Advances in Monte Carlo Techniques with Application to Lattice Protein Aggregation

by

Yuanwei Xu

Thesis

Submitted in partial fulfilment of the requirements for the

degree of

Doctor of Philosophy in Scientific Computing

University of Warwick, Centre for Scientific Computing

March 2016

THE UNIVERSITY OF
WARWICK

Contents

List of Tables

List of Figures

Acknowledgments

I would like to thank my supervisor, Professor P. Mark Rodger, for his guidance throughout the completion of my research degree and, especially, for his patience at explaining unfamiliar concepts that I frequently came across during the start of my research, when I faced huge interdisciplinary barrier.

I also thank my parents for providing immense support, both financially and mentally, in order for me to pursue my degree. On several occasions during my time at Warwick when I felt discouraged, my mother's heartfelt encouragement and her words of wisdom helped me regain my confidence.

I would also like to mention some inspiring textbooks that guided me through the learning process and helped pave the way for understanding important concepts in biophysics, statistical mechanics and various computational methods. These books provide insights through simple yet revealing models, and let me see the connection with my mathematical background. I really benefited a lot from reading these books: *Statistical Physics of Biomolecules: An Introduction* by Daneil Zuckerman, *Molecular Driving Forces: Statistical Thermodynamcis in Biology, Chemistry, Physics and Nanoscience* by Ken A. Dill and Sarina Bromberg, and *Monte Carlo Strategies in Scientific Computing* by Jun S. Liu.

Finally, I would like to thank the Centre for Scientific Computing, for supporting me to attend training courses and present at conferences, and for providing computational environment, resources, and technical support.

Yuanwei Xu

Declarations

This thesis is submitted to the University of Warwick in support of my application for the degree of Doctor of Philosophy. It has been composed by myself and has not been submitted in any previous application for any degree. The results of Chapter 5 have been published by the author [68].

Abstract

Motivated by an intricate mechanism to transport folded proteins across biological membranes, known as the Twin-arginine translocation (Tat) pathway, we construct lattice protein models in an attempt to study the aggregation of the membrane protein *TatA*, which plays an integral role during active Tat translocation. We develop force field that characterizes intra- and inter-residue interactions, as well as how each residue interacts with its environment.

Although written with the Tat process in mind, this thesis is mainly devoted to developing efficient Monte Carlo schemes for biomolecular simulations, which are often challenged and impeded by complex energy landscapes. To tackle the local trap problem that is typical in Metropolis sampling, the idea of dynamic weighting is incorporated into the parallel tempering (PT) algorithm. Our results show that, when applied to the lattice-protein model, the modified PT algorithm is capable of locating the low energy state much more quickly, but does not produce reliable estimates for equilibrium expectations.

A modern method for free energy calculation, called the multistate Bennett acceptance ratio (MBAR) estimator, is reviewed from a statistical perspective, reminiscent of the underlying statistical theory which the method is based upon. Instead of adopting the common practice of using MBAR as a post-simulation analysis tool, we propose a new approach that integrates MBAR into simulation, allowing the simulation to benefit from the statistical optimality of the MBAR estimator. We show that the MBAR-enhanced Monte Carlo improves simulation efficiency of the lattice-protein aggregation model and, since it can also be applied to continuous models, provides a promising alternative to the study of more realistic systems.

The new method is then applied to our model of TatA, where the protein features both a transmembrane and an amphipathic helix. The effect of individual helices on dimerization was studied and problem with the move set was identified. In this thesis, we used *pull move* and translation move as our Monte Carlo trial moves. Implementation details of pull moves, which are often omitted by many researchers who use them for sampling configuration space, are given in Chapter 1. We show that, for our double-helix TatA model, pull moves are no longer efficient moves and therefore, for future study of more realistic systems, we point to several methods which all attempt to design efficient trial moves. Aggregation of more than two polymer chains was also considered in this thesis.

Chapter 1

Introduction

> A journey of a thousand miles
> begins with a single step.
>
> Laozi, *Tao Te Ching*

The work in this thesis was initially motivated by a biochemical process known as the Twin-arginine translocation (Tat) pathway, which is utilized by bacteria and plant chloroplasts for translocating folded proteins across membrane. It is an intriguing process in that the protein does not have to unfold to move through the permeative lipid bilayer. Of particular interest to us is a key step in Tat involving aggregation of certain membrane proteins that form the translocation channel. We shall present an overview of the Tat pathway in Chapter 6.

Although inspired by the Tat mechanism, this thesis focuses primarily on method development, in particular, on Monte Carlo methods in biomolecular modelling. Thus, our work is not limited to the Tat mechanism that motivates this study, it also provides a framework to study multi-chain dynamics, a prominent example of such is protein aggregation, which is an active area of research in biochemistry and biophysics due to its association with numerous human diseases. In this chapter, we briefly introduce protein aggregation in order to motivate the reader and to show potential applications of our work in this area. We then describe our lattice model and the force field that we used to characterize various interactions in the multi-polymer system. The move set adopted in our Monte Carlo simulations will also be discussed.

In Chapter 2, we review some commonly used Monte Carlo (MC) techniques as a preparation for the following chapters. Specifically, the Metropolis algorithm, parallel tempering (PT) and multicanonical simulation will be examined. A perhaps unfamiliar paradigm of MC simulation to many researchers in statistical physics and computational chemistry, the dynamic weighting Monte Carlo, will also be discussed. We will also review some other related methods, such as the Wang-Landau algorithm and transition matrix Monte Carlo.

Having introduced the dynamic weighting framework, we present in Chapter 3 our study of a new algorithm which incorporates the idea of dynamic weighting into parallel tempering, and a comparison between this modified PT algorithm and the bare PT.

Chapter 4 looks at two of the methods used to analyze multiple equilibrium simulation data, the weighted histogram analysis method and the multistate Bennett acceptance ratio (MBAR) estimator. Both can be recast as statistical problems, in particular, the latter method has its root in extended bridge sampling theory [59] and, as such, will be reviewed from a statistical perspective.

In Chapter 5, we describe a new approach to calculate density of states using the MBAR estimator. The estimated density of states can then be used to guide subsequent MC simulations. We use a combination of parallel tempering, MBAR and multicanonical sampling as a demonstration of the method. This is an exemplar where simulation techniques are combined with an analysis method that has been proven to be statistically optimal, thereby providing a synergy between the existing methods.

Finally, in Chapter 6, we apply the method in Chapter 5 to more complex models as our first steps towards a better understanding of membrane protein *TatA* aggregation in the Tat mechanism. We construct lattice models that capture essential structural features of TatA, namely its double-helix nature. We study how aggregation might be affected by these secondary structures and, in the meanwhile, identify problems in simulation with this extra complexity and propose strategies to alleviate them.

We believe our work contributes to the development of Monte Carlo methodology and will provide valuable insights to researchers in the Tat community who wish to utilize simulation tools, to computational scientists working on applications in statistical mechanics, and to statisticians developing new Monte Carlo algorithms.

1.1 Protein Aggregation

Proteins are macromolecules that are essential to the functioning of living organisms. The primary structure of protein consists of a chain of amino acids that are coded by genes. To be fully functional, protein must fold into a specific three-dimensional structure called the native structure. The native structure corresponds to the global minimum of the free energy landscape [11]. Because of the presence of an enormous amount of possible conformations, how proteins find their native structure in a very short timescale has long been a mystery. It is now clear that rather than a systematic search, a protein only sample a small number of conformations to reach its native structure [15]. The correct folding to the native state depends both on interactions among different residual pairs of the molecule and on multiple contributing factors from the crowded cell [14]. Since many proteins do not fold in isolation and additional proteins in the environment may affect the kinetics of the folding process of a synthesized protein, misfolding can occur and aggregates can be formed. These misfolded proteins give rise to the loss of biological functions if not degraded properly by the cell.

A recent review revealed that one type of aggregate, amyloid fibrils, was linked with approximately 50 disorders including such neurodegenerative diseases as Alzheimers disease, Spongiform encephalopathies and Parkinsons disease [32]. Due to its medical significance, increasingly more research, experimental as well as computational approaches, have been focusing on the study of amyloid structures. The commonly used computational approaches for such study have been molecular dynamics (MD) [64], although Monte Carlo approaches are also used in lattice-protein models [10, 27]. While high-resolution MD simulations provide a more detailed description about the structural stability of the aggregate, the timescales used are still too short to study the assembly process, thus low-resolution models can be a promising approach for the study of aggregate assembly [67].

Whereas avoidance of aggregation is central to living systems in the case of amyloids, there are also functional aggregates that are not always pathogenic. For instance, the application part of this thesis concerns the oligomerization of membrane protein TatA that assists the translocation of folded proteins across membrane.

1.2 Lattice model and force field

1.2.1 A lattice model with implicit membrane

The two-letter H-P model [35] was used as a starting point to construct our lattice model. It is perhaps the simplest model to represent a protein. We consider a three-dimensional H-P model where a protein is represented as a sequence of non-overlapping beads on a cubic lattice with periodic boundary, and the type of each bead can be either hydrophobic (H) or hydrophilic (P). [1] We say two beads *contact*

[1] A third type of bead, H2, will be added later to model components of the amphipathic helix in our TatA model in Chapter 6.

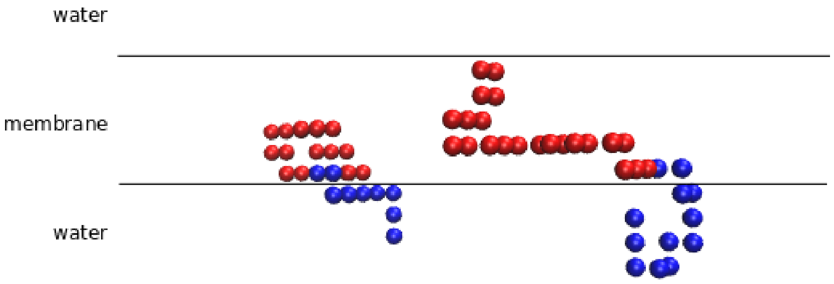


Figure 1.1: Side view of the simulation environment for the basic model. The boundaries of membrane define a hydrophobic region inside. The H-P beads of the two polymers are colored—hydrophobic beads in red and hydrophilic beads in blue.

if they are adjacent on lattice but are not bonded in the chain. H-H contacts are favoured over P-P or H-P contacts so the chain tends to expose hydrophilic beads at the surface and pack hydrophobic ones in the interior. The H-P model thus simulates the effect of protein folding in water, where water is modeled implicitly.

To simulate the membrane environment, we define a hydrophobic region inside the simulation box, as illustrated in Figure 1.1. The membrane is implicit in that phospholipids are not used to model a real membrane, instead, the environment is represented by a one-particle lattice site energy term that is either hydrophobic (membrane) or hydrophilic (water). This is the basic model and will be used in Chapter 3 and Chapter 5.

The real TatA protein has both a transmembrane helix (TMH) and an amphipathic helix (APH). This double-helix feature was encoded in our force field (Section 1.2.2) and can be turned on and off. While the TMH spans the membrane normal, the APH lies in parallel with the membrane surface, so we further define a membrane-water interface on both sides of the membrane, with the bottom interface serving as a compartment of APH (Figure 1.2). This model will be used in Chapter 6 and, except for Section 6.3, all of our models consist of two polymers.

1.2.2 Characterizing the force field

We consider three types of interactions: intra-polymer, inter-polymer and the implicit interactions associated with membrane, water and interface. The total potential energy is the sum of the energies defined by these three types of interactions, i.e. $E = E_{\text{intra}} + E_{\text{inter}} + E_{\text{im}}$. Both intra-polymer and implicit interactions are defined as the sum of terms corresponding to each individual polymer, the inter-polymer

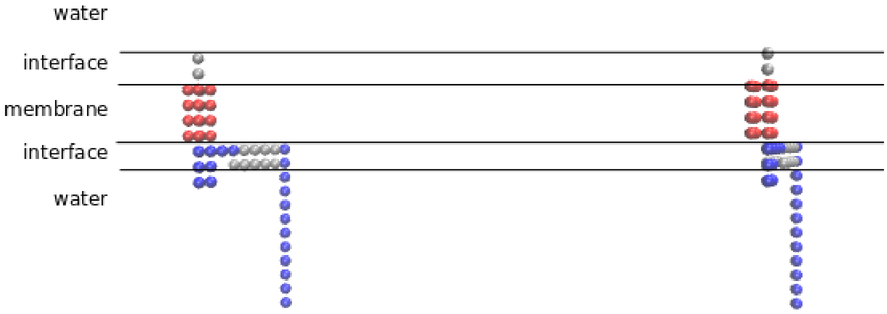


Figure 1.2: Side view of the simulation environment for the TatA model to be used in Chapter 6. The boundaries of different regions are indicated. The beads that form helix are packed in a spiral pattern in a 3D cubic lattice. The transmembrane helix is colored in red and the amphipathic helix is colored in gray in the bottom interface.

interaction is the sum of interactions between all polymer pairs. In other words,

$$E_{\text{intra}} = \sum_k E^k_{\text{intra}}, \qquad E_{\text{im}} = \sum_k E^k_{\text{im}},$$

and

$$E_{\text{inter}} = \sum_{s<t} E^{s,t}_{\text{inter}},$$

where k, s, t indexes polymers.

For intra-polymer interaction, we consider interactions of beads that not only in contact but also some distance apart. Of course, there is a cut-off distance beyond which there will be no interactions. When helices are not modeled and interface is absent, i.e. the basic model, the strength of interaction between a pair of beads is determined by both the type (H or P) and the environment (membrane or water). In water, an H-H contact is more favourable than an H-P or P-P contact; and in membrane, a P-P contact is more favourable than a P-H or H-H contact. When interface is present, i.e. models in Chapter 6, and one or both helices are modeled, we make two additions. We add a hydrogen bond (H-bond) term which will favour specifically numbered beads in the chain to attract each other and form a helix-like arrangement. Second, based on the H-bond condition, we construct a hydrophobic transmembrane helix and an H2-amphipathic helix where H2 is a new type of bead that tends to stay in the interface region and has the same properties as an H bead has in water. The APH lies horizontal with membrane surface.

The implicit interaction of a polymer with its environment is defined such that H beads are favoured in membrane, P beads are favoured in water and H2 beads are favoured in interface.

Finally, the inter-polymer interaction between polymers m and n is defined such that P beads from polymer m attract P beads from polymer n if they are both in membrane, and the same cut-off distance as specified in intra-polymer interaction applies; similarly, H beads from both polymers attract each other in water. The precise definitions of various terms in the force field are listed in Appendix A.

1.3 Move set

The move set consists of pull moves [37] and translation moves. A translation move just shift the entire chain with some random lattice sites, and is needed to study aggregation in our multi-polymer system. Specifically, one of the four directions corresponding to $\pm x$, $\pm y$ is chosen, and the chain is shifted with an amount uniformly chosen between 1 and 10 lattice sites. Pull moves are advanced moves, and they will be discussed in detail in this section.

1.3.1 Pull moves

For the purpose of Monte Carlo simulations, it is desirable that the move set has some nice properties. We say that a move set is *reversible* if for any configuration A and any configuration B obtained from applying a move in the set to A, there is a reverse move to get back to A from B. And, a move set is *complete* if any two configurations can be reached through a sequence of moves in the move set, this guarantees that all configurations have a non-zero probability to be visited. The word *ergodic* may also be used to describe this property. As we will see in Section 2.1, these are essential properties that a move set must possess in order to preserve the detailed balance condition. Note that reversibility and ergodicity are also used to describe Markov chains, to avoid confusion, in this section these are used in the context of a move set.

Pull moves have been widely used as trial moves in lattice polymer Monte Carlo simulations. The move set was shown to be reversible and complete[37], although a later paper pointed out that in general reversibility is violated [26]. We describe pull moves in a two-dimensional grid, realize that a three-dimensional generalization is straightforward.

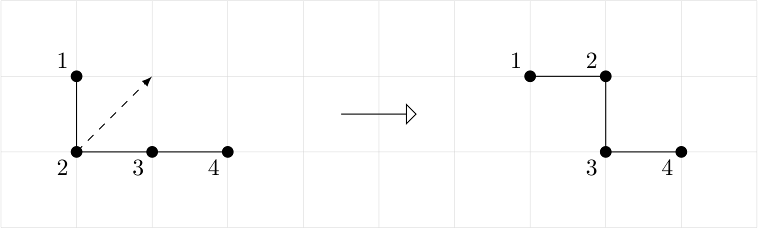


Figure 1.3: Pull move terminates after displacement of one bead. Bead 2 is selected, an empty lattice site adjacent to bead 3 and diagonally adjacent to bead 2 is chosen (pointed by the dashed arrow) to be the new location of bead 2. Since the resulting configuration is valid, pull move stops.

Generally speaking, a pull move starts by creating a square in the chain and successively pull the beads along until an existing square is undone or until the terminal bead is reached when there is no such square along the path. The chain can be pulled in either directions. If there are N beads in the chain and we have numbered them from 1 to N, then pulling *upwards* means that bead N is the last bead to move if needed to; and pulling *downwards* means that bead 1 is the last bead to move if needed to. We consider pull moves in the downward direction, pull moves in the upward direction can be implemented analogously. A lattice site is *empty* if there is no bead occupying it. *Overlaps* are not acceptable, that is, a lattice site can be occupied by one and only one bead. We next define a *valid* configuration to be a non-overlapped configuration such that beads that are adjacent in the chain are adjacent in the grid.

Having introduced these terms, we now describe details of pull move. Suppose some non-terminal bead i has been selected from a valid configuration in a two-dimensional grid, an empty lattice site diagonally adjacent to bead i and adjacent to bead $i+1$ is chosen to place bead i. If after this move the chain is already in a valid configuration, then the pull move terminates (Figure 1.3); otherwise the lattice site corresponding to the forth corner of the square defined by old position of bead i, bead $i+1$ and new position of bead i must be empty for a pull move to continue, in which case we place bead $i-1$ in this empty site and successively pull the beads with lower indices two lattice positions ahead, that is, bead $i-2$ is moved to the previous position of bead i and so on, until a valid configuration is reached (Figure 1.4). If terminal bead N has been chosen, then beads N and $N-1$ are placed at any two free locations connecting bead N, the rest of the beads are then moved two positions ahead until a valid configuration is reached (Figure 1.5).

Because pull move stops early whenever possible, we see that it is local in that the number of beads to be moved is minimal, this typically results in higher

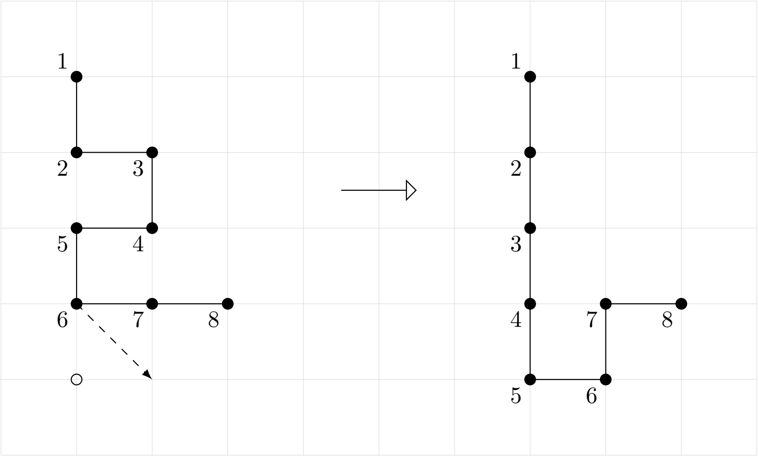


Figure 1.4: Pull move terminates after displacement of more than one bead. Bead 6 is selected, an empty lattice site adjacent to bead 7 and diagonally adjacent to bead 6 is chosen (pointed by the dashed arrow) to be the new location of bead 6. Since the forth corner of the square is empty (indicated by the empty circle in the figure), bead 5 is moved to this location, and then the rest of the beads upstream of the chain are successively moved two positions ahead, until a valid configuration is reached. In this case, bead 2 does not move because a valid configuration is formed after movement of bead 3.

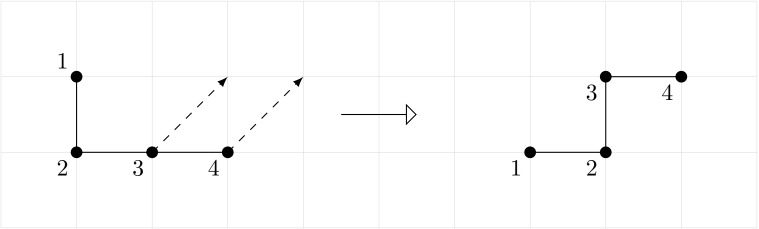


Figure 1.5: An example of pull move for terminal bead. Two free locations (pointed by the dashed arrows) are chosen to place beads 4 and 3. Beads with lower indices are pulled along the chain two lattice positions ahead.

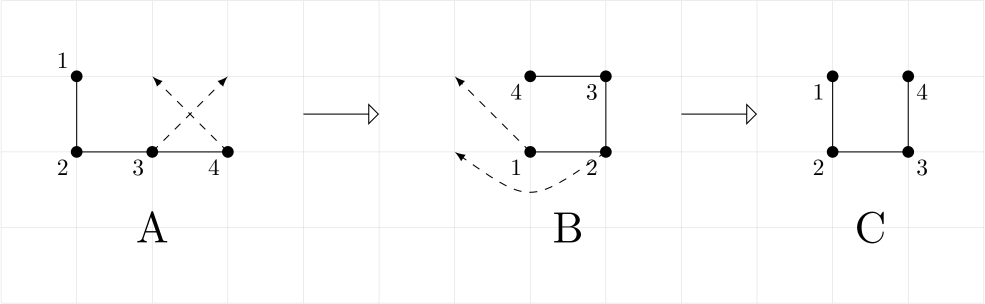


Figure 1.6: An example of irreversible pull move for terminal bead. Three configurations are labelled as A, B and C from left to right. For A, a particular end move results in B. To reverse this move, we must pull in the other direction, but the reverse move (shown by the dashed arrows in B) would stop early because bead 4 does not need to move.

Monte Carlo acceptance probability compared to moves which displace many beads in a configuration.

It has been shown, however, that a subset of pull moves are in fact not reversible, and those irreversible moves are precisely the end moves that result in a hook at the end of the chain [26]. An example is shown in Figure 1.6, here, the initial configuration is the same as in Figure 1.5 but a different location is chosen to place the terminal bead, this move is irreversible due to the local nature of pull moves. To fix this issue, we can simply exclude those irreversible moves from the move set without breaking ergodicity [26].

In the following we discuss the implementation details of pull moves for lattice Monte Carlo simulations. We notice that these details are often ignored or omitted in many literatures concerning MC studies of lattice polymers. However, an improper implementation can lead to simulation bias and inaccurate predictions of the system. This is particularly relevant when the goal is to obtain a full equilibrium sampling of the system rather than finding the ground state configuration.

1.3.2 Implementation details

In most Monte Carlo simulations in statistical physics, we often require that detailed balance be satisfied. Many people in the field [44, 58, 66] assume this is the case by adopting the simplified Metropolis acceptance criterion. However, in doing so, we need to ensure that the forward move and reverse move are equally likely. As we shall see, this can easily be violated in the case of pull moves, and so we need to be careful about their implementation.

Notice first that the only source of uncertainty in pull moves comes from the start of the move. For pull move in the downward direction, if a non-terminal bead i is selected as “pull bead”, we must choose a lattice site that is diagonally adjacent to i and adjacent to $i+1$; and if a terminal bead N is selected, then we must choose two lattice sites to place beads N and $N-1$. By definition, configurations with overlapped beads are not acceptable, so they tend to be excluded in the implementation of pull moves, that is, only valid configurations are proposed. We show how this easily leads to violation of equal probability assumption on trial moves.

Consider again the two configurations in Figure 1.4, and denote the left and right configurations by L and R, respectively. If only valid configurations are proposed, then there is only one choice to go for, as indicated in the figure, since the other choice would cause an overlap with bead 4. Hence, assuming that the pull bead is selected with equal probability, the chance of getting from L to R is 1/8 (1/8 × 1). On the other hand, to get from R to L, we need to choose a site diagonally adjacent to bead 3 and adjacent to bead 2. Since there are two empty sites that we can choose, the chance of getting from R to L is 1/16 (1/8 × 1/2).

The above discussion suggests that in order to preserve detailed balance (Section 2.1), we must count overlapped configurations as trial moves, even though they are destined to rejection. With this observation in mind, we propose the following strategy for implementation of pull moves in lattice Monte Carlo simulations. Let C and C' denote respectively the configurations before and after pull move, and consider pulling downwards and bead i has been selected as pull bead.

```
function PULLMOVE($C$)
    if $i == 1$ then PULLMOVE(C)
    else if $i == N$ then
        do terminal move and get $C'$
        if $C'$ is overlapped then
            return $C$
        end if
        if beads $N$ and $N-3$ are adjacent in $C'$ then PULLMOVE(C)
        end if
    else
        select a neighbour $j$ of $i+1$
        if $j$ is empty and diagonally adjacent to $i$ then
            do pull move and get $C'$
        else
            return $C$
        end if
```

```
17:     end if
18:     if $C'$ is overlapped then
19:         return $C$
20:     else
21:         return $C'$
22:     end if
23: end function
```

In the above procedure, line 8 specifies the condition for a hook that needs to be checked in order to exclude irreversible pull moves. Note also that before returning the new configuration C', we check if it is overlapped, and the old configuration is returned if that is the case. After calling PULLMOVE, we can then check if C has been modified. If yes we proceed to calculate the Metropolis acceptance probability, otherwise we reject the move. Instead of returning C, we could return any reference configuration whose potential energy is infinity, and we chose the old configuration just for simplicity.

Chapter 2

Simulation methods—A review

In this chapter, we review some Monte Carlo methods commonly used in molecular modeling, and that are closely related to our study. In molecular simulation, Monte Carlo methods will almost always be Markov Chain Monte Carlo (MCMC) as drawing samples directly from the underlying distribution is close to impossible. The basic idea of MCMC is to simulate a Markov chain whose stationary distribution is the target distribution. Thus, if we run the simulation long enough, the samples generated can be regarded as from the target distribution and statistical inferences can then be drawn. Of course, the samples are correlated and cannot beat independent samples in terms of statistical efficiency. This implies that one often has to collect many more samples than one would for independent sampling in order to achieve similar statistical error. Under mild regularity conditions, the ergodic theorem guarantees that the sample average still converges to its expected value as sample size tends to infinity. The theorem plays the same role as the law of large numbers does for independent and identically distributed samples.

As it is well known by many researchers that simple Metropolis Monte Carlo can get trapped indefinitely in a local energy basin, we will also introduce the concept of weighted MCMC which attempts to tackle the local trap problem, but which also involves more sophisticated transitions rules.

2.1 The Metropolis Algorithm

Throughout, we let $\pi(\mathbf{x})$ denote the target distribution, where $\mathbf{x}$ contains the Cartesian coordinates of the system which is 3N dimensional for a total of N atoms, although internal coordinates can also be used, such as bond lengths, bond angles, dihedral angles and so on. In the canonical ensemble, the distribution that we want to sample from is proportional to the Boltzmann factor, that is

$$\pi(\mathbf{x}) \propto \exp(-U(\mathbf{x})/k_B T)\,, \tag{2.1}$$

where $U(\mathbf{x})$ is the potential energy function, T is the temperature and k_B is the Boltzmann constant. For the ease of numerical modeling, we assume that we are working under a unit system such that $k_B = 1$.

Constructing a Markov chain such that it has an invariant distribution π is actually not as complicated as one might think. Metropolis et al. [48] proposed a simple yet powerful construction that has been cornerstone of almost all MCMC methods since developed. The Metropolis algorithm iterates the following two steps:

- If the current configuration is $\mathbf{x}$, propose a new configuration $\mathbf{x}'$ according to some unbiased trial move.
- Accept $\mathbf{x}'$ with probability $\min\{1, \frac{\pi(\mathbf{x}')}{\pi(\mathbf{x})}\}$.

Later, Hastings [28] generalized the algorithm to asymmetric proposal functions. That is, if the trial move is biased, one has to correct for it by accepting the new configuration with probability

$$\alpha(\mathbf{x}, \mathbf{x}') = \min\{1, \frac{\pi(\mathbf{x}')T(\mathbf{x}', \mathbf{x})}{\pi(\mathbf{x})T(\mathbf{x}, \mathbf{x}')}\}$$

Here, $T(\mathbf{x}, \mathbf{x}')$ is the proposal function. In the Metropolis algorithm, we simply have $T(\mathbf{x}, \mathbf{x}') = T(\mathbf{x}', \mathbf{x})$. Note that the unknown normalizing constant of π cancels out in the acceptance probability. To verify that π is indeed the invariant distribution, we are required to show that

$$\int \pi(\mathbf{x})A(\mathbf{x}, \mathbf{y})d\mathbf{x} = \pi(\mathbf{y}), \tag{2.2}$$

where $A(\mathbf{x}, \mathbf{y})$ is the actual transition probability and is equal to the product of the proposal and the acceptance probability; that is

$$A(\mathbf{x}, \mathbf{y}) = T(\mathbf{x}, \mathbf{y})\alpha(\mathbf{x}, \mathbf{y})\,.$$

Equation (2.2) is sometimes referred to as general balance. A sufficient condition for it is the so called detailed balance:

$$\pi(\mathbf{x})A(\mathbf{x}, \mathbf{y}) = \pi(\mathbf{y})A(\mathbf{y}, \mathbf{x}). \tag{2.3}$$

The detailed balance equation (2.3) simply states that at equilibrium, the probability of observing transition $\mathbf{x} \to \mathbf{y}$ is the same as observing them in reverse order. Some authors also refer to this condition as microscopic reversibility. It is straightforward to verify that (2.3) holds for the Metropolis algorithm, which implies that (2.2) holds. Hence, the Metropolis algorithm preserves π as an invariant distribution. Once the chain has reached the equilibrium regime, the dependent samples generated can be treated as draws from the canonical distribution π.

For the multi-polymer system we are interested in (see Section 1.2), the Metropolis algorithm may be implemented as Algorithm 1 below.

Algorithm 1 The Metropolis algorithm for the multi-polymer system

Let the current state be $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)})$, at step $t+1$:

- Randomly select a polymer i from $\{1, \ldots, n\}$ and conduct a pull move or translation move to x_i while keeping other polymers the same as previous step to obtain a new configuration $\mathbf{x}' = (x_i', x_{-i}^{(t)})$.
- Accept $\mathbf{x}'$, that is, set $\mathbf{x}^{(t+1)} = \mathbf{x}'$, with probability $\min\{1, \exp(-\Delta U/k_B T)\}$, where $\Delta U = U(\mathbf{x}') - U(\mathbf{x})$. Otherwise set $\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)}$.

Recall that in a standard Gibbs sampler, one achieves the task of drawing samples from a multivariate distribution by iteratively sampling from its conditional distributions. To make the Gibbs sampler reversible, a component of $\mathbf{x}$ is picked randomly and an update is drawn from its distribution conditional upon all remaining components. This is referred to as the random scan Gibbs sampler [42]. Even though its origin is quite different from the Metropolis algorithm, the Gibbs sampler can be viewed as a special MCMC algorithm in that the proposal functions are just the conditional densities $\pi(\mathbf{x}_i|\mathbf{x}_{-i})$. One could imagine that with this choice it would be much less noisy than using some arbitrary proposal function $T(\mathbf{x}, \mathbf{y})$. Note that there is no accept/reject step in a Gibbs sampler. In fact it can be easily shown that the acceptance probability is always one. In practice, if sampling from the conditional distribution of some components is impossible or difficult to realize, one could always replace it by a Metropolis step.

