Towards understanding the self-assembly of complicated particles via computation

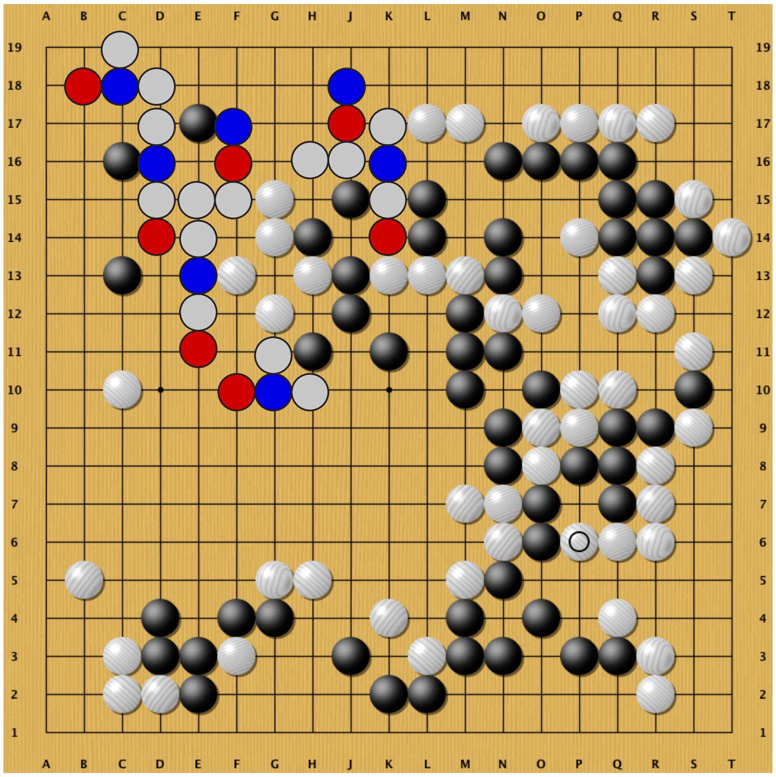
by

Eric Jankowski

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Chemical Engineering)
in The University of Michigan
2012

Doctoral Committee:

Professor Sharon C. Glotzer, Chair
Professor John H. Holland
Professor Ronald G. Larson
Professor Robert M. Ziff
Peter J. Woolf



Detailed and balanced

For Meghan,

ACKNOWLEDGEMENTS

I owe my deepest gratitude to my advisor, Professor Sharon Glotzer, for her support and guidance during my time at Michigan. Her vision and encouragement have in no small way influenced my development as a professional and the quality of my work. I am particularly grateful for the freedom to pursue any hunch or idea, which has made our lab an empowering and fun place to be.

To all of the Glotzer lab "peeps", I am fortunate to have you as friends and mentors. Drs. Chris Iacovella, Aaron Keys, Stephanie Teich-McGoldrick, Aaron Santos, Chetana Singh, Joshua Anderson, and Magnus Bergroth, thanks for showing me the ropes and indulging my many curiosities. Drs. Trung Dac Nguyen, Amir Haji Akbari Balou, and Sean Langelier, thank you for being the best classmates and competitors. I have thoroughly enjoyed trying to keep up with you and eating far too much whenever we get the chance. To Dr. Greg van Anders, Dr. Daphne Klotsa, Dr. Michael Engel, Dr. Carolyn Phillips, Nguyen Nguyen, Daniel Ortiz, Khalid Ahmed, Ben Apland, Jeremy Kapala, Ben Schultz, Ryan Marson, Ines Pons, Antonio Osorio, Andres Millan, Tony Sheh, Eric Irrgang, and Ross Smith, thank you for our ceaselessly interesting discussions and collaborations. I thank Karen Coulter for keeping the cogs in this machine greased. I consider the lab my extended family.

I am grateful to the Graduate Assistance in Areas of National Need fellowship from the Department of Education, the National Defense Science, and Engineering Graduate fellowship funded by the Air Force office of Scientific Research and the James S. McDonnell Foundation for supporting my research. I am indebted to the

Michigan Department of Chemical Engineering and its Graduate Teaching Fellowship for an invaluable opportunity to develop as an instructor with the brightest students and kindest support network.

I am grateful for my many teachers and mentors at the University of Michigan. Thanks to Prof. Solomon for being a role model, advisor, collaborator, and advocate for me ever since my first chemical engineering course. I thank the support of my committee members, Prof. John Holland, Prof. Robert Ziff, Prof. Ronald Larson, and Dr. Peter Woolf who all serve as great inspirations and kind helping hands. I thank Dr. Rick Riolo and my friends in the Center for Study of Complex Systems for providing additional mentoring, support, and fantastic discussions on topics outside of engineering.

The Center for Advance Computing at UM, and its administrators Brock Palen, Andy Caird, and Matt Britt have done an incredible job providing technical support and keeping our cluster alive. I am thankful for your help. I am also thankful for the thousands of open source software developers whose efforts frequently fly under the radar, but whose contributions to the ecosystem are crucial. Thank you.

To my bicycle-loving friends of Common Cycle, the Michigan Cycling team, and the local bicycle shops, I am obliged to you for tactile learning experiences and the chances to meet amazing people in beautiful places.

To the Ann Arbor Go club and the worldwide go community, if I were constrained to do only one thing, it would be playing go. The spirit of friendly competition and desire to find the most beautiful moves with you has truly informed and inspired my work.

I am grateful for the extensive network of friends I have come to know through Ann Arbor's peculiarly acronym-laden social scene. The Eat at Joe's crew, DFSC, A2Geeks, A2R2, MES, VDN, BBT, and NSP have enriched my studies and for that I am thankful.

To my family, in whose wacky marinade of ingenuity I am fortunate to have steeped. Creating and exploring universes small and large in our home was my first love. With your support and encouragement I am so lucky to have never stopped doing this.

