated after $N = 20,000$ vertices; see **Blue** – R_{GD}, **Black** – R_g (Prüfer), and **Red crosses** – R_g (the Sequential Algorithm). Calculations directly from the Prüfer sequence, however, were able to sample trees with up to 125,000 vertices; see **Pink** – R_{M1L} and **Green** – R_{LL}. The linear regression lines, shown in gray, illustrate that the average size of the polymer R scales as $\sim N^{1/4}$ – i.e., these polymers are randomly branched.

5.5 Graph distance (GD) to 3D size

Using our Prüfer-sequence algorithm and the theoretical tools described in the previous section, we have numerically calculated several 3D size measures of ideal randomly branched polymers (iRBP). Numerical results corresponding to four families of iRBPs with different degree distributions, $\{p_\nu = \frac{n_\nu}{N}\}$, are shown in Fig. 5.5.

The results in Fig. 5.5 are ensemble averages of size measures for iRBPs composed of up to $N = 120,000$ vertices. For every value of N we have analyzed 100 tree graphs. Fig. 5.5a shows the results obtained for branched polymers containing (on average) equal proportions of two-fold ($\nu = 2$) and three-fold ($\nu = 3$) vertices, i.e., $p_2 : p_3 = 1 : 1$. The fraction of leaves in this tree graph's family, p_1, follows from the normalization condition $\sum_\nu p_\nu = 1$ and Euler's "sum rule" $\sum_\nu n_\nu \nu = 2N - 2$, where n_ν is the number of vertices of degree ν. For large N, this leads to $p_1 = p_2 = p_3 = 1/3$.

Fig. 5.5b shows the results for random "RNA-like" polymers. The vertex-degree distribution corresponding to this family of iRBPs derives from earlier calculations [Gopal 2014] of RNA secondary structure. Specifically, Boltzmann weighted ensembles of 200 secondary structures corresponding to 200 different randomly generated 7000nt-long random sequence RNAs with uniform nt composition, (i.e., a total of 4×10^4 configurations), were generated using the RNAsubopt program from the Vienna package [Lorenz 2011]. Upon analyzing their degree distribution it was found that 5^{th} or higher order vertices (i.e., nt-loops) are extremely rare. Lumping their fraction into p_4, the degree distribution corresponding to this family (in the limit of large N) is given by: $p_1 = \frac{26}{126}, p_2 = \frac{75}{126}, p_3 = \frac{24}{126}, p_4 = \frac{1}{126}$. The two other iRBP families are: (i) Cayley trees of 3-fold vertices, i.e., trees consisting of vertices of degree 3 or 1 (i.e., leaves), with $p_1 = p_3 = \frac{1}{2}$ and (ii) Similar to the iRBPs in Fig. 5.5a, but with a larger fraction of the 2-fold vertices: $p_1 = \frac{1}{6}, p_2 = \frac{2}{3}, p_3 = \frac{1}{6}$. The data points of the 3D size metrics shown in Fig. 5.5 represent the statistical averages corresponding to 100 randomly generated tree graphs for each value of N.

Their meaning and calculation procedures are as follows:

- R_{GD} : Recall, $R_{GD} \equiv \sqrt{\langle GD \rangle}$ is the square root of the ensemble average of the graph diameter. In this calculation, starting with an arbitrary tree graph with the given $\{p_\nu\}$, ensembles of iRBP tree graph were generated using Prüfer shuffle, and their GD determined by the Mathematica [Wolfram Research 2012] built-in function GraphDiameter. The function utilizes popular algorithms [Dijkstra 1959, Floyd 1962, Warshall 1962] to compute graph distances from the adjacency matrix [Diestel 2000] of the tree graph (see below).

- R_g (Prüfer): This metric is the ensemble averaged radius of gyration, calculated using the Kramers formula, Eq. (5.3). Here too, iRBP tree graphs were generated using Prüfer shuffling, but vertex-to-vertex distances were calculated using a custom program utilizing the adjacency matrix.

- R_g (Seq. Alg.): These calculations of R_g are based on an efficient algorithm called the "sequential algorithm" [Blitzstein 2011] to generate an ensemble of randomly-branched polymers all with with a given degree sequence $\mathbf{deg} = \{..., d_j, ...\}$. In the algorithm bonds are drawn between pairs of vertices such that one of the vertices has at least two unsaturated edges and the other vertex has a single unsaturated edge. The former is chosen with probability proportional to $d_j - 1$, while the latter has the lowest index of the vertices with a single unsaturated edge. The degree of each vertex in the new bond is reduced by one, the degree sequence is updated, and a new pair of vertices is selected. One continues this way until only two vertices remain –each with one unsaturated edge– whereupon a bond connecting these last two vertices completes the tree. Repeating this procedure for the same degree sequence yields an ensemble of trees (all with the same degree sequence) that are uniformly distributed [Blitzstein 2011]. Once the trees were generated, the R_gs were evaluated based on Kramers theorem using the custom program mentioned above.

- R_{LL}: $R_{LL} \equiv \sqrt{\langle \overline{LLD} \rangle}$, where $\langle \overline{LLD} \rangle$ is the ensemble average of $\overline{LLD}$, the mean

leaf-to-leaf distance. In calculating R_{LL} we have used Prüfer shuffing to generate RBP ensembles, and LLDs were evaluated following the analysis of the Prüfer sequences described previously in section 5.3.1. One simplification employed in this calculation is that in evaluating LLD for a given tree graph we have not included all leaf-to-leaf paths, but rather only those originating from the first (rightmost) element of the Prüfer sequence, (like those shown in Fig. 5.2). In other words, we have sampled only a fraction of all the leaf-to-leaf paths. Note, however, that in the ensemble of randomly generated tree graphs the identity of the vertex appearing in position 1 of the Prüfer sequence is also arbitrary, implying that the R_{LL}s shown in Fig. 5.5 are faithful statistical averages of LLDs. The simplification just mentioned facilitates the calculation.

- R_{M1L}: Similar to R_{GD} this quantity measures the ensemble average of the maximal 3D end-to-end distance of a branched polymer. We calculated it subject to the same computational simplification employed in our calculation of R_{LL}. Explicitly, from the paths used to calculate R_{LL} we picked the longest leaf-to-leaf path (originating from the Prüfer-sequence's first element), obtaining the $M1LD$ (see e.g.,Fig. 5.2). Using $\langle M1LD\rangle$ to denote the ensemble average of $M1LD$ we then define $R_{M1L} = \sqrt{\langle M1LD\rangle}$. Clearly $M1LD \leq GD$, and hence $R_{M1L} \leq R_{GD}$, as is apparent from Fig. 5. Later we will show that the GD can be approximated by the average $M1LD$ corresponding to the top decile of $M1LD$.

All five measures of 3D size of the iRBPs in Fig. 5.5 show the expected scaling relation $R \sim N^{1/4}$. The computational procedures yielding the first three measures –R_{GD}, R_g (Prüfer), and R_g (Seq. Alg.)– utilize the Adjacency Matrix (AM) to represent the branching pattern of the tree graph. For a tree with N vertices the AM is a sparse $N \times N$ matrix whose ij-th element is 1 if a bond connects vertices i and j, and 0 otherwise [Diestel 2000]. In terms of CPU time, these calculations are considerably faster than our calculations of R_{LL} and R_{M1L} based on the determination of LLDs from the Prüfer-sequence; see section 5.6

for details. Nonetheless, the AM based calculations require the computer to allocate $\sim N^2$ bytes of random access memory (RAM) to handle a tree of N vertices. On the other hand, extracting graph distances from a Prüfer sequence of N elements requires the allocation of only $\sim N$ bytes. Thus, while Mathematica's GraphDiameter function and the Sequential Algorithms methods are faster than our custom Prüfer algorithm, the latter is far more efficient for large values of N.

Indeed, as reflected by the range of N values in Fig. 5.5, when the trees became large ($N > 20,000$ vertices) the AM based calculations (i.e., R_{GD} and R_g) required more memory than was available in our system (Intel Xeon CPU @ 2.50 GHz with 32 GB RAM) and the calculations were forced to terminate, obviating the utility of the GraphDiameter or the Sequential Algorithm. In contrast, our programs were able to comfortably calculate well beyond this limit.

5.5.1 Approximating the GD from the Prüfer sequence

Though we were not able to compute the GD directly from the Prüfer sequence, we found we could estimate it using the $ML1LD$. Recall, the $ML1LD$ is the largest leaf-leaf distance in the tree such that one of the leaves is connected to the first element of the **p** sequence. From this restriction it necessarily follows that, on average, the $GD > ML1LD$. The average of the top decile of $ML1LD$s, however, provides a good estimate of the ensemble-averaged GD, as illustrated in Fig. 5.6 for each family of branched polymers considered in this study (the $ML1LD$ and GD were calculated in units of graph edges E).

Calculation of the GD relies on the adjacency matrix [Dijkstra 1959, Floyd 1962, Warshall 1962] and is thus of limited use for very large trees ($N > 20,000$ in our case) because the system must allocate a significant amount of memory to handle the tree. In Fig. 5.5 we showed the $ML1LD$ could be computed for trees involving 100,000 vertices. Hence, one can calculate the GD for very large trees using, as a "back door", the $ML1LD$s of the top decile.

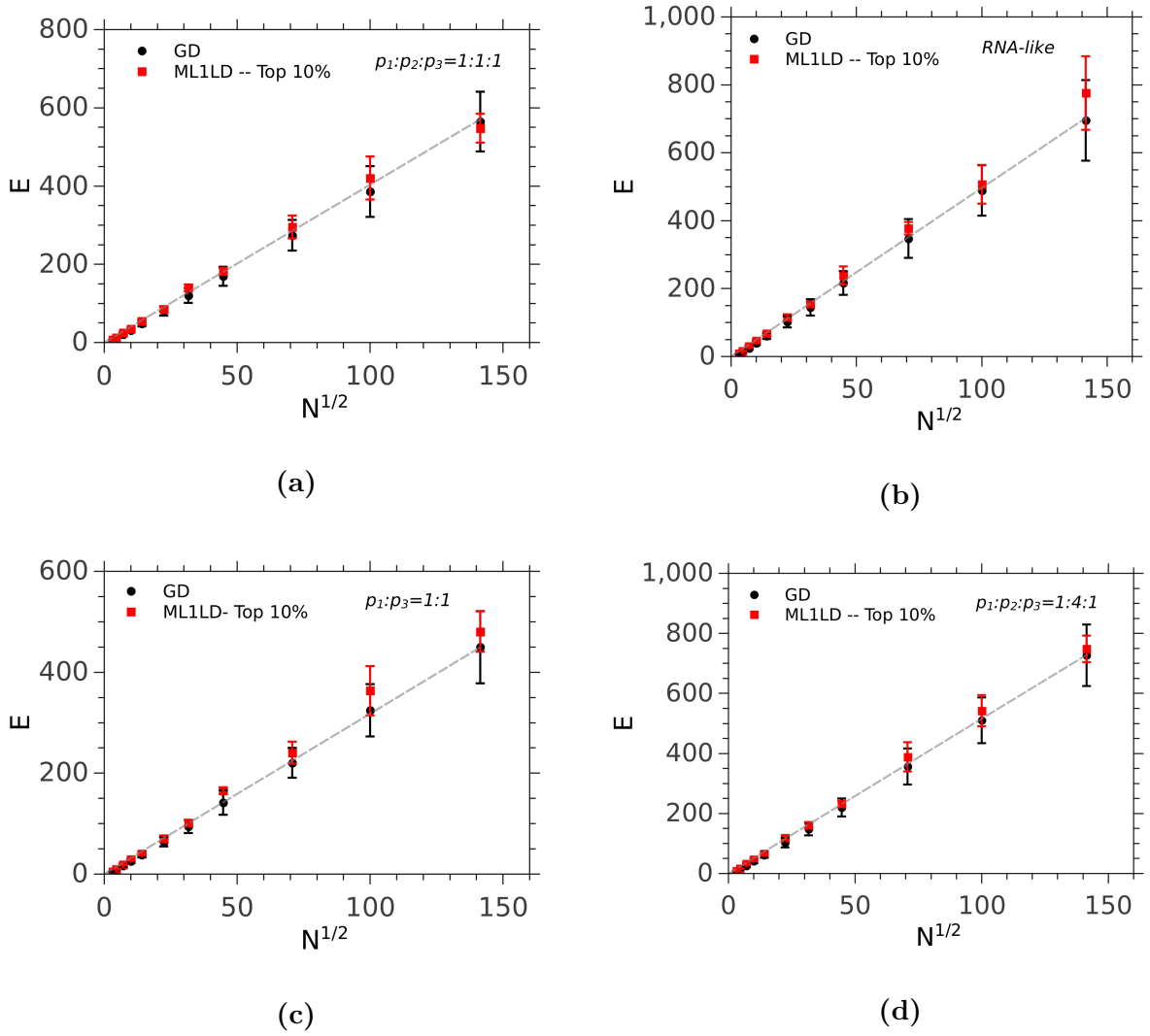

Figure 5.6: The average of the top deciles of the $ML1LD$ (red squares) in the RBP ensemble are approximately equal to the GD (black circles), for vertex degree distributions **(a)** $p_1 = p_2 = p_3 = 1/3$,**(b)** "RNA-like" $(p_1 = 26/126; p_2 = 75/126; p_3 = 24/126; p_4 = 1/126)$, **(c)** $p_2 = 0; p_1 = p_3 = 1/2$, **(d)** $p_1 = p_2 = 1/6; p_2 = 4/6$. The GD, measured in units of graph edges E, is proportional to $N^{1/2}$, consistent with $R_g \sim GD^{1/2}\ N^{1/4}$.