In this respect, the algorithm for the study of the multi-polymer system presented above may be viewed as a random scan Metropolized Gibbs sampler. The name is deceptive though since no Gibbs updates are applied here, as the conditionals are simply not available.

2.2 Parallel Tempering

The problem with Metropolis sampling is that it can easily get trapped in local energy minima, because of the presence of many energy barriers in the free energy landscape. Many methods have been proposed to cope with this difficulty. In this section we shall review a method of great importance to our study—the parallel tempering (PT) method, but first the simulated tempering (ST) method will be reviewed as it is closely related to PT.

To enhance sampling, Marinari and Parisi [45] proposed the simulated tempering algorithm, in which a temperature index k is augmented to the state space $\mathcal{X}$ so that the distribution is defined on $\mathcal{X} \times I$, where $I = \{1, \ldots, K\}$ for a total of K temperatures $T_1 < \ldots < T_K$. One then samples from a mixed canonical ensemble

$$\pi(\mathbf{x}, k) \propto \exp(-U(\mathbf{x})/T_k). \tag{2.4}$$

The idea is that by heating up the distribution, the sampler is able to explore much wider configuration space and escape from local energy basin. Sampling of (2.4) can be implemented by first fixing k and perform a Metropolis update on $\mathbf{x}$, and then fixing $\mathbf{x}$ and perform an update on k, which could be a random walk on all temperature levels. When making a temperature transition, the acceptance probability has to be governed by the Metropolis acceptance criterion:

$$P(k \to k') = \min\{1, \frac{\pi(\mathbf{x}, k')}{\pi(\mathbf{x}, k)}\} = \min\{1, \frac{Z_k}{Z_{k'}} \exp(-U(\mathbf{x})(\frac{1}{T_{k'}} - \frac{1}{T_k}))\},$$

where $Z_k = \int \exp(-U(\mathbf{x})/T_k) d\mathbf{x}$, the partition function at temperature T_k. The important thing to note here is that because $Z_k/Z_{k'}$ are unknown, one needs to estimate these constants beforehand, often through some pilot studies, in order to actually implement ST.

The need to specify these constants can be very inconvenient in many applications, and one advantage of PT is that this step is completely omitted. While many researchers attribute the method to Hukushima and Nemoto [30] and refer to it also as the exchange Monte Carlo method, the idea was actually proposed earlier by Geyer [22] in the context of statistical inference. Briefly, PT works by running several Markov chains in parallel, and allowing swaps, either at each iteration or at random, between states of neighbouring temperatures to speed up mixing of chains.

Formally, the state space for PT now becomes the joint product space $\prod_{k=1}^{K} \mathcal{X}_k$, which contains K configurations as replicas with K being the number of temperatures used in the simulation. One then samples from a joint distribution that is the product of each marginal distribution:

$$\pi(\mathbf{x}_1, \ldots, \mathbf{x}_K) = \prod_{k=1}^{K} \pi_k(\mathbf{x}_k),$$

where each π_k has Boltzmann weight at temperature T_k.

Now, suppose the current state is $\mathbf{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_K)$, and an attempt to swap some neighbouring states $\mathbf{x}_k$, $\mathbf{x}_{k-1}$ is proposed so that $\mathbf{X}' = (\mathbf{x}_{1:k-2}, \mathbf{x}_k, \mathbf{x}_{k-1}, \mathbf{x}_{k+1:K})$, then the acceptance probability becomes:

$$P(\mathbf{X} \to \mathbf{X}') = \min\{1, \frac{\pi_{k-1}(\mathbf{x}_k)\pi_k(\mathbf{x}_{k-1})}{\pi_{k-1}(\mathbf{x}_{k-1})\pi_k(\mathbf{x}_k)}\}. \qquad (2.5)$$

Note that the normalizing constants cancel out since they both equal $\prod_{k=1}^{K} Z_k$.

It is then easy to write down the PT algorithm for our multi-polymer system (Algorithm 2, page 15). The initial state $(\mathbf{x}_1^{(0)}, \ldots, \mathbf{x}_K^{(0)})$ can be simply a set of identical replicas of a single configuration.

Algorithm 2 Parallel tempering algorithm for the multi-polymer system

Let the current state be $(\mathbf{x}_1^{(t)}, \ldots, \mathbf{x}_K^{(t)})$, at step $t+1$,

- With probability α_0, do a parallel step. Update each $\mathbf{x}_k^{(t)}$, $k = 1, \ldots, K$ according to its Metropolis step, that is, obtain $(\mathbf{x}_1^{(t+1)}, \ldots, \mathbf{x}_K^{(t+1)})$ from Algorithm 1 on page 13.
- Otherwise, attempt a swap. Randomly choose a neighbouring pair k, $k-1$ from $\{1, \ldots, K\}$ and set $(\mathbf{x}_1^{(t+1)}, \ldots, \mathbf{x}_K^{(t+1)}) = (\mathbf{x}_{1:k-2}^{(t)}, \mathbf{x}_k^{(t)}, \mathbf{x}_{k-1}^{(t)}, \mathbf{x}_{k+1:K}^{(t)})$ with probability $\min\{1, \exp\left[(U(\mathbf{x}_k) - U(\mathbf{x}_{k-1}))(\frac{1}{T_k} - \frac{1}{T_{k-1}})\right]\}$.

Even though PT is a powerful algorithm to simulate bead-polymer systems, in practical use one still needs to be concerned with the spacing between temperatures, because an improper choice can greatly affect the performance of the algorithm. Clearly, as can be seen from Algorithm 2, the probability of making a swap depends on both temperature and energy difference. On one hand, the difference between adjacent temperatures should not be too large, otherwise the sampler will suffer from low exchange probabilities due to high free energy barriers; on the other hand, even if the temperature difference is small, the exchange probability could still be low if there is a possible phase transition that results in a large energy difference. We may try inserting more temperature levels, but we often do not know at what temperature the underlying phase change occurs in the first place! Furthermore, using many temperatures will decrease computational efficiency as the time needed to traverse through all temperatures increases. In fact, the expected waiting time of a round trip increases roughly as the order of K^2 for a temperature ladder of size K [39].

2.3 Dynamic weighting Monte Carlo

We know from the end of Section 2.2 that the traversal time increases with the number of temperature levels, hence few temperature levels should be used to reduce the computational cost. However, the waiting time for a swap increases exponentially with temperature difference, suggesting the use of many temperatures to ensure frequent swaps. This "waiting time dilemma" motivated us to consider a new type of MC algorithm—the dynamic weighting Monte Carlo (DWMC), that is fundamental different from regular MCMC algorithms. It was first introduced in [65], with a theoretical study of its properties in [41].

Here we review the basic idea of DWMC. To tackle the local trap problem commonly observed in Metropolis algorithm, Wong and Liang [65] proposed to run a "weighted Markov chain" by augmenting the state space with a weight variable w and allow large transitions that are usually rejected in Metropolis-like algorithms. Here w can be viewed as an importance weight used to correct for the bias introduced by such transitions. They designed a new type of transition rule and proposed some special moves that preserve this new rule in order to justify the use of the weighted average $\sum_{i=1}^{N} w^{(i)} A(\mathbf{x}^{(i)}) / \sum_{i=1}^{N} w^{(i)}$ to estimate the equilibrium expectation of some observable $A(\mathbf{x})$. By applying this new Monte Carlo method to some optimization problems such as the traveling salesman problem and neural network training, they were able to obtain better results compared to other methods.

Let us take a closer look at the logic behind DWMC. First, the concept of correctly weighted samples is defined as:

Correctly weighted samples A set of weighted samples $\{\mathbf{x}^{(i)}, w^{(i)}\}_{i=1}^{N}$ is called correctly weighted with respect to π if the joint probability density $f(\mathbf{x}, w)$ satisfies

$$\int_0^{+\infty} w\, f(\mathbf{x}, w)\, dw \;\propto\; \pi(\mathbf{x}).$$

Now, suppose we have *independent and identically distributed* (iid) samples $\{\mathbf{x}^{(i)}, w^{(i)}\}_{i=1}^{N}$ that are correctly weighted with respect to π, then it is easily seen that $\sum_{i=1}^{N} w^{(i)} A(\mathbf{x}^{(i)}) / \sum_{i=1}^{N} w^{(i)}$ is a consistent estimator for $\mathrm{E}_{\pi} A(\mathbf{x})$. In fact, by the strong law of large numbers,

$$\frac{1}{N} \sum_{i=1}^{N} w^{(i)} A(\mathbf{x}^{(i)}) \to \mathrm{E}_f[wA(\mathbf{x})] \quad a.s. \text{ as } N \to \infty.$$

Let Γ denote the configuration space, we have

$$\begin{aligned} \mathrm{E}_f[wA(\mathbf{x})] = \int_0^{+\infty} \int_\Gamma f(\mathbf{x}, w)\, w\, A(\mathbf{x})\, d\mathbf{x} dw &= \int_\Gamma A(\mathbf{x}) \int_0^{+\infty} w\, f(\mathbf{x}, w) dw d\mathbf{x} \\ &= c \int \pi(\mathbf{x})\, A(\mathbf{x})\, d\mathbf{x}, \end{aligned}$$

where c is some proportionality constant and the last equality holds because the samples are correctly weighted. Similarly,

$$\frac{1}{N} \sum_{i=1}^{N} w^{(i)} \to \mathrm{E}_f[w] = \int_\Gamma \int_0^{+\infty} w\, f(\mathbf{x}, w) dw d\mathbf{x} = c \int_\Gamma \pi(\mathbf{x}) d\mathbf{x}$$

Combining these equations we obtain

$$\frac{\sum_{i=1}^{N} w^{(i)} A(\mathbf{x}^{(i)})}{\sum_{i=1}^{N} w^{(i)}} \to \mathrm{E}_\pi A(\mathbf{x}) \quad \text{as } N \to \infty. \tag{2.6}$$

Of course, in many cases it is unrealistic to generate iid samples, and we are already familiar with MCMC techniques which achieve the task of generating (correlated) samples by evolving a Markov chain. In that spirit, in DWMC a weighted Markov chain is simulated and one hopes that, after some equilibration period, the samples generated are correctly weighted with respect to the target distribution π. So, just like in standard MCMC where the invariant distribution is preserved in each iteration, in DWMC one seeks to maintain the correct weightedness of the sample in each iteration. This new invariance principle is referred to as *invariance with respect to importance weighting* (IWIW) in the literature. Some special moves that satisfy IWIW either exactly or approximately were proposed [65]. For example, the *Q-type* move operates as in Algorithm 3, where a is some constant greater than one.

From the *Q-type* move, we can see that when rejection occurs, the associate weight increases, enabling the chain to escape from the local mode.

Although this framework of DWMC seems appealing, there are two important factors we are yet to address. First, we want to make sure that the weight

Algorithm 3 *Q-type* Move

Let the current state be $(X^{(t)},\, W^{(t)}) = (x,\, w)$.

- Propose y according to the proposal $T(x, y)$ and compute the Metropolis ratio

$$r(x, y) = \frac{\pi(y)T(y, x)}{\pi(x)T(x, y)}$$

- Draw $U \sim \mathrm{Unif}(0, 1)$ and set

$$(X^{(t+1)},\, W^{(t+1)}) = \begin{cases} (y, \max\{1, wr(x, y)\}) & \text{if } U \leq \min\{1, wr(x, y)\} \\ (x, aw) & \text{otherwise} \end{cases} \tag{2.7}$$

process so defined is stable; and second, that (2.6) holds in some sense—if this is the case, then we would have theoretical support for the convergence of DWMC, as we already have in standard MCMC through the ergodic theorem. It turns out that these two aspects of DWMC become the main issue of the method and cause difficulties not only in theoretical investigation, but also in practical use, as we shall demonstrate in Chapter 3. To handle the first issue, it was shown in [41] that, by a suitable modification of the weight process, $\{(X^{(t)}, \log W^{(t)})\}_t$ induced by the *Q-type* move has a unique equilibrium distribution. However, the weight process $\{W^{(t)}\}_t$ was shown to have infinite mean. Because of this, it is not at all trivial to establish the (weak) convergence of (2.6). To deal with this second issue, Liu, Liang, and Wong then proposed the *stratified truncation* procedure to process the raw weights generated from DWMC, and provided a theoretical justification for this procedure. As the name suggests, first the samples are stratified, then the weights within each stratum are truncated. More precisely, suppose we wish to estimate the equilibrium expectation of some observable $A(\mathbf{x})$ and have collected the weighted samples $\{A(\mathbf{x}^{(i)}),\, w^{(i)}\}_{i=1}^{N}$, then,

- the samples are stratified according to $A(\mathbf{x})$ such that within each stratum the range of $A(\mathbf{x})$ is not too large and the sizes of strata are comparable;
- the weights within each stratum are trimmed down to the $(100-k)$th percentile ($k = 1$ or 2), that is, let

$$w_{\mathrm{tm}}^{(i)} = \min\{w^{(i)},\, w^{*}\}, \text{ for each } w^{(i)} \in S,$$

  where w^{*} is the $(100-k)$th percentile of the weights in stratum S; and finally,

- the estimate for $\mathrm{E}_\pi A(\mathbf{x})$ is obtained by computing $\sum_{i=1}^N w_{\mathrm{tm}}^{(i)} A(\mathbf{x}^{(i)}) / \sum_{i=1}^N w_{\mathrm{tm}}^{(i)}$.

Essentially, weights that belong to the largest magnitude portion (say, 2%) in each stratum are replaced by the 98th percentile of the weights in the respective stratum. There are several implementation details to consider given the procedure outlined above, such as the number of strata to use and how to choose a suitable size of each stratum. We will explore how this procedure can influence the result of estimation in Chapter 3.

Further developments [38] within the DWMC framework include a scheme which augments the state space of Algorithm 3 to a population of the weighted pair $(\mathbf{x}, w)$, and in each iteration, after the dynamic weighting moves have been applied to each individual pair, $(\mathbf{x}, w)$, in the population, it uses a population control procedure that essentially replicates those individuals with large weights and discards those with small weights, and the individuals are then properly reweighted to avoid introducing bias. Incorporation of the population control step ensures finite mean of the weights and is thus crucial to this scheme. However, we note that although convergence of this scheme has been justified in [38], the theorem proved therein requires the population size at each iteration to tend to infinity, so in practice this population dynamic weighting scheme needs to maintain a large population for it to be accurate. This can be infeasible for our multi-polymer system because each individual in the population would be a joint configuration of all of the polymers in the system.

As another note, since IWIW is trivially satisfied in Metropolis-type moves, we see that standard MCMC can be viewed as a special case of DWMC with the (irrelevant) weight variable unchanged at each iteration. This observation leads to a generalization of DWMC that allows us to mix the usual Metropolis moves with DW moves. An application of such is presented in the next chapter where we incorporate the dynamic weighting idea into parallel tempering simulations.

2.4 Multicanonical simulation

By running multiple Markov chains in parallel with tempered distributions, a parallel tempering simulation is able to move across free energy barriers and an enhanced sampling can often be achieved at the lowest temperature. Until now we have been focusing on the temperature spacing problem, which is related to the actual implementation of the algorithm; another problem intrinsic to the method is the sampling of rare configurations, such as configurations that define the transition states in a potential energy surface. For example, problems can arise in systems involving phase transitions, where we typically observe a low exchange probability between low and high energy states, even with small temperature spacing. Ultimately, this is due to the Boltzmann weight in the canonical ensemble, which is the ensemble we have been using. The multicanonical simulation (MUCA) [6] takes a different perspective by sampling from a modified ensemble in which the energy is approximately uniformly distributed; that is, the multicanonical density is proportional to the inverse of the density of states:

$$\pi_{\text{mu}}(\mathbf{x}) \propto \frac{1}{\Omega(U(\mathbf{x}))} \,. \tag{2.8}$$

Because $\Omega(U)$ is not known *a priori*, the actual simulation only samples from an approximation $\hat{\pi}_{\text{mu}}$ of (2.8). Had it been known there would be no need to do simulations because we can compute all thermodynamics from the density of states. Thus, the idea of MUCA is to iteratively construct a sequence of approximations $\hat{\pi}^n_{\text{mu}}$ $(n = 1, 2, \ldots)$:

$$\hat{\pi}^n_{\text{mu}} \propto (\hat{\Omega}^n(U(\mathbf{x})))^{-1}, \ n = 1, 2, \ldots$$

such that $\hat{\pi}^n_{\text{mu}} \approx \pi_{\text{mu}}$ when n is large.

In practice, this is usually done by running many small-scale simulations where each simulation yields an approximation $\hat{\pi}^n_{\text{mu}}$, until one is satisfied with some Nth simulation which produces an approximately flat energy histogram. By approximately flat we mean the sampler is able to visit all energy regions relatively frequently, and a difference of up to a factor of ten is often deemed to be acceptable [5]. One can then run a longer simulation with the weights $\hat{\pi}^N_{\text{mu}}$ and obtain estimates with respect to the canonical ensemble through importance reweighting.

The key to successful implementation of MUCA therefore depends on the recursion rule used to update $\hat{\pi}_{\text{mu}}$. Initially, $\hat{\pi}^1_{\text{mu}}$ is set to 1, indicating there is no prior information about the system and the sampling of every configuration is equally likely. A simple update rule proceeds by giving to each $\hat{\Omega}^n(U_m)$, where $\{U_m\}_{m=1}^M$ is a discretization of energy U, a weight proportional to the observed energy histogram in bin m:

$$\hat{\Omega}^{n+1}(U_m) \propto \hat{\Omega}^n(U_m) H^n_m, \tag{2.9}$$

where H^n_m is the observed count in bin m from the nth simulation and the proportionality constant is irrelevant. We may write

$$\hat{\Omega}^{n+1}(U_m) = \hat{\Omega}^n(U_m)\frac{H_m^n}{H_{\text{exp}}}, \tag{2.10}$$

where H_{exp} is the expected count per bin and is equal to the total number of samples divided by the number of bins. The logic behind (2.10) is clear: if $\frac{H_m^n}{H_{\text{exp}}} > 1$ then bin m is oversampled, so in order to drive the sampler towards constant behaviour we increase $\hat{\Omega}^n(U_m)$, so that bin m is likely to be sampled less in the next round, and vice versa if $\frac{H_m^n}{H_{\text{exp}}} < 1$.

There are several problems with this simple recursion. First, there is always statistical noise associated with H_m^n, and this noise is erroneously treated as a correction factor for the density of states. An extreme situation is when we feed into our simulation the exact density of states: in this case the new update is doomed to be worse because all that we added is statistical noise. Another drawback is that each update is based only on the most recent H_m^n and historical data from previous simulations are ignored. Also, if H_m^n is zero then the multicanonical density is undefined.

A modified recursion which takes into account these problems was proposed by Berg [4]. In its original formulation, the approximation to the target density π_{mu} was written under a new parameterization:

$$\hat{\pi}_{\text{mu}} \propto e^{-S(U)} = e^{-b(U)U+a(U)}, \tag{2.11}$$

where $S(U)$ is the microcanonical entropy, $b(U)$ is the microcanonical temperature, given by the derivative of S with respect to U, and $a(U)$ is the fugacity. The weight to be used at the $(n+1)$th simulation follows once $b^{n+1}(U)$ and $a^{n+1}(U)$ have been determined from the nth simulation:

$$\hat{\pi}_{\text{mu}}^{n+1} \propto e^{-b^{n+1}(U)U+a^{n+1}(U)}. \tag{2.12}$$

Although this may seem complicated by introducing additional parameters, note that only one of them, say $b(U)$, is a "real" parameter because $a(U)$ follows from (2.11) and the fact that $b = \frac{\partial S}{\partial U}$.

Therefore, the modified recursion only involves the determination of $b^{n+1}(U)$, and this is done in a way that historical knowledge about the parameter is properly incorporated. Essentially, the update b^{n+1} not only uses data from the nth simulation, but also combines b^n which encapsulates the history of the previous $n-1$ simulations. A weight proportional to the inverse of the variance of b^n is used as a guide to combine the most recent and historical simulations.

This recursion scheme addresses most problems typical of the simple recursion (2.9), but does not eliminate them. For example, statistical noise is still present since the estimate of the variance is based on finite and often very short simulations. In his paper, Berg used around 9000 simulations with 1000 MC steps each to study a 10-state Potts model and claimed that using frequent iterations[1] was capable of increasing the stability of the result. However, because short simulations generally yield larger statistical uncertainties than longer simulations, it is unclear whether we should use more iterations with fewer steps per simulation or fewer iterations with more steps per simulation, given the same amount of CPU time. Fortunately, with the estimates of the density of states derived from our new method, this multicanonical recursion is often not necessary.

Apart from the actual recursion step, another notable problem with MUCA is that it often requires human input to guide the simulation. This is related to the fact that the weights stay put during an iteration. It is only after one iteration finishes that the weights get updated. Since a MUCA simulation starts by assigning to each configuration an equal weight, it may not be able to visit low energy region of the system under an affordable time, and hence proper guesses of $\hat{\Omega}^n(U)$ near the ground state are often needed in the course of the simulation.

As mentioned in the beginning of this section, the method of tempered distributions such as ST and PT are often effective in moving across energy barriers and thus exploring configuration space more rapidly. This is one of the motivations for combining the strength of PT and MUCA, and how this can be done efficiently will be discussed in Chapter 5.

2.5 Other related methods

2.5.1 The Wang-Landau algorithm

Clearly, many other Monte Carlo methods exists in addition to those mentioned above. One method, closely related to the multicanonical approach, is the Wang-Landau algorithm [62]. It can be viewed as an adaptive MCMC algorithm. Configurations are still sampled with weights proportional to the inverse of the density of states; however the density of states is updated on the fly according to some modification factor $f > 1$, whose purpose is to help produce a flat energy histogram in a relatively short time. The update has the form $\Omega(U) \leftarrow f\,\Omega(U)$ once an energy level has been visited, implying that energies that have been seen will be less likely

[1] Here, and only in this section, an iteration refers to one of such many simulations.

to be seen again, thus enabling a quick exploration of all energy levels. One then iterates the process by systematically decreasing f (e.g. $f \leftarrow \sqrt{f}$) until it is essentially identical to one, at which point the Wang-Landau sampler effectively reduces to a MUCA simulation without recursion step. In general, in order to generate a flat histogram, more MC steps are needed as the modification factor becomes close to one. The resulting density of states will be a good approximation to the true value, and can thus be used to calculate thermodynamic quantities. An important distinction from the multicanonical approach is that in the Wang-Landau algorithm, the density of states is refined to a much higher accuracy, so that it can be used to estimate quantities such as free energy and entropy which are not directly accessible from conventional Monte Carlo simulations. This implies that the condition to stop the simulation is more stringent than that in MUCA, as the price for obtaining an accurate estimate of the density of states. Since the Wang-Landau sampler again starts with the initial weights being equal to one, as is the case in MUCA simulation, it is possible to adopt our method in Chapter 5 to reduce the number of Monte Carlo steps needed to generate flat histograms.

2.5.2 Transition Matrix Monte Carlo

Apart from the need to specify a criterion for the "flatness" of the histogram and to choose a suitable schedule for decreasing the modification factor, the Wang-Landau algorithm has the deficiency that it eventually saturates, meaning that further iterations do not improve the results once a limiting accuracy has been reached [69]. Due to this limitation, attempts have been made to incorporate ideas from transition matrix Monte Carlo, see for example [55] and [43].

In transition matrix Monte Carlo [19, 20, 63], one updates instead of the density of states the transition probabilities between macrostates. Here energy is our macrostate but it can also be other thermodynamic variables depending on the ensemble. Starting from the detailed balance equation (2.3), we define the transition probability to energy U', given that the current energy is U:

$$A(U, U') = \frac{1}{\Omega(U)} \sum_{U(\mathbf{x})=U} \sum_{U(\mathbf{x}')=U'} A(\mathbf{x}, \mathbf{x}'),$$

i.e. $A(U, U')$ is the microcanonical average of $\sum_{U(\mathbf{x}')=U'} A(\mathbf{x}, \mathbf{x}')$ over configuration $\mathbf{x}$ with energy U, where the sum is taken over all configurations, or microstates, with energy U'. With this definition and the familiar detailed balance equation with respect to microstate, we recover the detail balance equation with respect to the macrostate energy:

$$p(U)A(U,U') = p(U')A(U',U). \tag{2.13}$$

In the infinite-temperature case and with $p(U)$ given by the Boltzmann weight times the density of states, we get,

$$\Omega(U)A_\infty(U,U') = \Omega(U')A_\infty(U',U), \tag{2.14}$$

where $A_\infty(U,U')$ denotes the infinite-temperature transition from U to U' and is independent of the acceptance probability $\alpha(\mathbf{x},\mathbf{x}')$. The fact that $A_\infty(U,U')$ depends only on $T(\mathbf{x},\mathbf{x}')$, the proposal move probability, allows us to estimate it during a Monte Carlo run by recording move statistics.

In multicanonical and Wang-Landau simulation, the acceptance probability

$$\alpha(\mathbf{x},\mathbf{x}') = \min\{1, \frac{\Omega(U)}{\Omega(U')}\}$$

is used to generate a flat histogram. In view of (2.14), however, the transition matrix Monte Carlo uses

$$\alpha(\mathbf{x},\mathbf{x}') = \min\{1, \frac{A_\infty(U',U)}{A_\infty(U,U')}\}$$

with $A_\infty(U,U')$ estimated periodically through the move statistics collected so far. As the energy histogram becomes more flat, the estimate for $A_\infty(U,U')$ becomes more accurate, and hence the estimate for density of states can be extracted by solving equation (2.14).

Even though TMMC does not directly reference the density of states and is only concerned with infinite-temperature transition probabilities, problems can arise in the early stage of the simulation when many energy values have not been visited, and thus making an estimate for $A_\infty(U,U')$ undefined [55]. To combine the best of both methods, namely the quick exploration of macrostates in Wang-Landau and the continuous improvement in the estimation of density of states in TMMC, Shell, Debenedetti, and Panagiotopoulos proposed a hybrid scheme in which move statistics are recorded in a Wang-Landau run and, periodically, the transition probabilities based on the current move statistics are used to obtain a refreshed estimate of density of states—a process they refer to as “refreshing”.

2.5.3 Remarks

For these density-of-states-based (DoS-based) methods, simulations are generally started with the "disordered" state of the system where proposal moves leading to unvisited states are definitely accepted to enable exploration of the whole energy spectrum. In some cases, there could be physical and geometric constraints on the systems of interest and it may be more convenient to just focus on a subset of the energy spectrum. One reason that we use parallel tempering as the first stage of our simulation instead of directly applying those DoS-based methods is because we wish to preserve certain features of the molecule of interest, and of the medium in which it resides. For instance, to investigate the effect secondary structure has on the aggregation of TatA molecule, we need both to maintain a proper shape of the helix and to make sure that that part of the molecule stays within the membrane during the course of the simulation. Since both the medium and structure of the molecule are encoded in our force field, an infinite-temperature simulation would inevitably sample the entire energy spectrum and explore part of phase space which may not be interesting to us. On the other hand, we can control and monitor the structural integrity of the molecule in a PT simulation by choosing a suitable temperature ladder.

Chapter 3

An exploration of parallel tempering with dynamic weighting

> The infinite! No other question has ever moved so profoundly the spirit of man.
>
> David Hilbert

In this chapter we develop a modified version of PT by incorporating the idea of dynamic weighting, we call it parallel tempering with dynamic weighting (PTDW). The main difference from standard PT is that the probability of making an exchange is no longer governed by the Metropolis rule as prescribed in (2.5), and it is allowable to make exchanges which would otherwise be rejected. An additional weight variable must be included to properly account for the bias incurred by this non-Metropolis move. We first present the modified PT algorithm, followed by a numerical study that compares PTDW with bare PT, and a test of the efficacy of the methods.

3.1 The PTDW algorithm

Motivated by [39], where the author introduced a simulated tempering with dynamic weighting (STDW) algorithm, we present a PT version with the "swap step" guided by the *Q-type* move (see Algorithm 3 on page 18). Note that here in Algorithm 4 the weight stays unchanged in the parallel step, and is updated according to the

Q-type move in the swap step.

Algorithm 4 PTDW algorithm(Q-type) for the multi-polymer system

Let the current state be $(\mathbf{x}_1^{(t)} \dots \mathbf{x}_K^{(t)}, w^{(t)})$, at step $t+1$,

- With probability α_0, do a parallel step. Update each $\mathbf{x}_k^{(t)}$, $k = 1, \dots, K$ according to its Metropolis step, that is, obtain $(\mathbf{x}_1^{(t+1)}, \dots, \mathbf{x}_K^{(t+1)})$ from Algorithm 1 on page 13. Let $w^{(t+1)} \leftarrow w^{(t)}$.
- Otherwise, randomly choose a neighboring pair $k-1$ and k from $\{1, \dots, K\}$ and draw $u \sim \text{Uniform}(0,\, 1)$, update

$$(\mathbf{x}_{1:K}^{(t+1)}, w^{(t+1)}) = \begin{cases} (\mathbf{y},\, \max\{1, w^{(t)} r^{(t)}\}) & \text{if } u \le \min\{1, w^{(t)} r^{(t)}\} \\ (\mathbf{x}_{1:K}^{(t)},\, a w^{(t)}) & \text{otherwise} \end{cases}$$

where $\mathbf{y}$ is the state with $\mathbf{x}_{k-1}^{(t)}$ and $\mathbf{x}_k^{(t)}$ swapped in $\mathbf{x}_{1:K}^{(t)}$, $a > 1$ and $r^{(t)} = \exp\left[(U(\mathbf{x}_{k-1}^{(t)}) - U(\mathbf{x}_k^{(t)}))(\frac{1}{T_{k-1}} - \frac{1}{T_k})\right]$ is the Metropolis acceptance ratio.

Another dynamic weighting move, the *R-type* move, was also proposed and shown to satisfy IWIW exactly [41]. [1] It is listed in Algorithm 5, from which we can see that rejections increase the chance of escaping from the current state, as is the case in *Q-type* move. Algorithm 4 can then be adapted to implement an R-type instead of Q-type move. However, as alluded in [41] and also verified by our simulation, variation of the resulting weights from an R-type move can be much larger than that of an Q-type move, thus exacerbating the quality of estimates constructed from those weights. [2] Henceforth, *Q-type* move is used as our dynamic weighting move.

3.2 Numerical study

The goal in this section is to numerically test if the new DWMC framework, in particular, the PTDW algorithm introduced above, does produce consistent results with that of a bare PT approach. If it does, then this new Monte Carlo scheme may be a competitive alternative to the study of systems having complex energy landscapes, such as the TatA protein aggregation model that we are interested in.

[1] The *Q-type* move only approximately satisfies IWIW.

[2] In fact, in our numerical study where *Q-type* move was used, the weights range from 10^{-2} to 10^{14} and already has sample variance of order 10^{24}. This observation confirms numerically the infinite-mean nature of the weights in dynamic weighting Monte Carlo.

Algorithm 5 *R-type* move

Let the current state be $(X^{(t)}, W^{(t)}) = (x, w)$.

- Propose y according to the proposal $T(x, y)$ and compute the Metropolis ratio

$$r(x,y) = \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}$$

- Draw $u \sim \text{Unif}(0, 1)$ and set

$$(X^{(t+1)}, W^{(t+1)}) = \begin{cases} (y, wr(x,y)+1) & \text{if } U \le wr(x,y)/(wr(x,y)+1) \\ (x, w(wr(x,y)+1)) & \text{otherwise} \end{cases}$$

It is observed, in our example study, that the new PTDW scheme found the low energy state much faster than the (bare) PT method. However, we also show that although weight distribution is stable in PTDW, property estimates are unstable; and while stratified truncation (ST) stabilizes these estimates, they disagree with the corresponding PT estimates, and a further study provides evidence that is against the PTDW estimates.