My greatest appreciation is for my wife, Meghan. You bring me delight and humility.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

LIST OF APPENDICES

Appendix

ABSTRACT

Towards understanding the self-assembly of complicated particles via computation

by

Eric Jankowski

Chair: Sharon C. Glotzer

We develop advanced Monte Carlo sampling schemes and new methods of calculating thermodynamic partition functions that are used to study the self-assembly of complicated "patchy " particles. Patchy particles are characterized by their strong anisotropic interactions, which can cause critical slowing down in Monte Carlo simulations of their self-assembly. We prove that detailed balance is maintained for our implementation of Monte Carlo cluster moves that ameliorate critical slowing down and use these simulations to predict the structures self-assembled by patchy tetrominoes. We compare structures predicted from our simulations with those generated by an alternative learning-augmented Monte Carlo approach and show that the learning-augmented approach fails to sample thermodynamic ensembles. We prove one way to maintain detailed balance when parallelizing Monte Carlo using the checkerboard domain decomposition scheme by enumerating the state-to-state transitions for a simple model with general applicability. Our implementation of checkerboard Monte Carlo on graphics processing units enables accelerated sampling of thermodynamic properties and we use it to confirm the fluid-hexatic transition observed at high packing fractions of hard disks. We develop a new method, bottom-up building block assembly, which generates partition functions hierarchically. Bottom-up building block assembly provides a means to answer the question of which structures are favored at a given temperature and allows accelerated prediction of potential energy minimizing structures, which are difficult to determine with Monte Carlo methods. We show how the sequences of clusters generated by bottom-up building block assembly can be used to inform “assembly pathway engineering”, the design of patchy particles whose assembly propensity is optimized for a target structure. The utility of bottom-up building block assembly is demonstrated for systems of CdTe/CdS tetrahedra, DNA-tethered nanospheres, colloidal analogues of patchy tetrominoes and shape-shifting particles.

CHAPTER I

Introduction

1.1 Motivation

The paper that this thesis is printed upon or the screen on which it is being viewed are both materials whose physical properties are determined by the arrangement and interactions of their constituent building blocks. In the case of paper, the particular network of cellulose fibers in a given sheet determines how easy it is to tear, how quickly it will soak up water, or how smooth it is to the touch. The arrangement of aluminum, silicon, and oxygen in the glass of a liquid crystal display determine its scratch-resistance and clarity, and the ordering of rod-like molecules determines the color of each pixel. Technologies such as paper and computer screens are enabled by our ability to create materials whose physical properties match the specifications imposed by a particular application. Improving our ability to design and control the structure of materials will enable new technologies with potential for impact in medicine, sustainability, and consumer products.

Controlling the structure of a material can be difficult when it is made up of trillions of particles, each of which might be one-onethousandth the diameter of a human hair. If 100-nanometer square particles are used to make a 1m×1m sheet, one hundred trillion (1×10^{14}) particles would need to be precisely placed. Placing these particles sequentially, the same way bricks are laid down to make a wall, is an infeasible manufacturing technique. Even if one million particles could be placed each second, it would take over three years to assemble the aforementioned sheet.

Instead of sequentially placing building blocks by hand, a more practical solution is to have the particles assemble themselves. A collection of particles suspended in a solution may spontaneously form ordered structures as the particles attract, collide and rearrange[1–3]. In thermally equilibrated systems, this process is known as free energy minimization, and results from simultaneous potential energy minimization and entropy maximization. In principle, self-assembly allows arbitrarily large materials with customized structures to be synthesized from a "soup" of building blocks, so long as the building block interactions are chosen properly.

The particles that can now be used as the building blocks for self-assembly have enormous variety, as do the structures that arise from them. Polymers composed of chemically distinct monomer blocks, such as polystyrene and poly-butadiene, have been used to assemble micelles, cylinders, sheets, and bicontinuous diamond networks[4]. Rods, cubes, tetrahedra, prisms, and branched particles can be made from gold, silver, platinum, and cadmium alloys, which can self-assemble into wires, sheets, and helices [5–14]. Recent work has shown that polydisperse metallic nanoparticles can self-assemble to form larger, more monodisperse aggregates[15]. Lithographic techniques have demonstrated success in synthesizing large amounts of monodisperse micron-sized particles [16, 17]. The interactions of these particles can be modified by coating them with silica[18], polyethylene glycol [19], small ligands such as trizma and serinol [20], or by chemical addition of single-stranded DNA[21]. Colloidal building blocks composed of spheres[22–29], ellipses[30], or aggregates of surfactant-stabilized polymers[31] are another set of particles whose interactions can be tuned with solvent mixture, salts, and small polymer additives[32–34]. The assembly of structures composed from DNA or DNA-functionalized particles is an active area of research, demonstrating control over metallic nanoparticles, proteins, colloids, and assemblies of DNA with carefully designed sequences of nucleotides[2, 35–40]. Nanoparticles that have non-spherical shapes and directional interactions are examples of complex *patchy particles*, and are the focus of this work.

Creating new materials from patchy particles has potential for transformative technologies now and for developing even more advanced materials in the future. Particles containing magnetite nanoparticles are attractive candidates for MRI contrast agents [41, 42], cancer therapeutics [43], as vehicles for controlled drug delivery [44], and as components of electrorheological clutches[45]. Self-assembled platinum coated carbon nanotubes can be used to improve the performance of proton exchange membranes in fuel cells[46], and agglomeration of spherical nanoparticles can strengthen body armor[47]. The same rules that govern the self-assembly of the particles participating in the aforementioned applications also govern the dynamics of biological self assembly. The spontaneous formation of virus capsids [48–50], the growth and destruction of microtubules within the cell cytoplasm [51], and the assembly of ribosomes which synthesize protein chains in every cell [52] are all results of self-assembly. Understanding the link between building block interactions and the structures that emerge from them is not only useful for creating new materials, but is central to our understanding of life.

1.2 Background

Our ability to study self-assembly today was largely enabled by advances in statistical mechanics theory in the late 1800's. Work by Clausius[53], Maxwell[54], Boltzmann[55], and Gibbs[56] defined the concepts of entropy and thermodynamic ensembles, and provided mathematical connection between the macroscopic properties of materials and the microscopic details of the building blocks that comprise them. The microscopic details for a collection of N classical particles are their positions $\{\vec{x_1}, \vec{x_2}, ..., \vec{x_N}\} = \vec{x_1}^N$ and momenta $\{\vec{p_1}, \vec{p_2}, ..., \vec{p_N}\} = \vec{p}^N$, which define a *microstate*.