5.6 Computational time and memory usage of Prüfer

We measured the system's wall-clock time (in seconds) using Mathematica's built-in function AbsoluteTiming [Wolfram Research 2012] to test the efficacy of generating trees using the Prüfer shuffling method. More explicitly, the wall-clock time for the calculation of the R_g of a single tree using the Prüfer shuffling method and the Sequential Algorithm [Blitzstein 2011] was measured for the families of branched polymers considered in this work. The results of this "race" are summarized in Table 5.1 where from this we can see that the Prüfer shuffling method consistently lags behind the Sequential Algorithm over a wide range of N.

Table 5.1: Computation Times for the Calculation of R_g

	$p_3 : p_1 = 1 : 1; p_2 = 0$		$p_1 : p_2 : p_3 = 1 : 1 : 1$		$p_1 : p_2 : p_3 = 1 : 4 : 1$		RNA-like	
	Seq. Alg. (s)	Prüfer (s)	Seq. Alg. (s)	Prüfer (s)	Seq. Alg. (s)	Prüfer (s)	Seq. Alg. (s)	Prüfer (s)
N								
10	0.00116	0.00425	0.00117	0.00624	0.00141	0.00425	0.00136	0.00416
20	0.00184	0.00729	0.00201	0.0107	0.00199	0.00726	0.00246	0.00714
50	0.00445	0.0184	0.00512	0.0296	0.00599	0.0196	0.00599	0.0194
100	0.0104	0.0643	0.0112	0.0921	0.00993	0.0635	0.00114	0.0644
200	0.0313	0.212	0.0297	0.347	0.0260	0.219	0.0278	0.210
500	0.147	1.86	0.158	1.90	0.119	1.16	0.127	1.12
1000	0.517	8.39	0.543	8.75	0.476	5.33	0.507	4.42
2000	2.16	38.1	2.00	46.4	2.00	19.8	1.88	18.6
5000	16.9	42.1	16.6	30.1	14.6	49.6	15.9	19.0
10000	65.8	295	46.9	89.3	40.2	193	43.0	132

5.6.1 Prüfer based calculations are memory efficient

In this section we validate our claim that Prüfer based calculations (e.g. the LLD and the $ML1LD$) are memory-wise more efficient than canonical calculations involving the adjacency matrix. A tree with N vertices is typically stored as an $N \times N$ adjacency matrix and, it was found, requires $\sim N^2$ bytes of memory to be allocated. On the other hand, the same tree's Prüfer sequence (containing $N-2$ elements) requires less memory and scales as N bytes.

We investigated the memory demands of the tree's adjacency matrix and Prüfer sequence for the four families of branched polymers. Shown in Fig. 5.7, are log-log plots of the bytes of memory (Mem.) used to store a tree with N vertices, using the adjacency matrix (in black) and Prüfer sequence (red). The slopes of the linear regression lines through the adjacency matrix and Prüfer sequence data points were 2 and 1, respectively. The largest tree we could generate using the adjacency matrix was 20,000 vertices; a tree of this size consumed ≈ 32 GB of memory (i.e. the system limit).

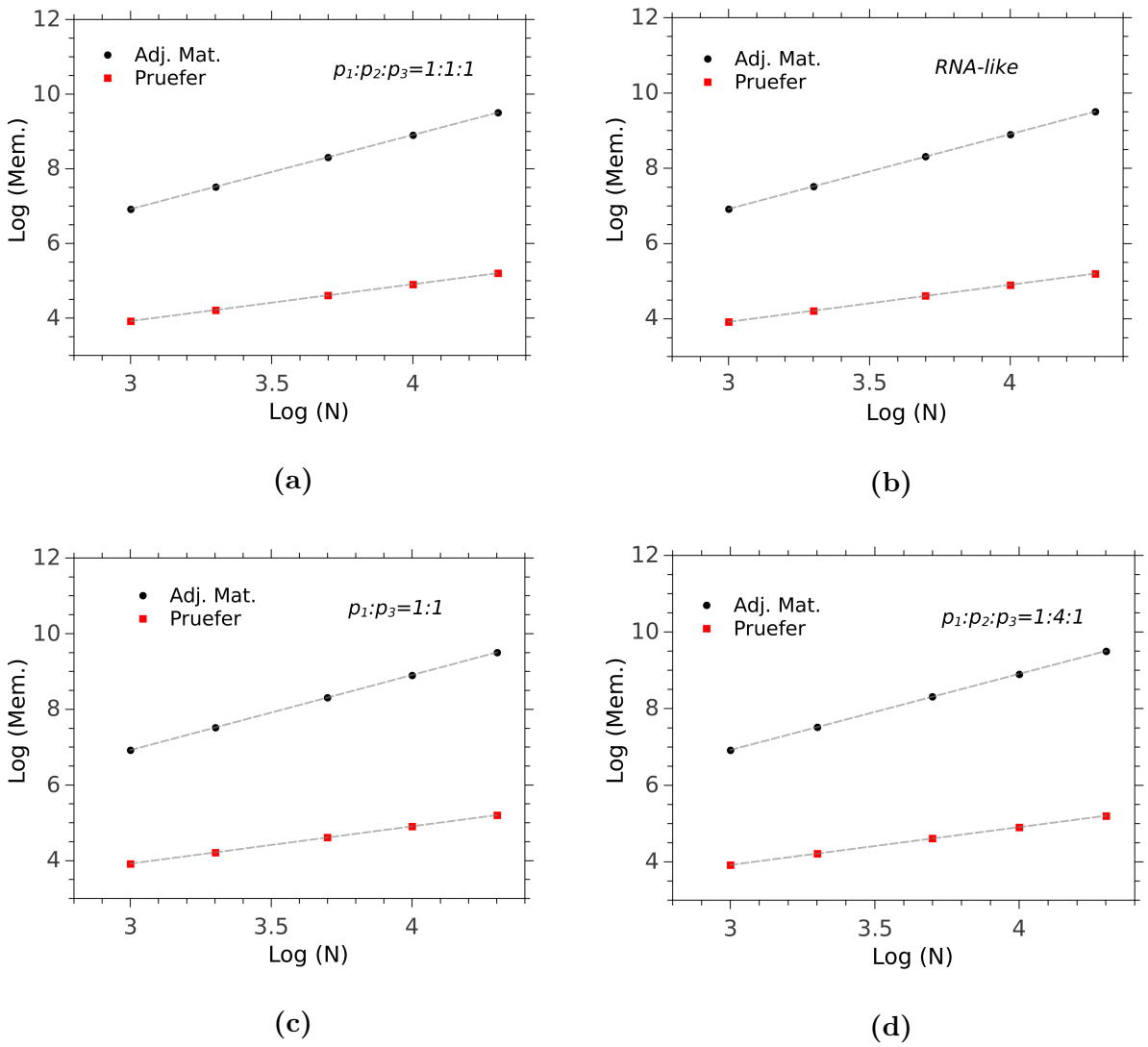

Figure 5.7: For a tree with N vertices, more bytes of memory (Mem.) are allocated for the tree's adjacency matrix (black) than for its corresponding Prüfer sequence (red); data shown on a log-log scale. For the four familes of branched polymers (**a-d**), it was found that the slopes of the linear regression lines through the adjacency matrix and Prüfer sequence data points were 2 and 1, respectively. That is, the adjacency matrix requires $\sim N^2$ bytes, while the Prüfer sequence requires $\sim N$ bytes.

5.7 Conclusions

The results presented in this section show that our Prüfer-based algorithm is slower yet, memory-wise, more efficient than the other algorithms we have used for calculating branched polymer sizes. It is not unlikely that professional programming could upgrade our "home-made" method to make it both faster and even more memory efficient. Nevertheless, presenting yet another algorithm for calculating the size of randomly branched polymers was not the primary goal in this work. We regard the Prüfer shuffle procedure as an elegant way to compare different types of branched polymeric structures, primarily RNA. The present work represents a step forward toward our more challenging goal of understanding the branching patterns associated with the secondary structures (tree graphs) of random and non-random RNA sequences, which are qualitatively different from those of ideal randomly branched polymers.

Appendix A

Distribution functions for 1D random adsorption

Suppose there are n 1D lattices, each with M binding sites, and a total of P proteins distributed randomly on them. We are interested in the probability of finding a lattice with at least j_o proteins.

The total number of ways of distributing P proteins on the n lattices is just $\binom{nM}{P}$. The number of ways of distributing P proteins on n lattices such that one lattice has at least j_o proteins is $\sum_{j=j_o}^{M} \binom{M}{j}\binom{nM-M}{P-j}$. So, the probability of observing our subsystem with at least j_o proteins is simply the fraction, $f(j_o)$, where $f(j_o) = \dfrac{\sum_{j=j_o}^{M} \binom{M}{j}\binom{nM-M}{P-j}}{\binom{nM}{P}}$. In the limit the number of lattices and protein in the system large –for fixed values of M, j_o, and the coverage fraction $\sigma = \dfrac{P}{nM}$– the factor $\dfrac{\binom{nM-M}{P-j}}{\binom{nM}{P}}$ reduces to $\sigma^j(1-\sigma)^{M-j}$. More explicitly,

when $n >> 1$ and $P >> j$ such that σ remains constant:

$$\begin{aligned}
\frac{\binom{nM-M}{P-j}}{\binom{nM}{P}} &= \frac{(nM-M)!P!(nM-P)!}{(P-j)!(nM-M-P+j)!(nM)!} \\
&= \frac{P^j(nM-P)!}{(nM)^M(nM-M-P+j)!} \\
&= \frac{P^j(nM-P)!}{(nM)^M(nM-P-(M-j))!} \\
&= \frac{P^j(nM-P)^{M-j}}{(nM)^M} \\
&= \frac{P^j(nM)^{M-j}(1-\sigma)^{M-j}}{(nM)^M} \\
&= \frac{P^j(1-\sigma)^{M-j}}{(nM)^j} \\
&= \sigma^j(1-\sigma)^{M-j}.
\end{aligned}$$

Thus, $f(j_o; M, \sigma)$ becomes:

$$f(j_o; M, \sigma) = \sum_{j=j_o}^{M} \binom{M}{j} \sigma^j (1-\sigma)^{M-j}. \tag{A.1}$$

The probability $p(j_o)$ of finding a lattice with exactly j_o proteins, we truncate the sum in Eq. (A.1) after the first term:

$$p(j_o; M, \sigma) = \binom{M}{j_o} \sigma^{j_o} (1-\sigma)^{M-j_o}, \tag{A.2}$$

which is the binomial distribution.