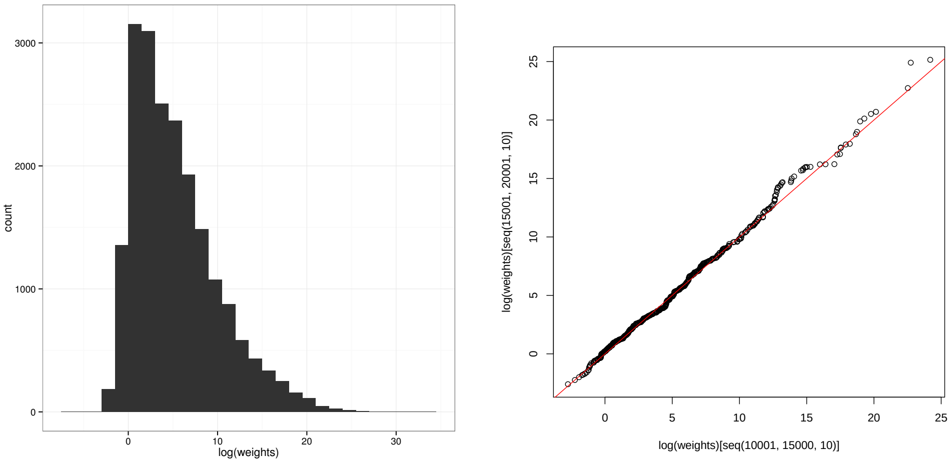
3.2.1 Unstable property estimates from PTDW

We know from Section 2.3 that the infinite-mean nature of the weight variable is certainly not a desirable property. To find out in a real example whether this has a severe impact on the estimation results, we study the basic model described in Section 1.2. To be consistent, the same set of temperatures was used in both PTDW and PT simulations, this along with observed swap rates of the PT simulation are listed in Table 3.1. While it is entropically favourable for the polymers to move independently, there is an energetic tendency for the hydrophilic tails of both polymers to interact with each other within the membrane and form a dimer. The equilibrium of the system is a balance between these two driving forces. At high temperatures, the entropy dominates and monomer state is predominant, and at low temperatures, the energy dominates and dimer state is predominant. There is an energy barrier between these two states and this is reflected by the low swap rate between T_2 and T_3 in Table 3.1.

We first examine the distribution of the (log) weights of PTDW to check that it is indeed a stable distribution, as mentioned in Section 2.3. One way to do this is to partition the log weights into contiguous blocks and compare their distribution functions. In Figure 3.1, we show the histogram of log weight and a

Temperature	T_1		T_2		T_3		T_4		T_5
	0.3		0.48		0.85		1.3		2.0
Swap rates		0.38		0.05		0.56		0.52	

Table 3.1: Temperatures and observed swap rates for the PT simulation. The PTDW simulation used the same set of temperatures.



(a)

(b)

Figure 3.1: Histogram of $\log w$ from the PTDW simulation (Figure 3.1a) and q-q plot of log weights corresponding to MC steps 10001—15000 (horizontal axis) and to steps 15001—20001 (vertical axis) (Figure 3.1b). The line $y = x$ is plotted in red in the q-q plot.

Quantile-Quantile (q-q) plot for two contiguous sets of log weights. The q-q plot is an effective graphical technique to inspect if two sets of data share a common distribution. The axes of a q-q plot are quantiles of the datasets. For continuous distributions the α-quantile ($\alpha \in [0, 1]$) is simply given by $F^{-1}(\alpha)$, where F is the distribution function, so, for example, the 0.5-quantile is just the median. As we can see in Figure 3.1b, the quantile points fall along the line $y = x$, suggesting that the two sets of log weights have similar distributions.

The mean potential energy of the system was used as a property for comparison, since it can be obtained easily from the simulations without additional calculations. To monitor convergence, we plot cumulative average of the estimates at each temperature. The results are shown side-by-side in Figure 3.2. In the case of PTDW, the cumulative average is defined as the weighted average:

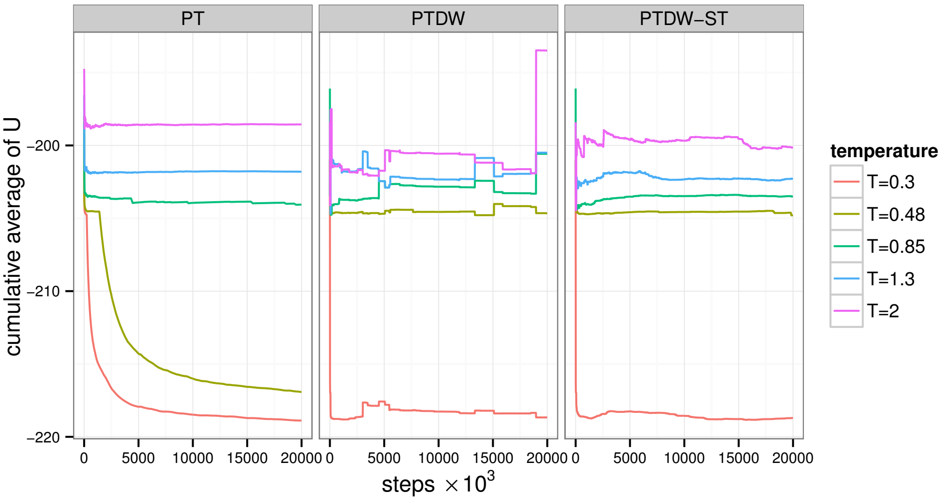


Figure 3.2: Cumulative average of potential energy computed by the method of PT (left), PTDW with raw weights (middle), and PTDW with stratified truncation (right). It is worth noting that because *stratified truncation* was applied based on the energy values of each temperature trajectory, the stratified weights became dependent on temperature; whereas when it is not applied, the same set of weights, namely the raw weights, are used in property estimates at all temperatures.

$$\overline{U}_n(\mathbf{x}) = \frac{\sum_{i=1}^{n} w^{(i)}\, U(\mathbf{x}^{(i)})}{\sum_{i=1}^{n} w^{(i)}}.$$

From the PT plot we see a much quicker convergence at higher temperatures than at lower temperatures; whereas in the case of PTDW (no stratified truncation), the behaviour is irregular, especially at higher temperatures ($T \geq 0.85$). We note that changes in energy as simulation goes along are negligible compare to changes in DW weights, which can be as large as 10^{14}, and so these “abrupt jumps” must be due to the large variability in the weights, where adding one observation with its associated weight significantly changes the values of $\overline{U}_n(\mathbf{x})$ accumulated from previous observations. So we see that, for the system we are interested in, raw weights from PTDW cannot be used to estimate expectations.

Post-processing the raw weights with the stratified truncation procedure, denoted as PTDW-ST, was conducted and the results are shown in the rightmost of Figure 3.2. Although one can play with different settings of the strata sizes, here we used 50 strata with equal number of samples in each stratum. It is observed that instability of the cumulative average is reduced considerably, however, the energy estimates do not agree with those from the PT simulation—most notably they disagree entirely at $T = 0.48$, with PT estimate being typical of the dimer state and PTDW-ST estimate being typical of the monomer state. The fact that there is a large discrepancy between the PT and the PTDW-ST estimates at this temperature let us presume that at least one of the two methods is converging to the wrong value. In the next section, we present evidence against the PTDW-ST estimate.[3]

Despite not being able to estimate averages reliably, the PTDW method did show its capability to quickly locate the low energy state, which can be seen by the rapid decrease of $\overline{U}_n(\mathbf{x})$ at $T = 0.3$, irrespective of whether stratified truncation was applied. In contrast, more than 5×10^6 MC steps were needed for equilibration in the PT method.

3.2.2 A model-dependent strategy for verification and efficient sampling

This subsection is related to Chapter 5, but rather than being generic, we shall present a model-dependent strategy to address the issue of inconsistent property estimates between PT and PTDW. The idea is the same as in multicanonical sampling, except that we now feed into the simulation the weights that are *derived from* a PT simulation. Specifically, first the density of states (DoS) is estimated, and then a biased simulation is run with weights determined by this estimated density of states. If the biased simulation produces a sufficiently enhanced sampling across all relevant regions of energy values, then satisfactory results can be obtained by reweighting to the canonical ensemble; otherwise, one has to either apply the kind of recursion rules discussed in Section 2.4, or use some advanced algorithms such as those introduced in Section 2.5, in order to further refine the DoS.

Fortunately, that our model is a two-state system and coexistence between dimer and monomer states is observed in one of the temperatures means that we can estimate the DoS separately for each state and then combine them together. By doing that, we were able to achieve sufficient sampling so that multicanonical recursions need not be implemented. We can then compare the new results with those of PT and PTDW all together.

More precisely, let $p(U|\beta)$ denote the probability density function of energy U at inverse temperature β, given by

[3] While working on this, we improved the method used for verification and realized that the new method not only serves the purpose of verification but can be used also as a generic Monte Carlo method—this will be the content of Chapter 5.

$$p(U|\beta) = Z(\beta)^{-1}\Omega(U)e^{-\beta U}, \tag{3.1}$$

where $Z(\beta) = \int \Omega(U)e^{-\beta U}dU$ is the configurational partition function and $\Omega(U)$ is the density of states. Assume the PT simulation has been run on a total of K temperatures. We discretize U in the sampled energy range and let $\{U_m\}_{m=1}^{M}$ be the midpoints of energy bins, that is, each bin is of the form $[U_m - \Delta U/2, U_m + \Delta U/2)$, where ΔU is the bin width. Here the range $[U_1,\ U_M]$ should cover both dimer and monomer (dispersed) states of the system: from the PT plot in Figure 3.2, this means that $U_1 \approx -219$ and $U_M \approx -198$.

Replacing $p(U_m|\beta_k)\Delta U$ by the observed frequency of bin m at inverse temperature β_k $(k = 1, \ldots, K)$, we get an estimate of $\Omega(U_m)$ from the PT simulation at temperature level k,

$$\hat{\Omega}_{km} = Z(\beta_k)e^{\beta_k U_m}H_{km}/(N\Delta U), \tag{3.2}$$

where H_{km} is the histogram count of energy bin m at temperature level k and N is the number of samples at each temperature.

To within a normalization constant, $\hat{\Omega}_{km}$ can be determined by

$$\hat{\Omega}^{\circ}_{km} = \frac{H_{km}e^{\beta_k U_m}}{\sum_{m=1}^{M} H_{km}e^{\beta_k U_m}}. \tag{3.3}$$

For each k, only a subset of energy pertinent to that temperature is sampled, and we will get many zeros in $\hat{\Omega}^{\circ}_{km}$ because no data is observed in the corresponding bins under a finite simulation. Two non-zero regions can be identified in the matrix $H_{m\times k}$: one with low energy, low temperature and corresponds to the dimer state (upper-left of $H_{m\times k}$); the other one with high energy, high temperature and corresponds to the dispersed state (lower-right of $H_{m\times k}$). The full $H_{m\times k}$ matrix of a replicate PT simulation is listed in Table B.1.

To proceed, we restrict (3.3) to ranges of k and m corresponding to the two states and calculate $\hat{\Omega}^{\circ}_{km}$ for each state. This puts the two domain DoS's on a different scale so we need a scaling factor s to reconstruct the overall DoS:

$$\hat{\Omega}(U_m) = \begin{cases} \hat{\Omega}_1(U_m) & \text{if } U_m \in \mathcal{U}_1 \\ s\,\hat{\Omega}_2(U_m) & \text{if } U_m \in \mathcal{U}_2, \end{cases} \tag{3.4}$$

where $\mathcal{U}_1$ and $\mathcal{U}_2$ are the energy ranges of dimer and dispersed states, respectively, $\hat{\Omega}_1$ is calculated via (3.3) but restricted to $\mathcal{U}_1$, and similarly for $\hat{\Omega}_2$. We can approximate s if both states are sampled in one of the temperatures. By the equation of relative

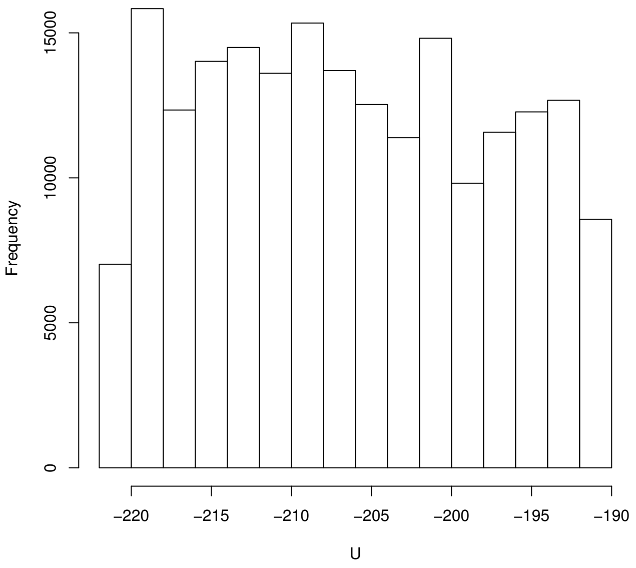


Figure 3.3: Energy histogram of one of the 10 independent biased simulations. The weights of the biased simulation were determined by the replicate PT simulation.

population of the two states, we have,

$$\frac{p_1}{p_2} = \frac{\int_{\mathcal{U}_1} \Omega(U) e^{-\beta U} dU}{\int_{\mathcal{U}_2} \Omega(U) e^{-\beta U} dU} \approx \frac{\sum_{m \in \mathcal{M}_1} \hat{\Omega}_1(U_m) e^{-\beta U_m}}{s \sum_{m \in \mathcal{M}_2} \hat{\Omega}_2(U_m) e^{-\beta U_m}}. \tag{3.5}$$

Since coexistence is observed at $T = 0.85$ in our PT simulation, the factor s can be estimated by substituting $(0.85)^{-1}$ for β into (3.5) and replacing p_1/p_2 by the observed relative frequency. The biased simulation can then be run with weights proportional to $1/\hat{\Omega}(U_m)$.

To avoid risk of self-fulfilling, a replicate PT simulation with the same specifications as the one conducted in Section 3.2.1 was run, and the procedure described above was followed. Specifically, we have run 10 independent biased simulations, each used 5×10^6 MC steps, in order to get an idea of the statistical errors of estimation.

Figure 3.3 shows the energy histogram of one such run, from which we can see that the energy range covers both dimer and dispersed states, and each energy bin is sampled relatively frequently. Therefore, multicanonical recursions need not to be implemented, and property estimates can be obtained by importance reweighting.

Figure 3.4 plots the energy estimates and associated statistical errors from

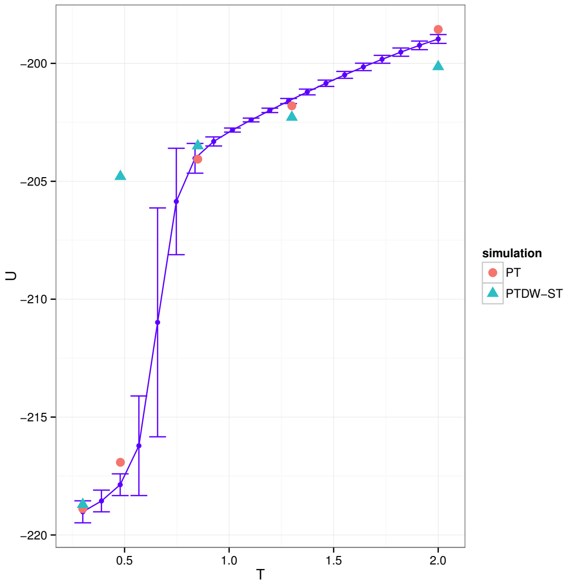


Figure 3.4: Estimates of the mean potential energy from the biased simulation (blue), PT (circle) and PTDW-ST (triangle). The half-length of the error bar equals two times the standard deviation of the estimates obtained from the 10 independent runs.

the biased simulations, against the estimates from the original PT and PTDW-ST simulations in Section 3.2.1. We see that at $T = 0.48$, the PTDW-ST estimate is a long way from the statistically insignificant region of the biased simulation, thus giving a high confidence to reject the PTDW-ST estimate.

3.3 Concluding remarks

Our work in this chapter confirms the issues raised in Section 2.3. The dynamic weighting Monte Carlo framework introduces many theoretical difficulties and it is hard to establish a similar theorem to the ergodic theorem in standard Markov Chain Monte Carlo which guarantees the convergence of averages. The fact that the weighting variable may have infinite mean is a sign of risk when attempting to use its realizations to estimate equilibrium averages. Even if stratified truncation has been used to post-processing the weights, our results of a lattice-polymer example show that property estimates are still unreliable. It should be noted, however, that the PTDW scheme found the low energy state much faster than the PT method, suggesting that the scheme may be used as an optimization procedure, such as searching for the lowest energy configuration. This is also supported by the fact that the dynamic weighting method has been successfully applied to certain optimization problems [41, 65].

We are not aware of any practitioners in the field of computational chemistry and physics having applied the dynamic weighting Monte Carlo approach to their own problems. As stated in [41] already,

> “The waiting time infinity in the standard Metropolis process now manifests itself as an ‘importance weight infinity’ in the dynamic weighting process”.

Chapter 4

Analysing simulation data

Computer simulations have become indispensable in modern scientific research with more and more enlightening results being published in almost all areas of science. Nowadays many researchers who use computer simulation as a mean of research have access to High Performance Computing (HPC). These facilities provide massive computation power and multiple processors so that the running time is significantly reduced for a well-parallelized job. Nevertheless, theoretical advances in methodology are still very important and valuable; and for some cases it might not be straightforward to parallelize the code for the specific problem at hand.

From the methodology side, improvement on accuracy of the results can be obtained either through an improvement of the simulation method, or through an optimal way of utilizing the amount of information in the output. The former is done at simulation stage and is the focus of Chapter 2 and Chapter 3, while the latter is done at estimation stage and is the focus of this chapter. We should emphasize, however, that the two themes are not completely separate, as we will be demonstrating that the method developed in analysis can be applied back to simulation to achieve great efficiency improvement.

The purpose of this chapter is therefore twofold: to provide a foundation for the next chapter by reviewing some analysis methods aimed at improving the statistical quality of the data, and to show that advances in statistical methods play an important role in computational chemistry.

4.1 The Weighted Histogram Analysis Method

The Weighted Histogram Analysis Method (WHAM) is an extension of the multiple histogram equations of Ferrenberg and Swendsen [18], and was applied for the first time to molecular simulation data for free energy and potential of mean force (PMF) calculations in [34]. It can also be used to calculate thermodynamic averages [12].

Below we review the derivation of WHAM using the conventional approach. An alternative maximum likelihood approach is described by Bartels [2]. The conventional derivation of WHAM relies on the following simple identity in statistics which we shall refer to as the optimal rule:

Optimal rule Let $\mathbf{X}$ denote the random sample $\{X_j\}_{j=1}^N$, and let $\{W_i(\mathbf{X})\}_{i=1}^k$ be k independent estimators of some interesting quantity θ, with $\mathrm{Var}(W_i) = \sigma_i^2$, then the best estimator of θ, in the sense that it has the lowest variance among all weighted estimators, is given by

$$W_{\mathrm{opt}} = \frac{\sum_{i=1}^k (\sigma_i^2)^{-1} W_i}{\sum_{i=1}^k (\sigma_i^2)^{-1}}, \tag{4.1}$$

with

$$\mathrm{Var}(W_{\mathrm{opt}}) = \frac{1}{\sum_{i=1}^k (\sigma_i^2)^{-1}}. \tag{4.2}$$

We see that the optimal estimator is weighted by the inverse of the variance of each individual estimator. When the variance of an estimator is small, it contributes more to the resulting optimal estimator, and vice versa.

Suppose we have carried out K independent simulations at temperatures $T_1, \ldots, T_k$, and we are interested in inference about thermodynamic properties at some temperature, which may not be one of those temperatures used in simulation, from *all* simulation data. The idea is to derive an estimate for the density of states from each simulation and combine them using the optimal rule (4.1).

In the canonical ensemble at inverse temperature β, we can write down the probability density function of the potential energy U as

$$p(U|\beta) = Z(\beta)^{-1}\Omega(U)e^{-\beta U}, \tag{4.3}$$

where

$$Z(\beta) = \int \Omega(U)e^{-\beta U}dU \tag{4.4}$$

is configurational partition function, $\Omega(U)$ is the density of states.

We consider a discretization of U across the sampled energy range. Let M be the total number of bins, with $\{U_m\}_{m=1}^M$ being the midpoints of each bin. Then,

$$\Omega(U_m) = p(U_m|\beta)Z(\beta)e^{\beta U_m}. \tag{4.5}$$

For simulation at inverse temperature $\beta_k,\ k = 1, \ldots, K$, the value that the probability density function takes at U_m can be approximated by

$$p(U_m|\beta_k) \approx \frac{H_{km}}{N_k \Delta U} \tag{4.6}$$

where H_{km} is the histogram count of energies in bin m from simulation k, N_k is the total number of samples from simulation k and ΔU is the bin width. Substituting (4.6) into (4.5) we get an estimator for the density of states at U_m from simulation k:

$$\begin{aligned} \hat{\Omega}_{km} &= \frac{H_{km}}{N_k \Delta U} Z(\beta_k) e^{\beta_k U_m} \\ &= \frac{H_{km}}{N_k \Delta U} e^{\beta_k U_m - f_k} \end{aligned} \tag{4.7}$$

where $f_k = -\ln Z(\beta_k)$ is the dimensionless free energy at T_k.

If we can determine the variance of H_{km}, then we can apply the optimal rule to combine the estimators from all temperatures. To that end, we make an additional assumption that the samples themselves are independent. This simplifies the derivation of WHAM but is unrealistic as data generated from Monte Carlo or molecular dynamics simulations are often highly correlated. The correlated nature of the data can be accounted for by introducing statistical inefficiency terms.

Under independence assumption, H_{km} follows a binomial distribution with parameters N_k and p_{km}, where p_{km} is the probability of a sample drawn in bin m from simulation k. Instead of estimating it directly by the observed frequency, we express $\hat{p}_{km}$ in terms of the yet to be determined optimal estimator $\hat{\Omega}_m$:

$$\hat{p}_{km} = \hat{\Omega}_m e^{f_k - \beta_k U_m} \Delta U. \tag{4.8}$$

Note that we have dropped the subscript k in $\hat{\Omega}_{km}$ that indexes simulations.

If ΔU is small and the samples are sparsely spread across all the bins, then a further assumption is that $p_{km} \ll 1$. Then,

$$\begin{aligned} \mathrm{Var}(H_{km}) &= N_k p_{km}(1 - p_{km}) \\ &\approx N_k \hat{p}_{km} \\ &= N_k \hat{\Omega}_m e^{f_k - \beta_k U_m} \Delta U \end{aligned} \tag{4.9}$$

and, from (4.7),

$$\begin{aligned}\mathrm{Var}(\hat{\Omega}_{km}) &= \left(\frac{e^{\beta_k U_m - f_k}}{N_k \Delta U}\right)^2 \mathrm{Var}(H_{km}) \\ &\approx \frac{\hat{\Omega}_m e^{\beta_k U_m - f_k}}{N_k \Delta U} \qquad (4.10)\end{aligned}$$

Now, according to the optimal rule, we obtain the optimal estimator for $\Omega(U_m),\ m = 1, \ldots, M$,

$$\hat{\Omega}_m = \frac{\sum_{k=1}^{K} H_{km}}{\sum_{k=1}^{K} N_k \Delta U e^{f_k - \beta_k U_m}}. \qquad (4.11)$$

If we approximate the integral appearing in the partition function $Z(\beta_k)$ by a finite sum we can write down f_k as

$$f_k = -\ln \sum_{m=1}^{M} \hat{\Omega}_m e^{-\beta_k U_m} \Delta U \qquad (4.12)$$

Equation (4.11) and (4.12) together define the WHAM equations and are the operational form used in [34]. The dimensionless free energy f_k can be solved self-consistently by iterating through the equations with an initial value of f_k, say $f_k = 0$ for all k.

Also, the variance of $\hat{\Omega}_m$ can be obtained from (4.2):

$$\mathrm{Var}(\hat{\Omega}_m) = \frac{\hat{\Omega}_m}{\sum_{k=1}^{K} N_k e^{f_k - \beta_k U_m} \Delta U} \qquad (4.13)$$

where the $\hat{\Omega}_m$ on the right of the equation should be interpreted as the solution of the WHAM equations (4.11) and (4.12).

To summarize, we state once more the assumptions used to derive WHAM equations:

1. The simulations are independent.

2. The samples collected from each simulation are independent.

3. For each simulation, samples are sparsely distributed across all bins.

As mentioned before, the second assumption is unrealistic in most molecular simulations. In order to extend WHAM equations so that they are applicable to correlated samples, a statistical inefficiency term g_{km} is needed. In fact, g_{km} is the number of correlated samples required to perform an independent sampling with respect to bin m in simulation k [12]. In other words, if we associate each bin with an experiment that counts the number of samples in that bin, then the effective sample size of the mth experiment is N_k/g_{km}. By applying this correction, we get a more general version of WHAM equations:

$$\hat{\Omega}_m = \frac{\sum_{k=1}^{K} (g_{km})^{-1} H_{km}}{\sum_{k=1}^{K} (g_{km})^{-1} N_k e^{f_k - \beta_k U_m} \Delta U}, \tag{4.14}$$

with the corresponding variance

$$\mathrm{Var}(\hat{\Omega}_m) = \frac{\hat{\Omega}_m}{\sum_{k=1}^{K} N_k (g_{km})^{-1} e^{f_k - \beta_k U_m} \Delta U}. \tag{4.15}$$

The statistical inefficiency g was included in the original derivation of Ferrenberg and Swendsen [18] but omitted in Kumar et al. [34]. They claimed that for many biomolecular systems with no phase transitions, g_{km} is independent of k, and hence, cancel out in (4.14), so the equation reduces to (4.11). The effect of neglecting g_{km} was further examined in [12], where the authors found that, when using their approach to apply WHAM to parallel tempering simulations, for the same energy bin, g_{km} could differ by up to two orders of magnitude for different k.

The main purpose of [12], however, is to provide a detailed analysis of the use of WHAM for simulated and parallel tempering simulations, which are of interest to our study.

Since one automatically obtains data from all temperatures in a PT simulation, it is tempting to apply WHAM to the analysis of such data. Unfortunately, the first two assumptions are both violated if one apply directly the WHAM equations to PT data organized by temperatures. Even if effectively uncorrelated samples are used for each temperature, the samples between temperatures are not independent because of the "swap step" in the algorithm, so the first assumption is violated. The third assumption is also likely to be violated, as each temperature may sample a different region of the configuration space, resulting in uneven distribution of energy bins. To rectify these issues, Chodera et al. suggested that the data be organized not by temperatures, but by "replicas". Each such replica can be viewed as an independent simulated tempering simulation from which an estimator for the density of states as well as its uncertainty can be derived. The optimal rule is then applied to combine the estimators from all replicas to produce a single best estimator for the density of states.

There are limitations of WHAM and also some practical issues when taking into account correlations in simulation data.

First, the bin width ΔU needs to be chosen carefully. Clearly, it should not be too large otherwise many of our approximations will not hold and the estimator for the density of states, $\hat{\Omega}_m$, will be very poor—think of approximating a function with a few distinct values. On the other hand, a small bin width will cause the number of independent samples in each bin to decrease. This will increase the relative uncertainty of the probability of that bin, since, from (4.8) and (4.13), we have

$$\frac{\mathrm{Var}(\hat{p}_{km})}{\hat{p}_{km}^2} = \frac{1}{\hat{\Omega}_m \Delta U \sum_{k=1}^{K} N_k e^{f_k - \beta_k U_m}},$$

which is large when ΔU is small.

Second, the statistical inefficiency g is not trivial to calculate. By definition, $g = 1 + 2\tau$, where τ is the integrated auto-correlation time and is effectively a sum of the lag-t auto-correlation functions $C_t,\ t = 1, \ldots, N-1$ with N being the total sample size. It was shown that the accuracy of C_t deteriorates as the lag t increases [72], so truncations are often involved in calculating τ. For example, Chodera et al. [12] truncate the sum when C_t first crosses zero.

In addition, it is difficult to assess the quality of the estimator that was used to estimate the expectation of an observable of interest. More precisely, let $A(\mathbf{x})$ be some thermodynamic quantity which is a function of system coordinates $\mathbf{x}$. The expectation of $A(\mathbf{x})$ with respect to the Boltzmann distribution π, expressed in terms of integration over energy, is given by,

$$\mathrm{E}_\pi A(\mathbf{x}) = \frac{\int \Omega(U) e^{-\beta U} \bar{A}(U) dU}{\int \Omega(U) e^{-\beta U} dU}, \tag{4.16}$$

where

$$\bar{A}(U') = \frac{\int \delta(U(\mathbf{x}) = U') A(\mathbf{x}) d\mathbf{x}}{\Omega(U')}$$

is the average of $A(\mathbf{x})$ over those configuration with $U(\mathbf{x}) = U'$.

Equation (4.16) was estimated in [12] by discretizing U and replacing both integrals with finite sums. Although the authors discussed the statistical uncertainty of the estimator, their approach involves the calculation of several statistical inefficiency terms which are not trivial to calculate. Furthermore, apart from knowing the uncertainty, or variance, of an estimator, we also need to know its "biasedness", that is, how much does the expectation differ from the true value? An unbiased estimator is such that the expectation equals the true value for all sample sizes. If, under infinite sample size, the estimator is still biased, then it probably should not be used.

4.2 The Multistate Bennett Acceptance Ratio estimator

One of the most important tasks in computational chemistry and physics is the calculation of free energy difference. Much effort has been made towards efficient calculations of free energy. The Weighted Histogram Analysis Method discussed in Section 4.1 was a relatively recent method. Earlier methods include one-sided exponential averaging [71], the Bennett acceptance ratio (BAR) method [3] and umbrella sampling [60], among others. The Multistate Bennett Acceptance Ratio method (MBAR) [56] is a generalization of BAR to multiple thermodynamic states and has many advantages over existing methods. In particular, it does not require discretization of energy and thus removes the bias introduced by binning in WHAM. It has also been shown that the estimator is asymptotically unbiased and has the lowest variance among all commonly used reweighting estimators [59].

Actually, the mathematical formulation of the theory of MBAR estimator was based on the work of statisticians [33, 47, 59]. When Bennett [3] published the acceptance ratio method for free energy calculation, it was difficult for researchers outside the field of computational physics to appreciate its generality. It was not until twenty years later that Meng and Wong [47] independently discovered an important identity for which the BAR method is a special case. Subsequent research in the statistics community have established an extension of the identity to multiple densities and proved the optimality of the resulting estimator [59]. The result was rediscovered and applied back to free energy calculation in [56], only four years later.

In this section we will take a somewhat different approach and review BAR and MBAR from a purely statistical perspective, realizing that the task of estimating free energy difference can be formulated as estimating ratios of normalizing constants. This observation in some sense draws attentions of statisticians because the task is often encountered in statistical procedures as well, such as computing likelihood ratios and Bayesian inference [47].