A system at equilibrium is defined by an *ensemble* of microstates that are consistent with some fixed thermodynamic variables. For example, a set of microstates with the same number of particles N confined to volume V with energy E defines a *microcanonical* ensemble. Other ensembles, including the *canonical (N, V, T)* ensemble and *grand canonical (μ, V, T)* ensemble allow energy and the number of particles to vary. For a given ensemble, the likelihood of observing a particular microstate γ depends upon its weight. In the microcanonical ensemble a microstate's weight is

$$w_{N,V,E}(\gamma) = \Omega_\gamma \tag{1.1}$$

where Ω_γ is the number of ways of selecting $\vec{x}^N$ and $\vec{p}^N$ that give energy E. For the canonical ensemble, the weight of a microstate is

$$w_{N,V,T}(\gamma) = \Omega_\gamma \exp\left(\frac{-(U(\vec{x}^N) + K(\vec{p}^N))}{k_B T}\right) \tag{1.2}$$

where $k_B T$ is Boltzmann's constant, $U(\vec{x}^N)$ is the potential energy and $K(\vec{p}^N)$ is the kinetic energy. The sum over all microstate weights for an ensemble is known as the *partition function*, and for the canonical ensemble the partition function is

$$Q(N, V, T) = \int \exp\left(\frac{-(U(\vec{x}^N) + K(\vec{p}^N))}{k_B T}\right) d\vec{x}^N d\vec{p}^N \tag{1.3}$$

and the probability of observing a particular microstate is

$$P(\gamma) = \frac{w(\gamma)}{Q}. \tag{1.4}$$

Evaluating or approximating a system's partition function is a central challenge of statistical mechanics, as it can be used to derive all of the system's physical properties, including heat capacity, compressibility, and structure. The contributions to Q from

$\vec{p}^N$ can be calculated analytically by

$$K(\vec{p}^N) = \sum_{i=1}^{N} \frac{p_i^2}{2m_i} \tag{1.5}$$

where i represents the index of a particle. Determining contributions from the configurational portion of the partition function

$$Z(N, V, T) = \int \exp\left(\frac{-U(\vec{x}^N)}{k_B T}\right) d\vec{x}^N \tag{1.6}$$

which is also expressed as a sum over microstates

$$Z(N, V, T) = \sum_{\gamma} \Omega_\gamma \exp\left(\frac{-U_\gamma}{k_B T}\right) \tag{1.7}$$

is much more challenging than calculating K because changing the position of one particle x_i might also change the potential energies of every other particle in the system. This renders $Z(N, V, T)$ analytically intractable for all but the simplest systems.

Computer simulations provide a convenient way to approach analytically intractable problems and advances in computational hardware and algorithms have enabled studies of systems ranging from catalyst kinetics [57] and oil-water mixtures[58], to adaptive cognitive systems [59, 60]. Simulations that numerically integrate (1.6) have played an important role in the study of self-assembly since their use in 1953 to calculate the hard disk equation of state [61]. Other early Monte Carlo simulations predicted flocs in colloidal systems[62] and configurations of polymer chains[63]. Early molecular dynamics simulations considered hard spheres [64] and configurations of model protein molecules[65]. Recent simulations that apply statistical mechanics to problems in self-assembly have elucidated control over tethered nanoparticle structure[66–70], quasicrystal formation[71, 72], and how interaction anisotropy can be used to control structure[3, 73, 74]. Exploiting new computational hardware such as graphics processing units (GPUs)[75] and implementing new techniques on them[76, 77] will advance our ability to engineer new materials using self-assembly.

1.3 Overview

In this work we present new theoretical developments and computational tools that are used to study the self-assembly of complicated particles. In Chapter II we review Monte Carlo methods that are used to perform self-assembly simulations in subsequent chapters and present new work detailing Monte Carlo parallelization for GPUs. In Chapter III we compare techniques for studying anisotropic tetromino self-assembly and develop a new tool, bottom-up building block assembly (BUBBA), for studying sequences of clusters. In Chapter IV we extend BUBBA to systems of building blocks with continuous degrees of freedom and nonzero temperatures and use it to predict free energy minimizing structures for a set of patchy particles. In Chapter V we demonstrate how BUBBA is useful for designing patchy particles that have a good chance of assembling a target structure by using it to inform the engineering of assembly pathways. In Chapter VI we summarize systems of particles for which BUBBA has been used successfully. In Chapter VII we conclude and present future avenues of research.

CHAPTER II

Methods

In this chapter we discuss the details of the computational tools used throughout this work. This chapter is organized as follows. In Section 2.1 we detail the Metropolis Monte Carlo simulation scheme. In Section 2.2 we explain the implementation of cluster moves used extensively in our Monte Carlo simulations. In Section 2.3 we prove constraints for checkerboard domain decomposition schemes used to perform Monte Carlo simulations on GPUs.

2.1 Monte Carlo Simulations

The naïve way of numerically integrating (1.6) is to generate configurations randomly, calculating their weights according to $\Omega_\gamma \exp(-U_\gamma/k_B T)$, and summing them. Unfortunately, this method is impractical, as for even a collection of 100 indistinguishable particles occupying sites on a 50×50 lattice there are over 6.67×10^{181} possible configurations to check. It is important to note that many configurations might not be physically realizable, such as those where two particles share a site, and that others might be substantially more energetically favorable than others. The Metropolis sampling scheme [61] avoids unnecessarily checking thermodynamically irrelevant configurations by generating a sequence of configurations that asymptotically approaches the underlying equilibrium distribution to be sampled. This is accomplished through the careful generation and acceptance of trial moves which adhere to the following constraints:

1. The trial moves are Markovian
2. The system is ergodic
3. Detailed balance is maintained

Here we denote the initial configuration for a system as γ_0 and the configuration after t trial moves have been attempted as γ_t. The Markovian constraint for a Monte Carlo simulation is satisfied when γ_{t+1} depends only upon γ_t. Another way of stating this is that the system has no memory; the next configuration depends only upon the current configuration. The ergodicity constraint is satisfied if every possible configuration can be reached from any other configuration by a finite number of trial moves. The ergodicity constraint ensures that only one distribution of states is asymptotically approached as the simulation progresses. The constraint of detailed balance ensures that at equilibrium, each elementary process is balanced by its reverse process. That is, the probability of observing state α and transitioning to state β is equal to the probability of observing state β and transitioning to state α:

$$P(\alpha)\pi(\alpha \to \beta) = P(\beta)\pi(\beta \to \alpha) \tag{2.1}$$

where $\pi(o \to n)$ is the probability of generating a trial move from old state o to new state n. As we shall see in Section 2.3 and Chapter III, violating any of these conditions can introduce sampling artifacts.