Appendix B

Canonical partition function for 1D lattice system

We follow the method and notation of Hill [Hill 1960] in order to compute $Q(n, M, \epsilon, T)$ –the canonical partition function of a 1D lattice with M sites, n occupied sites, nearest-neighbor potential energy ϵ, and temperature T. Let us denote the number of nearest-neighbor pairs by n_{11}. Instead of keeping track of the number of occupied-unoccupied pairs n_{01}, it is clearer to keep track of the number of groups of occupied sites[1], n_g ($n_g = \frac{n_{01}}{2}$). The relation between n, n_{11}, n_g is

$$n = n_{11} + n_g \tag{B.1}$$

and the energy of n proteins distributed on M sites with n_{11} pairs is

$$n_{11}\epsilon = (n - n_g)\epsilon. \tag{B.2}$$

Let $g(n, M, n_g)$ denote the number of configurations with exactly n_g, so the weight of these configurations is $g(n, M, n_g) \exp\left[\frac{-(n-n_g)\epsilon}{kT}\right]$. $Q(n, M, T)$, the sum of states of n proteins

[1] A group of occupied sites on either end of the lattice does not present a problem if we insist the occupied site on the end have an unoccupied neighbor. This boundary condition is reasonable for our system.

distributed on M sites is

$$Q(n, M, T) = \sum_{n_g=1} g(n, M, n_g) \exp\left[\frac{-(n - n_g)\epsilon}{kT}\right], \tag{B.3}$$

where the sum is over all possible values of n_g. The upper bound of n_g is the lesser of n and $M - n + 1$. We now derive an explicit expression for g.

There are n_g groups of occupied sites, which must each have at least one occupied site. This means we are free to distribute the remaining proteins amongst the n_g groups. Mathematically, we write this as

$$\frac{(n - n_g + n_g - 1)!}{(n - n_g)!(n_g - 1)!} = \frac{(n-1)!}{(n - n_g)!(n_g - 1)!} = \binom{n-1}{n_g - 1}. \tag{B.4}$$

We compute the number of ways of distributing the unoccupied sites in a similar manner. There are $n_g + 1$ groups of unoccupied sites, which differ from the occupied sites in that the unoccupied sites on the end can have zero members. This means we have two more unoccupied sites to distribute, so the number of unoccupied sites we can distribute among the $n_g + 1$ groups is $M - n - n_g - 1 + 2$. So there are in total

$$\begin{aligned}\frac{(M - n - n_g - 1 + 2 + n_g + 1 - 1)!}{n_g!(M - n - n_g + 1)!} &= \frac{(M - n + 1)!}{n_g!(M - n - n_g + 1)!} \\ &= \binom{M - n + 1}{n_g}\end{aligned} \tag{B.5}$$

ways of distributing the unoccupied sites; $g(n, M, n_g)$ is the product of equation (B.4) and (B.5):

$$g(n, M, n_g) = \binom{n-1}{n_g - 1}\binom{M - n + 1}{n_g}. \tag{B.6}$$

Appendix C

Supporting Information for the Competition Experiment

C.1 Equilibrium analysis of the competition experiments

Free energy (ΔA), energy (ΔE), and entropy ($T\Delta S$) differences as outlined in section 2.3 of the text were calculated for all the simulated competition experiments. Specifically $\Delta A(M_1; M) = A_f(M_1; M) - A_i(0; M)$ was calculated as a function of M_1 – the number of capsid proteins (CP) on one of the trees, with M denoting the total number of CP available to both trees. ΔE and $T\Delta S$ are defined similarly.

With the exception of "the long tree vs two short trees" competition, Eq. (4.1) was used to compute the free energy ΔA (see Eq. (4.2)). The corresponding calculation for the competition involving trees of different length was done using Eq. (4.6). Using Eqs. (3.20) and (4.4), ΔE was computed for competitions involving trees of the same length. For the competition involving one long tree and two short trees, $\Delta E = \left\langle E(M_1^f; M) \right\rangle - \langle E(0; M) \rangle$ where $\langle E(M_1; M) \rangle$ is the average energy of the system when M_1 CP are bound to the long tree and the remaining CPs are distributed among the two short trees. That is,

$$\langle E(M_1;M)\rangle = \langle E_1(M_1)\rangle + \frac{\sum_{M_2=0}^{M-M_1} Q_2(M_2)Q_3(M-M_1-M_2)\left[\langle E_2(M_2)\rangle + \langle E_3(M-M_1-M_2)\rangle\right]}{Q_{tot}(M-M_1)} \tag{C.1}$$

with $Q_{tot}(M-M_1)$ the normalizing factor. Finally, ΔS is derived from $T\Delta S = \Delta E - \Delta A$.

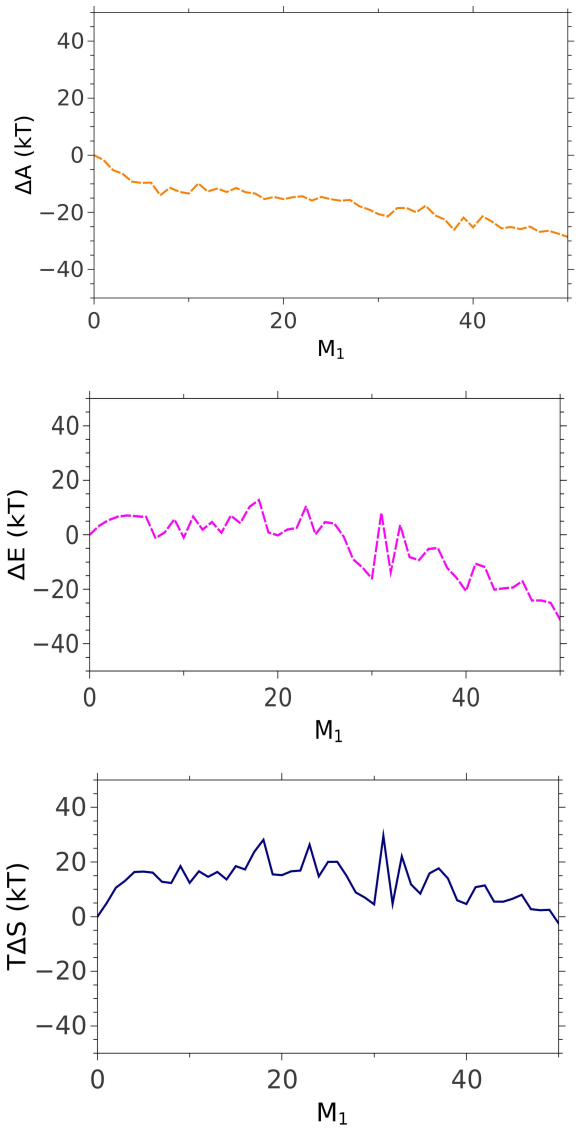

Figure C.1: Long tree vs. two short trees: Changes in the thermodynamic functions in the competition between the 50-vertex tree and two 25-vertex trees. ΔA (top), ΔE (middle), $T\Delta S$ (bottom) as a function of M_1 – the number of CP particles on the long tree, where M_1 ranges from 0 to 50. The free energy is minimal when $M_1 = 50$, consistent with the kinetic (particle exchange) simulations for this competition showing that practically all CP end up on the long tree (see Fig. 4.4 of main text). The dominant contribution to the free energy change is the energy gained by the condensation of CP on the long tree.

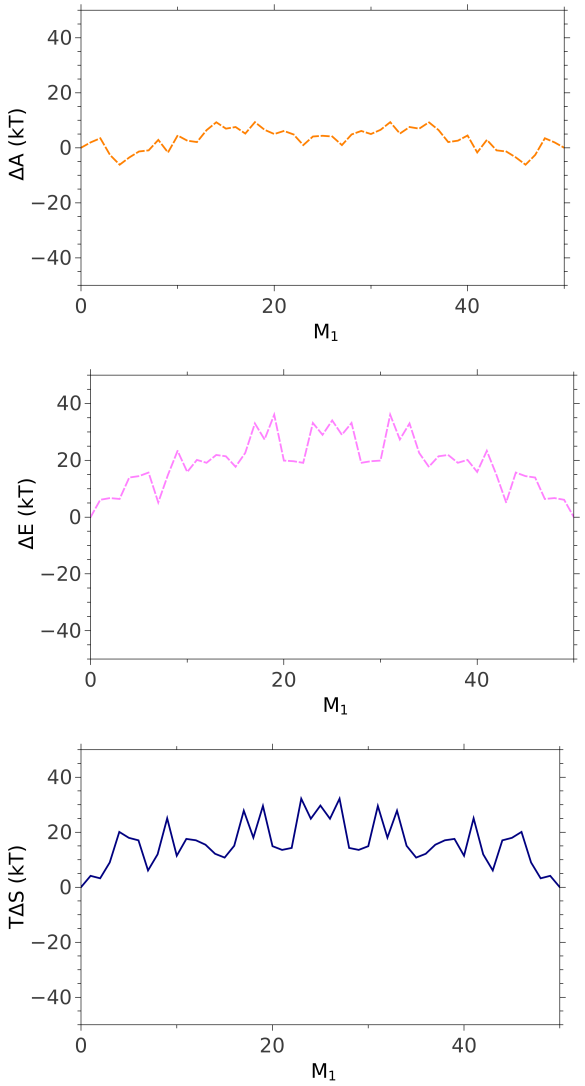

Figure C.2: Long tree vs. long tree: Changes in the thermodynamic functions in the competition between two identical 50-vertex trees. ΔA (top), ΔE (middle), $T\Delta S$ (bottom) as a function of M_1– the number of CP particles on one of the trees, where M ranges from 0 to 50. The free energy has two minima ($M_1 = 5$ and $M_1 = 45$) separated by an "activation" barrier of about 15 kT (measured by $A(15;50) - A(5;50)$), implying the system is stable to large fluctuations. The particle exchange simulations for this competition show that practically all CP end up on one of the long trees, remaining that way over the course of the simulation (see Fig. 4.5 of main text).

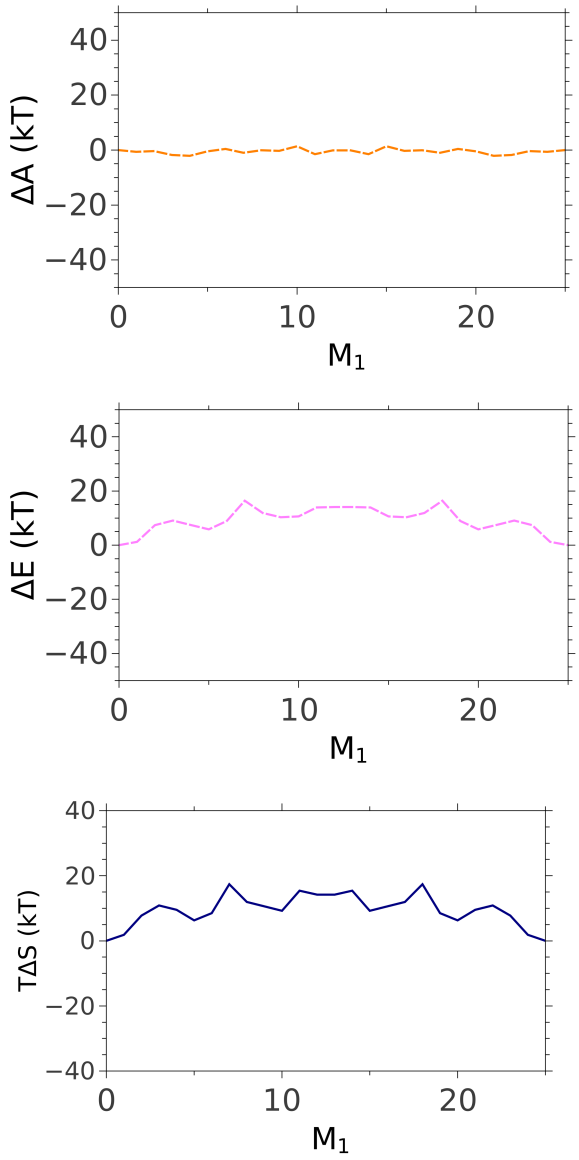

Figure C.3: Short tree vs. short tree: Changes in the thermodynamic functions in the competition between two identical 25-vertex trees. ΔA (top), ΔE (middle), $T\Delta S$ (bottom) as a function of M_1 – the number of CP particles on one of the short trees, where M ranges from 0 to 25. The free energy is essentially has essentially no minima (i.e., no activation barrier), consistent with the frequent and sizeable CP fluctuations seen in the particle exchange simulations for this competition (see Fig. 4.5) of main text).