Let us start with two thermodynamic states, 1 and 2, and we are interested in estimating their free energy difference. An example related to our work would be a system at two different temperatures in the canonical ensemble. Because the partition function $Z(\beta_i)$ $(i = 1, 2)$ is just a normalizing constant, we use a simplified notation c_i. The distribution associated with state i is then,

$$p_i(\mathbf{x}) = \frac{q_i(\mathbf{x})}{c_i},$$

where $q_i(\mathbf{x})$ is known and $c_i = \int q_i(\mathbf{x})d\mathbf{x}$. The reduced free energy difference is the logarithm of the ratio of normalizing constants:

$$\Delta f_{12} = f_2 - f_1 = \ln \frac{c_1}{c_2}.$$

The goal is then to estimate the ratio $r = c_1/c_2$ efficiently given draws (may be dependent) from both densities.

Bennett proposed to estimate r as a ratio of canonical averages:

$$r = \frac{c_1}{c_2} = \frac{\mathrm{E}_2[q_1(\mathbf{x})\alpha(\mathbf{x})]}{\mathrm{E}_1[q_2(\mathbf{x})\alpha(\mathbf{x})]} \tag{4.17}$$

where E_i denotes expectation with respect to p_i and $\alpha(\mathbf{x})$ is some arbitrary function. He then chose α to minimize the variance of the estimator for $\ln r$. Many previous methods can be regarded as special cases of (4.17). For example, taking $\alpha(\mathbf{x}) = q_2^{-1}(\mathbf{x})$ gives the importance sampling identity $r = \mathrm{E}_2[q_1(\mathbf{x})/q_2(\mathbf{x})]$. The key identity (4.17) was independently discovered and extensively studied later by Meng and Wong [47]. They also considered its generalizations to multiple states.

Now, given draws from both densities, $\{\mathbf{x}_{1n}\}_{n=1}^{N_1}$ and $\{\mathbf{x}_{2n}\}_{n=1}^{N_2}$, the Monte Carlo estimator of (4.17) is given by

$$\hat{r}_\alpha = \frac{N_2^{-1}\sum_{n=1}^{N_2} q_1(\mathbf{x}_{2n})\alpha(\mathbf{x}_{2n})}{N_1^{-1}\sum_{n=1}^{N_1} q_2(\mathbf{x}_{1n})\alpha(\mathbf{x}_{1n})}, \tag{4.18}$$

where we used subscript α to make it explicit that the estimator depends on the choice of the function α. The next rational step is then to choose such α that minimizes the relative mean squared error (MSE) of the estimator. It was proved in [47] that, under the assumption that the configurations are statistically independent, the optimal α that minimizes the asymptotic MSE of $\ln \hat{r}_\alpha$ is given by

$$\alpha_{\mathrm{opt}} \propto \frac{1}{N_1 p_1 + N_2 p_2} = \frac{1}{N_1 q_1 + N_2 q_2 r}, \tag{4.19}$$

with the corresponding minimum error

$$\left[\int \frac{N_1 N_2\, p_1\, p_2}{N_1\, p_1 + N_2\, p_2}\, d\mathbf{x}\right]^{-1} - \frac{1}{N_1} - \frac{1}{N_2}. \tag{4.20}$$

Note that the optimal α depends on the unknown ratio r, so the optimal estimator can not be obtained directly. Nevertheless, an iterative scheme can be applied by plugging α_{opt} into (4.18) and, starting with an initial value of $r^{(0)}$, compute iteratively the next estimate $r^{(t)},\ t = 1, 2 \ldots$. It was shown that the resulting sequence is convergent and that the limit, $\hat{r}_{\mathrm{opt}}$, has an asymptotic mean squared error given by (4.20) [47]. In other words, the iteration of the form $r^{(t+1)} = g(r^{(t)})$, with g defined by equations (4.18) and (4.19), has a unique fixed point whose statistical error is the same as the minimum error one would get for the optimal but infeasible α_{opt}. It is also informative to see that if p_1 and p_2 were identical, then the minimum error would be 0. In other words, the BAR estimator will be exact if the two densities completely overlap.

When data from K ($K > 2$) thermodynamic states are available, one might be interested in estimating the ratios $r_i = c_1/c_i,\ i = 2, \ldots, K$. In fact, this is precisely what we will be doing for a parallel tempering simulation, and efficient estimation of these ratios constitutes a key component in the new sampling scheme to be described later.

For the multistate case, a straightforward solution would be estimating each ratio r_i via the BAR estimator, using samples from p_1 and p_i. However, one might ask whether we can do better by using samples from *all* K densities in estimating *each* r_i. Meng and Wong introduced the idea of "bridge sampling" in an attempt to extend the key identity (4.17) to cases where multiple densities are involved. Their idea was based on the observation that if p_1 and p_2 do not have sufficient overlap but p_3 overlaps with both p_1 and p_2, then instead of estimating r_2 directly through p_1 and p_2, one could estimate it indirectly via p_3, representing the product estimation $c_1/c_2 = (c_1/c_3)(c_3/c_2)$. The statistical quality of the indirect estimator should be much better, because each pair of densities in the product now have significantly more overlap. Therefore, p_3 can be viewed as a "bridge" between p_1 and p_2. This approach of extension using estimating equations has been shown by Tan [59] to be consistent with a maximum likelihood approach [33]. We state the main result of the extension.

Consider a generalized version of (4.17):

$$\frac{c_1}{c_i} = \frac{\mathrm{E}_i[q_1(\mathbf{x})\alpha_{ij}(\mathbf{x})]}{\mathrm{E}_1[q_i(\mathbf{x})\alpha_{ij}(\mathbf{x})]}, \qquad 2 \le i \le K,\ 1 \le j \le K,\ j \ne i, \tag{4.21}$$

with the Monte Carlo estimator:

$$\hat{r}_i = \frac{N_i^{-1}\sum_{n=1}^{N_i} q_1(\mathbf{x}_{in})\alpha_{ij}(\mathbf{x}_{in})}{N_1^{-1}\sum_{n=1}^{N_1} q_i(\mathbf{x}_{1n})\alpha_{ij}(\mathbf{x}_{1n})}. \tag{4.22}$$

Then the optimal α_{ij} that minimizes the asymptotic variance of $\hat{r}_i$ is given by

$$\alpha_{ij}(\mathbf{x}) = \frac{N_j c_j^{-1}}{\sum_{k=1}^{K} N_k c_k^{-1} q_k(\mathbf{x})}, \tag{4.23}$$

and that the bridging sampling estimator $\hat{r}_i$ is consistent and asymptotically normal.

Replacing r_i with c_1/c_i, equation (4.23) can be rewritten as

$$\alpha_{ij}(\mathbf{x}) = \frac{N_j r_j}{\sum_{k=1}^{K} N_k r_k q_k(\mathbf{x})}, \tag{4.24}$$

where, again, the optimal choice depends on the unknown ratios $\{r_i\}_{i=2}^{K}$, since $r_1 = 1$ by definition.

As an extension to (4.18) and (4.19), equations (4.22) and (4.24) define a set of $K-1$ estimating equations which can be solved self-consistently for $\hat{r}_i$. If $K = 2$, then there is only one such function α that needs to be determined and (4.24) reduces to (4.19), so we see that the MBAR estimator is indeed an extension of BAR to multiple states.

Not only can MBAR be used to estimate free energy differences, it can also be used to estimate equilibrium expectations at almost any thermodynamic state [56]. The idea is, not surprisingly, to think of the expectation as ratio of normalizing constants of some "fictitious" states. More precisely, the expectation of some observable $A(\mathbf{x})$ with respect to some state s is

$$\mathrm{E}_s A(\mathbf{x}) = \frac{\int q_s(\mathbf{x}) A(\mathbf{x}) d\mathbf{x}}{\int q_s(\mathbf{x}) d\mathbf{x}},$$

where s may not necessarily be one of the K states already sampled, in which case we treat $N_s = 0$. Let $\tilde{q}(\mathbf{x}) = q_s(\mathbf{x})A(\mathbf{x})$ and $\tilde{c} = \int \tilde{q}(\mathbf{x}) d\mathbf{x}$, then

$$\mathrm{E}_s A(\mathbf{x}) = \frac{\tilde{c}}{c_s}.$$

We then augment the set of estimating equations to include the new "state" with normalizing constant $\tilde{c}$ and another one with normalizing constant c_s, if s is an unsampled state. The corresponding sample sizes for the new "states" are set to zero so that no additional iterations are needed and expectations along with their uncertainties can be computed efficiently.

It is important to keep in mind that, as in the case of WHAM, both BAR and MBAR have assumed that the samples are uncorrelated both within and between states. Although one might still apply MBAR to correlated data, any statistical errors associated with the estimates so derived will be unreliable [56].

An interesting observation of the connection between MBAR and WHAM can be seen if we notice that the MBAR estimator of the dimensionless free energy at state i,

$$\hat{f}_i = -\ln \sum_{j=1}^{K} \sum_{n=1}^{N_j} \frac{q_i(\mathbf{x}_{jn})}{\sum_{k=1}^{K} N_k q_k(\mathbf{x}_{jn}) \exp(\hat{f}_k)}, \tag{4.25}$$

is precisely the f_i of Equation (21) in [34], if we substitute $q(\mathbf{x})$ with the Boltzmann weight $\exp(-\beta U(\mathbf{x}))$. However, the equation in [34] was not a direct consequence of WHAM which would otherwise require constructing histograms, but rather, a convenient formula proposed by the authors for the calculation of f_i directly from the data, by treating each data point as occupying its own "bin" with a bin width of zero.

Based on this observation, we see that the MBAR estimator for free energies coincides with the WHAM estimator when the bin width is reduced to zero. However, a zero bin width cannot be used in the derivation of WHAM, as the density of states from each simulation cannot be constructed when $\Delta U = 0$. On the other hand, the derivation of (4.25) was based on extended bridge sampling theory and, as a consequence, has the desired optimality properties.

Chapter 5

Efficient calculation of density of states using MBAR

In this chapter, we present a new method to calculate the density of states using the MBAR estimator. A combination of PT and MUCA will be used to demonstrate the efficiency of our method in a statistical model of sampling from a two-dimensional normal mixture and also in a physical model of aggregation of lattice polymers. While MBAR has been commonly used for final estimation of thermodynamic properties, our numerical results show that the efficiency of estimation with our new approach, which uses MBAR as an intermediate step, often improves upon conventional use of MBAR. We also demonstrate that it can be beneficial in our method to use full PT samples for MBAR calculations in cases where simulation data exhibit long correlation.

The work in this chapter has been published under the title "Improved estimation of density of states for Monte Carlo sampling via MBAR" (see [68]).

5.1 Introduction

Generally speaking, the MC methods used to study physical and chemical systems can be broadly classified into two categories: temperature-based and energy-based. In temperature-based methods, the system is simulated at one or several predefined temperatures and the Boltzmann weight is used; examples of such methods include the classical Metropolis algorithm, simulated tempering and parallel tempering. One deficiency of these methods, apart from the actual sampling, is that quantities like the free energy and entropy are not directly accessible. Fortunately, advances in free energy calculation greatly facilitate analysis of simulation data with much higher statistical efficiency than earlier methods. For example, in the Weighted Histogram Analysis Method (WHAM), the density of states (DoS) are estimated by optimally combining estmates from each simulation after discretizing energy with a suitable resolution; and in the Multistate Bennett Acceptance Ratio (MBAR) method, the problem of calculating free energy differences is treated as a problem in estimating the ratio of normalizing constants and it uses extended bridge sampling theory to derive statistical estimators that are proven to be optimal. MBAR removes discretization of energy, is capable of directly producing estimates of free energy and equilibrium expectations, and thus obviates the need to calculate the density of states.

In contrast to temperature-based methods, energy-based methods usually work in a generalized ensemble which is independent of temperature. We have come across such methods in Section 2.4 and Section 2.5, and we see that they all try to achieve an even sampling of energy by iteratively refining the DoS. One could either reweight these biased samples to obtain properties with respect to the canonical ensemble or use the DoS produced from the final iteration to calculate thermodynamic properties, in which case the histogram should be sufficiently flat to provide acceptable accuracy.

While it is generally considered a merit that methods like the Wang-Landau algorithm allow for a quick exploration of the whole energy spectrum, there are situations where this is not always desirable. For example, the range of potential energy in complex systems could span several orders of magnitude. Two implications are that computationally, the time needed to traverse all energy levels in a random walk increases as the square of energy range; and that practically, it might be the case that only part of the configuration space, hence a subset of all accessible energy levels, are of interest. This suggests that instead of directly applying energy-based methods, which assumes no prior knowledge about the system, we may initially run a temperature-based simulation and then, based on the information we have collected, apply one of the energy-based methods to a reduced range of energy.

We report an approach to derive estimates of the density of states from the MBAR estimator. In WHAM, these estimates come out naturally because histograms are used. However, since WHAM solves a self-consistent equation concerning the DoS and free energy, this discretization will introduce error to free energy estimates which in turn causes DoS estimates to be inaccurate. In contrast, MBAR does not require discretization of energy so this error in free energy is removed.

To illustrate how this idea can be applied in practice, we use a combination of PT and MUCA in two examples: a statistical example to demonstrate the correctness of the method, and a lattice-polymer example to demonstrate the efficiency and utility of the method in real physical models.

Use of both PT and MUCA, rather than either method alone, is beneficial to the kind of model we are interested in. Although parallel tempering is a powerful algorithm to simulate bead-polymer systems, which are often characterized by many local energy minima, it can become inefficient in situations where the system undergoes a phase change that resembles a first-order phase transition. Because there is a steep change in energy, the transition rate between low and high energy states can be low even if the chosen temperature difference is small. In contrast, by sampling from the multicanonical ensemble, the sampler can move freely in energy space because a flat energy histogram will be produced with good estimates of the density of states. As the weights used in multicanonical simulation are *a priori* unknown, they are normally set to be equal to one at the start of the simulation, indicating that the system is started from the disordered state with all configurations equally likely. This means that the sampler may have difficulty sampling low energy configurations whose phase space volume is proportionally smaller than high energy configurations, and so it may take some time to produce "working estimates" of the density of states.

This suggests that there can be merit in using an estimated DoS from PT as the weights of a MUCA simulation. This idea was used, although in a different context, in Section 3.2.2 of Chapter 3. In general, without having to partition the DoS based on system characteristics, one could use the WHAM method by following the advice of [12]. We show, however, that an efficient alternative is to use the MBAR estimator instead.

5.2 Estimating density of states using MBAR

Suppose independent canonical simulations have been carried out at K temperatures. Consider a discretization of U in the energy range sampled from the K simulations. Instead of including all energy levels that have been seen, it is possible to ignore those that are close to the high energy end of the spectrum and so are rarely sampled. In this way one can reduce the range of the spectrum to include only interesting system events, e.g. phase transitions, although results associated with the highest temperature distribution are then likely to be inaccurate. Following the same notations as Section 3.2.2, but bear in mind that k now indexes simulations that are not necessarily from PT, we have,

$$\hat{\Omega}_{km} = Z(\beta_k) e^{\beta_k U_m} H_{km} / (N_k \Delta U), \tag{5.1}$$

where N_k is the number of samples from simulation k and may be different for each simulation.

We can write down estimates of $\log \Omega(U_m)$ from each temperature simulation by taking the logarithm of (5.1):

$$\begin{aligned}
\log \hat{\Omega}_{1m} &= \log Z(\beta_1) + \beta_1 U_m + \log H_{1m} - \log(N_1 \Delta U) \\
\log \hat{\Omega}_{2m} &= \log Z(\beta_1) + \log \frac{Z(\beta_2)}{Z(\beta_1)} + \beta_2 U_m + \log H_{2m} - \log(N_2 \Delta U) \\
&\vdots \\
\log \hat{\Omega}_{Km} &= \log Z(\beta_1) + \log \frac{Z(\beta_K)}{Z(\beta_1)} + \beta_K U_m + \log H_{Km} - \log(N_K \Delta U).
\end{aligned} \tag{5.2}$$

Because it is only needed to determine $\log \Omega$ up to an additive constant, we see that the first term $\log Z(\beta_1)$ can be ignored, and only $\log Z(\beta_k)/Z(\beta_1)$, $k = 2, \ldots, K$ need to be estimated; but these are precisely the dimensionless free energy differences. We can therefore use the MBAR estimator to best estimate these quantities.

Note that there will be K independent estimates of the density of states in (5.2). Since MBAR also yields uncertainty estimates of the free energy differences, it followes naturally that the estimates $\log \hat{\Omega}_{km}$ should then be weighted inversely proportional to their variances.

5.3 Working with parallel tempering simulation

The parallel tempering simulation simultaneously simulates the system at multiple temperatures that form a temperature ladder. A key step in PT is the exchange of configurations between neighbouring temperatures to speed up the mixing of chains simulated at low temperatures, thus enabling the lowest temperature chain to escape from local energy basins with the help of high temperature chains, whose distributions are more flat.

Because a PT simulation yields data from all temperatures, it is natural to think of applying the method in Section 5.2. However, the exchange step that makes PT effective also introduces correlations between temperature trajectories. This violates the independence assumption in Section 5.2, so subsampling is needed to remove the correlation (Section 5.3.1). In Section 5.3.2, however, we show that there are situations in which it is justifiable to use this method to estimate the DoS from the full PT dataset.

5.3.1 Using a subsampled PT trajectory

To deal with the correlation introduced in PT, a reordering of temperature trajectories by so called replicas may be applied if we have recorded the history of temperature swaps. Here each replica contains blocks of configurations sampled at different temperatures and are nearly independent [12]. In doing so, the main contribution to the correlation now comes from within each replica, and this is the correlation that results from correlated sampling in MCMC simulations. Subsampling with a suitable statistical inefficiency $g > 1$ can then be applied to each replica to obtain effectively uncorrelated data. We point out that once g is known, we can use it to subsample the original temperature trajectory because it is equivalent to first subsampling the replicas and then permuting the subsampled replica back by temperature.

Instead of constructing multiple replicas, we use a subsampling strategy suggested by Chodera.[1] We construct a new time series defined by $u_t = \sum_{k=1}^{K} \beta_k U(\mathbf{x}_{kt})$, from each temperature trajectory in the PT simulation, and then use the statistical inefficiency of $\{u_t\}_{t=1}^{N}$ for subsampling. The rationale is that if the reduced potential of a single-temperature simulation provides a practical estimate for the relaxation time of the trajectory, as suggested by Shirts and Chodera [56], then an extension to the multi-temperature case should be given by the u_t defined above, where the exponential of $-u_t$ effectively gives the overall relative probability of observing a sample in the product state space of the parallel tempering simulation.

5.3.2 Using a full PT trajectory

We note that the estimating equations of MBAR can still be applied to correlated datasets, but the estimated uncertainties will no longer be valid [56]. Thus if one wishes to report statistical uncertainties of any MBAR estimator, subsampling is required whenever simulation data are correlated. However, there are reasons why using full PT samples may still be an option here. First, our DoS estimates are not used to produce thermodynamic properties; rather, they are the weights to be used in the subsequent MUCA simulation. Second, MBAR is not used as a final step to obtain free energy differences or other thermodynamic quantities; it is only

[1] Personal communication.

used as an intermediate step to estimate the log ratio of partition functions in (5.2). Evidently, the optimal combination of $\log \Omega$ estimators based on their variances will no longer apply if full PT samples are used. However, if we had subsampled data with long correlations, the resulting subsample size would be small and the uncertainty estimates may still be unreliable. This is because MBAR estimators are derived under the asymptotic limit, and so the estimated standard deviation will only reflect its true value when the sample size is large. Furthermore, a result from statistics [52] states that the variance of the Monte Carlo estimator for the expectation of some function of state, constructed from a subsampled Markov chain, is no smaller than the variance of the estimator constructed from the full Markov chain. An implication of this result is that variance of a full sample MBAR estimator may be smaller than the variance of a subsample MBAR estimator.

From the above observations, a simple alternative to using a subsampled PT trajectory, which also avoids complications associated with computing statistical inefficiency, is to feed into MBAR the full dataset from the PT simulation, ignore uncertainty estimates of the computed free energy differences, and average over the resulting estimates in (5.2). In the last step, if bin width ΔU is small, then we may only average over those $\log \hat{\Omega}_{km}$ whose corresponding entries in H_{km} are greater than some small integer, say 1 or 2, because the calculation is likely to be unreliable if there is only one observation in the bin.

5.4 Numerical Study

In this section, we present two examples where PT samples interesting regions of configuration space but suffers from low swap rate in the vicinity of a transition temperature. We apply the method in Section 5.2 to accurately estimate the density of states from the PT simulation and then run MUCA simulations with those estimated weights. We show that, for these two examples, the additional MUCA simulation is trivial in the sense that it will produce a more or less flat energy historgram sufficient for analysis, without needing to implement multicanonical recursions as in [4].

The first example (Section 5.4.1) is designed to mimic a phase transition by sampling from a mixture of two-dimensional normal distributions with suitably chosen parameters. The advantage of using a statistical model is that exact results can be obtained relatively easily through numerical integration, so that we know whether different simulation methods perform correctly. The second example (Section 5.4.2), inspired by reality, concerns the simulation of the aggregation of

Temperature	T_1		T_2		T_3		T_4
	0.4		0.5		2.0		3.0
Swap rates		0.29		0.01		0.28	

Table 5.1: Temperatures used and associated PT swap rates observed in the statistical model.

lattice-polymers in an implicit membrane and water environment. Since this model exhibits a phase transition between aggregated and dispersed states, the statistical model may be viewed as a simplified, low-dimensional analog of the physical model, with the benefit of knowing correct solutions.

To facilitate our study, we have used the `pymbar` module of Shirts and Chodera [56] for MBAR calculations. The PT and MUCA simulations were run with our own code.

5.4.1 A statistical example

Consider a mixture of bivariate normal distributions defined by

$$\pi \sim 0.5N(\boldsymbol{\mu}_1, \Sigma_1) + 0.5N(\boldsymbol{\mu}_2, \Sigma_2),$$

where the mean vector and covariance matrix of the two distributions are given by $\boldsymbol{\mu}_1 = [0, 0]^T,\ \Sigma_1 = \text{diag}[0.01, 0.01]$ and $\boldsymbol{\mu}_2 = [2, 2]^T,\ \Sigma_2 = \text{diag}[2, 2]$. We define the energy function to be $U(\mathbf{x}) = -\log \pi(\mathbf{x})$ and implement a PT sampler that samples from $\pi_T \propto \exp(-U(\mathbf{x})/T)$ with temperatures listed in Table 5.1. We then followed the procedure described in Sections 5.2–5.3 to estimate the density of states from the PT simulation. Specifically, we used both subsampled PT (Section 5.3.1) and full PT (Section 5.3.2) to obtain these estimates. Once this was done, MUCA simulations were run with weights proportional to the inverse of the estimated density of states. In this example, 10 independent MUCA simulations with different initial configurations randomly generated from π were run. We refer the approach that uses subsampled PT as PTMBARMUCA, and the approach that uses full PT as FPTMBARMUCA.

The parameters of π were chosen to mimic a broad high energy state and a narrow low energy state. A plot of the energy surface is shown in Figure 5.1. Because we are interested in quantities that vary with temperature, our goal is not just to sample from π, which corresponds to $T = 1$, but to ensure efficient crossing between the two states. This is different from common practice in statistics where one would use a temperature ladder $1 = T_1 < \ldots < T_K$ and only the lowest temperature

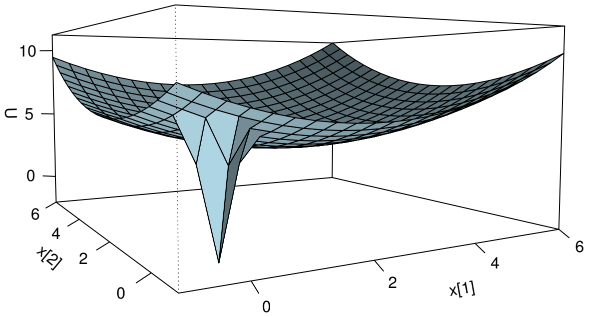


Figure 5.1: The energy surface in the statistical example: $U(\mathbf{x}) = -\log \pi(\mathbf{x})$.

distribution is of interest. Instead, in our setting the temperatures can be chosen as any positive values, and the relatively large difference between T_2 and T_3 was chosen to test if our method can sustain large gaps in the temperature ladder across a phase transition.

A trace plot of the last 2000 samples generated from all temperatures in the PT simulation is shown in Figure 5.2. It is clear that π_T becomes more localized as T is close to 0, with almost all probability mass concentrated at the first normal distribution in π, and it becomes flatter when T is large. The swap rate between T_2 and T_3 in the PT simulation was observed to be only 1% (Table 5.1 on page 53). Because the interval $[T_2, T_3]$ contains $T = 1$, which is when $\pi_T = \pi$, the observed frequency implies that transitions between the two modes in π are rare under the currrent parameter setting.

Two properties were calculated: the mean potential energy $\langle U(\mathbf{x})\rangle$ and heat capacity C_{v}. For a given inverse temperature $\beta = 1/T$,

$$\langle U(\mathbf{x})\rangle_T = \frac{-\int \pi(\mathbf{x})^\beta \log \pi(\mathbf{x}) d\mathbf{x}}{\int \pi(\mathbf{x})^\beta d\mathbf{x}},$$

and

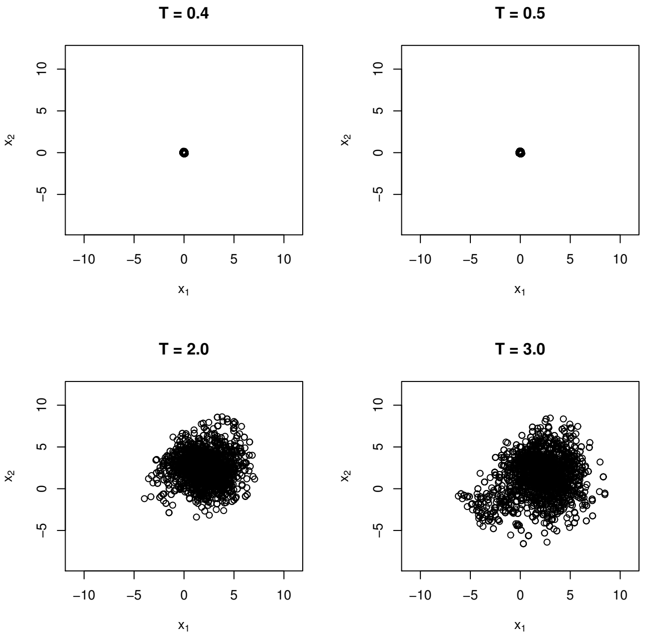


Figure 5.2: Trace plot of the last 2000 samples of the PT simulation for the statistical example.

$$(C_{\rm v})_T = \frac{\langle U^2(\mathbf{x})\rangle_T - \langle U(\mathbf{x})\rangle_T^2}{T^2}.$$

We note that both $\langle U(\mathbf{x})\rangle_T$ and $(C_{\rm v})_T$ can be calculated through numerical integration, and so correct results are known. The `R` package `cubature` was used to perform the integration. For all integration results, the estimated relative errors were of order 10^{-5}. In Figure 5.3, we show separately the estimated potential energy and heat capacity across a series of temperatures between $T_1 = 0.4$ and $T_4 = 3.0$ using both the subsampled and full PT trajectory, along with exact integration results. Clearly, the estimates show good agreement with the correct values of potential energy and heat capacity, whether or not subsampling is used.

For comparison, we also used MBAR to obtain directly the estimates as well as uncertainties of thermodynamic quantities of interest, which was the original purpose of MBAR when it was introduced in [56]. We note that although its use for PT simulations was not addressed there, a subsampling strategy as mentioned in Section 5.3.1 can be applied to obtain effectively uncorrelated data. This approach will be refered to as PTMBAR.

Since exact numerical integration results are known, a detailed error analysis can be conducted to investigate the quality of estimation for different methods. We

Temperature	T_1		T_2		T_3		T_4		T_5
	0.3		0.48		0.85		1.3		2.0
Swap rates		0.43		0.02		0.54		0.52	

Table 5.2: Temperatures used and associated PT swap rates observed in the lattice-polymer model.

used the Mean Squared Error (MSE) as a quality measure. The MSE of an estimator is defined as the average squared deviation between the estimator and its true value, and so takes into account both the variance and the bias of the estimator. To inspect the performance of the methods in different temperature ranges, we show pictorially the MSE of potential energy (Figure 5.4a) and heat capacity (Figure 5.4b) across all temperatures for all methods.

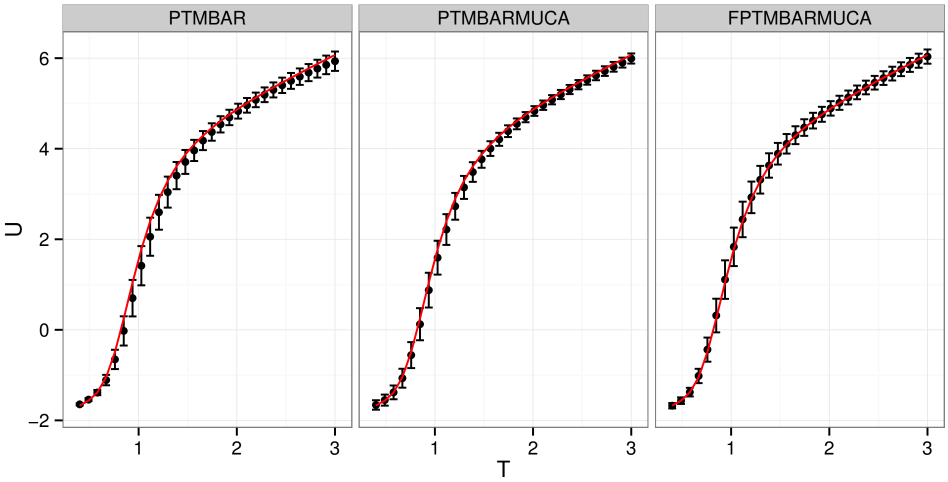
The MSE plots suggest that both PTMBARMUCA and FPTMBARMUCA have consistently smaller MSEs than PTMBAR across almost all temperatures, in particular, the MSEs are significantly smaller when T is around 1. In addition, larger MSE was observed in both energy and heat capacity estimates of PTMBAR when T is close to 3; as is also reflected in the PTMBAR plots in Figure 5.3, where relatively large error bars are observed near $T = 3$ and, in particular, deviations from the exact values of heat capacity exceed one standard deviation.

5.4.2 A lattice-polymer study

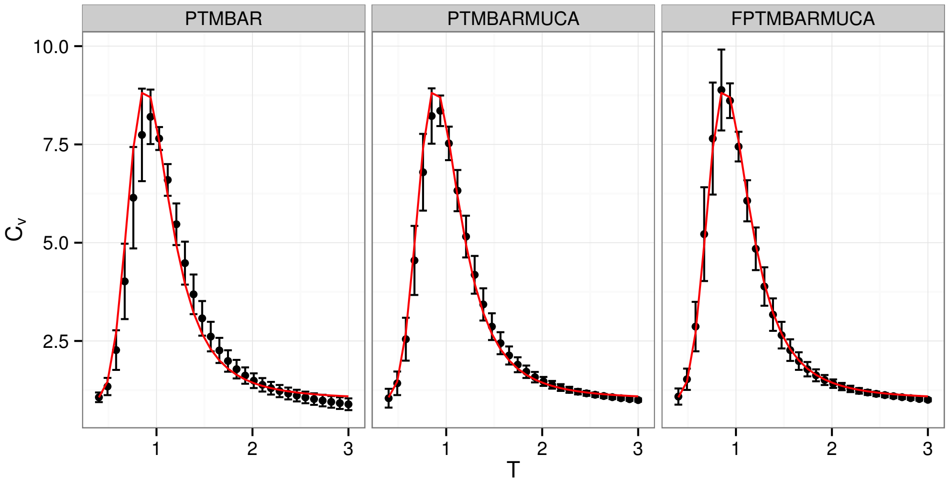
We now study the lattice-polymer model in Section 1.2. For the current method validation study we restrict our attention to a simpler lattice polymer which exhibits no secondary structure. A PT simulation was run with a total of 3×10^7 iterations, the temperature ladder and associated swap rates for this model are listed in Table 5.2.