Simulations that adhere to the above constraints can be elegantly concise when implemented in computer code and have low memory footprint. Coupled with the fact that for many systems, equilibrium distributions can be sampled in minutes on modern processors, the Metropolis Monte Carlo scheme is an attractive candidate for many materials simulations[78–80]. However, it can be the case that for complex materials systems, especially patchy particles, additional enhanced sampling techniques may be required to ensure ergodic sampling over the time scales we are willing to wait.

2.2 Detailed Balance-Obeying Cluster Moves

One problem that arises is the critical slowing down that occurs in the vicinity of a phase transition. Simulations become effectively nonergodic near phase transitions because of the diminishing probability of accepting local moves. This problem has been resolved in the Ising model using simultaneous flipping of spin domains[81] and an extension that works on a single block of spins[82]. The rejection-free method was extended to continuous sysems[83, 84], which employs the mirroring of clusters about a central pivot point and enables enhanced sampling of dense systems. Non-rejection-free cluster moves have also been developed which aim to mimic physical dynamics[85], and which are particularly useful for systems of patchy particles which can experience long agglomeration times in the absence of cluster moves. In this section we detail our implementation of cluster moves used throughout this work. Many of the variables used in this section are chosen specifically for the lattice models used in later sections, but the general expressions remain the same for continuous systems.

We begin by expanding (2.1) with $P(\gamma) = \frac{e^{-\beta U_\gamma}}{Z}$

$$\frac{e^{-\beta U_o}}{Z} R_{gen}(o \to n) R_{acc}(o \to n) = \frac{e^{-\beta U_n}}{Z} R_{gen}(n \to o) R_{acc}(n \to o) \tag{2.2}$$

where $\beta = \frac{1}{k_b T}$, $R_{gen}(\alpha \to \beta)$ is the rate of generation of a trial move, and $R_{acc}(\alpha \to \beta)$ is the probability of accepting that trial move. Rearranging:

$$\frac{R_{acc}(o \to n)}{R_{acc}(n \to o)} = e^{-\beta(U_n - U_o)} \frac{R_{gen}(n \to o)}{R_{gen}(o \to n)} \tag{2.3}$$

Generating a trial move involves selecting a particle at random for use as a cluster seed, generating (growing) a cluster from the seed, and selecting a trial displacement or rotation at random. In every state, the probability of any particle being chosen as the seed particle is simply $\frac{1}{N}$ and the probability of choosing a particular move (translating in one of the four cardinal directions or rotating clockwise or anticlockwise) is $\frac{1}{6}$. As these two factors do not depend upon the state, (2.3) reduces to:

$$\frac{R_{acc}(o \to n)}{R_{acc}(n \to o)} = e^{-\beta(U_n - U_o)} \frac{P^n_{gen}(C|s)}{P^o_{gen}(C|s)} \tag{2.4}$$

where $P^{\gamma}_{gen}(C|s)$ is the probability of generating cluster C in state γ given the seed particle s. Cluster generation begins by defining a cluster C based upon seed particle s and calling a function we name NextInCluster that is defined in Algorithm 1.

Algorithm 1 Function NextInCluster(seed s, Cluster C) adds particles to a cluster in a depth-first search tree

```
Require: seed $s$, Cluster $C$
  for all particles $l$ that shares an edge with particle $s$ do
    if $l$ has not been added to a cluster then
      Add $l$ to cluster $C$ with probability $p_{s,l}$
      if $l$ is added to $C$ and the size of $C < S_{max}$ then
        NextInCluster($l$,$C$)
```

Here, the probability $p_{s,l}$ is the probability that particles s and l are part of the same cluster. This is also known as the bond probability. We define $p_{s,l} = 0.5$ if particles s and l share at least one edge and 0 otherwise. Recursively adding neighboring particles to a cluster in this way can generate very large clusters in dense areas. This, combined with the fact that cluster moves can be initiated from any particle means that large clusters are more likely to be selected in dense systems. To account for this, we terminate the growth of a cluster once it has reached a maximum size $S_{max} = \frac{1}{rand}$, where $rand$ is a random number chosen uniformly between 0 and 1. By sequentially growing clusters using the method mentioned above, the probability of creating a cluster C given seed particle s and the path chosen over its neighbors is the product of the $p_{k,l}$ over the path selected times the product of $(1 - p_{k,l})$ over the cluster's interface. However, there are many paths over which cluster C can be grown. Therefore, the probability of generating a cluster becomes:

$$P_{gen}(C|s) = \left(\sum_i \prod_{k,l \in C} p_{k,l} \right) \prod_{\substack{k \in C \\ l \notin C}} (1 - p_{k,l}) \tag{2.5}$$

where the sum over i is the sum of all paths that result in formation of cluster C. Note that this sum is the same for cluster C regardless of state, so (2.4) reduces to:

$$\frac{R_{acc}(o \to n)}{R_{acc}(n \to o)} = e^{-\beta(U_n - U_o)} \frac{\prod_{\substack{k \in C \\ l \notin C}} (1 - p^n_{k,l})}{\prod_{\substack{k \in C \\ l \notin C}} (1 - p^o_{k,l})} \tag{2.6}$$

where $p^{\alpha}_{k,l}$ is the probability that particles k and l are in the same cluster in state α. We take the minimum of (2.6) and unity to be the acceptance probability for a Monte Carlo move:

$$\mathrm{acc}(o \to n) = \min \left(1, e^{-\beta(U_n - U_o)} \frac{\prod_{\substack{k \in C \\ l \notin C}} (1 - p^n_{k,l})}{\prod_{\substack{k \in C \\ l \notin C}} (1 - p^o_{k,l})} \right). \tag{2.7}$$

For our on-lattice simulations where it is computationally inexpensive to determine particle nearest neighbors and hence grow clusters efficiently, this results in very efficient software. In an optimized version of the code used in Chapters III-V, our simulations achieve 1×10^5 cluster moves per second and the number of cluster moves per second is independent of the system size until the program memory exceed the CPU cache size (32 KB for one core of the Intel Core 2 Duo chips used here). Empirically we find optimal performance with fewer than 2000 particles in a 200 × 200 simulation box.