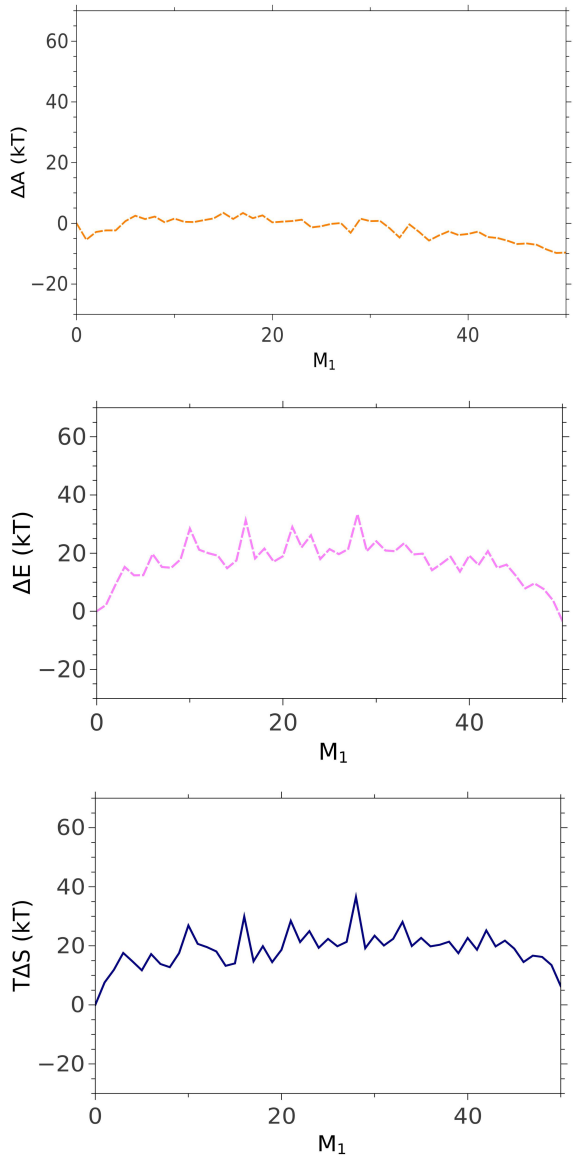

Figure C.4: Compact tree vs. extended tree: Changes in the thermodynamic functions in the competition between compact and extended 50-vertex trees. ΔA (top), ΔE (middle), $T\Delta S$ (bottom) as a function of M_1 – the number of CP on the compact tree, M ranging from 0 to 50. ΔA reaches its lowest values (≈ -10 kT), Close to saturation (i.e. $M_1 \approx 50$), consistent with the kinetic (particle exchange) simulations for this competition – showing that almost all CPs end up on the compact tree (see Fig. 4.7); $\Delta E \approx 0$, implying that the CP binding to the compact tree is mainly driven by the gain in conformational entropy of the "CP-free" extended tree.

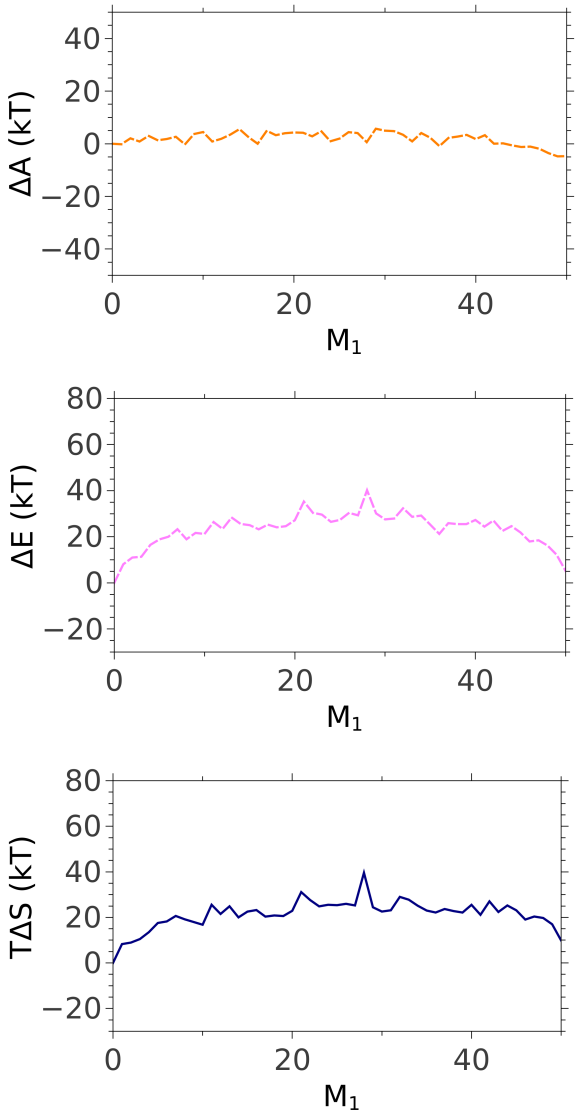

Figure C.5: Branched tree vs. Linear tree: Changes in the thermodynamic functions in the competition between compact and linear 50-vertex trees. ΔA (top), ΔE (middle), $T\Delta S$ (bottom) as a function of M_1 – the number of CP on the compact tree, M ranging from 0 to 50. ΔA reaches its lowest values (≈ -5 kT) close to saturation (i.e. $M_1 \approx 50$), consistent with the corresponding temporal evolution in Fig. 4.9 showing that almost all CPs end up on the compact tree. Near saturation ΔE is positive, indicating the energy is higher in the equilibrated state than it is initially. This energy barrier is overcome by the large entropy gain in the system as noted in Table 4.1.

C.2 Dependence of CP distribution on CP-CP interaction energy

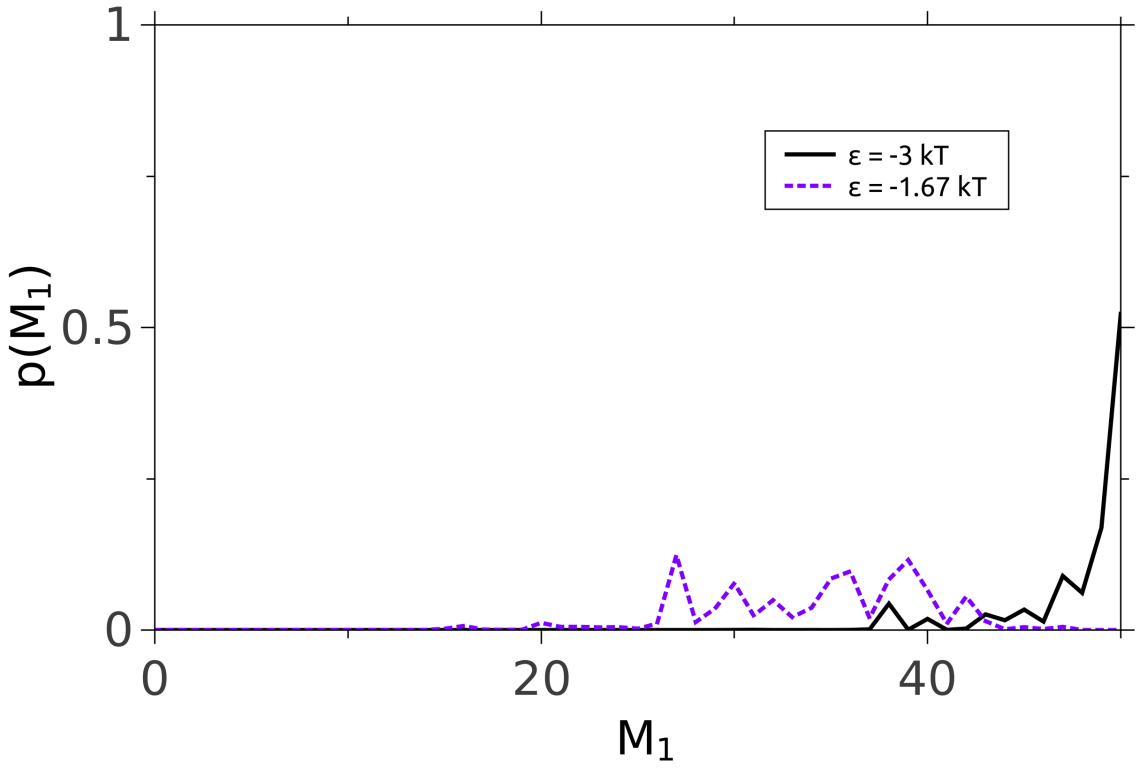

Figure C.6: Long tree vs. two short trees: Equilibrium probability distributions for the competition between a long tree and two short trees as a function of M_1 CP particles on the long tree with $\epsilon = -3$ kT (black) and $\epsilon = -1.67$ kT (purple). The stronger the interaction, the more CP particles end up on the long tree; notably, the longer tree still "wins" the competition (i.e. ends up with more than 25 CP particles) when the interaction is relatively weak.

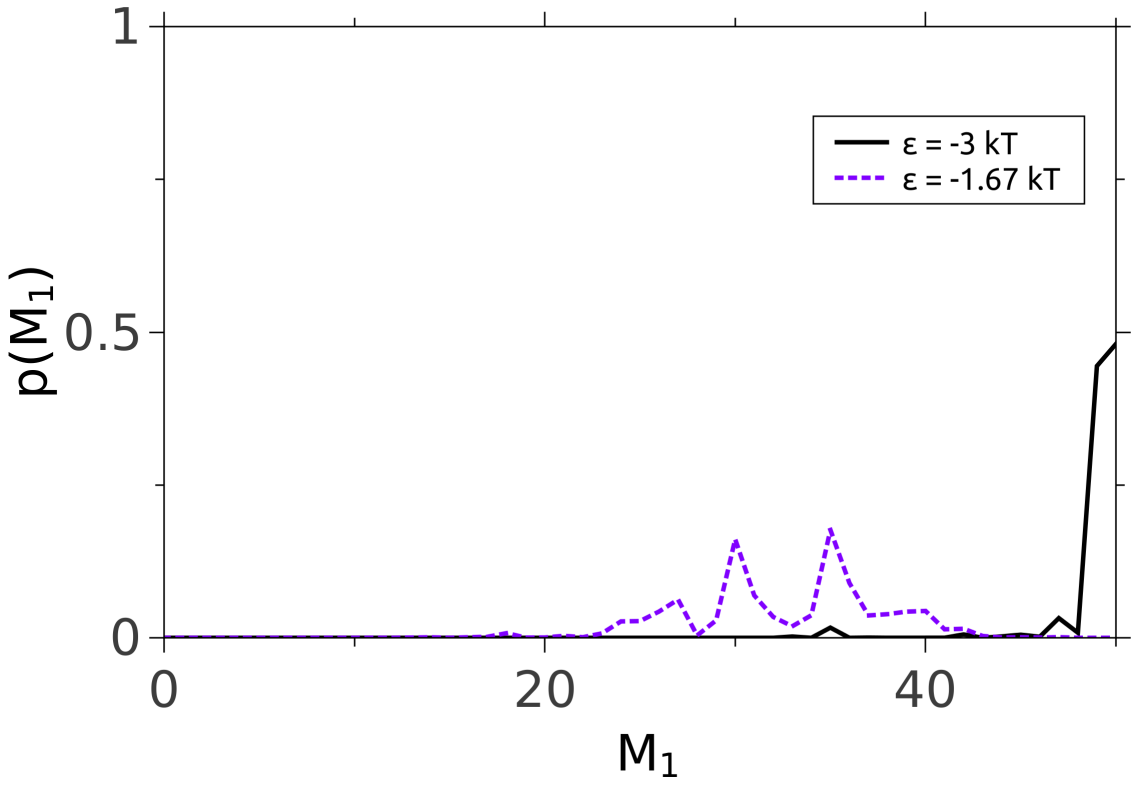

Figure C.7: Compact tree vs. Extended tree: Equilibrium probability distributions for the competition between the compact tree and the extended tree as a function of M_1 CP particles on the compact tree for both strong ($\epsilon = -3$ kT; black) and weak ($\epsilon = -1.67$ kT; purple) CP-CP interactions. CP particles preferentially bind to the compact tree even when ϵ is relatively weak. The compact tree's ability to steal CP, however, is considerably diminished for weaker ϵ.