In Section 5.4.1 we were mainly concerned with the correctness of the methods, and hence used a simple model which could be solved exactly. In this section we turn to the question of efficiency by using a model with more realistic complexity. As a result, while we continue to compare three approaches (PTMBAR, PTMBARMUCA and FPTMBARMUCA), we proceed as follows: we used all 3×10^7 iterations of PT for analysis with PTMBAR, whereas for the latter two approaches only the first 2×10^7 iterations were used to estimate the DoS, and this was then followed by an additional 1×10^7 iterations of MUCA. Hence the computational effort is roughly equal for all three methods.

In order to obtain error estimates, we ran 10 independent MUCA simulations using different initial configurations that belonged to both aggregated and dispersed

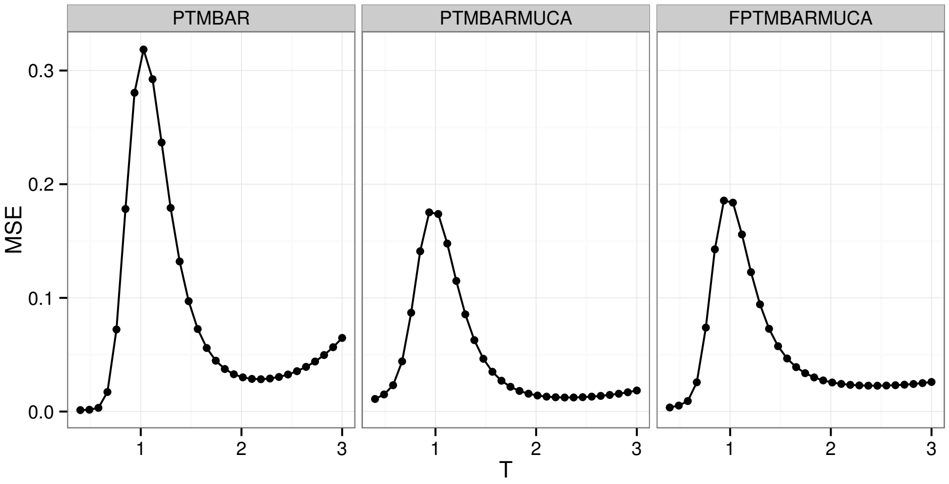


(a)

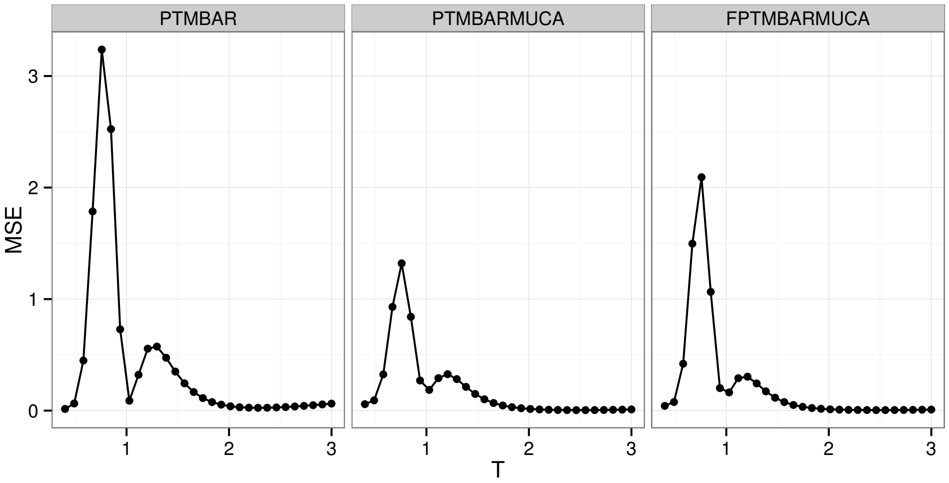


(b)

Figure 5.3: Estimated potential energy (Figure 5.3a) and heat capacity (Figure 5.3b) from three methods. The exact numerical integration results are shown in red curves. In FPTMBARMUCA, averages of $\log \hat{\Omega}_{km}$ for each bin m over all temperatures k with $H_{km} > 2$ were used. The error bar was calculated as follows: For PTMBAR, this was the analytical standard deviation of the MBAR estimator; For PTMBARMUCA and FPTMBARMUCA, this was the sample standard deviation of the estimates obtained from the 10 independent MUCA runs.



(a)



(b)

Figure 5.4: MSE of potential energy (Figure 5.4a) and heat capacity (Figure 5.4b) calculated from the corresponding plot in Figure 5.3. It can be seen that the MSEs of potential energy are comparable between both versions of our new method. The heat capacities show a difference in the peak near T = 0.75, with a smaller MSE for PTMBARMUCA than for FPTMBARMUCA; this must be due to smaller variance with PTMBARMUCA since Figure 5.3b shows that the bias is smaller in FPTMBARMUCA. Both methods have smaller MSE than PTMBAR.

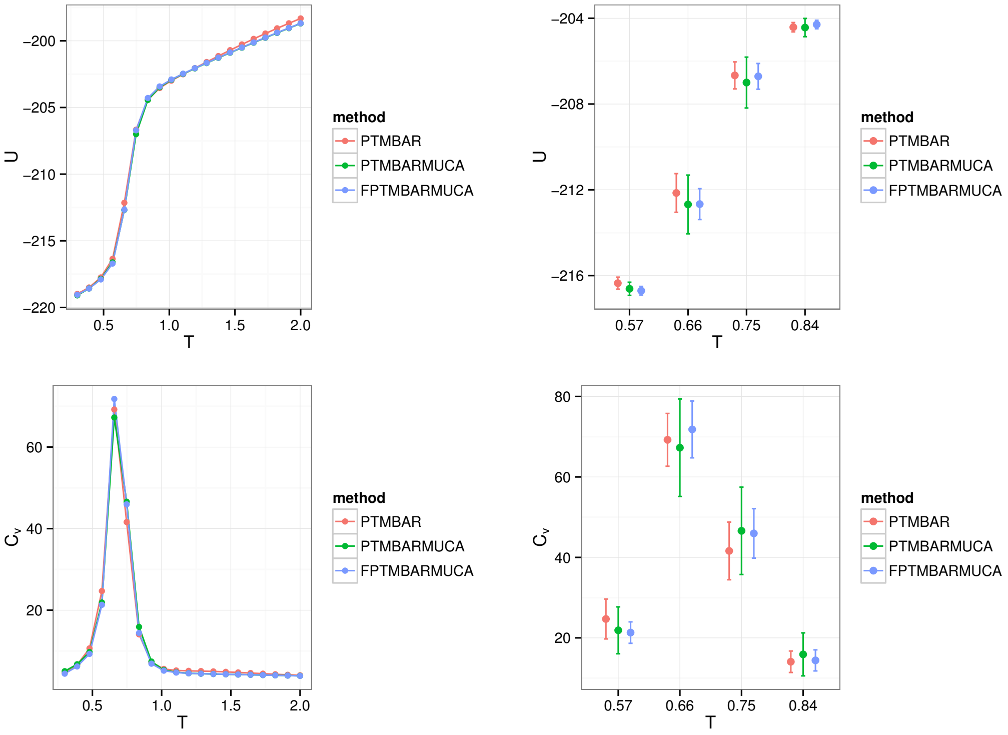


Figure 5.5: Estimated mean potential energy (top row) and heat capacity (bottom row) along with their uncertainties for the lattice polymer aggregation model. Results are overlayed. The methods with additional MUCA simulations only used half as many PT samples as the method of PTMBAR. In FPTMBARMUCA, an average of the equations in (5.2) over non-zero entries of H_{km} was used. The plots in the second column show the uncertainties (one standard deviation) of the corresponding property estimates on the left. The error bars were computed in exactly the same way as shown in Figure 5.3, and they are manually shifted left and right ($\pm\delta$) to avoid obstruction.

states. The results of potential energy and heat capacity calculated from the three methods are shown in Figure 5.5. Since exact values are not known *a priori* in this case, the results are overlayed to check for self-consistency. Uncertainty estimates are also compared for four temperature points which cover the peak region of the heat capacity curve, i.e. the transition between aggregated and dispersed states.

The first point to notice is that energy and heat capacity estimates agree very well for all three methods; in particular, the peak of the heat capacity occurs around $T = 0.66$, and differences between the mean values calculated by the methods are not statistically significant at any of the four temperature points considered. Note that these temperatures were purposely chosen around the transition temperature since property estimates at these temperatures are most likely to suffer from incomplete sampling of the two states. The uncertainty plots suggest that the method of PTMBARMUCA has larger statistical errors than PTMBAR; however, with FPTMBARMUCA, that is, with the DoS estimated using full PT trajectory, we obtain errors that are no worse, and often smaller, than PTMBAR.

Whether or not subsampling should be performed to derive the DoS appears to be affected by the strength of correlation in the PT trajectory. The statistical inefficiency in the lattice polymer model is about 12 times larger than that in the statistical model of Section 5.4.1. Although more iterations were used, the polymer model posed a more significant sampling problem because the dimension of the configuration space got much bigger. Hence, for PTMBARMUCA, the subsample size may still be inadequate for obtaining reliable estimates of the uncertainties of the terms in (5.2). On the other hand, taking averages over the estimates would seem to be a better option here, but would be too arbitrary in the statistical model.

Lastly, we mention that although we used multiple MUCA simulations and calculated standard deviation of the estimates obtained from each run, a resampling technique known as bootstrap [17] can be used to obtain error estimates and replace multiple runs which, however, do serve a useful purpose for convergence check. In bootstrap methods, one generates by sampling with replacement many sets of bootstrapped samples called *resamples* from the original data, and use the distribution of the resamples to approximate the distribution of the population. When we have dependent data, as is the case for most Monte Carlo simulations, the resamples need also preserve the dependence structure and the so called block bootstrap methods [51] can be used. We refer interested readers to [13] for a comprehensive acount of bootstrap methods.

5.5 Conclusion

The MBAR estimator exhibits superior statistical properties and has been widely used in free energy calculations invloving multiple equilibrium states. We proposed an approach that makes use of MBAR to calculate the density of states, and showed how this could be applied to data from parallel tempering simulations. Subsequent MUCA simulations which use this estimated density of states were shown to converge rapidly, without the need for multicanonical recursions. In this way, MBAR "optimally connects" PT and MUCA simulations and constitutes an important and integrated part of the simulation stage, rather than being confined to its more usual role as a post-simulation analysis tool. Our numerical study of a statistical model showed that the method was formally correct when compared with exact numerical integration results. We then used the method to study polymer aggregation in a lattice model and compared it with the traditional method of using MBAR to analyse simulation data. We observed that even when we applied our method to the first half of generated PT data, we were able to obtain comparable and even better results than the traditional method. Our results therefore suggests that it can be more beneficial and efficient to do analytical calculations, e.g. deriving MUCA weights through MBAR, than simply running longer Monte Carlo simulations.

When system size is large, it is useful to parallelize the MUCA step by dividing into overlapping intervals the sampled energy range of the PT simulation, perform MUCA simulations on each energy interval with weights already determined by the PT step, and join the resulting histograms on each interval. Adjacent energy intervals are chosen to overlap to reduce the "boundary effect" [62].

Because in our method MBAR is not used to produce final estimates, nor the associated uncertainties, of physical properties, there is some leeway in how it can be applied. In particular, it is possible to use full PT samples for MBAR DoS calculations, i.e. without first subsampling to remove the intrinsic correlations in the MC trajectory. This aspect was explored in both of our examples. Clearly, it is natural to subsample the data because we can then properly combine the estimators of the log density of states. However, the optimality of such estimators decreases as subsample size shrinks, and hence if correlation is long, the full sample strategy of Section 5.3.2 may be preferred. The conventional usage of MBAR to report statistical uncertainties would preclude such possibility.

Chapter 6

Application to the Twin-Arginine Translocation mechanism

> Essentially, all models are wrong,
> but some are useful.
>
> George E. P. Box

In this chapter, results of applying our integrated, MBAR-enhanced MC approach to some more realistic models will be presented. Such models could be considered as our first steps towards a better understanding of TatA aggregation in membrane, which is an integral part in the Tat mechanism. In Section 6.1, we give an overview of the Tat pathway and address the significance of the computer simulation study reported in this chapter. Specifically, attempts were made to investigate the role of the helix components in the TatA molecule (Section 6.2), and to study the case where there are more than two polymers (Section 6.3). Problems with simulation efficiency when using the current force field and move set are identified, and strategies that can help alleviate such problems are proposed for further study. We note that even with these difficulties, we were still able to address polymers with some elements of secondary structure using the MBAR-enhanced MC approach.

6.1 The Tat pathway

The Twin-arginine translocation (Tat) pathway is one of two major pathways cells have for transporting proteins across membranes. It is involved in the export of proteins across bacterial cytoplasmic membranes and across the thylakoid membranes in plant chloroplasts, and is essential for bacterial pathogenesis and for plant photosynthesis [36]. The translocated proteins are referred to as substrates; these are proteins that need to be transported to perform their functions either within the cell or in extracellular space. One distinctive feature of the Tat mechanism is that substrate proteins are transported in a folded manner, contrary to the general secretory (Sec) pathway which transports proteins in an unfolded state [36]. The name *Tat* is an acronym for "twin-arginine translocation" and comes from the unique, consensus twin-arginine (RR) motif that is a key feature of the amino acid sequence of the signal peptide that triggers Tat translocation. Major components of the Tat translocon are membrane proteins from the TatABC family; these are small integral membrane proteins that, when forming complexes that have the right structural organization, allow the folded substrate to be translocated without compromising the permeability of the lipid bilayer. TatA consists of a single transmembrane helix (TMH) and an amphipathic helix (APH) that lies along the membrane surface. The two helices form approximately a right angle and are connected by a small loop. A schematic representation of the TatA protein is shown in Figure 6.1. TatC consists of six TMHs and has limited conformational flexibility [7]. TatB has similar structure to TatA and the two are best discriminated by their biochemical behaviour: whereas TatA proteins oligomerize to form the translocation channel, TatB proteins form a 1 : 1 complex with TatC and play a role in substrate recognition prior to the transient translocation process [7].

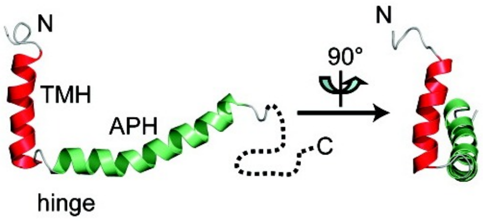


Figure 6.1: Ribbon representation of TatA protein, details of the composition of amino acids are omitted. The TatA molecule consists of a transmembrane helix (red) near the N-terminal tail, a hinge region, an amphipathic helix (green) at the membrane-water interface and an unstructured C-terminal tail (dashed line). The structure has been determined in atomic resolution by NMR spectroscopy [29]. (Reprinted with permission from [29]. Copyright (2010) American Chemical Society)

A diagram illustrating the potential key steps involved in the Tat pathway is shown in Figure 6.2. It should be noted that although a TatA oligomer is depicted in

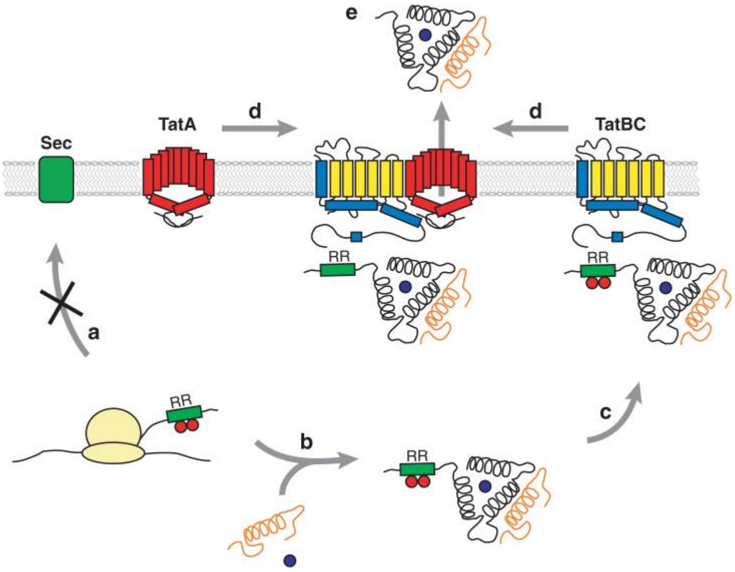


Figure 6.2: Diagram showing Tat targeting and translocation. The order of events are labelled from (a) to (e). The bottom left corner shows that while the substrate is being synthesised from the ribosome, the RR signal peptide is inserted. The insertion of signal peptide and/or the binding of additional helper proteins (molecular chaperones, shown in red circles) prevent the substrate from targeting the Sec pathway [31] (a). After folding, additional subunits (orange) and cofactor (blue circle) are added (b) and the substrate is recognized by the TatBC complex through the signal peptide (c). This then appears to trigger the aggregation of TatA and a TatABC complex, which serves as the active translocation site, is formed (d). The substrate protein is then translocated through a pore constituting TatA proteins (e). Once transport is completed, the signal peptide is removed from the substrate. (figure courtesy of [36])

the figure, the oligomer dissociates and returns to dispersed state once translocation is completed, since a persistent oligomer would have undesirable consequences for the cell, such as ion leakage [1]. Also, experimental results show that the TatA channel can vary its diameter to accommodate substrates of different sizes [23], implying that translocation is mediated by oligomerization of variable amounts of TatA monomer.

The fact that Tat is able to transport folded proteins makes it particularly challenging compared to Sec pathway, because membrane must maintain a permeability barrier to ions and small molecules during transport. As an example, the *Escherichia. coli* Tat pathway is able to transport substrates of up to 70 Å in diameter, whereas an unfolded polypeptide chain is only about 12 Å in diameter [8]. Due to this distinctive feature that the Tat mechanism possesses, models of the dynamics of the translocation process have been proposed (see [50] for a review),

	TMH	APH
case 1	0	0
case 2	1	0
case 3	0	1
case 4	1	1

Table 6.1: There are four possible cases concerning helices within our TatA model, as listed here, where 1/0 indicates on/off of the helix. Computationally, these can be achieved by manipulating relevant parameters in the force field (see Appendix A).

including one that predicts a local weakening of the membrane that is sufficient for the substrate to move through [9].

Clearly, the structure of the TatA oligomer is crucial to understanding the Tat mechanism. However, due to the transient nature of active translocation complex, it is difficult to conduct experimental analyses and thereby establish its structural organization during translocation. Computer simulation can be an indispensable complementary tool in elucidating the Tat mechanism. Indeed, molecular dynamics simulations have been applied to investigate the stability of experimentally proposed solubilized TatA oligomer structure in membrane [53]; however it is unclear whether TatA assembly in the native membrane environment will result in the same oligomeric structure as determined in a detergent solution.

Our lattice models, on the other hand, allow us to simulate the assembly process in an (implicit) membrane environment and explore its equilibrium properties. It is realized that once the equilibrium behaviour of TatA aggregation is adequately explained by our simulation, we can then model how the TatBC complex and the substrate interact with, and influence, the aggregation of TatA within the Tat process.

6.2 Helices and TatA aggregation

Among the interesting questions concerning TatA aggregation are whether, and how, the aggregation is affected by the secondary structure of TatA. Our lattice TatA model (Figure 6.3) provides us with a way to look at this problem by performing simulations with different combinations of the TatA secondary structure (Table 6.1).

We shall consider the four cases listed in Table 6.1 separately, and concentrate on two-chain systems in this section. For each case, the same amino acid sequence was used. The model and move set described in sections 1.2 and 1.3 were used, and the force field parameters are listed in Appendix A. For clarity, we reiterate

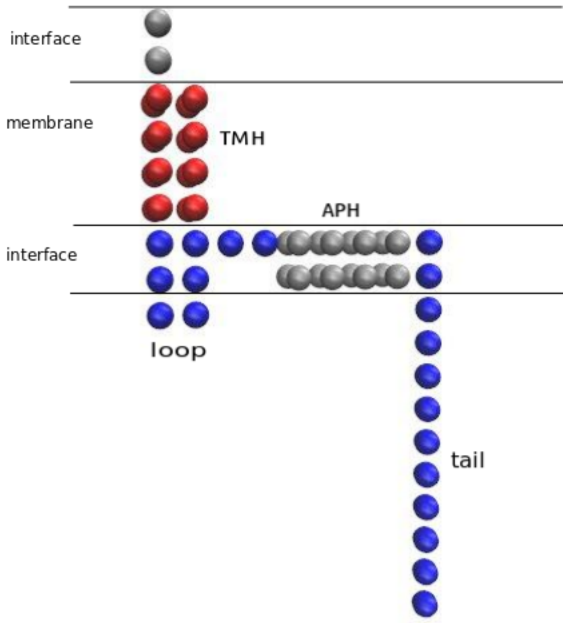


Figure 6.3: Lattice model of TatA. Four regions are identified: transmembrane helix (TMH), loop, amphipathic helix (APH) and the hydrophilic tail. Boundaries of membrane and interface are also shown.

that we are using an adaptation of the three-dimensional HP model with 1) implicit membrane and interface, 2) amphipathic bead type (H2) to supplement hydrophobic (H) and polar (P) beads, and 3) interactions that favour helical contacts where relevant. Note that even in cases where at least one of the helices is absent, the membrane-water interfaces were still kept for consistency.

6.2.1 Case 1: both TMH and APH are absent

Here, and in other cases, we monitor six observables of interest: the heat capacity (C_v), the inter-polymer contribution to total potential energy (U_{inter}), the number of inter-polymer tail-tail contacts (N_{tt}), the number of inter-polymer tail-loop contacts (N_{tl}), the number of inter-polymer contacts ($N_{\text{inter}} = N_{\text{tt}} + N_{\text{tl}}$), and the number of intra-polymer P-P contacts per chain (N_{intra}). Because of the way the force field is defined, the P-P contacts in N_{intra} must happen within the membrane, so there is a competition within the membrane between intra- and inter-polymer P-P contacts.

A parallel tempering simulation was carried out first, followed by MBAR calculations to obtain optimal density of states estimates, which were then fed into multicanonical simulations. Specifically, we used full PT dataset for MBAR density of states calculation, i.e. the FPTMBARMUCA approach in Section 5.3.2 was used. More simulation details are listed in Appendix C. The results for this case are shown in Figure 6.4.

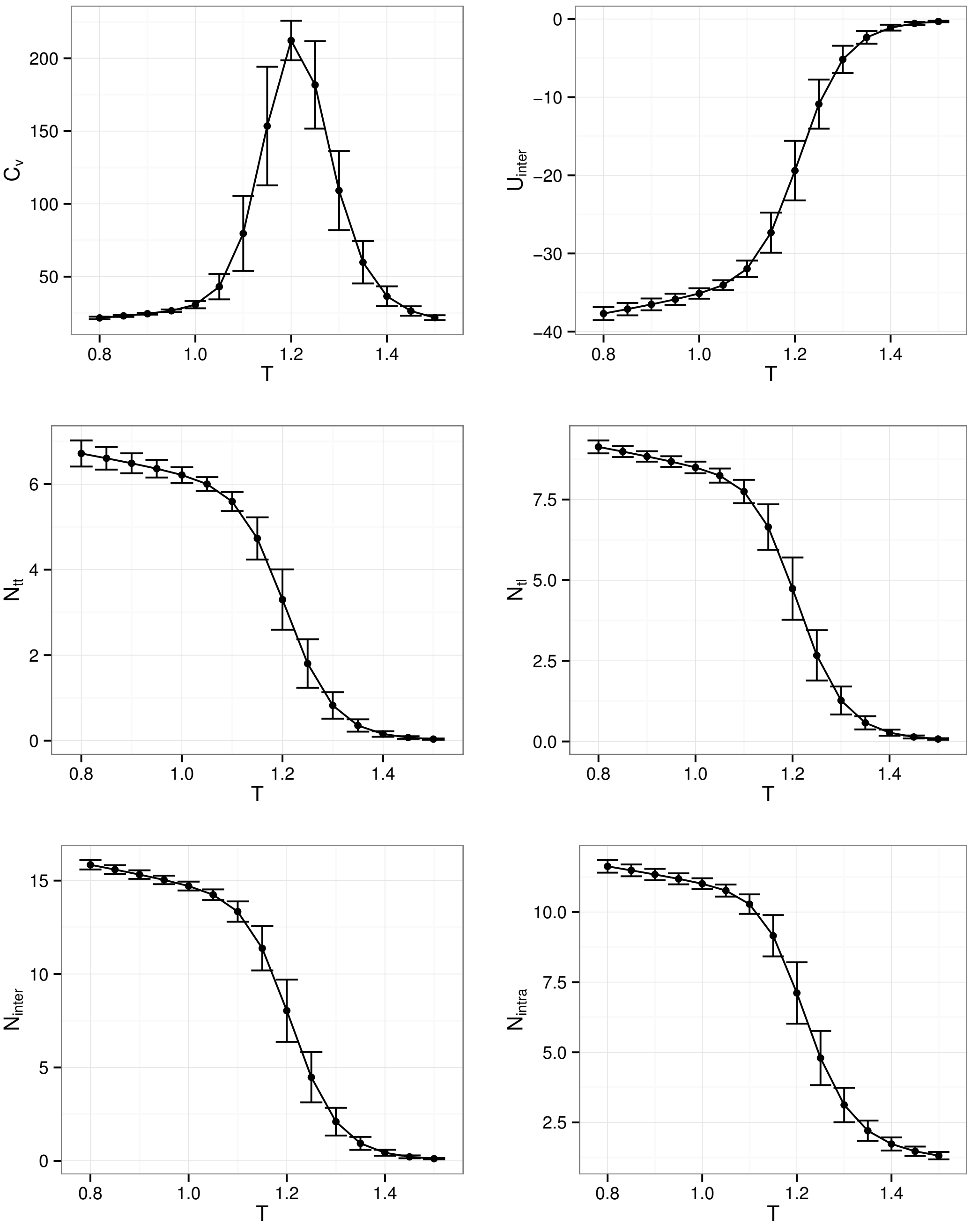
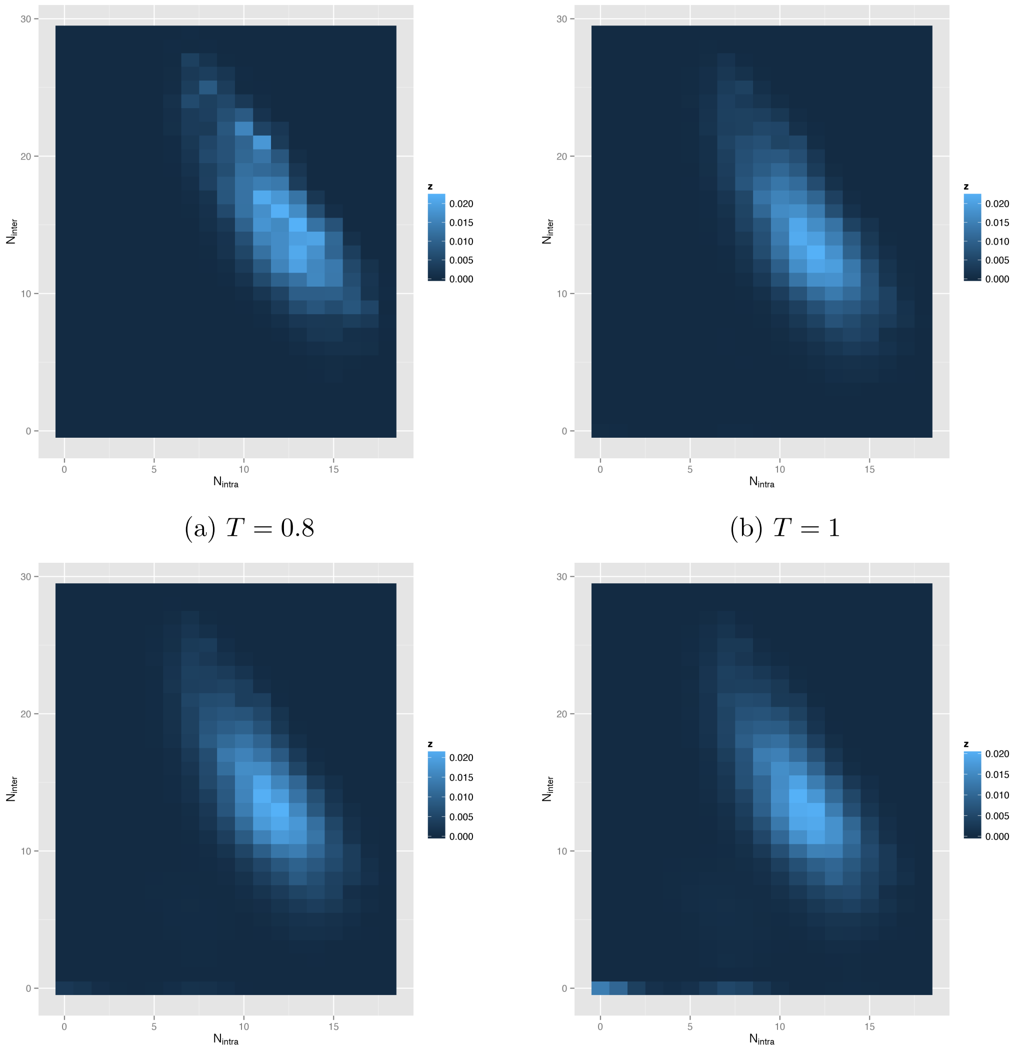


Figure 6.4: Property estimates for case 1 (no helices). The six properties are the heat capacity (C_v), the inter-polymer contribution to total potential energy (U_{inter}), the number of inter-polymer tail-tail contacts (N_{tt}), the number of inter-polymer tail-loop contacts (N_{tl}), the number of inter-polymer contacts (N_{inter}) and the number of intra-polymer P-P contacts per chain (N_{intra}). The error bar was calculated as one standard deviation of the estimates over 10 independent MUCA runs, each of which has starting configuration chosen from either aggregated or dispersed state.

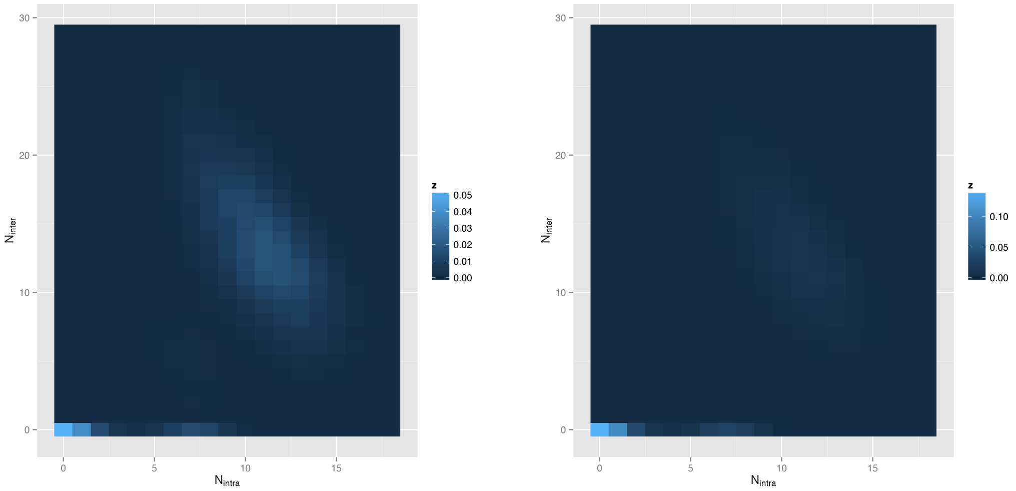


(a) $T = 0.8$

(b) $T = 1$

(c) $T = 1.05$

(d) $T = 1.1$



(e) $T = 1.15$

(f) $T = 1.2$

Figure 6.5: Case 1 (no helices) level plots (with rectangular tiling) showing the probability of (N_{intra}, N_{inter}) taking different values in a grid, at different temperatures. The brighter the tile, the higher its probability.