2.3 Monte Carlo on Graphics Hardware

Taking advantage of the high performance computing capabilities of graphics processing units has enabled dramatic improvements in performance for molecular dynamics simulations[75, 76, 86]. Achieving similar speedups for Monte Carlo simulations is attractive for the larger system sizes and expanded problem sets it could be applied to. Perhaps the central challenges in implementing a massively multithreaded Monte Carlo algorithm is implementing a domain decomposition scheme that obeys detailed balance. Implementations exist for asynchronous algorithms[87, 88], but these are not applicable to the simultaneous instruction multiple data (SIMD) architectures we target. Previous work on lattice models have demonstrated elements of domain decomposition, but do not obey detailed balance[89, 90]. Other work has claimed that detailed balance cannot be obeyed in parallel Monte Carlo[91], and demonstrate domain decomposition schemes[92, 93] that are alternatives to the checkerboard[87] approach. The checkerboard scheme is attractive for SIMD hardware because it optimally parallelizes trial moves and maps naturally to the GPU architecture. In this section we describe and implement a checkerboard domain decomposition scheme for NVIDIA graphics processing units and prove it adheres to detailed balance.[1]

We demonstrate and prove our algorithm using the 2×2 lattice model depicted in Figure 2.1. While the clarity of our proof is facilitated through the use of this lattice model, it applies to both continuous and lattice Monte Carlo models. The crux of the proof is showing that transitions from state-to-state are equally sampled in general, and that when certain transitions are occasionally blocked, their reverse moves are identically blocked. In our 2×2 lattice model we consider four sites arranged in a square with indexes 0 (lower left), 1 (lower right), 2 (upper left) and 3 (upper

[1] This section is adapted from Reference [94] J. Anderson, E. Jankowski, T. Grubb, M. Engel, and S.C Glotzer, Calculation of the hard sphere equation of state on graphics hardware. *in prep.*

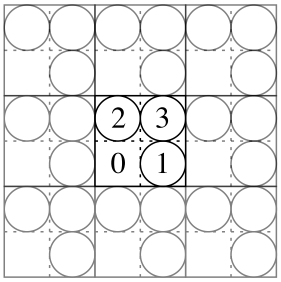


Figure 2.1: 2×2 square lattice model with periodic boundary conditions. Circles represent occupied lattice sites. With three occupied sites there are four possible microstates for indistinguishable particles, which are distinguished by the location of the vacancy. The microstate pictured shows the vacancy in site 0. Reproduced from Reference [94].

right) with periodic boundaries. Three of the lattice sites are filled with "particles", represented by circles in Figure 2.1, which have no interaction with other particles or the lattice, but which can move between neighboring sites. In this model, no site is more likely to be vacant than any other site, and we would expect a Monte Carlo simulation, whether serial or parallel, to show the probability of observing the vacancy in a given site to be $\frac{1}{4}$.

The commonly used "checkerboard" domain decomposition scheme used in parallel Monte Carlo implementations [87, 89] will be used here as well. In this scheme, the simulation volume is divided into square (or in three dimensions, cubic) cells, each of which can contain zero to n_p particles, where n_p depends upon the model and the size of the cell. In one *Monte Carlo sweep* (MCS) a Metropolis Monte Carlo move is attempted in each cell. In our 2D checkerboard scheme we decompose our simulation box into four cells, which results in one site being assigned to each cell. We assign each cell to a checkerboard set with an identifier $cid \in \{a, b, c, d\}$ based upon the x

and y coordinates of the cell:

$$\begin{aligned}
cid = a &: (x \in even) \text{ and } (y \in even) \\
cid = b &: (x \in odd) \text{ and } (y \in even) \\
cid = c &: (x \in even) \text{ and } (y \in odd) \\
cid = d &: (x \in odd) \text{ and } (y \in odd)
\end{aligned} \tag{2.8}$$

. One Monte Carlo sweep consists of four sub-sweeps; one sub-sweep for each of the four checkerboard sets. In a sub-sweep over one checkerboard set, trial moves are attempted for a randomly selected particle in each cell of the checkerboard in parallel. The spatial decomposition of our model depicted in Figure 2.2 is the simplest instantiation of the checkerboard scheme, with each checkerboard set containing one site.

The checkerboard domain decomposition scheme ensures that the parallel moves occurring simultaneously for the cells in one checkerboard set obey detailed balance. That is, so long as the interaction range and maximum trial displacement of a particle is less than half of the cell side length, each parallel trial move maintains local detailed balance within the sub-sweep. Complications arise, however, when a particle moves out of one cell and into a cell in which a trial move is being attempted on a subsequent sub-sweep within the same Monte Carlo sweep. As an example, consider the transition from state 4 to state 1 Figure 2.2. In state 4, the particle in site 0 will have a trial move applied to it by checkerboard a. Because the lattice site to the North and South (site 2) is occupied, there is a $\frac{1}{2}$ probability that a trial move will be generated and accepted that moves the particle into site 1. After moving to site 1, checkerboard b could move the particle. Note that here, “state” refers to the state of the model being implemented on the GPU at the onset of a Metropolis move kernel call, defined by particle positions and which checkerboard is attempting to perform moves in parallel. On the subsequent sub-sweep (state 1 in Figure 2.2) we must decide whether to allow

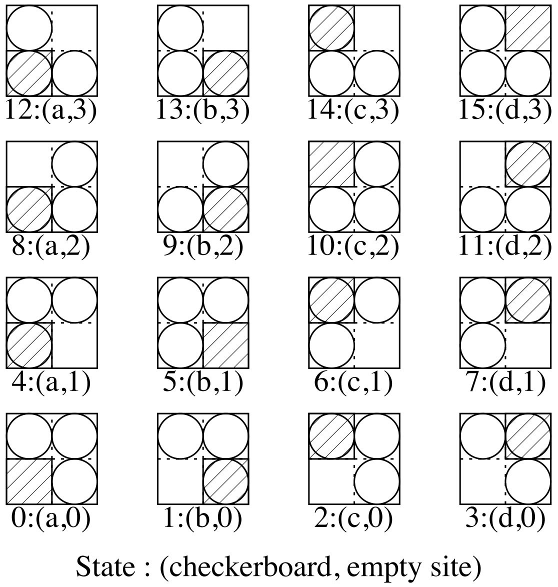


Figure 2.2: The checkerboard domain decomposition for the lattice model in Figure 2.1 has one checkerboard handling trial moves out of each of the four sites. The sixteen possible states that can occur during a MC sweep are enumerated above, with the active checkerboard $\{a, b, c, d\}$ indicated by a shaded square. Reproduced from Reference [94].