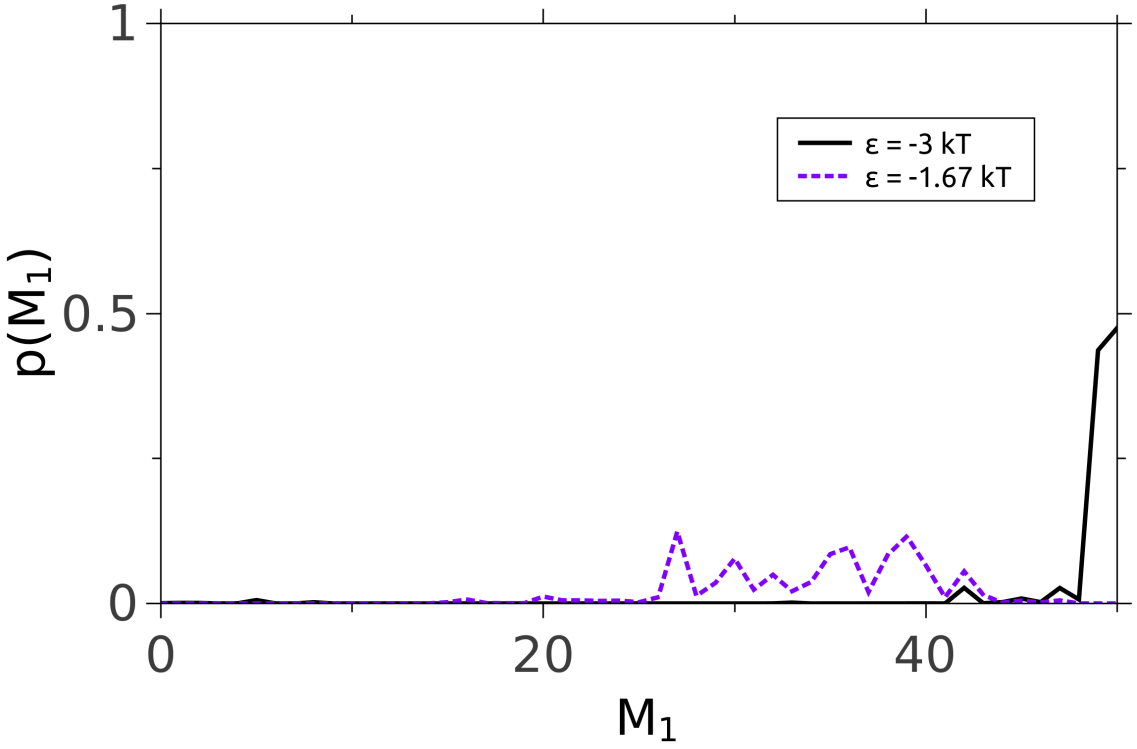

Figure C.8: Branched tree vs. Linear tree: Equilibrium probability distributions for the competition between the branched and linear trees as a function of M_1 CP particles with $\epsilon = -3$ kT (black) and $\epsilon = -1.67$ kT (purple). The branched tree "wins" the competition (i.e. ends up with more than 25 CP particles) even when the interaction energy is relatively weak.

C.3 Dependence of CP distribution on CP: Binding Site ratio

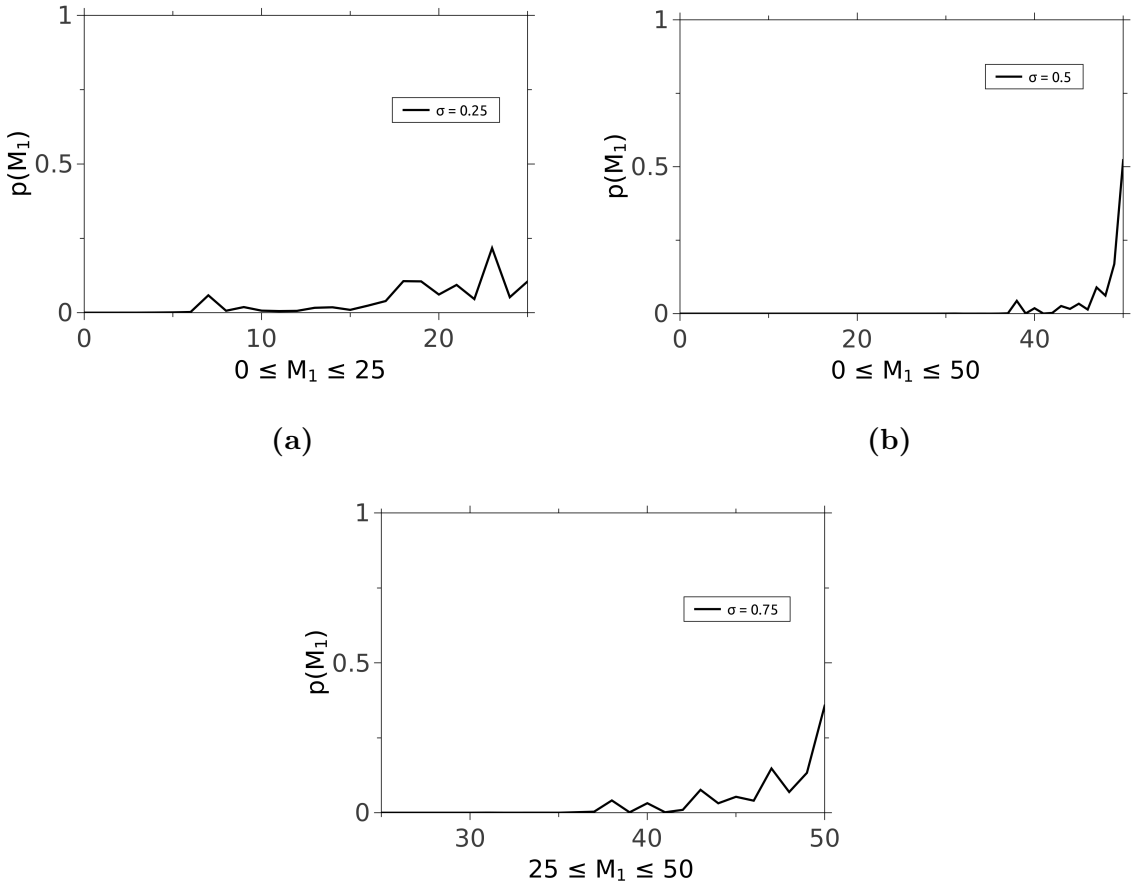

Figure C.9: Equilibrium probability distributions for the competition between a long tree and two short trees as a function of M_1 CP particles on the long tree for different $\sigma = \dfrac{M_{tot}}{L_{tot}}$, where M_{tot} and L_{tot} denote the total number of CP particles and binding sites in the system, respectively. For this competition $L_{tot} = 50$ and in the three cases considered [$\sigma = 0.25$ (a); $\sigma = 0.50$ (b); $\sigma = 0.75$ (c)], the long tree steals almost all the CP particles from the short trees, though this effect is attenuated for σ deviating from 0.5.

C.4 Equilibrium CP distribution between two identical trees

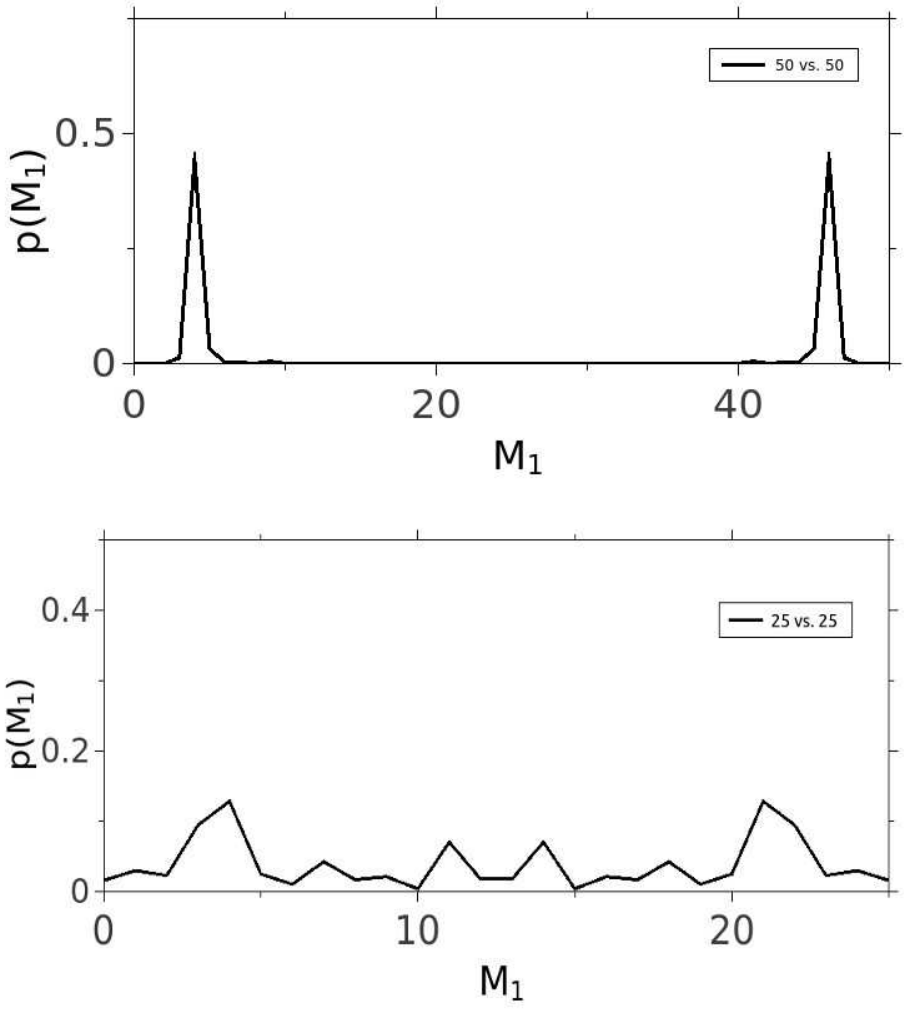

Figure C.10: The equilibrium probability distributions between two long trees (top) and two short trees (bottom), show that it is equally probable to find most of the particles on either tree. However, the kinetic simulations show the equilibrium distribution is only realized for the short trees. Particle swings between the long trees did not occur on the time scale of the simulation once most of the particles were bound to one tree. See Fig. 4.5 in the main text for the corresponding particle exchange simulations.

C.5 Changes of thermodynamic properties of each tree

In this section we extend the equilibrium analysis presented above by calculating $\Delta A_1(M;L)$, $\Delta E_1(M;L)$, and $T\Delta S_1(M;L)$ for every tree in the competition experiments. These calculations are done using Eq. (3.20) to directly evaluate the "free" energy difference

$$\Delta A_1(M;L) = -kT\frac{\ln(Q(L,M,T))}{\ln(Q(L,0,T))} \tag{C.2}$$

The corresponding values of the average energy of the tree $\Delta E_1(M;L) = \langle E_1(M)\rangle - \langle E_1(0)\rangle = \langle E_1\rangle$ were calculated using Eq. (3.19). Then $\Delta S_1(M;L)$ follows from $T\Delta S_1(M;L) = \Delta E_1(M;L) - \Delta A_1(M;L)$.

Clearly, the total free energy of a two-tree system composed of two polymers P_1 and P_2 of equal length L, is given by

$$\Delta A(M_1;M) = \Delta A_1(M_1;L) + \Delta A_2(M - M_1;L) \tag{C.3}$$

with an analogous definition for $\Delta E(M_1;M)$. Calculating the thermodynamic changes (i.e. $\Delta A_1(M;L)$, $\Delta E_1(M;L)$, and $T\Delta S_1(M;L)$) of each tree individually as a function of M, allows us to better understand how transferring CP to one tree offsets the energy/entropy gain (or loss) of the other tree.