The heat capacity shows two distinct states with a transition temperature centred at $T = 1.2$, but spanning 1.0 to 1.4. Also, the inter-polymer energy tends towards 0 as T increases, with almost no interactions between the two polymers when $T > 1.4$. Hence we see that dispersed state dominates high temperatures ($T > 1.2$) and the dimer state dominates low temperatures ($T < 1.2$). All other quantities, i.e. the various contacts, show the same trend as the temperature varies: the number of contacts decreases as T increases. Now, if we focus on a particular temperature in the range $T < 1.2$, we can see that N_{tt} and N_{tl} are negatively correlated, meaning that if we have a bunch of samples $(N_{\text{tt}}, N_{\text{tl}})$ *at a particular temperature*, it would then be the case that a larger N_{tt} implies a smaller N_{tl}, and vice versa. In a statistical sense, this means that the variance of N_{inter}, i.e. the sum of N_{tt} and N_{tl}, is smaller than the sum of the variances of N_{tt} and N_{tl}. As the error bars shown in Figure 6.4 are one standard deviation, N_{inter} should have error bars that are no smaller than either of N_{tt} and N_{tl} if the two were uncorrelated; but clearly, this is not the case when T is low. For example, at $T = 0.8$, the error bar in N_{inter} is smaller than both of the error bars in N_{tt} and N_{tl}. Hence, we see that the chains "trade" tail-tail contacts for tail-loop contacts when they form a dimer.

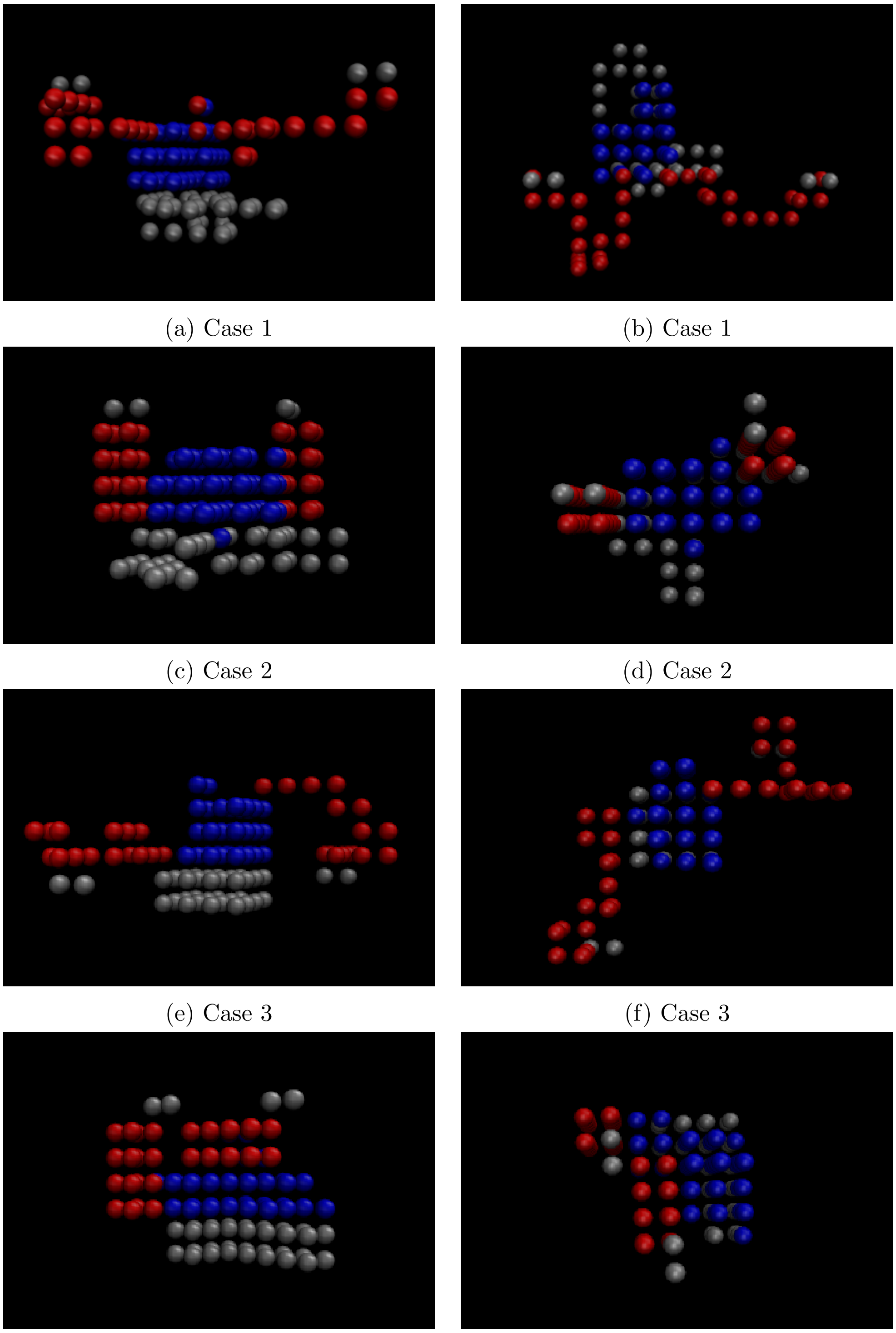
In addition, comparing N_{inter} with N_{intra}, we see that when $T < 1.2$, more inter-polymer contacts were observed on average than the number of intra-polymer contacts. In particular, at $T = 0.8$, N_{inter} is about 5 contacts larger than N_{intra}.

To illustrate how the distributions of both N_{inter} and N_{intra} change with temperature, we took one of the MUCA simulations and calculated the probability of the pair (N_{intra}, N_{inter}) taking on various values across the range of temperatures. The results are shown in level plots (Figure 6.5), where the "z-value" is the probability. We can immediately identify regions of high probability from these plots. For example, Figure 6.5a shows the distribution of (N_{intra}, N_{inter}) at the lowest temperature $T = 0.8$, and we find that the three most populated states are centred on (13, 15), (12, 16) and (11, 17). From Figure 6.5d, we can see a clear presence of N_{inter} being both zero and positive, indicating coexistence of both dispersed and dimer states. And, at $T = 1.05$ (Figure 6.5c), although we can still discern regions where $N_{\text{inter}} = 0$ has positive probability, the dimer state dominates; similarly, the monomer state dominates at $T = 1.2$ (Figure 6.5f).

Typical dimer snapshots for this and other three cases are shown in Figure 6.6. We also observed that for Cases 2 and 4, at least 94.7% of transmembrane helical contacts were maintained in the simulated temperature range; for Cases 3 and 4, at least 96.3% of amphipathic helical contacts were maintained; for Cases 1 and 3, at least 99.5% of hydrophobic (H) beads were within the membrane region and for Cases 1 and 2, at least 97.3% of amphipathic (H2) beads were within the interface region.

6.2.2 Case 2: only TMH is present

Now we switch to Case 2, where the transmembrane helix is present but the amphipathic helix is not. The respective plots are shown in Figure 6.7 and Figure 6.8. From Figure 6.7, we see that observations we made about the various contacts in Case 1 hold also in Case 2. The transition temperature is around $T = 1.15$, lower than that observed in case 1, suggesting that dimerization occurs at a slightly lower temperature when the transmembrane helix is present. The probability distributions of (N_{intra}, N_{inter}) at different temperatures (Figure 6.8) show that the dimer state is substantially populated when $T < 1.05$. The three most frequent combinations of the pair at $T = 0.8$ (Figure 6.8a) are (10, 17), (12, 15) and (10, 18).



(a) Case 1

(b) Case 1

(c) Case 2

(d) Case 2

(e) Case 3

(f) Case 3

(g) Case 4

(h) Case 4

Figure 6.6: Dimer snapshots for all cases. Side views are shown on the left and top-down views are shown on the right. Membrane and interface are not shown.

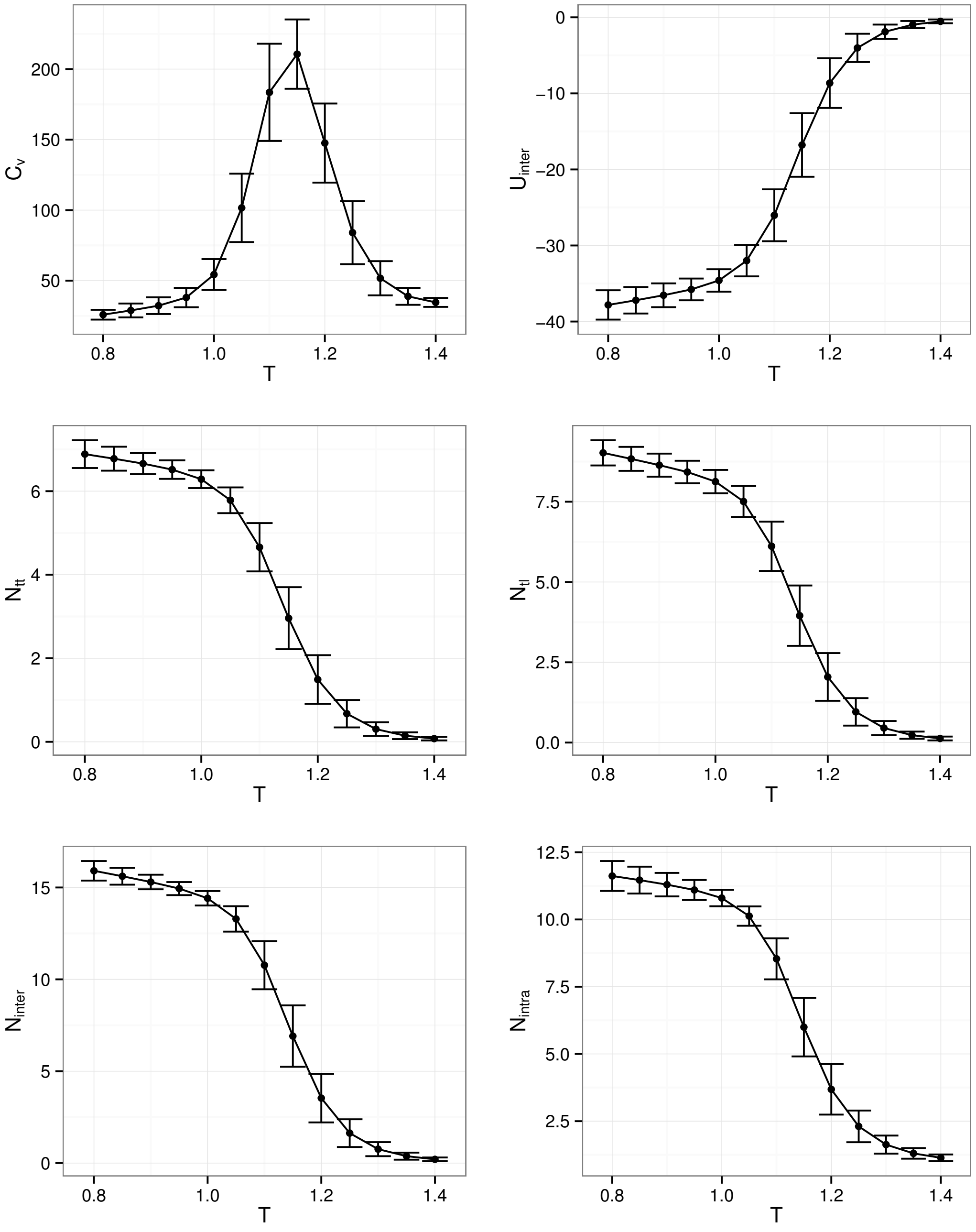
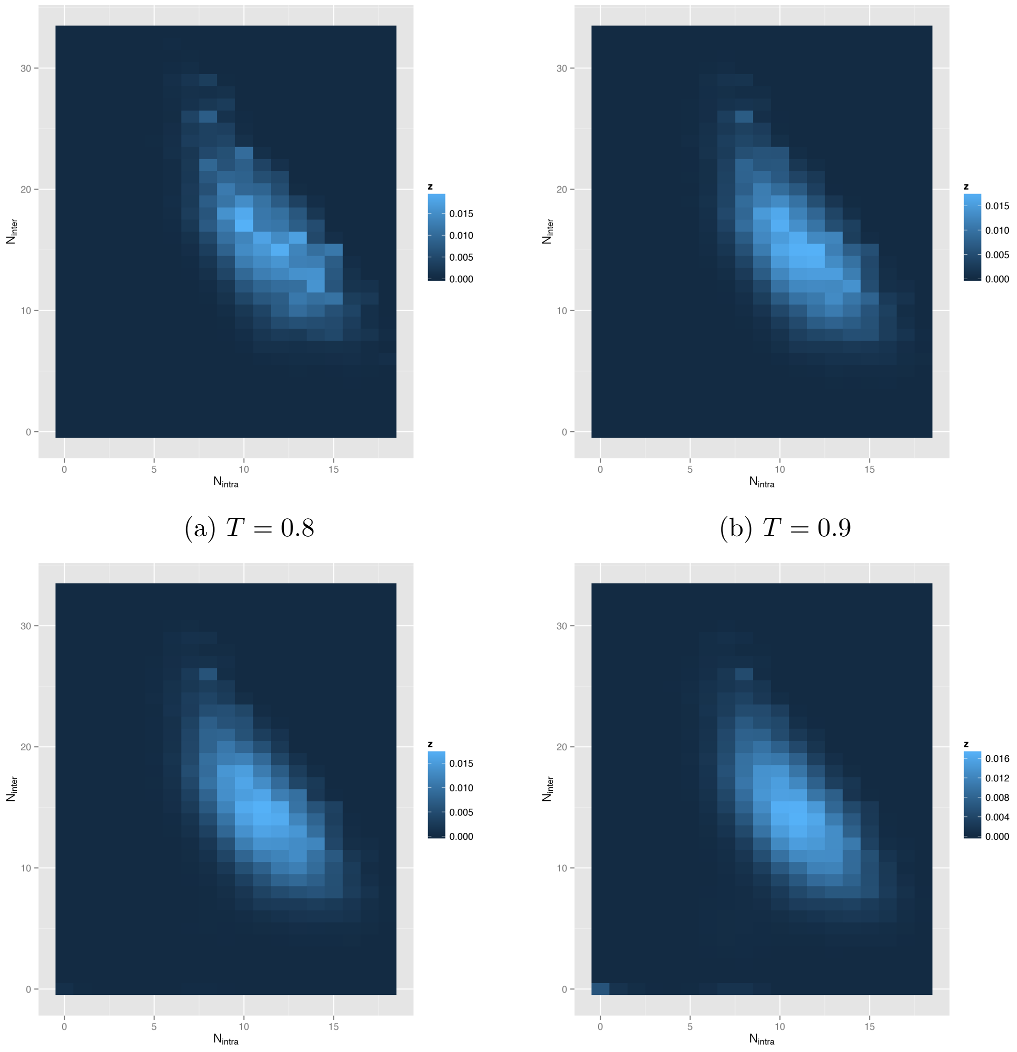


Figure 6.7: Property estimates for case 2 (only TMH).

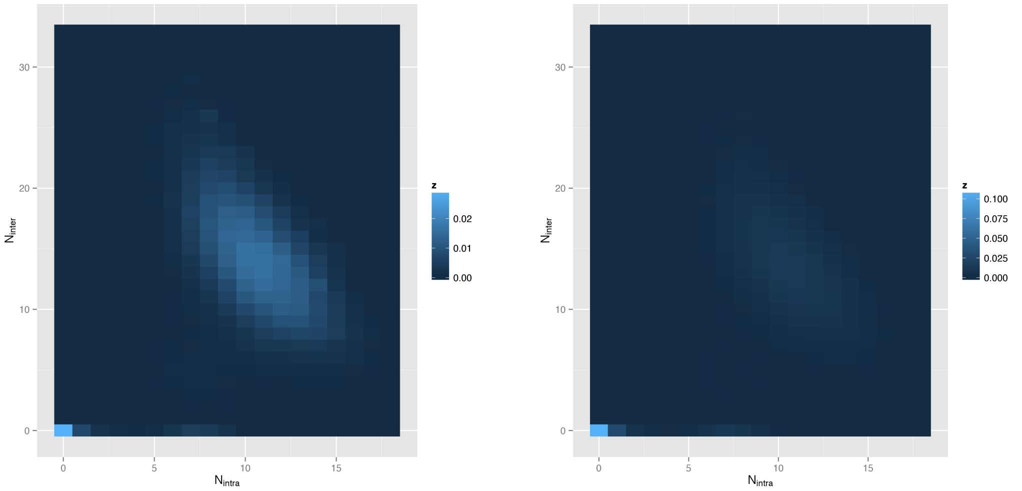


(a) $T = 0.8$

(b) $T = 0.9$

(c) $T = 0.95$

(d) $T = 1$



(e) $T = 1.05$

(f) $T = 1.1$

Figure 6.8: Case 2 (only TMH) level plots, showing the probability of (N_{intra}, N_{inter}) taking different values in a grid, at different temperatures.

6.2.3 Case 3: only APH is present

In this case the results for the estimated properties in the simulated temperature range are shown in Figure 6.9 and the probability distributions of (N_{intra}, N_{inter}) at various temperatures are shown in Figure 6.10. Since we did not know the approximate location of the transition temperature beforehand, the PT temperature ladder was chosen similar to the first two cases. It is notable that the heat capacity does not reflect the complete transition given the temperature ladder used, as the transition to the dispersed state is partly missing in the plot. Nevertheless, it can still be seen that the transition temperature is around $T = 1.5$, significantly higher than those observed in Cases 1 and 2. At $T = 1.6$, while the population is dominated by the monomer state, the average inter-polymer energy is negative and the average number of inter-polymer contact is positive, suggesting that dimers are still present at this temperature. Larger error bars were observed in tail-tail and tail-loop contacts, N_{tt} and N_{tl}, when $T < 1.3$; in contrast, the corresponding error bars in the sum N_{inter} were much smaller, this observation implies a strong negative correlation between N_{tt} and N_{tl}. The probability distributions of (N_{intra}, N_{inter}) show that the dimer state is substantially populated when $T < 1.35$. The three most frequent combinations of the pair at $T = 0.9$ (Figure 6.10a) are (12, 15), (11, 16) and (11, 17).

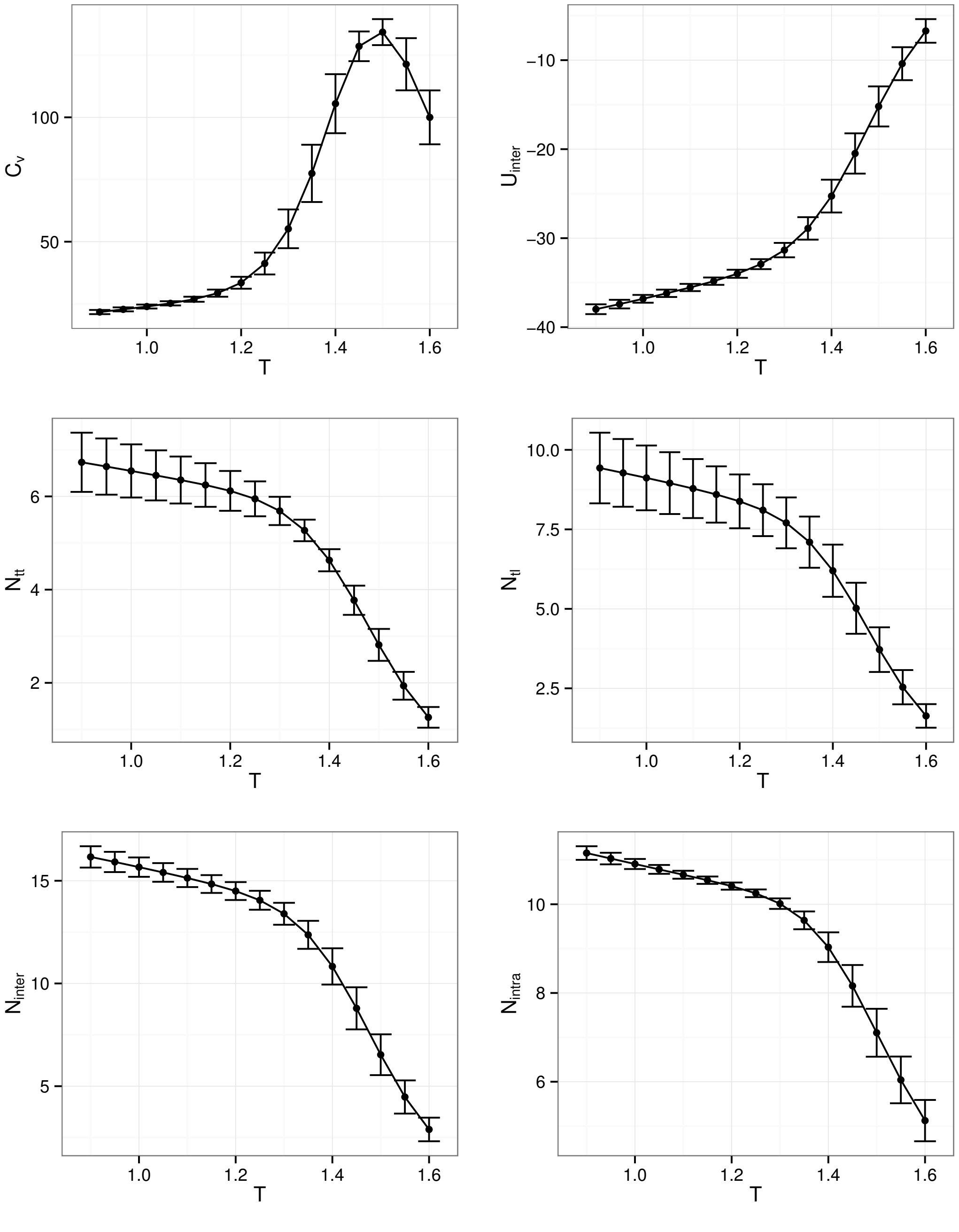
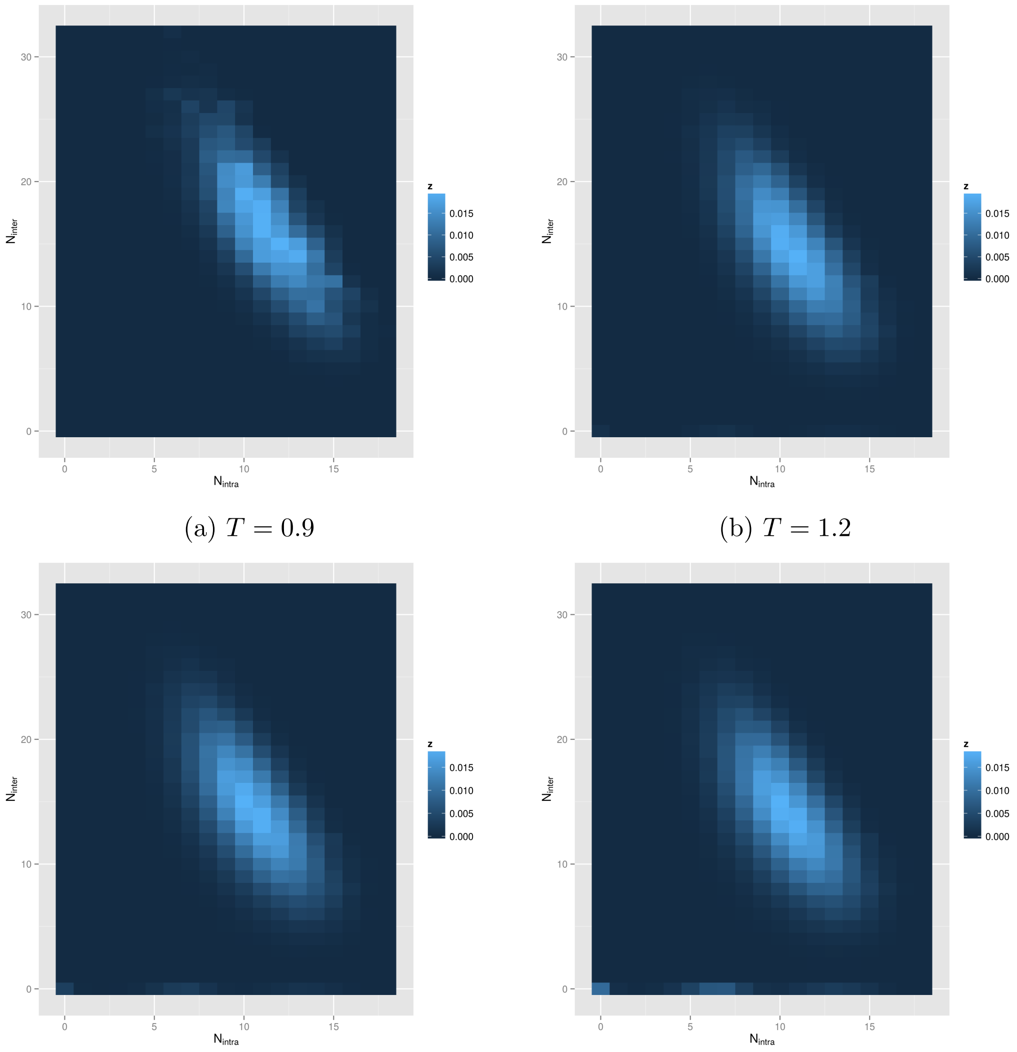


Figure 6.9: Property estimates for case 3 (only APH).

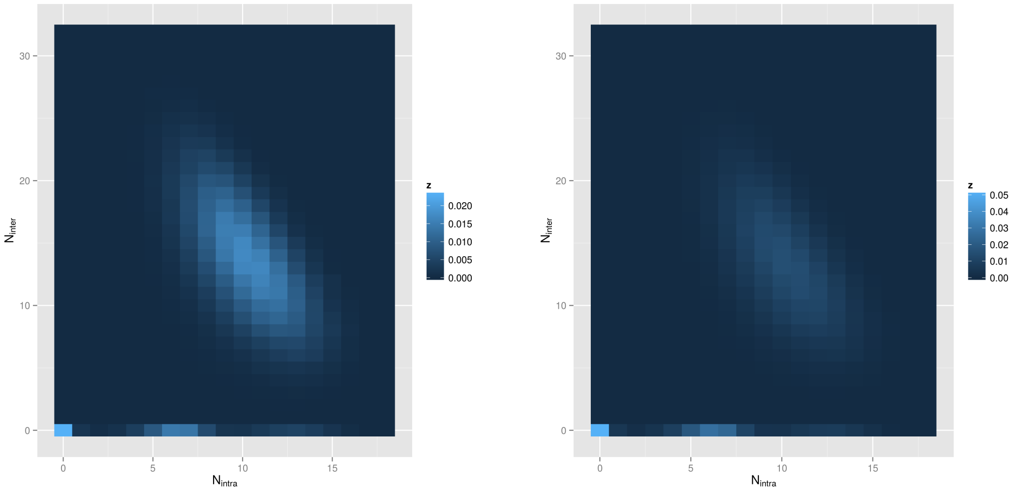


(a) $T = 0.9$

(b) $T = 1.2$

(c) $T = 1.25$

(d) $T = 1.3$



(e) $T = 1.35$

(f) $T = 1.4$

Figure 6.10: Case 3 (only APH) level plots, showing the probability of (N_{intra}, N_{inter}) taking different values in a grid, at different temperatures.

6.2.4 Case 4: both TMH and APH are present

Similar to Case 3, the transition to dispersed state is partly missing as shown in the heat capacity plot in Figure 6.11. However, a lower transition temperature (near $T = 1.35$) was observed compared with Case 3. The quantities N_{tt} and N_{tl} are still negatively correlated, although weaker than the negative correlation observed in Case 3. At $T = 1.4$, there are about 6 inter-polymer contacts; whereas at the same temperature in Case 3, there are still more than 10 inter-polymer contacts, and there is almost no inter-polymer contacts at this temperature in Cases 1 and 2. The probability distributions of (N_{intra}, N_{inter}) (Figure 6.12) show that the dimer state is substantially populated when $T < 1.25$. The three most frequent combinations of the pair at $T = 1.1$ (Figure 6.12a) are (9, 24), (9, 23) and (10, 21), showing that more inter-polymer, and less intra-polymer, contacts were observed than the corresponding contacts in the first three cases.

Recapitulating all four cases, our results suggest that the amphipathic helix tends to favour dimerization, since a higher temperature is needed to disrupt it; and while the transmembrane helix hinders dimerization (lower transition temperature T_{trans}), the amphipathic helix appears to be the stronger effect, giving a bigger shift in T_{trans} in Case 3 compared with Case 2, and resulting in an increase in T_{trans} when both helices are present.