the particle that has just entered the cell to be eligible for a trial move, or not. In our off-lattice simulations we find that we must not allow particles that have entered during the current MCS to be eligible for trial moves, otherwise particles develop a systematic bias to be located near cell corners and edges, where they experience higher mobility. However, preventing newly-entered particles from participating in a trial move appears to violate the Markov condition: that only the current state of a system and not its history should determine its subsequent state. We shall show later that symmetric violation of the Markov condition can be used to ensure detailed balance for pairs of MCSs.

Given the four checkerboard indices and four possible locations of the vacant site, there are sixteen possible states that can be encountered during a sub-sweep in our model (Figure 2.2). Here we show that sequential iteration over the four checkerboards ($a \to b \to c \to d \to a$) during a MCS breaks detailed balance. Transitions from state to state occur via trial moves of particles out of the “active” checkerboard. The full set of possible transitions when checkerboards are updated sequentially ($a \to b \to c \to d \to a$) can be specified as a matrix

$$A = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & h & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & h & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 \\
0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 & h & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & h & 0 & 0 & 0 & h \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix} \tag{2.9}$$

where the elements $A_{i,j}$ indicate the probability of transitioning from state i to state j and $h = 0.5$. Here, i and j refer to the row and column indices of A, respectively, and are zero-indexed beginning in the upper left. For example, the “1” located at $i = 3$ and $j = 0$ indicates that sub-sweep state 0 will follow sub-sweep state 3 every time. Note that the sum of each row sums to 1 and that for this transition matrix we do not prevent reverse moves.

Given any initial condition vector v, the expected probability of observing state i

after n sub-sweeps can be calculated by

$$p = vA^n. \tag{2.10}$$

For sequential updates we calculate p numerically using

$$p^* = v(A^n + A^{n+1} + A^{n+2} + A^{n+3})/4 \tag{2.11}$$

to average over the four checkerboards. To determine the probability of cell 0 containing the empty site $P(0)$ we calculate p^* with $n = 1024$ using MATLAB. Summing $P(0) = p_0^* + p_1^* + p_2^* + p_3^*$ we find that $P(0) = \frac{2}{15}$. Analogously, we find $P(1) = \frac{11}{30}$, $P(2) = \frac{2}{15}$ and $P(3) = \frac{11}{30}$ for any legal v. These P show substantial bias for the vacant site to be located in cells 1 and 3, inconsistent with the expected $\frac{1}{4}$ probability. If we prevent reverse moves (the transition chains $4 \rightarrow 1 \rightarrow 6$ and $14 \rightarrow 11 \rightarrow 12$ are no longer legal) we find $P(0) = P(2) = 0.1739$ and $P(1) = P(3) = 0.3261$. This shows that sequentially performing checkerboard sub-sweeps fails to sample the expected probability distribution regardless of whether or not reverse moves are allowed.

Parallel checkerboard sub-sweeps can maintain detailed balance if, for each MCS, the order of checkerboard sub-sweeps are randomized and reverse moves are prevented. During a shuffled MCS, each checkerboard attempts a sub-sweep once, but the order in which the four checkerboards do so is determined each time using a Fischer-Yates shuffle[95]. By randomly shuffling checkerboard order for each MCS, the transition matrix A is expanded to

$$B = \begin{bmatrix}
0 & t & t & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s & 0 & s & s & s & 0 & s & s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
s & s & 0 & s & 0 & 0 & 0 & 0 & s & s & 0 & s & 0 & 0 & 0 & 0 \\
t & t & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & s & s & s & 0 & s & s & s & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & t & 0 & t & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & t & t & 0 & t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & s & s & s & 0 & 0 & 0 & 0 & 0 & s & s & s & 0 \\
0 & s & s & s & 0 & 0 & 0 & 0 & 0 & s & s & s & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & 0 & t & t & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & t & 0 & t & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s & s & s & 0 & s & s & s & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & t & t \\
0 & 0 & 0 & 0 & s & 0 & s & s & 0 & 0 & 0 & 0 & s & 0 & s & s \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & s & s & 0 & s & s & s & 0 & s \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & t & t & t & 0
\end{bmatrix} \tag{2.12}$$

where $t = \frac{1}{3}$ and $s = \frac{1}{6}$. For example, state 4 from Figure 2.2 can now transition to states 1, 2, 3, 5, 6, or 7 with probability $\frac{1}{6}$, to account for the three possible checkerboards to update next, times the two possible locations of the vacant site. Using transition matrix B in place of A in (2.11) we find $P(0) = P(1) = P(2) = P(3) = \frac{1}{4}$ as expected for this model when we prevent reverse moves.

Here, preventing reverse moves is essential for correct sampling. As mentioned above, failing to prevent reverse moves (moves where a particle moved from one checkerboard is moved by a subsequent checkerboard within one MCS) alters the structure of the system sampled, with particles having an unphysical spatial bias for the corners of the checkerboards in our 2D hard disk simulations. Preventing reverse moves is also essential for maintaining detailed balance for this scheme, as symmetric reverse move preventions cancel each other out. The sequences of states that are prevented during one MCS are

$$\begin{aligned}
4 &\to 1 \to \text{any} \\
1 &\to 4 \to \text{any} \\
2 &\to 8 \to \text{any} \\
8 &\to 2 \to \text{any} \\
13 &\to 7 \to \text{any} \\
7 &\to 13 \to \text{any} \\
14 &\to 11 \to \text{any} \\
11 &\to 14 \to \text{any}
\end{aligned} \tag{2.13}$$

with each prevented chain $a \to b \to x$ cancelling the chain $b \to a \to x$. The first line of (2.13) for our model can be read “if state 1 follows state 4, no trial move from state 1 will be accepted by the current sub-sweep”. The symmetry of these prevented moves is precisely the condition of detailed balance, showing that the rates of forward and reverse moves from any one state to any other are identical. However, because one reverse move might be prevented during a MCS, detailed balance is only strictly upheld over pairs of MCS. This proof contradicts the idea in [92] that domain decomposition Monte Carlo cannot be validated with traditional Markov chain theory. That is to say, by preventing reverse moves *and* shuffling the order over which checkerboards are updated, detailed balance is upheld over pairs of MC sweeps.