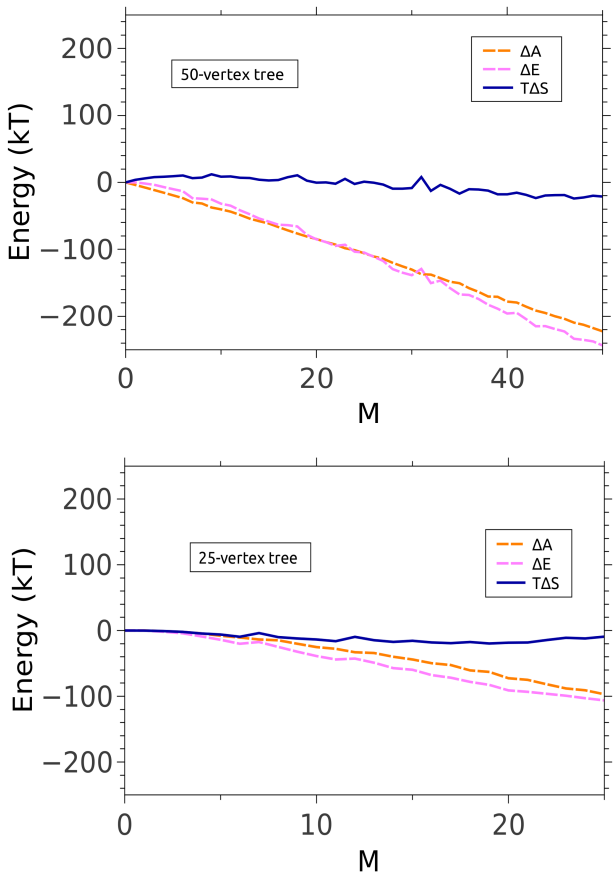

Figure C.11: Long tree and the short tree: Changes in the thermodynamic functions of a 50-vertex tree (top) and 25-vertex tree (bottom) as a function of the number of CP bound to their backbones. ΔA_1 (orange), ΔE_1 (pink), and $T\Delta S_1$ (blue) as a function of M – the number of CP on the long tree (top; M ranges from 0 to 50) and, separately, on the short tree (bottom; M ranges from 0 to 25). The thermodynamic differences of the small and large trees can be compared by accounting for the size difference (i.e. increasing the changes in free energy, energy, and entropy of the smaller tree by a factor of two). The energy is lower when all the CP particles are bound to the long tree (≈ -240 kT) than if they were all bound to the two short trees (-210 kT $= 2 \times -105$ kT). Furthermore, the entropy gain associated with the smaller trees (20 kT $= 2 \times 10$ kT) transferring its particles to the larger tree is approximately equal and opposite to the entropy loss of the larger tree (-20 kT).

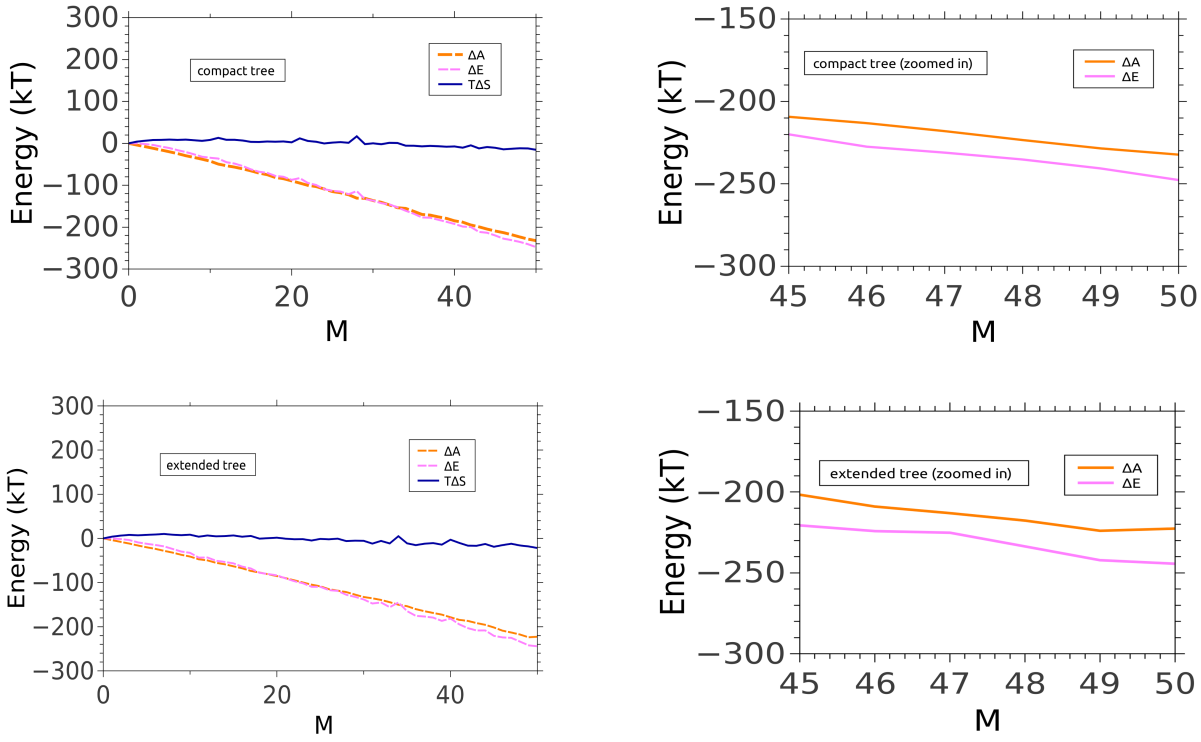

Figure C.12: Changes in the thermodynamic functions of the 50-vertex compact tree (top) and 50-vertex extended tree (bottom) as a function of the number of CP bound to their backbones. ΔA_1 (orange), ΔE_1 (pink) , and $T\Delta S_1$ (blue) as a function of M – the number of CP particles on the compact tree (top left) and, separately, on the extended tree (bottom left), M ranges from 0 to 50. ΔA_1 and ΔE_1 of the compact tree (top right) and of the extended tree (bottom right) are shown around the region close to saturation (i.e. M varies from 45 to 50). When both trees have $M = 50$ CP particles, the free energy of the compact tree is lower (-230 kT) than that of the extended tree (-220 kT). At this coverage, however, the energy of the compact tree is only slightly lower than the extended tree, implying the entropy is the predominant driving force in this process.

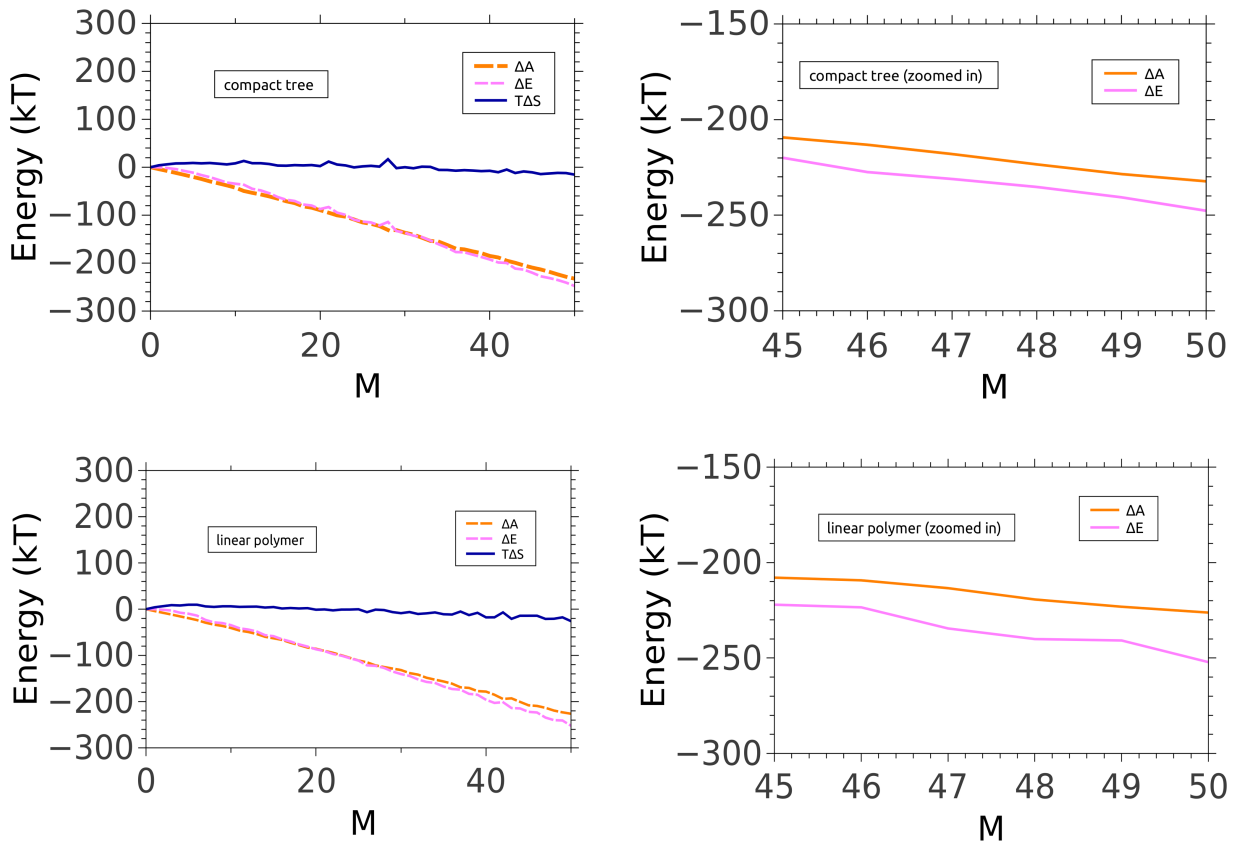

Figure C.13: Changes in the thermodynamic functions of the 50-vertex compact tree (top) and 50-vertex linear tree (bottom) as a function of the number of CP bound to their backbones. ΔA_1 (orange), ΔE_1 (pink) , and $T\Delta S_1$ (blue) as a function of M – the number of CP particles on the compact tree (top left) and, separately, on the linear tree (bottom left), M ranges from 0 to 50. ΔA_1 (orange) and ΔE_1 (pink) of the compact tree (top right) and of the extended tree (bottom right) around the region close to saturation (i.e. M varies from 45 to 50). When $M = 50$, the free energy of the compact tree is lower than that of the linear polymer by about 5 kT. However, for this value of M, the energy of the compact tree is higher, which is duly compensated by a substantial gain in entropy as noted in Table 4.1.

C.6 The radius of gyration of the CP-dressed trees

The radius of gyration (R_g) of each tree was computed as a function of M – the number of CP bound to its backbone. CP-CP interactions are attractive ($\epsilon = -3kT$) and short-ranged (between sites separated by a distance equal to the lattice spacing), favoring compact tree conformations, i.e., those with small R_g. The (two-dimensional) R_g was calculated using the common definition

$$R_g^2 = \frac{1}{L}\sum_{i,j} d_{i,j}^2 \tag{C.4}$$

where d_{ij} is the two-dimensional Euclidian distance between vertices i and j. The average radius of gyration $\langle R_g \rangle$ was calculated as a function of M using Eq. 3.19.