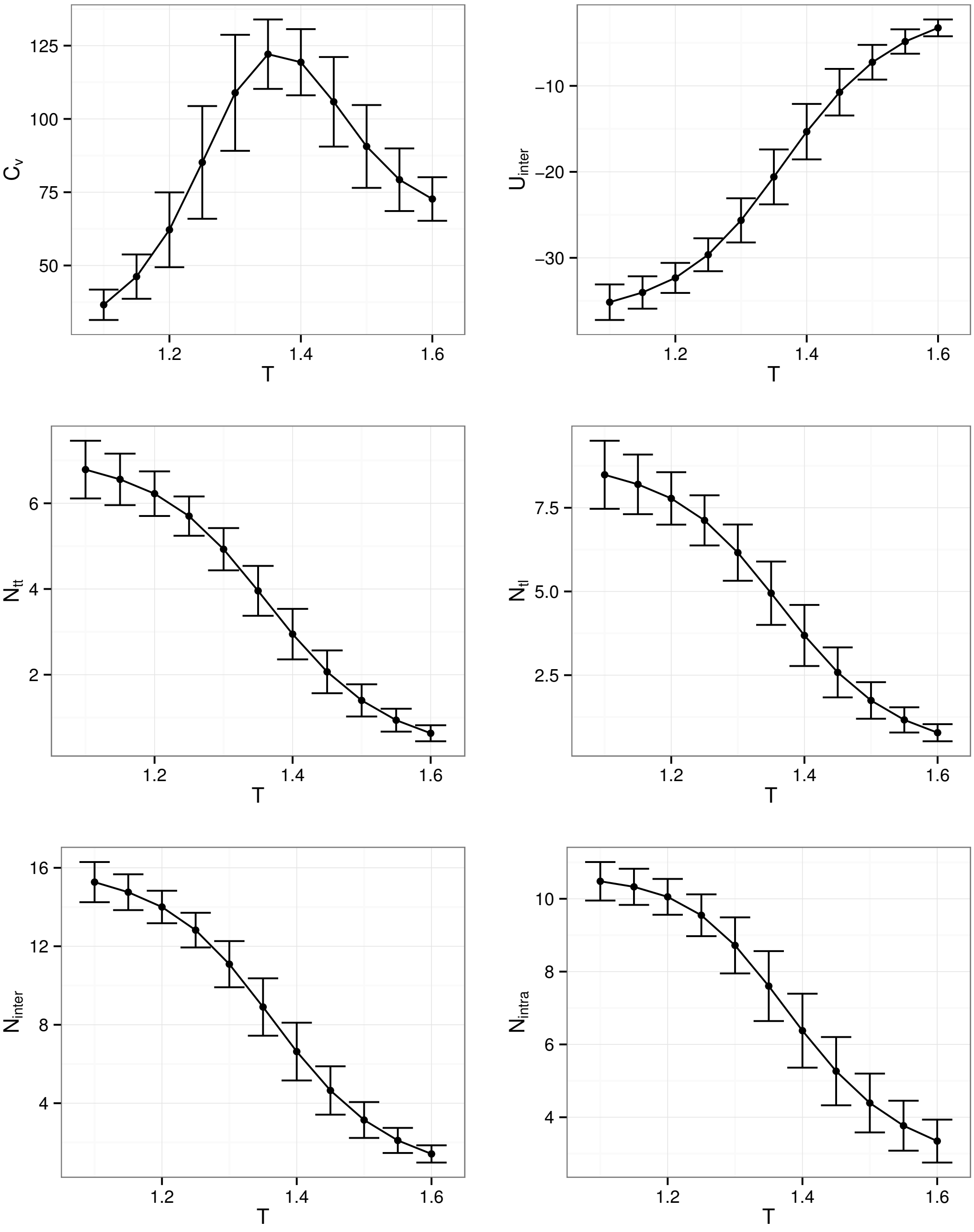
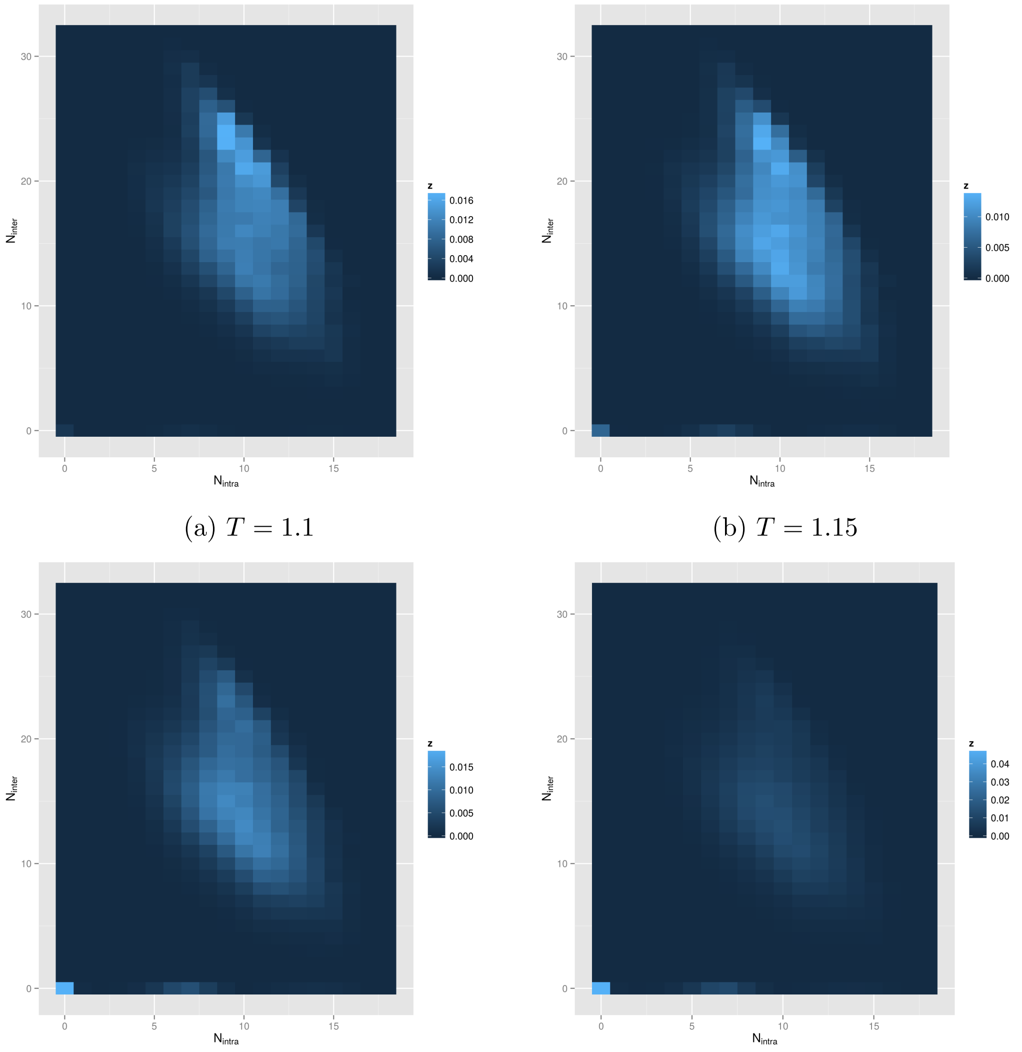


Figure 6.11: Property estimates for case 4 (both TMH and APH).



(a) $T = 1.1$

(b) $T = 1.15$

(c) $T = 1.2$

(d) $T = 1.25$

Figure 6.12: Case 4 (both TMH and APH) level plots, showing the probability of (N_{intra}, N_{inter}) taking different values in a grid, at different temperatures.

6.3 Aggregation of more than two polymers

So far we have been considering only two polymers in our system. Like we mentioned before, the TatA transport channel can adopt variable sizes during substrate translocation, hence it is interesting as well as desirable to study properties of multi-chain systems. As a straightforward extension to Section 6.2, we double the number of polymers while keeping the concentration fixed and consider in this section the aggregation of four identical polymer chains in membrane.

As can be imagined, we now have a system that is more complex than the two-chain models, because we could have not just fully aggregated (tetramer) and fully dispersed states, but also partially aggregated states in which only two or three chains associate, as illustrated in Figure 6.13.

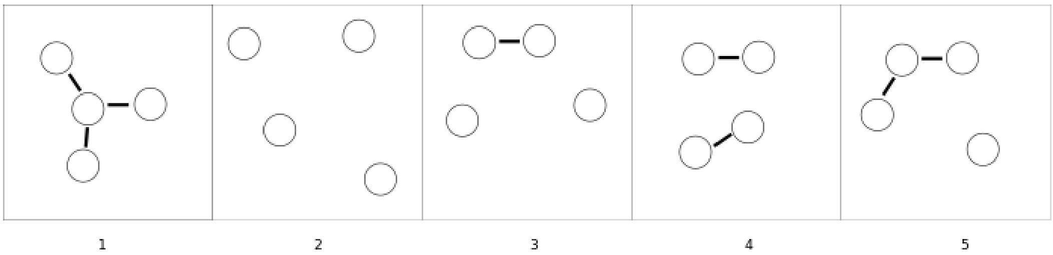
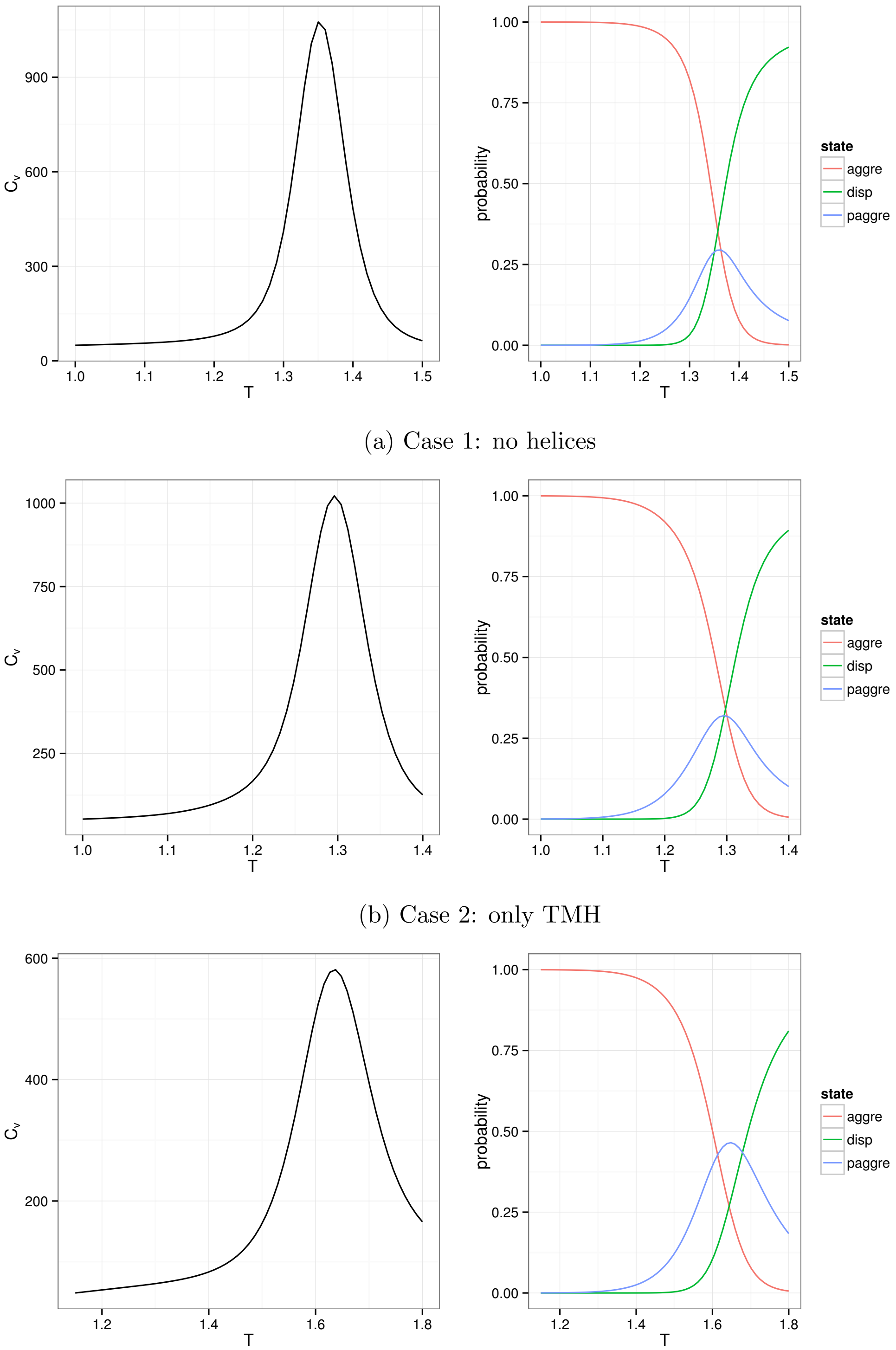


Figure 6.13: A schematic drawing showing possible aggregation states in a four-chain system—1: tetramer; 2: dispersed; 3-5: partially aggregated. Here each circle represents a polymer, and two polymers interact if they are connected by an edge. In our analysis we do not differentiate between states 3–5, and simply refer to them collectively as the partially aggregated state.

In this section, we again study different combinations of TatA secondary structure. In particular, Cases 1–3 of Section 6.2 will be considered, while Case 4 is best studied when some more efficient trial move is used in place of, or in conjunction with, pull moves (see discussion below). The MBAR-enhanced Monte Carlo was again used and a MUCA simulation which sampled sufficiently frequently all relevant energy regions was chosen for analysis. In Figure 6.14, we plot for all three cases the heat capacity and the fraction of each state (tetramer, dispersed or partially aggregated) across temperatures.

Notice first that a single peak is observed in heat capacity, suggesting that the system is still a two-state system, even though partially aggregated state can occur. The transition temperatures for Cases 1–3 are $T = 1.355$, $T = 1.296$ and $T = 1.638$, respectively; this result is consistent with the conclusions we made on the two-chain models in Section 6.2, namely, that the amphipathic helix (APH)



(a) Case 1: no helices

(b) Case 2: only TMH

(c) Case 3: only APH

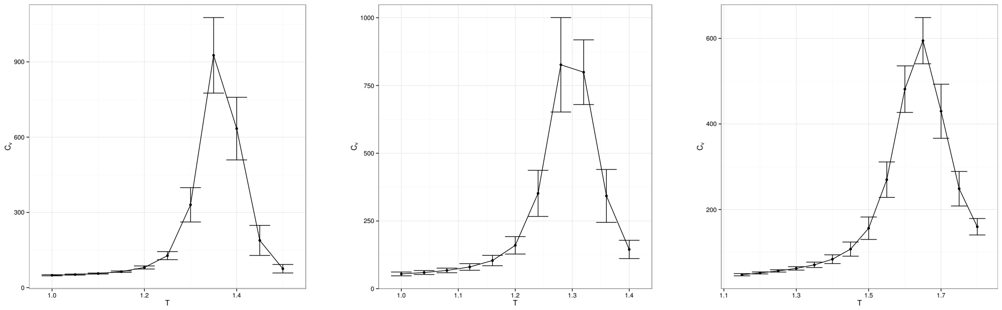
Figure 6.14: Heat capacity and fractions of aggregated (tetramer), dispersed and partially aggregated states across temperatures. All three cases are shown—(a): no helices, (b): only transmembrane helix and (c): only amphipathic helix.

favours aggregation whereas the transmembrane helix (TMH) disfavours it, with the amphipathic one being the stronger effect.

Conspicuous from Figure 6.14 is the lack of error bars. While 10 independent MUCA runs were performed for each of the three cases, we found that for Case 2 (only TMH), not all runs sampled the entire energy space relatively frequently, and so the statistics collected were insufficient to produce reliable data. The pull move acceptance rate of the corresponding case of tetramer model shown in Appendix C is indicative of that of the good runs; in fact, 3 out of the 10 runs have acceptance rates 0.092, 0.116 and 0.125, while the rest range from 0.16 to 0.19. Excluding the 3 runs with the lowest acceptance rates, we plot the heat capacity of Case 2 with error bars in Figure 6.15, along with the other two cases, i.e. no helices and only APH, which used all 10 runs to produce the error bars.

It is also informative to compare energy trace of the run having the lowest acceptance rate, with that of a good run that samples all energy range relatively frequently, this is shown in Figure 6.16. We see that Run 1 appears to be trapped indefinitely in tetramer state beyond $n = 4 \times 10^5$, suggesting existence of kinetically trapped entanglements. Entanglement of chains has been modelled in semicrystalline polyethylene in polymer simulations [21, 49], here our results appear to indicate that it is less likely to be trapped in entanglements with an “amorphous” transmembrane helix than with a preformed TMH. The reason for this might be due to the following observation from the tetramer snapshots shown in Figure 6.17: In Case 2 (Figure 6.17b), the TMH beads (red) maintain helical shape because of the strong interaction encoded in the helices, the hydrophilic beads (blue) comprising the loop and tail regions (Figure 6.3) form a compact configuration surrounded by TMH, and it is possible that the tetramer be trapped in certain collection of configurations and the only way to get out of it is by deforming one or several helices; on the other hand, when there is no helical interaction, the TMH beads can move more freely (Figure 6.17a), and the different arrangements of the TMH beads might open pathways to escape from entanglements.

To find out quantitatively what contributes to the change in heat capacity, we examine the fractions of various states as they vary with temperature; these are shown in the right panels of Figure 6.14. It can be seen that the temperature where both tetramer (red) and dispersed (green) states are equally populated, i.e. where their fractions intersect, and the temperature where the fraction of partially aggregated state is maximum, almost coincide with the transition temperature. The values of the fractions at the transition temperature (T_{trans}) were read-off and shown in Table 6.2.



(a) Case 1: no helices

(b) Case 2: only TMH

(c) Case 3: only APH

Figure 6.15: Heat capacity for three cases with error bars. In plotting Figure 6.15b, the three runs with the lowest acceptance rates were excluded. The half length of the error bar equals one standard deviation.

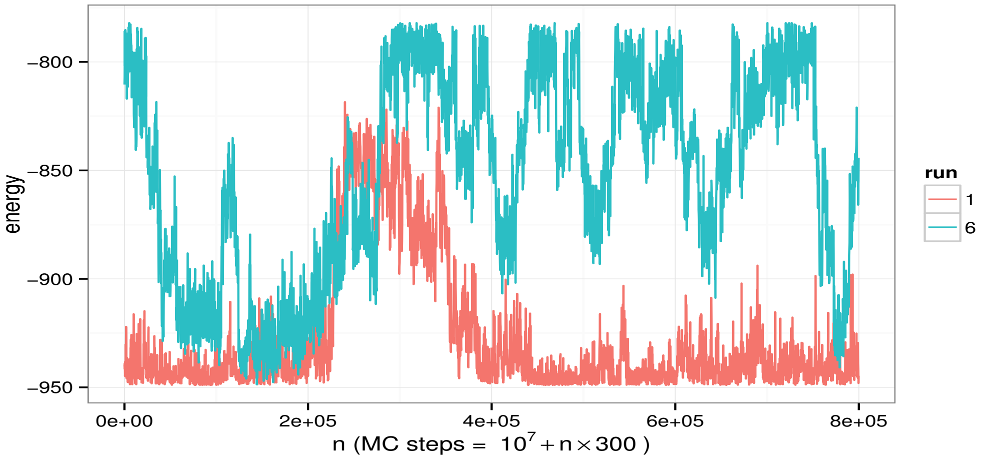
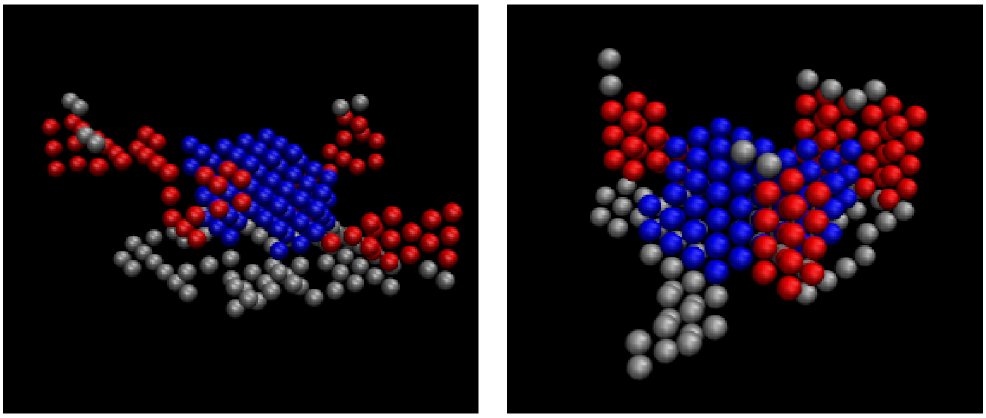


Figure 6.16: Energy versus Monte Carlo step for the run with the lowest acceptance probability (Run 1), and a run with acceptance probability 0.184 (Run 6).

	T_{trans}	fraction		
		tetramer	dispersed	par. aggre.
Case 1	1.355	0.375	0.331	0.294
Case 2	1.296	0.358	0.321	0.320
Case 3	1.638	0.298	0.240	0.462

Table 6.2: Fractions of tetramer, dispersed and partially aggregated states at the transition temperature for the three cases.



(a) no helices

(b) with TMH

Figure 6.17: Tetramer snapshot for Case 1 (a) and Case 2 (b).

An interesting observation is that, when only the APH is present, there are more partially aggregated population than either of tetramer and dispersed populations at the transition temperature. On the other hand, when there are no helices or only the TMH is present, the fraction of partially aggregated state (F_{paggre}) is no more than either the fraction of tetramer state (F_{aggre}), or the fraction of dispersed state (F_{disp}).

Another observation from Table 6.2 is that if we compare across the three cases, we see both F_{aggre} and F_{disp} decrease, and F_{paggre} increases, from zero-helix to TMH and to APH case. This implies that, at the transition temperature, the population of partial aggregates increases when either helix is present, and that the APH results in a larger increase. Similarly, both tetramer and dispersed population decrease when either helix is present, and the APH results in a larger decrease.

Clearly, for each of the three cases, F_{aggre} decreases and F_{disp} increases when T increases. However, Figure 6.14 also shows that the decrease in F_{aggre} is the sharpest in Case 1 among all three cases, suggesting that either helix tends to slow down the transition. Furthermore, F_{paggre} decreases when T is away from T_{trans}, and this decrease has a slower rate than the decrease of F_{aggre} (F_{disp}) as T increases (decreases) from T_{trans}.

Next, we show inter- and intra-polymer contacts as functions of temperature, as we have done in the analysis of two-chain models. In Section 6.2, we decomposed inter-polymer contact into tail-tail and tail-loop contributions, here in four-chain case, we shall consider only the total number of inter-polymer contacts, defined as the sum of contacts from all pairs of polymers. As shown in Figure 6.18, inter-polymer contacts (N_{inter}) clearly dominate intra-polymer contacts (N_{intra}) for all

	T	N_{inter}	N_{intra}
Case 1	1	58.5	9.3
Case 2	1	55.8	9.5
Case 3	1.15	58.5	8.8

Table 6.3: Average values of inter- and intra-polymer contacts at the lowest temperature in each case.

three cases, in contrast to what we observed in the two-chain models, where N_{inter} was only marginally greater than N_{intra}. Table 6.3 shows estimated average values of N_{inter} and N_{intra} at the lowest temperature in the respective case.

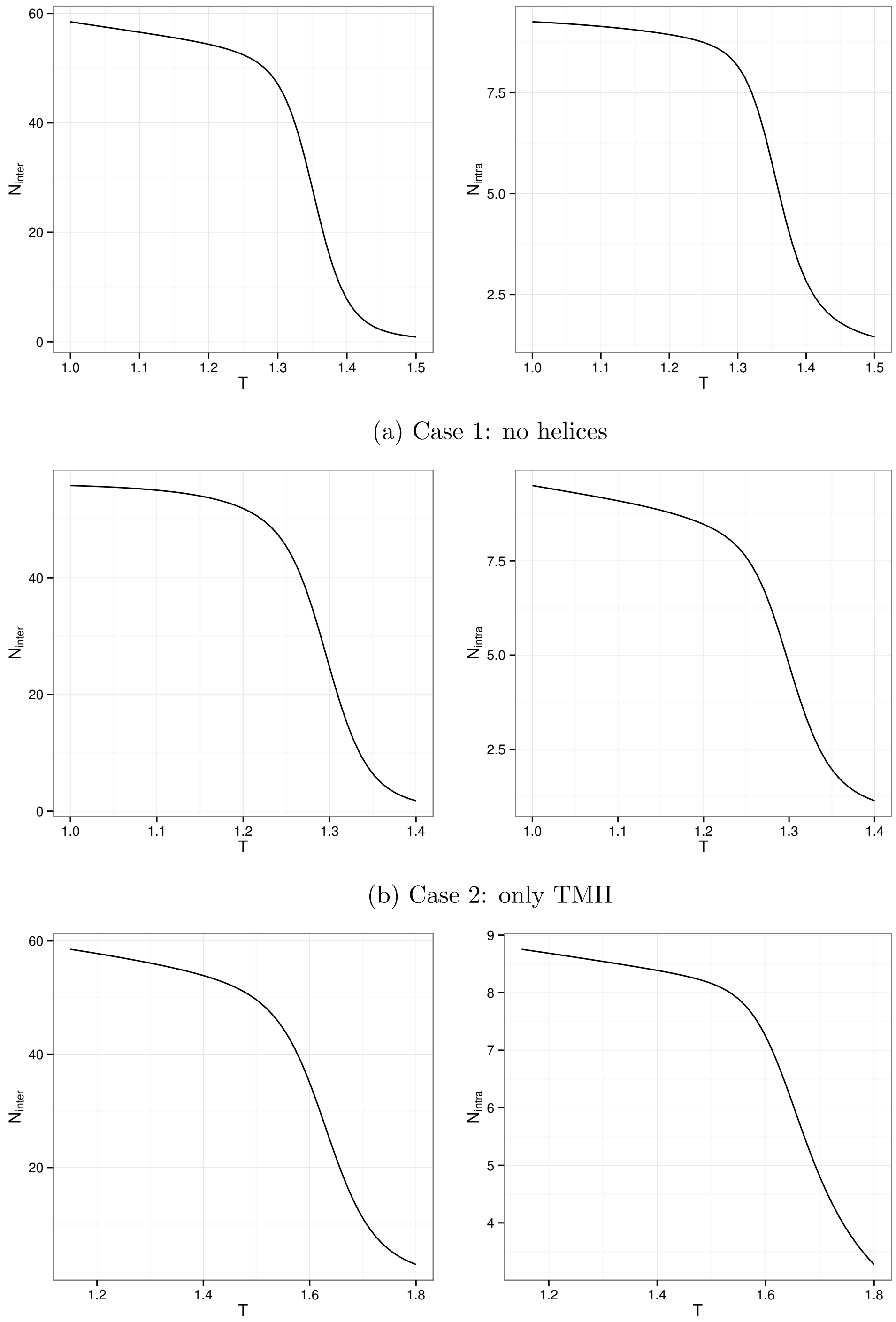
6.4 Discussion and future directions

Compared to the basic, illustrative model in Section 5.4.2, Chapter 5, the TatA models considered in the chapter requires more Monte Carlo iterations, and hence are more time-consuming. Also, the rejection rate increases from zero-helix (Case 1) to one-helix (Cases 2 and 3), and from one-helix to two-helix (Case 4) (see Table C.1, Appendix C). Given the extra complexity in these models, namely the helical structure, interface region and additional amphipathic type bead, we identify several factors that are responsible for efficiency loss and for the added simulation burden.

First, we note that the simulated energy range, defined as the distance between the first and the last energy bin, was 83 for Case 1 in Section 6.2.1, and was 44 in the basic model in Section 5.4.2; hence, although both models did not consider helices, the former would require much longer time for the MUCA simulation to converge. The simulated energy ranges for Cases 2–4 were 90, 97 and 99, respectively, so even longer simulations are needed.

Second, for the TatA models in this chapter, the length of each polymer increased by 20 beads compared to the basic model, these added beads are mainly the amphipathic type beads. Because the chains are longer, the number of accessible configurations increases, and this increase can be substantial especially for the four-chain models in Section 6.3. Computationally, the energy calculations are more costly in the four-chain models, because in each iteration we need to compute three more intra-polymer terms and five more inter-polymer terms, since in the four-chain case there are six possible pair interactions compared to only one in the two-chain case.

In addition, an important reason particularly relevant to helical cases is that pull move is too "arbitrary"—it has no idea where the membrane, interfaces and



(a) Case 1: no helices

(b) Case 2: only TMH

(c) Case 3: only APH

Figure 6.18: Inter-polymer (N_{inter}) and intra-polymer (N_{intra}) contacts as functions of temperature. All three cases are shown—(a): no helices, (b): only transmembrane helix and (c): only amphipathic helix.

helical regions are. Because the strength of helical contacts was set to be high in the force field, reflecting our intention to preserve the helix-like structure given the simulated temperature range, proposal moves which displace beads comprising the helix are likely to be rejected. Recall that a pull move starts by randomly choosing a "pull bead" and pull the chain either upwards or downwards. In one-helix case, pull move in one of the directions will encounter helix and the move is likely to be rejected; and, on top of that, more beads are devoted to helix in two-helix case, rendering a larger probability of the "pull bead" being one that makes up the helix. Similar argument applies if transmembrane/amphipathic beads are pulled out of the membrane/interface region, in which case a large energy penalty would be incurred. Frequent rejections result in long correlation in the trajectory and, hence, large statistical errors of property estimates.

6.4.1 Designing efficient move set — CBMC, HMC and FRESS

How could we improve upon the current situation? To study more complex and realistic models, we need an efficient move set. As noted in the previous paragraph, the problem with pull move is that it does not recognize the structure of the molecule as well as the membrane/interface environment set by the force field, and leads to lots of wasted moves. One way to deal with this problem, therefore, is to incorporate *a priori* information from the force field into proposal moves, so most of the moves will be devoted to interesting subset of the configuration space.

One method that comes into our mind is the configuration bias Monte Carlo (CBMC) [57]. An interesting observation is that the CBMC approach, although developed in the context of polymer chain simulations, can be viewed as a Metropolized independence sampler with the proposal generated through sequential importance sampling (SIS) [39]. Here, the word "independence" does not mean independent samples but comes from the use of a proposal function which does not depend on the current state, i.e. $T(\mathbf{x}, \mathbf{y}) = g(\mathbf{y})$. The generation of $\mathbf{y}$ is through a SIS strategy and this step is key to CBMC—essentially, it preferentially reconstructs the molecule and assigns a proper weight to it.

The Rosenbluth method [54] is a demonstration of SIS in one of the early problems in polymer simulation: finding the mean squared extension of a polymer of specified length in a self-avoiding random walk in lattice space. A naive solution to this problem would be growing the polymer "uniformly". In other words, if the current length of the polymer is n beads, the $(n+1)$th bead is chosen uniformly at one of the neighbors of n. If that position has already been occupied, the growing process is restarted from the beginning; otherwise we continue until the desired length is reached. The procedure is repeated many times to obtain a collection of samples to be used for averaging. Although correct, this naive approach is extremely inefficient because even for a modest length polymer, the number of restarts can be formidably large. One might consider avoiding previously visited positions by only choosing empty neighbors of n in the above process, in fact, this is precisely what Rosenbluth and Rosenbluth did, but in doing so, a weight needs to be assigned to correct for the bias. In the end, one obtains not just a collection of samples but their associated weights and weighted average should be used instead of the simple average in the naive method.

Inspired by the Rosenbluth method, we may replace pull move by a SIS-based move. We note that various adaptations of SIS have been proposed in different fields [24, 40], in particular, variants of the Rosenbluth method have emerged, such as the scanning method [46] and the *Prune-Enriched Rosenbluth Method* [25]. In light of the current TatA model, we may also treat the helix segment as a block and regenerate it in one update as opposed to one step at a time. The idea of using strides of more than one step was suggested by [61].

Our discussion above suggests a way to solve the current problem by including explicitly the information of the secondary structure of TatA in trial moves. While this may be efficient for the current model, it cannot be easily generalized. So a perhaps better option would be designing an adaptive proposal and letting it "learn" from the underlying potential energy function. The CBMC method can be used in this context by adding one bead at a time, with the new bead placed favouring low energy positions. More precisely, suppose the current (partial) configuration is $\mathbf{x}_t = (x_1, \ldots, x_t)$ and let its energy be U_t. To place bead $t + 1$, we look at all available neighboring sites of x_t that are not occupied and let j denote one of such sites, then x_{t+1} is placed at j with probability proportional to $\exp(-(U_{t+1}^j - U_t)/T)$, where U_{t+1}^j is the energy of the configuration with x_{t+1} placed at position j.

Although the CBMC approach can be applied to lattice polymer models, having to regrow the polymer one residue at a time in each proposal step is still quite demanding computationally, especially when the length of the polymer is large. The method of hybrid Monte Carlo (HMC) [16] is similar in spirit to CMBC in that it utilizes information from the potential energy function in each proposal step. In HMC, one augments the configuration space with a fictitious momentum variable $\mathbf{p}$ and evolves the current configuration by doing a molecular dynamics (MD) step, such as the Verlet or the leapfrog algorithm. Under the HMC framework, we can define *Hamiltonians* on the phase space:

$$H(\mathbf{x}, \mathbf{p}) = U(\mathbf{x}) + K(\mathbf{p})$$

and

$$H'(\mathbf{x}, \mathbf{p}) = U'(\mathbf{x}) + K(\mathbf{p}),$$

where $U(\mathbf{x})$ is the potential energy of the system and $K(\mathbf{p})$ is the kinetic energy. The first Hamiltonian is used in the accept/reject step and the second one, in which $U'(\mathbf{x})$ may be different from $U(\mathbf{x})$, is used to evolve the current configuration. The U' is often chosen to be similar to but easier to explore than the true potential U, thereby allowing HMC to follow the dynamics of the potential. Since HMC is designed for off-lattice models, it is not obvious how the framework may be adapted to lattice space.