We validate our domain decomposition scheme by using it to simulate hard disks in two dimensions and comparing the pressure vs. density equation of state with that

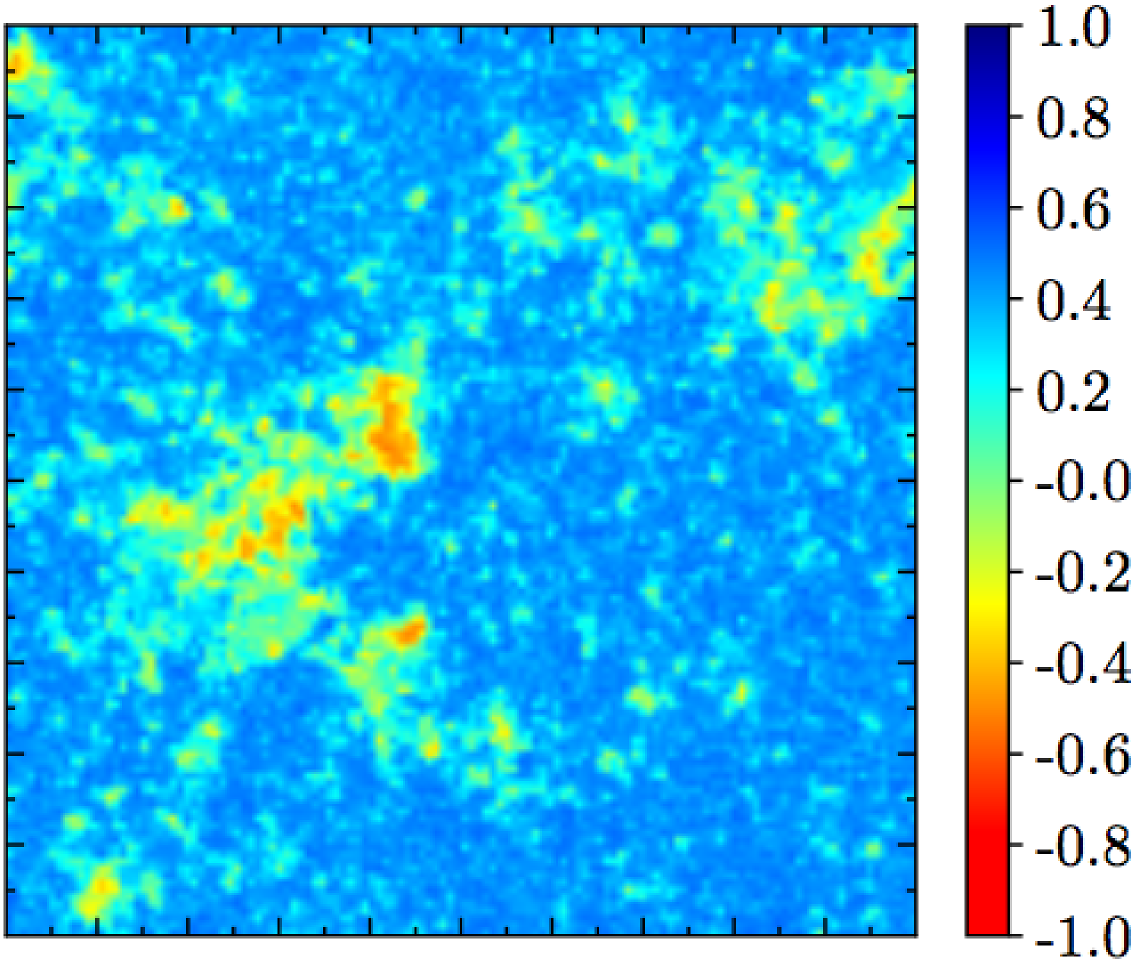


Figure 2.3: One million hard disk simulation snapshot run on an NVIDIA gpu using the domain decomposition described in this section. Blue areas indicate particles with hexatic ordering [96] and red indicates fluid. The packing fraction is 0.712. Reproduced from Reference [94].

calculated in Refernece [96]. The hard disk model is similar to the model described in Figure 2.1, but here disks have continuous degrees of freedom in the plane and one simulation cell can accommodate up to four disks. In our simulations, which take 23 days to run, we observe the same first order liquid-hexatic transition as observed in Refernece [96], but whose simulations take 9 months to run. A simulation snapshot of a one-million particle hard disk run is given in Figure 2.3.

2.4 Summary and Conclusions

In this chapter we describe the importance of detailed balance in providing a theoretical footing for equilibrium simulations. We prove two schemes that use advanced Monte Carlo moves do obey detailed balance, ensuring that the ensembles sampled by these methods are thermodynamic. The cluster moves used to speed up patchy particle self-assembly simulations are used extensively throughout this work and the GPU-enabled checkerboard scheme is being continuously improved and extended at the time of publication for this thesis.

CHAPTER III

BUBBA at Zero Temperature

In this chapter, we consider how increasingly complex particles are pushing the limits of traditional simulation techniques used to study self-assembly. We test the use of a learning-augmented Monte Carlo method for predicting low energy configurations of patchy particles shaped like "Tetris®" pieces (tetrominoes). We extend this method to compare it against Monte Carlo simulations with cluster moves and introduce a new algorithm - bottom-up building block assembly - for quickly generating ordered configurations of particles with a hierarchy of interaction energies.[1]

3.1 Introduction

Patchy particles, including those possessing intricate geometries and hierarchies of interaction energies, hold promise as the building blocks for next-generation materials and devices [3]. However, the complicated interactions between particles makes the prediction of their energy minimizing ordered arrangements challenging when using essentially all particle-based simulation methods rooted in statistical thermodynamics. This is particularly true of stochastic methods such as Monte Carlo (MC). Despite optimizations [98] that obey detailed balance such as cluster moves [84, 85], hybrid

[1] This chapter is adapted from Reference [97] E. Jankowski and S.C. Glotzer, A comparison of new methods for generating energy-minimizing configurations of patchy particles. *Journal of Chemical Physics*, 131(10):104104, 2009.