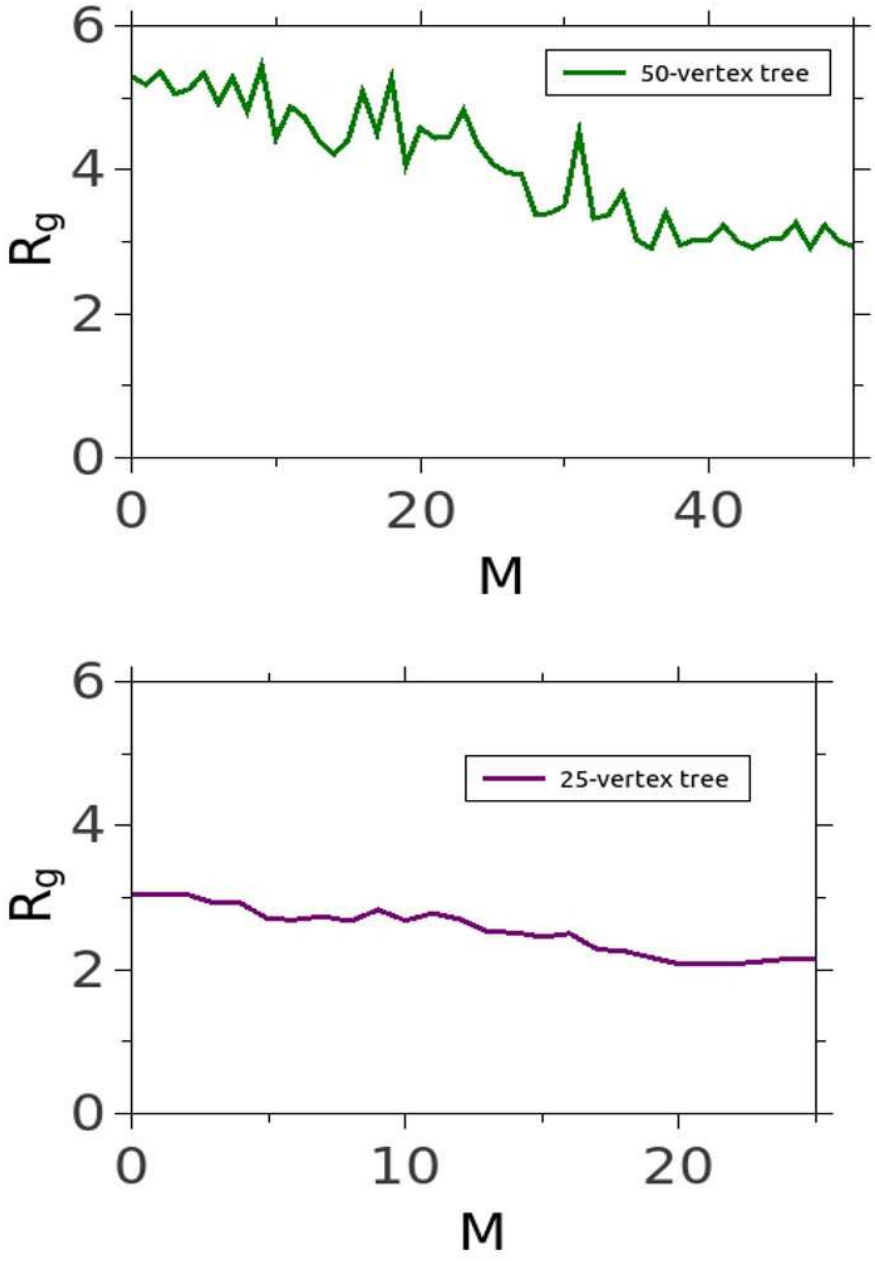

Figure C.14: Long tree and short tree: The average radius of gyration, $\langle R_g \rangle$, of the long tree (50-vertex) tree as a function of the number of bound CP, M (top), and the analogous plot for the short (25-vertex) tree (bottom).

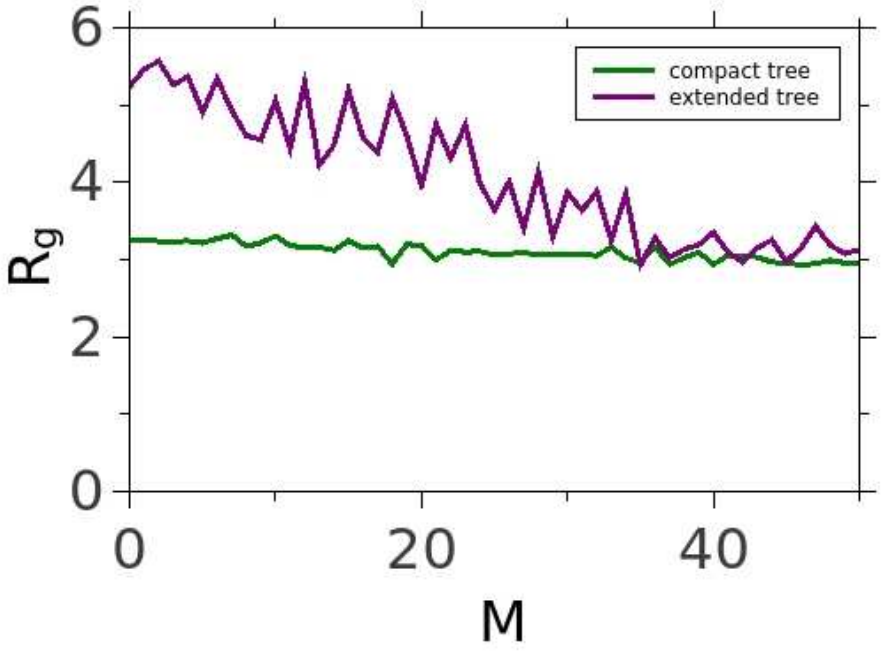

Figure C.15: Compact tree and the extended tree: The average radius of gyration, $\langle R_g \rangle$, of the compact and extended (both 50-vertex) trees as a function of their number of bound CP, M. The size of the compact tree is essentially independent of M; the size of the extended tree decreases significantly with increasing M.

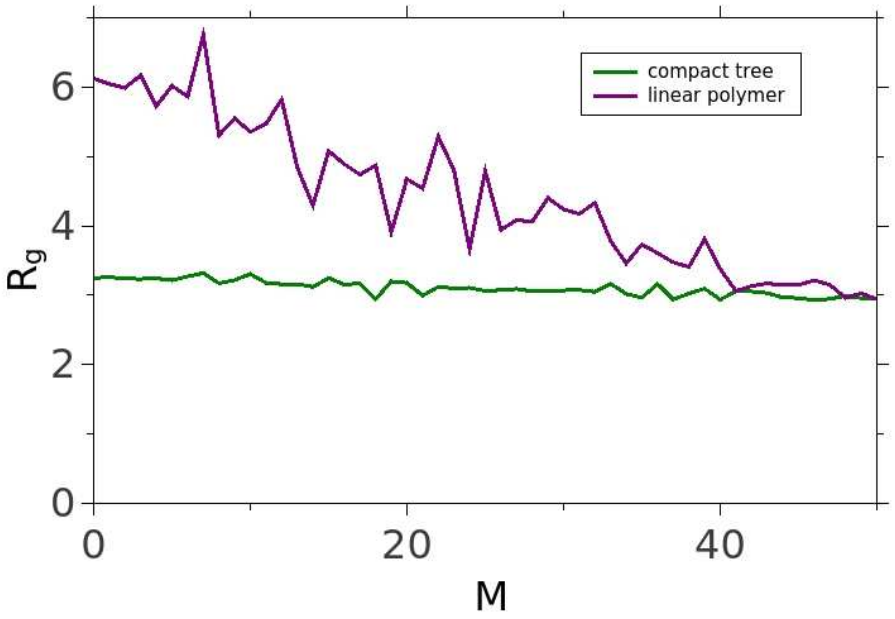

Figure C.16: Compact tree and the linear tree: The average radius of gyration, $\langle R_g \rangle$, of the compact and linear (both 50-vertex) trees as a function of their number of bound CP, M. $\langle R_g \rangle$ of the compact tree (green) and linear polymer (purple) as a function of M. The size of the compact tree is essentially independent of M; the size of the linear polymer depends on M, decreasing with increasing M.

Appendix D

Estimates of dimensions/energies of RNA and CP

D.1 RNA

D.1.1 Size

Average single bond length ≈ 0.15 nm (any general chemistry text book)

Average distance from 3'-3' carbon (six bonds) modelled as a random walk with six steps (1 nucleotide along the chain) ≈ 0.4 nm. Contour length of six bonds is 1 nm.

Size of nucleotide≈ 0.7 nm. Calculated by taking the cube root of the volume of a nucleotide in a duplex.

Average number of base pairs in a duplex ≈ 5 [Yoffe 2008]

Average number of nucleotides in a loop ≈ 10 [Yoffe 2008]

Rise/bp in a duplex ≈ 0.28 nm [van Holde 2005]

Contour length/bp in a duplex ≈ 0.283 nm

Diameter of duplex ≈ 2 nm [van Holde 2005]

Average length of RNA duplex ≈ 1.5 nm (measured along axis)

Average size of RNA loop ≈ 2-3 nm (calculated using self-avoiding/ideal polymer models)

Average end-to-end distance of 20 nt: 4 nm (assuming self-avoiding polymer in 3D); 3 nm (assuming ideal polymer).

D.1.2 Charge

Surface charge density of duplex $\approx -1 \frac{e}{\text{nm}^2}$

Charge density of a nt $\approx -1.4 \frac{e}{\text{nm}}$

D.1.3 Energy

Base-pair stacking energy depends on sequence, can be as weak as -1 kT for d(AT) and as strong as -4 kT for d(GC). [SantaLucia Jr. 1998]

D.2 CCMV capsid

D.2.1 Size

Outer radius of CCMV virus particle (T=3): 14.3 nm [Jeffrey 1995]

Outer surface area: 642 nm^2

Inner radius of CCMV virus particle (T=3): 10.5 nm [Jeffrey 1995]

Inner surface area: 346 nm^2

Pentamer-Hexamer angle (A-C dihedral angle): 142° [Jeffrey 1995]

Hexamer-Hexamer angle (B-C dihedral angle): 138° [Jeffrey 1995]

Angle between two pentamer faces in an icosahedron: 138.2°

D.3 CCMV CP

D.3.1 Size

Distance between C_α atoms: $\approx$ 0.45 nm[Fersht 1999]

Bond Angles: $\approx 120°$[Fersht 1999]

The average distance, modelling the three bonds between the C_α atoms as a freely rotating polymer, gives 0.3 nm [Rubinstein 2003]. Modelling as self-avoiding polymer in 3D gives 0.3 nm.

26 amino acids in flexible tail [Jeffrey 1995]

Contour Length of tail: 12 nm (using 0.45 nm); 7.8 nm (using 0.3 nm)

Average End-to-end distance of tail (using 0.3 nm): 1.5 nm (ideal polymer) 2 nm (self-avoiding polymer)

The surface of average RNA duplex could accommodate about 100 amino acids assuming each is 0.3 nm.

Number of times tail could wrap around duplex: twice (if contour is 12 nm) and once (if contour is 8 nm)

Size of CP (using outer radius): 1.8 nm; Size of CP-dimer: 3.6 nm

Size of CP (using outer radius) assuming equilateral triangle: 3 nm; size of CP-dimer: 5 nm

Size of CP (using inner radius): 1.4 nm; Size of CP-dimer: 2.8 nm

Size of CP (using inner radius) assuming equilateral triangles: 2 nm; size of CP-dimer: 4 nm.

D.3.2 Charge

10 basic residues in tail (6 Rs, 3 Ks, and N-terminus) [Jeffrey 1995]. BMV CP has one less basic residue in the tail.

Charge density of tail: $+0.8\frac{e}{\text{nm}}$ (using 0.45 nm); $+1\frac{e}{\text{nm}}$ (using 0.3 nm)

D.3.3 Energy

Energy of CP-dimer – CP-dimer interaction (pH ≈ 5): $\approx$ -6 kT [Johnson 2005]

RNA-CP-dimer binding energy: $\approx$ -20 kT

Bibliography

[Adolph 1977] K. Adolph and P. Butler. *Studies on the Assembly of a Spherical Plant Virus.* J. Mol. Biol., vol. 109, pages 345–357, 1977. (Cited on page 2.)

[Archer 2013] E. J. Archer, M. A. Simpson, N. J. Watts, R. O Kane, B. Wang, D. A. Erie, A. McPherson and K. M. Weeks. *Long-range architecture in a viral RNA genome.* Biochemistry, vol. 52, no. 18, pages 3182–3190, 2013. (Cited on page 60.)

[Athavale 2013] S. S. Athavale, J. J. Gossett, J. C. Bowman, N. V. Hud, L. D. Williams and S. C. Harvey. *In Vitro Secondary Structure of the Genomic RNA of Satellite Tobacco Mosaic Virus.* PLOS ONE, vol. 8, no. 1, pages 1–15, 2013. (Cited on page 60.)

[Bancroft 1967] J.B. Bancroft and E. Hiebert. *Formation of an infectious nucleoprotein from protein and nucleic acid isolated from a small spherical virus.* Virology, vol. 32, no. 2, pages 354–356, 1967. (Cited on page 2.)

[Blitzstein 2011] J. Blitzstein and P. Diaconis. *A sequential importance sampling algorithm for generating random graphs with prescribed degrees.* Internet Mathematics, vol. 6, no. 4, pages 489–522, 2011. (Cited on pages 75, 79.)

[Cadena-Nava 2012] R. Cadena-Nava, M. Comas-Garcia, R. Garmann, A. Rao, C. Knobler and W. Gelbart. *Self-Assembly of Viral Capsid Protein and RNA Molecules of Different Sizes: Requirement for a Specific High Protein/RNA Mass Ratio.* J. Virol., vol. 86, no. 6, pages 3318–3326, 2012. (Cited on pages 3, 6.)