More recently, an appealing Monte Carlo scheme, termed *fragment regrowth via energy-guided sequential sampling* (FRESS), was proposed [70]. It is originally implemented to search for the globally lowest energy conformation in hydrophobic-polar (HP) protein folding models and is therefore well-suited to lattice MC simulations. FRESS resembles CBMC in that regrowing the chain is also involved, as the name suggests. However, it differs from CBMC in two respects: 1) more often an internal segment is regrown, instead of regrowing the chain all the way up to the terminal residual each time; and 2) the segment to regrow has variable length. These two features equip FRESS with the capability to both explore configurations that are local and carry out more global moves, which allows the algorithm to jump out of local energy basins.

6.4.2 Incorporation of efficient move set in MBAR-enhanced MC

In Section 6.4.1, some strategies are outlined in order to cope with the sampling inefficiency we are experiencing in more complicated models. All of these strategies are concerned with designing efficient and clever trial moves. There is no reason why we cannot incorporate such move set into PT and MUCA trial moves in our MBAR-enhanced Monte Carlo method shown in Chapter 5. We end our discussion with a few comments about the generality of the method.

Although lattice models has been used in our work, the method can be used in off-lattice models as well; in fact, the statistical model of the bivariate normal mixture considered in Section 5.4.1 is an example. Also, note that in our method MBAR plays an important role in linking the two-stage Monte Carlo computations, and while we used parallel tempering and multicanonical sampling methods, other choices are available. For example, instead of running MUCA simulations and reweighting the data to obtain property estimates, we could use methods like the Wang-Landau algorithm discussed in Section 2.5 to further refine the density of states, using those estimates derived from MBAR as a guide.

We believe that, together with a suitably chosen move set, the MBAR-enhanced Monte Carlo is a promising approach for the study of aggregation of lattice proteins or even for more realistic models.

Appendix A

Specifications of the force field

As stated in Section 1.2.2, three types of interactions were considered: E_{intra}, E_{inter} and E_{im}. We provide detailed expressions of the various terms.

We first introduce the following list of notations.

1. n_k: the number of beads in polymer k.
2. x_i: the coordinate of bead i in a polymer. It defines a point in 3-D space.
3. c_i: the color of bead i in a polymer. When there is no interface, it is either hydrophobic (H) or hydrophilic (P); when there is interface, it is one of H, P or H2.
4. d_{ij}: the Euclidean distance between beads i and j.
5. ϵ_{ij}: the strength of interaction between beads i and j, as a linear function of d_{ij},
$$\epsilon_{ij} = \begin{cases} \frac{\epsilon_{\max}(d_{ij}-d_{\text{cut}})}{1-d_{\text{cut}}}, & 1 \leq d_{ij} < d_{\text{cut}} \\ 0, & d_{ij} \geq d_{\text{cut}} \end{cases}$$
where d_{cut} is the cut-off distance and $\epsilon_{\max}$ defines the maximum interaction when $d_{ij} = 1$.
6. M: the membrane region, $M = \{(x,y,z) \in \mathbf{Z}^3 : 0 < z < h_{\text{memb}}\}$, where $\mathbf{Z}^3$ denotes set of all 3-D integer point and h_{memb} defines the height of membrane.
7. C: the lower (cytoplasmic) side of the interface.
$C = \{(x,y,z) \in \mathbf{Z}^3 : -h_{\text{inter}} \leq z \leq 0\}$, where h_{inter} is the height of interface.
8. P: the upper (periplasmic) side of the interface.
$P = \{(x,y,z) \in \mathbf{Z}^3 : h_{\text{memb}} \leq z \leq h_{\text{memb}} + h_{\text{inter}}\}$

9. W: the water region. When there is no interface, it is everywhere else of M, i.e. $W = M^c$, the complement of set M; when there is interface, it is everywhere else of M, C and P, i.e. $W = (M \cup C \cup P)^c$.

10. δ: delta function. We define the following for membrane region and hydrophobic beads, the sets of notations $\{\delta_{\mathrm{WW}}, \delta_{\mathrm{W}}, \delta_{\mathrm{pp}}, \delta_{\mathrm{p}}\}$, $\{\delta_{\mathrm{CC}}, \delta_{\mathrm{PP}}, \delta_{\mathrm{C}}, \delta_{\mathrm{P}}, \delta_{\text{h2-h2}}, \delta_{\mathrm{h2}}\}$ are defined similarly for water region and hydrophilic beads, and for interface region and H2 beads, respectively.

$$\delta_{\mathrm{MM}}(i,j) = \begin{cases} 1, & \text{if } x_i, x_j \in M \\ 0, & \text{otherwise} \end{cases} \qquad \delta_{\mathrm{M}}(i) = \begin{cases} 1, & \text{if } x_i \in M \\ 0, & \text{otherwise} \end{cases}$$

$$\delta_{\mathrm{hh}}(i,j) = \begin{cases} 1, & \text{if } c_i = c_j = \mathrm{H} \\ 0, & \text{otherwise} \end{cases} \qquad \delta_{\mathrm{h}}(i) = \begin{cases} 1, & \text{if } c_i = \mathrm{H} \\ 0, & \text{otherwise} \end{cases}$$

It is understood and should be clear in the context that the delta functions also depends on polymer index k, s, t.

11. H_{b}: set which defines hydrogen bond condition.

$$H_{\mathrm{b}} = \{(i,j) : j - i = 5,\ j\%4 \neq 0 \text{ or } i\%4 = 0,\ (j+1)\%4 = 0,\ j - i = 3\}$$

where percent sign is the modulo operator. This requires the starting index of a sequence of beads comprising helix being a multiple of 4. The associated delta function for this set is denoted δ_{hbond}.

We are now ready to provide details of each individual terms in the potential:

$$\begin{aligned} E^k_{\mathrm{intra}} = -\sum_{j-i \geq 3} \quad & \delta_{\mathrm{MM}} \quad (\delta_{\mathrm{hh}}\, \delta_{\mathrm{hbond}}\, (f_1\, \epsilon_{\mathrm{hbond}}) + \delta_{\mathrm{pp}}\, \epsilon_{ij}) + (\delta_{\mathrm{CC}} + \delta_{\mathrm{PP}})\, \delta_{\text{h2-h2}}\, \delta_{\mathrm{hbond}} (f_0\, \epsilon_{\mathrm{hbond}}) \\ + \quad & \delta_{\mathrm{WW}} (\delta_{\mathrm{hh}} + \delta_{\text{h-h2}} + \delta_{\text{h2-h2}}) \epsilon_{ij}, \end{aligned}$$

$$E^k_{\mathrm{im}} = -\sum_{i=1}^{n_k} \delta_{\mathrm{M}}\, \delta_{\mathrm{h}}\, \epsilon_{\mathrm{hm}} + \delta_{\mathrm{W}}\, \delta_{\mathrm{p}}\, \epsilon_{\mathrm{pw}} + (\delta_{\mathrm{C}} + \delta_{\mathrm{P}}) \delta_{\mathrm{h2}}\, \epsilon_{\mathrm{h2inf}},$$

and

$$E^{s,t}_{\mathrm{inter}} = -\sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \delta_{\mathrm{MM}}\, \delta_{\mathrm{pp}}\, \epsilon_{ij} + \delta_{\mathrm{WW}}\, (\delta_{\mathrm{hh}} + \delta_{\text{h-h2}} + \delta_{\text{h2-h2}}) \epsilon_{ij}.$$

The summation in E^k_{intra} is taken over all pairs of beads in polymer k that are separated by at least 3 beads apart, which is the smallest number of beads required to

Parameter	Value				
	Chapter 3 & 5	Chapter 6			
		case 1	case 2	case 3	case 4
$\epsilon_{\mathrm{hbond}}$	0	0	4	4	4
f_1	0	0	1	0	1
f_0	0	0	0	1	1
ϵ_{hm}	4	4	4	4	4
$\epsilon_{\mathrm{h2inf}}$	-	4	4	4	4
ϵ_{pw}	1.2	1.2	1.2	1.2	1.2
ϵ_{max}	1	1	1	1	1
d_{cut}	1.7	1.7	1.7	1.7	1.7
h_{memb}	6	5	5	5	5
h_{inter}	-	1	1	1	1

Table A.1: Parameters used in the lattice polymer model.

form a contact in a rectangular lattice. In E^k_{intra}, $\epsilon_{\mathrm{hbond}}$ sets the strength of helical contact and f_1, f_0 are factors that control the strength of the transmembrane- and the amphipathic-helix, respectively. The ϵ_{hm}, ϵ_{pw} and $\epsilon_{\mathrm{h2inf}}$ in E^k_{im} reflect, respectively, the tendency of (1) a hydrophobic bead to stay in membrane, (2) a hydrophilic bead to stay in water, and (3) an H2-bead to stay in either side of the interface.

Table A.1 lists the parameter values used in the lattice polymer model.

Appendix B

The PTDW simulation

For technical reasons [41], the following modifications were made in the weight updating process:

1. If the Metropolis ratio is too small, then rejection does not cause the weight to change.
2. A random multiplier with mean 1 is included in the weight updating scheme.

More precisely, the log weight process of a Q-type move in the PTDW simulation has the form:

$$\log w^{(t+1)} = \begin{cases} \max\{0, \log w^{(t)} + \log r^{(t)}\} + \log v^{(t)} & \text{if } u \le \min\{1, w^{(t)} r^{(t)}\} \\ \log w^{(t)} + \log a + \log v^{(t)} & \text{if rejected and } r^{(t)} \ge \epsilon \\ \log w^{(t)} & \text{if rejected and } r^{(t)} < \epsilon, \end{cases}$$

where $v \sim \text{Uniform}(1-\delta, 1+\delta)$ and $a > 1$. In the simulation we used $\delta = 0.4$, $a = 2$ and $\epsilon = 10^{-10}$.

The histogram matrix of the PT simulation in Section 3.2.2 is listed in Table B.1.

Table B.1: The full $H_{m\times k}$ matrix, including all observed energy values, of the PT simulation in Section 3.2.2. Row labels (energy values) and column labels (temperatures) are typeset in bold. The entry H_{km} is the observed count of samples in bin m at temperature k, where a bin size of 0.5 is used.

	0.3	**0.48**	**0.85**	**1.3**	**2**
-220.25	5135	751	9	0	0

Table B.1— *Continued from previous page.*

	0.3	**0.48**	**0.85**	**1.3**	**2**
-219.75	48013	11046	26	0	0
-219.25	75586	34289	48	0	0
-218.75	40165	32597	429	0	0
-218.25	17345	28783	638	2	0
-217.75	9165	27181	550	2	0
-217.25	3012	20646	456	1	0
-216.75	957	11846	487	3	0
-216.25	290	7793	646	2	0
-215.75	236	9648	638	4	0
-215.25	62	4412	537	1	0
-214.75	26	4979	1121	9	1
-214.25	5	1605	756	10	0
-213.75	1	1757	988	15	0
-213.25	1	857	797	14	0
-212.75	1	852	907	15	1
-212.25	0	360	515	7	3
-211.75	0	329	761	23	1
-211.25	0	94	435	21	1
-210.75	0	90	570	26	2
-210.25	0	40	498	31	2
-209.75	0	34	366	21	2
-209.25	0	4	239	22	1
-208.75	0	3	223	14	1
-208.25	0	0	202	29	1
-207.75	0	1	133	19	5
-207.25	0	0	102	14	9
-206.75	0	0	96	21	3
-206.25	0	0	135	34	7
-205.75	0	0	382	148	36
-205.25	0	0	547	242	67
-204.75	0	3	89888	44136	12870
-204.25	0	0	951	601	222
-203.75	0	0	47079	37483	15311
-203.25	0	0	820	712	332
-202.75	0	0	1709	1850	1003

Table B.1— *Continued from previous page.*

	0.3	**0.48**	**0.85**	**1.3**	**2**
-202.25	0	0	21920	29278	16650
-201.75	0	0	1496	2652	1949
-201.25	0	0	9132	19308	15615
-200.75	0	0	3059	7455	6582
-200.25	0	0	1670	5140	5503
-199.75	0	0	3801	13727	15799
-199.25	0	0	1069	5366	7461
-198.75	0	0	1186	6808	10461
-198.25	0	0	602	4462	7938
-197.75	0	0	497	4398	9286
-197.25	0	0	331	3162	7556
-196.75	0	0	158	2288	6284
-196.25	0	0	158	2772	8385
-195.75	0	0	78	1419	5337
-195.25	0	0	71	1746	7162
-194.75	0	0	27	855	4182
-194.25	0	0	25	943	5348
-193.75	0	0	14	637	4041
-193.25	0	0	12	565	4303
-192.75	0	0	4	352	2872
-192.25	0	0	1	322	2896
-191.75	0	0	2	233	2407
-191.25	0	0	1	161	2193
-190.75	0	0	2	105	1492
-190.25	0	0	0	96	1714
-189.75	0	0	0	60	1089
-189.25	0	0	0	47	1181
-188.75	0	0	0	31	691
-188.25	0	0	0	27	804
-187.75	0	0	0	13	535
-187.25	0	0	0	15	589
-186.75	0	0	0	4	293
-186.25	0	0	0	9	387
-185.75	0	0	0	5	190
-185.25	0	0	0	2	222

Table B.1— *Continued from previous page.*

	0.3	**0.48**	**0.85**	**1.3**	**2**
-184.75	0	0	0	3	127
-184.25	0	0	0	2	149
-183.75	0	0	0	0	100
-183.25	0	0	0	0	91
-182.75	0	0	0	0	58
-182.25	0	0	0	0	44
-181.75	0	0	0	0	38
-181.25	0	0	0	0	37
-180.75	0	0	0	0	6
-180.25	0	0	0	0	15
-179.75	0	0	0	0	8
-179.25	0	0	0	0	9
-178.75	0	0	0	0	9
-178.25	0	0	0	0	5
-177.75	0	0	0	0	3
-177.25	0	0	0	0	3
-176.75	0	0	0	0	3
-176.25	0	0	0	0	4
-175.75	0	0	0	0	2
-175.25	0	0	0	0	2
-174.75	0	0	0	0	1
-174.25	0	0	0	0	0
-173.75	0	0	0	0	1
-173.25	0	0	0	0	1
-172.75	0	0	0	0	2
-172.25	0	0	0	0	1
-171.75	0	0	0	0	0
-171.25	0	0	0	0	0
-170.75	0	0	0	0	0
-170.25	0	0	0	0	0
-169.75	0	0	0	0	0
-169.25	0	0	0	0	0
-168.75	0	0	0	0	0
-168.25	0	0	0	0	1
-167.75	0	0	0	0	0

Table B.1— *Continued from previous page.*

	0.3	**0.48**	**0.85**	**1.3**	**2**
-167.25	0	0	0	0	1
-166.75	0	0	0	0	0
-166.25	0	0	0	0	1

Appendix C

Additional simulation infomation

The statistical example in Section 5.4.1, Chapter 5 used as the proposal distribution the bivariate normal distribution centered at the current configuration and with the identity covariance matrix. The same covariance matrix was used in proposal distributions across all temperature levels, and so no optimizations have been made. The PT sampler was run for 3×10^4 iterations and data were collected after an equilibration period of 10^4 iterations. The `timeseries` module of `pymbar` [56] was used to calculate the statistical inefficiency of the correlated PT trajectory; this was also used, where necessary, in Section 5.4.2. The bin width $\Delta U = 0.1$ was used in the statistical example; and each of the 10 independent MUCA runs used the same number of iterations and equilibration time as the original PT simulation.

The chain sequence used in Chapter 6 was $(P)_{12}(H2)_{16}(P)_8(H)_{16}(H2)_2$, where the notation $(X)_n$ means that bead type X is repeated n times. More details of the simulations done in this chapter are listed in Table C.1 on page 101. The majority of simulations were conducted on the Cluster of Workstations in the Centre for Scientific Computing, University of Warwick, and some simulations in Section 6.3 were run in Apocrita, a cluster hosted at the Queen Mary University of London.

	case	temperature ladder	box size	PT iters (per temperature)	MUCA iters	pull move acceptance in MUCA	approx CPU time per 10^7 iters (PT/MUCA)
Two-chain	1	0.8 1 1.2 1.5	90	3×10^7	7×10^7	0.31	7h/1h
	2	0.8 1 1.2 1.4		3×10^7	1.9×10^8	0.20	
	3	0.9 1.1 1.35 1.6		3×10^7	1.9×10^8	0.21	
	4	1.1 1.25 1.4 1.6		5×10^7	3.7×10^8	0.09	
Four-chain	1	1 1.15 1.3 1.5	114	3×10^7	1.9×10^8	0.28	9h/2h
	2	1 1.1 1.25 1.4		5×10^7	2.5×10^8	0.18	
	3	1.15 1.3 1.55 1.8		5×10^7	2.5×10^8	0.20	

Table C.1: Additional information for simulations in Chapter 6: temperature ladder, the size of the cubic simulation box with periodic boundary condition, number of parallel tempering iterations per temperature, number of multicanonical iterations, pull move acceptance rate in MUCA simulations, and approximate CPU time for 10^7 iterations in PT/MUCA. Here, an iteration refers to one Monte Carlo step, as in Algorithm 2, page 15. The number of attempted pull moves is 70 per 100 iterations, the remianing 30 is the number of translation move attempts.

Bibliography

[1] Felicity Alcock et al. “Live cell imaging shows reversible assembly of the TatA component of the twin-arginine protein transport system”. In: *Proc. Natl. Acad. Sci.* 110.38 (2013), E3650–E3659.

[2] Christian Bartels. “Analyzing biased Monte Carlo and molecular dynamics simulations”. In: *Chem. Phys. Lett.* 331.5 (2000), pp. 446–454.

[3] Charles H Bennett. “Efficient estimation of free energy differences from Monte Carlo data”. In: *J. Comput. Phys.* 22.2 (1976), pp. 245–268.

[4] Bernd A Berg. “Algorithmic aspects of multicanonical simulations”. In: *Nucl. Phys. B (Proc. Suppl.)* 63.1 (1998), pp. 982–984.

[5] Bernd A Berg. “Introduction to multicanonical Monte Carlo simulations”. In: *Fields Inst. Commun* 26.1 (2000), pp. 1–24.

[6] Bernd A Berg and Thomas Neuhaus. “Multicanonical algorithms for first order phase transitions”. In: *Phys. Lett. B* 267.2 (1991), pp. 249–253.

[7] Ben C Berks. “The Twin-Arginine Protein Translocation Pathway”. In: *Annu. Rev. Biochem.* 84 (2015), pp. 843–864.

[8] Ben C Berks, Frank Sargent, and Tracy Palmer. “The Tat protein export pathway”. In: *Mol. Microbiol.* 35.2 (2000), pp. 260–274.

[9] Thomas Brüser and Carsten Sanders. “An alternative model of the twin arginine translocation system”. In: *Microbiol. Res.* 158.1 (2003), pp. 7–17.

[10] Troy Cellmer et al. “Thermodynamics of folding and association of lattice-model proteins”. In: *J. Chem. Phys.* 122.17 (2005).

[11] Hue Sun Chan and Ken A Dill. “The protein folding problem”. In: *Physics today* 46.2 (1993), pp. 24–32.

[12] John D Chodera et al. “Use of the weighted histogram analysis method for the analysis of simulated and parallel tempering simulations”. In: *J. Chem. Theory Comput.* 3.1 (2007), pp. 26–41.

[13] Anthony Christopher Davison and David Victor Hinkley. *Bootstrap Methods and Their Application.* Cambridge: Cambridge University Press, 1997.

[14] Christopher M Dobson. “Protein folding and misfolding”. In: *Nature* 426.6968 (2003), pp. 884–890.

[15] Christopher M Dobson, Andrej Šali, and Martin Karplus. “Protein folding: a perspective from theory and experiment”. In: *Angewandte Chemie International Edition* 37.7 (1998), pp. 868–893.

[16] Simon Duane et al. “Hybrid Monte Carlo”. In: *Phys. Lett. B* 195.2 (1987), pp. 216–222.

[17] Bradley Efron. “Bootstrap methods: another look at the jackknife”. In: *Ann. Stat.* 7.1 (1979), pp. 1–26.

[18] Alan M Ferrenberg and Robert H Swendsen. “Optimized monte carlo data analysis”. In: *Phys. Rev. Lett.* 63.12 (1989), p. 1195.

[19] M Fitzgerald, RR Picard, and RN Silver. “Canonical transition probabilities for adaptive Metropolis simulation”. In: *Europhys. Lett.* 46.3 (1999), p. 282.

[20] M Fitzgerald, RR Picard, and RN Silver. “Monte Carlo transition dynamics and variance reduction”. In: *J. Stat. Phys.* 98.1-2 (2000), pp. 321–345.

[21] Ulf W. Gedde and Alessandro Mattozzi. “Long Term Properties of Polyolefins”. In: ed. by Ann-Christine Albertsson. Berlin, Heidelberg: Springer, 2004. Chap. Polyethylene Morphology, pp. 29–74.

[22] Charles J Geyer. “Markov Chain Monte Carlo Maximum Likelihood”. In: *In Computer Science and Statistics: Proceedings of the 23rd Symposium on the Interface; Keramidas, E. M., Kaufman, S. M., Eds.; Interface Foundation of North America: Fairfax Station, VA, 1991; pp 156-163.* Ed. by Elaine M Keramidas. Interface Foundation of North America, 1991.

[23] Ulrich Gohlke et al. “The TatA component of the twin-arginine protein transport system forms channel complexes of variable diameter”. In: *Proc. Natl. Acad. Sci.* 102.30 (2005), 10482–10486”.

[24] Neil J Gordon, David J Salmond, and Adrian FM Smith. “Novel approach to nonlinear/non-Gaussian Bayesian state estimation”. In: *IEE Proceedings F (Radar and Signal Processing)* 140.2 (1993), pp. 107–113.

[25] Peter Grassberger. “Pruned-enriched Rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000”. In: *Physical Review E* 56.3 (1997), pp. 3682–3693.

[26] D Györffy, P Závodszky, and A Szilágyi. “”Pull moves” for rectangular lattice polymer models are not fully reversible”. In: *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 9.6 (2012), pp. 1847–1849.

[27] Paul M Harrison et al. “Conformational propagation with prion-like characteristics in a simple model of protein folding”. In: *Protein Sci.* 10.4 (2001), pp. 819–835.

[28] W Keith Hastings. “Monte Carlo sampling methods using Markov chains and their applications”. In: *Biometrika* 57.1 (1970), pp. 97–109.

[29] Yunfei Hu et al. “Solution NMR structure of the TatA component of the twin-arginine protein transport system from gram-positive bacterium Bacillus subtilis”. In: *J. Am. Chem. Soc.* 132.45 (2010), pp. 15942–15944.

[30] Koji Hukushima and Koji Nemoto. “Exchange Monte Carlo method and application to spin glass simulations”. In: *J. Phys. Soc. Jpn.* 65.6 (1996), pp. 1604–1608.

[31] Rachael L Jack et al. “Coordinating assembly and export of complex bacterial proteins”. In: *The EMBO Journal* 23.20 (2004), pp. 3962–3972.

[32] Tuomas PJ Knowles, Michele Vendruscolo, and Christopher M Dobson. “The amyloid state and its association with protein misfolding diseases”. In: *Nat. Rev. Mol. Cell Biol.* 15.6 (2014), pp. 384–396.

[33] A Kong et al. “A theory of statistical models for Monte Carlo integration”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65.3 (2003), pp. 585–604.

[34] Shankar Kumar et al. “The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method”. In: *J. Comput. Chem.* 13.8 (1992), pp. 1011–1021.

[35] Kit Fun Lau and Ken A Dill. “A lattice statistical mechanics model of the conformational and sequence spaces of proteins”. In: *Macromolecules* 22.10 (1989), pp. 3986–3997.

[36] P.A. Lee, D. Tullman-Ercek, and G. Georgiou. “The Bacterial Twin-Arginine Translocation Pathway”. In: *Annu. Rev. Microbiol* 60:373-395 (2006).

[37] N. Lesh, M. Mitzenmacher, and S. Whitesides. “A Complete and Effective Move Set for Simplified Protein Folding”. In: *In Proceedings of the 7th Annual International Conference on Research in Computational Molecular Biology, Berlin, Germany, April 10-13, 2003; ACM Press: New York, 2003.* Berlin,

Germany, 2003. ISBN: 1-58113-635-8. DOI: 10.1145/640075.640099. URL: http://doi.acm.org/10.1145/640075.640099.

[38] Faming Liang. "Dynamically weighted importance sampling in Monte Carlo computation". In: *J. Am. Stat. Assoc.* 97.807–821 (2002).

[39] Jun S Liu. *Monte Carlo strategies in scientific computing.* Springer, 2008.

[40] Jun S Liu and Rong Chen. "Blind deconvolution via sequential imputations". In: *J. Am. Stat. Assoc.* 90.430 (1995), pp. 567–576.

[41] Jun S Liu, Faming Liang, and Wing Hung Wong. "A theory for dynamic weighting in Monte Carlo computation". In: *J. Am. Stat. Assoc.* 96.454 (2001), pp. 561–573.

[42] Jun S Liu, Wing H Wong, and Augustine Kong. "Covariance structure and convergence rate of the Gibbs sampler with various scans". In: *Journal of the Royal Statistical Society. Series B* (1995), pp. 157–169.

[43] Katie A Maerzke et al. "Simulating Phase Equilibria using Wang-Landau-Transition Matrix Monte Carlo". In: *Journal of Physics: Conference Series.* Vol. 487. 1. IOP Publishing. 2014, p. 012002.

[44] Martin Mann et al. "Classifying proteinlike sequences in arbitrary lattice protein models using LatPack". In: *HFSP Journal* 2.6 (2008), pp. 396–404.

[45] Enzo Marinari and Giorgio Parisi. "Simulated tempering: a new Monte Carlo scheme". In: *Europhys. Lett.* 19.6 (1992), p. 451.

[46] Hagai Meirovitch. "Scanning method as an unbiased simulation technique and its application to the study of self-attracting random walks". In: *Physical Review A* 32.6 (1985), pp. 3699–3708.

[47] Xiao-Li Meng and Wing Hung Wong. "Simulating ratios of normalizing constants via a simple identity: a theoretical exploration". In: *Statistica Sinica* 6.4 (1996), pp. 831–860.

[48] Nicholas Metropolis et al. "Equation of state calculations by fast computing machines". In: *J. Chem. Phys.* 21.6 (1953), pp. 1087–1092.

[49] Fritjof Nilsson et al. "Modelling tie chains and trapped entanglements in polyethylene". In: *Polymer* 53 (2012), pp. 3594–3601.

[50] Tracy Palmer and Ben C Berks. "The twin-arginine translocation (Tat) protein export pathway". In: *Nat. Rev. Microbiol.* 10.7 (2012), pp. 483–496.

[51] Dimitris N Politis and Joseph P Romano. "The stationary bootstrap". In: *J. Am. Stat. Assoc.* 89.428 (1994), pp. 1303–1313.

[52] Christian P Robert and George Casella. *Monte Carlo Statistical Methods.* 2nd ed. New York: Springer, 2004.

[53] Fernanda Rodriguez et al. “Structural model for the protein-translocating element of the twin-arginine transport system”. In: *Proc. Natl. Acad. Sci.* 110.12 (2013), E1092–E1101.

[54] Marshall N Rosenbluth and Arianna W Rosenbluth. “Monte Carlo calculation of the average extension of molecular chains”. In: *J. Chem. Phys.* 23.2 (1955), pp. 356–359.

[55] M Scott Shell, Pablo G Debenedetti, and Athanassios Z Panagiotopoulos. “An improved Monte Carlo method for direct calculation of the density of states”. In: *J. Chem. Phys.* 119.18 (2003), pp. 9406–9411.

[56] Michael R Shirts and John D Chodera. “Statistically optimal analysis of samples from multiple equilibrium states”. In: *J. Chem. Phys.* 129.12 (2008), p. 124105.

[57] Jörn Ilja Siepmann and Daan Frenkel. “Configurational bias Monte Carlo: a new sampling scheme for flexible chains”. In: *Mol. Phys.* 75.1 (1992), pp. 59–70.

[58] Adam D Swetnam and Michael P Allen. “Improved simulations of lattice peptide adsorption”. In: *Phys. Chem. Chem. Phys.* 11 (2009), pp. 2046–2055.

[59] Zhiqiang Tan. “On a likelihood approach for Monte Carlo integration”. In: *J. Am. Stat. Assoc.* 99.468 (2004), pp. 1027–1036.

[60] Glenn M Torrie and John P Valleau. “Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling”. In: *J. Comput. Phys.* 23.2 (1977), pp. 187–199.

[61] F. T. Wall, R. J. Rubin, and L. M. Isaacson. “Improved statistical method for computing mean dimensions of polymer molecules”. In: *J. Chem. Phys.* 27.1 (1957), pp. 186–188.

[62] Fugao Wang and David P Landau. “Efficient, multiple-range random walk algorithm to calculate the density of states”. In: *Phys. Rev. Lett.* 86.10 (2001), p. 2050.

[63] Jian-Sheng Wang and Robert H Swendsen. “Transition matrix Monte Carlo method”. In: *Journal of statistical physics* 106.1-2 (2002), pp. 245–285.

[64] Guanghong Wei, Normand Mousseau, and Philippe Derreumaux. “Computational simulations of the early steps of protein aggregation”. In: *Prion* 1.1 (2007), pp. 3–8.

[65] Wing Hung Wong and Faming Liang. "Dynamic weighting in Monte Carlo and optimization". In: *Proceedings of the National Academy of Sciences* 94.26 (1997), pp. 14220–14224.

[66] Thomas Wüst and David P. Landau. "Versatile Approach to Access the Low Temperature Thermodynamics of Lattice Polymers and Proteins". In: *Phys. Rev. Lett.* 102 (2009), p. 178101.

[67] Ying Xu, Dong Xu, and Jie Liang, eds. *Computational methods for protein structure prediction and modeling.* 1st ed. Vol. 2. Springer-Verlag, 2007.

[68] Yuanwei Xu and P. Mark Rodger. "Improved Estimation of Density of States for Monte Carlo Sampling via MBAR". In: *J. Chem. Theory Comput.* (2015). ISSN: 1549-9626. DOI: 10.1021/acs.jctc.5b00189.

[69] Qiliang Yan and Juan J de Pablo. "Fast calculation of the density of states of a fluid by Monte Carlo simulations". In: *Phys. Rev. Lett.* 90.3 (2003), p. 035701.

[70] Jinfeng Zhang, S. C Kou, and Jun S Liu. "Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo". In: *J. Chem. Phys.* 126.22 (2007).

[71] Robert W Zwanzig. "High-temperature equation of state by a perturbation method. I. nonpolar gases". In: *J. Chem. Phys.* 22.8 (1954), pp. 1420–1426.

[72] Robert Zwanzig and Narinder K Ailawadi. "Statistical error due to finite time averaging in computer experiments". In: *Phys. Rev.* 182.1 (1969), p. 280.