[Caspar 1962] D. Caspar and A. Klug. *Physical principles in the construction of regular viruses.* In Cold Spring Harbor symposia on quantitative biology, volume 27, pages 1–24, 1962. (Cited on page 2.)

[Comas-Garcia 2012] M. Comas-Garcia, R. Cadena-Nava, A. Rao, C.M. Knobler and W.M. Gelbart. *In Vitro Quantification of the Relative Packaging Efficiencies of Single-Stranded RNA Molecules by Viral Capsid Protein.* J. Virol., vol. 86, no. 22, pages 12271–12282, 2012. (Cited on pages 4, 32.)

[Comas-Garcia 2014] M. Comas-Garcia, R.F. Garmann, S.W. Singaram, A. Ben-Shaul, C.M. Knobler and W.M. Gelbart. *Characterization of Viral Capsid Protein Self-Assembly around Short Single-Stranded RNA.* J. Phys. Chem. B, vol. 118, no. 27, pages 7510–7519, 2014. (Cited on page 56.)

[de Gennes 1968] P.G. de Gennes. *Statistics of branching and hairpin helices for the dAT copolymer.* Biopolymers, vol. 6, no. 5, pages 715–729, 1968. (Cited on page 61.)

[Diestel 2000] R. Diestel. Graph theory. Springer-Verlag, 2000. (Cited on pages 75, 76.)

[Dijkstra 1959] E.W. Dijkstra. *A note on two problems in connexion with graphs.* Numerische Mathematik, vol. 1, no. 1, pages 269–271, 1959. (Cited on pages 75, 77.)

[Dykeman 2013] E.C. Dykeman, P.G. Stockley and R. Twarock. *Building a viral capsid in the presence of genomic RNA.* Phys. Rev. Lett. E, vol. 87, no. 2, 2013. (Cited on page 58.)

[Dykeman 2014] E. C. Dykeman, P. G. Stockley and R. Twarock. *Solving a Levinthal's paradox for virus assembly identifies a unique antiviral strategy.* Proc. Nat. Acad. Sci., vol. 111, no. 14, pages 5361–5366, 2014. (Cited on page 58.)

[Fang 2011] L. T. Fang, W.M. Gelbart and A. Ben-Shaul. *The size of RNA as an ideal branched polymer.* J. Chem. Phys., vol. 135, page 155105, 2011. (Cited on page 34.)

[Fersht 1999] A. Fersht. Structure and mechanisms om protein science. W.H. Freeman Co., 1999. (Cited on pages 112, 113.)

[Floyd 1962] R. W. Floyd. *Algorithm 97: Shortest Path.* Commun. ACM, vol. 5, no. 6, page 345, 1962. (Cited on pages 75, 77.)

[Fraenkel-Conrat 1955] H. Fraenkel-Conrat and R.C. Williams. *Reconstitution of Active Tobacco Mosaic Virus from its Inactive protein and nucleic acid components.* Proc. Nat. Acad. Sci., vol. 41, no. 10, pages 690–698, 1955. (Cited on page 2.)

[Frenkel 1996] D. Frenkel and B. Smit. Understanding molecular simulation: From algorithms to applications. Academic Press, 1996. (Cited on pages 17, 28, 37, 38.)

[Gan 2004] H. H. Gan, D. Fera, J. Zorn, N. Shiffeldrim, M. Tang, U. Laserson, N. Kim and T. Schlick. *RAG: RNA-As-Graphs database-concepts, analysis, and features.* Bioinfomatics, vol. 20, no. 8, pages 1285–1291, 2004. (Cited on pages 25, 34.)

[Garmann 2014a] R. F. Garmann, M. Comas-Garcia, M. Koay, J. Cornelissen, C. M. Knobler and W. M. Gelbart. *Role of electrostatics in the assembly pathway of a single-stranded RNA virus.* J. Virol., vol. 88, no. 18, pages 10472–10479, 2014. (Cited on page 57.)

[Garmann 2014b] R.F. Garmann, M. Comas-Garcia, A. Gopal, C.M. Knobler and W. M. Gelbart. *The Assembly Pathway of an Icosahedral Single-Stranded RNA Virus Depends on the Strength of Inter-Subunit Attractions.* J. Mol. Bio., vol. 426, no. 5, pages 1050 – 1060, 2014. (Cited on pages 3, 5.)

[Garmann 2015] R. F. Garmann, A. Gopal, S. S. Athavale, C. M. Knobler, W. M. Gelbart and S. C. Harvey. *Visualizing the global secondary structure of a viral RNA genome with cryo-electron microscopy.* RNA, vol. 21, no. 5, pages 877–886, 2015. (Cited on page 60.)

[Gonca 2014] E. Gonca, J. Wagner, P. Van Der Schoot, R. Podgornik and R. Zandi. *RNA topology remolds electrostatic stabilization of viruses*. Phys. Rev. E., vol. 89, no. 3, page 032707, 2014. (Cited on page 59.)

[Gopal 2014] A. Gopal, D. Egecioglu, A. M. Yoffe, A. Ben-Avinoam, A.L.N. Rao, C.M. Knobler and W.M. Gelbart. *Viral RNAs are unusually compact*. PLOS ONE, 2014. (Cited on pages 35, 42, 74.)

[Grassberger 1997] P. Grassberger. *Pruned-enriched Rosenbluth method: Simulations of Theta polymers of chain length up to 1000000*. Phys. Rev. E., vol. 56, page 3682, 1997. (Cited on page 30.)

[Grassberger 2002] P. Grassberger. *Go with the winners: a general Monte Carlo strategy*. Comp. Phys. Comm., vol. 147, pages 64–70, 2002. (Cited on page 30.)

[Hill 1960] T.L. Hill. An introduction to statistical thermodynamics. Addison-Wesley Inc., Mass., 1 édition, 1960. (Cited on page 86.)

[Jeffrey 1995] S. Jeffrey, S. Munshi, G. Wang, T. S. Baker and J. E. Johnson. *Structures of the native and swollen forms of cowpea chlorotic mottle virus determined by X-ray crystallography and cryo-electron microscopy*. Structure, vol. 3, no. 1, pages 63–78, 1995. (Cited on pages 112, 113.)

[Johnson 1997] J. Johnson and J. Speir. *Quasi-equivalent Viruses: A Paradigm for Protein Assemblies*. J. Mol. Biol., vol. 269, no. 3, pages 665–675, 1997. (Cited on page 3.)

[Johnson 2005] J. Johnson, J. Tang, Y. Nyame, D. Willits, M. Young and A. Zlotnick. *Regulating Self-Assembly of Spherical Oligmers*. Nano Lett., vol. 5, no. 4, pages 765–770, 2005. (Cited on pages 3, 22, 42, 113.)

[Khokhlov 1994] A.R. Khokhlov and A.Y. Grosberg. Statistical physics of macromolecules. Polymers and Complex Materials. American Institute of Physics, 1994. (Cited on pages 61, 72.)

[Li 1995] B. Li, N. Madras and A.D. Sokal. *Critical exponents, hyperscaling, and universal amplitude ratio for two- and three-dimensional self-avoiding walks.* J. Stat. Phys., vol. 80, no. 3-4, pages 661–754, 1995. (Cited on page 22.)

[Lorenz 2011] R. Lorenz, S. H. Bernhart, C. Höner zu Siederdissen, H. Tafer, C. Flamm, P. F. Stadler and I. L. Hofacker. *ViennaRNA Package 2.0.* Alg. Mol. Bio., vol. 6, no. 1, pages 1–14, 2011. (Cited on page 74.)

[Metropolis 1953] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller and E. Teller. *Equation of State Calculations by Fast Computing Machines.* J. Chem. Phys., vol. 21, pages 1087–1092, 1953. (Cited on pages 14, 38.)

[Moon 1970] J.W. Moon. Counting labelled trees. Canadian Mathematical Congress, 1970. (Cited on pages 61, 62.)

[Perlmutter 2013] J. D. Perlmutter, C. Qiao and M. F. Hagan. *Viral genome structures are optimal for capsid assembly.* eLife, vol. 2, 2013. (Cited on page 59.)

[Perlmutter 2015] J. D. Perlmutter and M. F. Hagan. *The role of packaging sites in efficient and specific virus assembly.* Journal of molecular biology, vol. 427, no. 15, pages 2451–2467, 2015. (Cited on page 58.)

[Pruefer 1918] H. Pruefer. *Neuer Beweis Eines Satzes Ueber Permutationen.* Archiv der Mathematik, 1918. See also wikipedia entry for Pruefer sequence. (Cited on pages 4, 50, 61.)

[Rosenbluth 1955] M. N. Rosenbluth and A. W. Rosenbluth. *Monte Carlo Calculation of the Average Extention of Molecular Chains*. J. of Chem. Phys., vol. 23, pages 356–359, 1955. (Cited on pages 28, 37.)

[Rubinstein 2003] M. Rubinstein and R. Colby. Polymer physics. Oxford University Press, Oxford, 1 édition, 2003. (Cited on pages 61, 70, 71, 113.)

[SantaLucia Jr. 1998] J. SantaLucia Jr. *A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics*. Proc. Nat. Acad. Sci., vol. 95, no. 4, pages 1460–1465, 1998. (Cited on page 112.)

[Speir 2006] J.A. Speir, B. Bothner, C. Qu, D.A. Willits, M.J. Young and J.E. Johnson. *Enhanced local symmetry interactions glglobal stabilize a mutant virus capsid that maints infectivity dynamics*. J. Virol., vol. 80, pages 3582–3591, 2006. PDB ID: 1ZA7. (Cited on page 2.)

[Tzlil 2005] S. Tzlil and A. Ben-Shaul. *Flexible Charged Macromolecules on Mixed Fluid Membranes: Theory and Monte Carlo Simulations*. Biophys. J., vol. 89, pages 2972–2987, 2005. (Cited on pages 17, 37.)

[van Holde 2005] K. E. van Holde, C. Johnson and P. Shing Ho. Principles of physical biochemistry. Prentice Hall, 2 édition, 2005. (Cited on page 111.)

[Voorhees 1992] P. W. Voorhees. *Ostwald ripening of two-phase mixtures*. Ann. Rev. Mat. Sci., vol. 22, no. 1, pages 197–215, 1992. (Cited on page 47.)

[Warshall 1962] S. Warshall. *A Theorem on Boolean Matrices*. J. ACM, vol. 9, no. 1, pages 11–12, January 1962. (Cited on pages 75, 77.)

[Wolfram Research 2012] Inc. Wolfram Research. Mathematica. Wolfram Research, Inc., version 9.0 édition, 2012. (Cited on pages 75, 79.)

[Wu 2013] B. Wu, J. Grigull, M. O. Ore, S. Morin and K. A. White. *Global organization of a positive-strand RNA virus genome.* PLOS Pathog., vol. 9, no. 5, page e1003363, 2013. (Cited on pages 59, 60.)

[Yoffe 2008] A. M. Yoffe, P. Prinsen, A. Gopal, C. M. Knobler, W. M. Gelbart and A. Ben-Shaul. *Predicting the sizes of large RNA molecules.* Proc. Nat. Acad. Sci., vol. 105, no. 42, pages 16153–16158, 2008. (Cited on pages 34, 111.)

[Zandi 2009] R. Zandi and P. Van der Schoot. *Size regulation of ss-RNA viruses.* Biophys. J., vol. 96, no. 1, pages 9–20, 2009. (Cited on page 46.)

[Zeng 2012] Y. Zeng, S. B. Larson, C. E. Heitsch, A. McPherson and S. C. Harvey. *A model for the structure of satellite tobacco mosaic virus.* J. Struct. Bio., vol. 180, no. 1, pages 110–116, 2012. (Cited on page 36.)

[Zimm 1949] B. H. Zimm and W. H. Stockmayer. *The Dimensions of Chain Molecules Containing Branches and Rings.* J. Chem. Phys., vol. 17, no. 12, pages 1301–1314, 1949. (Cited on page 61.)

[Zinke-Allmang 1992] M. Zinke-Allmang, L. C. Feldman and M. H. Grabow. *Clustering on surfaces.* Surface Science Reports, vol. 16, no. 8, pages 377–463, 1992. (Cited on page 47.)

[Zlotnick 2005] Adam Zlotnick. *Theoretical aspects of virus capsid assembly.* J. Mol. Recog., vol. 18, no. 6, pages 479–490, 2005. (Cited on page 45.)