MC [99], and configurational bias methods [100, 101], MC simulations can become intractably long when used to predict energy minimizing configurations of patchy particles with disparate interaction energies because trial move acceptance probabilities become so low that the simulation becomes stuck in a sub-optimally organized, kinetically arrested state. Consequently, there is renewed interest in the development of simulation methods for strongly-interacting, complex molecular and particle-based systems [85, 102].

A new method proposed by Troisi *et al.* is an important step in this direction. The authors introduce an adaptive, learning-augmented Monte Carlo (LAMC) approach intended to prevent the formation of sub-optimal local arrangements of model complex building blocks shaped like"Tetris ®" pieces (Figure 3.1) [103]. On a 2D lattice, these building blocks are examples of lattice animals [104]. They record the energies of the best (i.e. lowest energy) clusters up to a maximum size, updating these values any time a new, lower energy cluster is found. Any cluster of particles selected for a trial move that does not have an energy consistent with the optimal value for its size is forced to split apart. Biasing the formation of low-energy clusters in this way, the simulation learns about successively better clusters as it proceeds. By moving a best-energy (i.e. lowest energy) cluster of a given size as a single unit, Troisi *et al.*'s LAMC benefits from two performance improvements over traditional MC. The first improvement is that trial moves breaking apart the best-energy clusters are never attempted until a better energy cluster is discovered. Second, the simultaneous translations and rotations of best-energy clusters help overcome the long agglomeration times that occur in MC simulations without cluster moves. As a result, their method generates very low temperature ordered structures faster than traditional Monte Carlo [103].

However, by not allowing cluster moves for sub-optimal clusters, the LAMC algorithm loses some efficiency in exploring phase space. It is tempting to think that a Monte Carlo simulation that includes cluster moves for all clusters in addition to

Troisi *et al.*'s learning-augmented cluster bias will find energy-minimizing configurations faster than either LAMC or cluster Monte Carlo (cMC). In cMC, the random translation that would normally be applied to a single particle during a trial move is applied to a group of particles (cluster) that are defined based upon their relative positions. Rotations of a cluster can also be included. Therefore, a simulation that combines cluster moves and Troisi *et al.*'s learning augmentation would be expected to overcome long agglomeration times better than LAMC and would utilize fewer trial moves that attempt to break optimal configurations than cMC. Thus, such a method could potentially be more efficient than cMC for generating low temperature states.

This chapter is divided into three main sections. In the first section we develop a new learning-augmented cluster Monte Carlo (LAcMC) method following the above line of reasoning and compare it against cMC. In the second section, we are inspired by Troisi *et al.*'s bottom-up cluster bias approach to develop a new algorithm - bottom up building block assembly (BUBBA) - for quickly generating energy-minimizing configurations of complex building blocks. In the final section we provide an analysis of each method's ability to find low energy structures and their relative efficiencies, and discuss the theoretical and practical limitations of each method. In all sections, we utilize the same model system, which is described in the next section.

3.2 Model System

To facilitate comparison with previous work, we consider systems of particles identical to those invented by Troisi *et al.* [103]. Each rigid tetromino particle is made up of four subunits and is shaped like a Tetris ®piece. Each subunit is defined to be positive, negative, or neutral. All nearest neighbor subunits interact with a weak van der Waals-like attraction that has a potential energy value of $U = -\epsilon$. Additionally, the interaction of nearest-neighbor charged subunits is $U = \pm 10\epsilon$, depending upon their

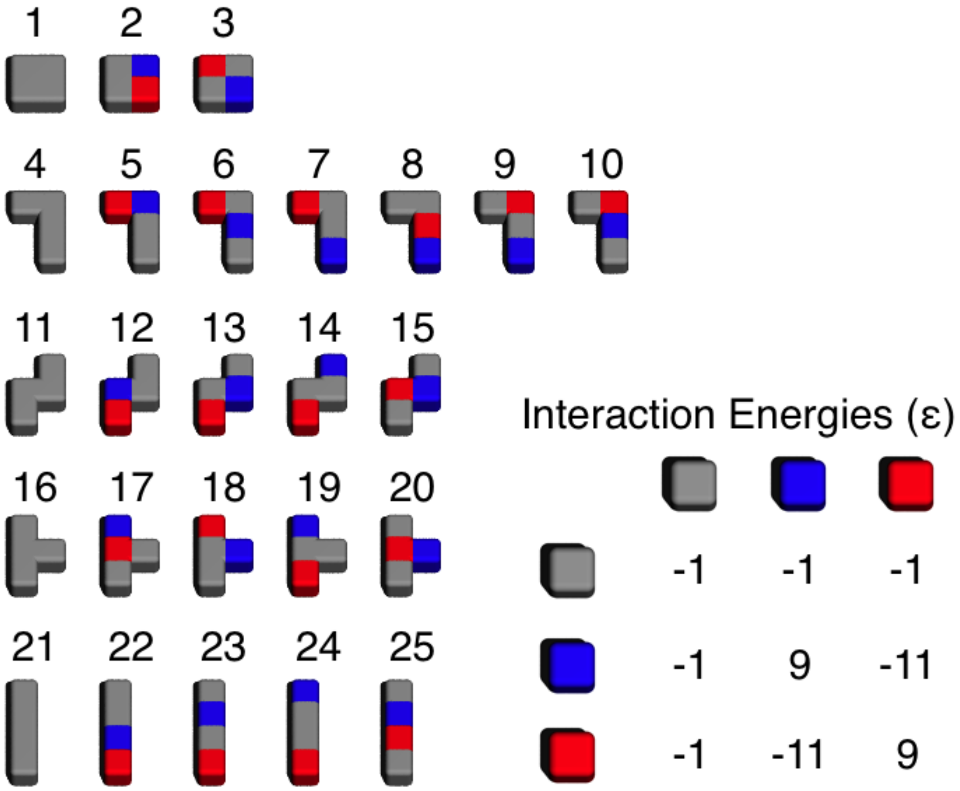


Figure 3.1: The 25 primitive particles studied in this work and in Troisi *et al.* [103]. Grey represents neutral subunits, red are positive, and blue are negative. Reproduced from Reference [97].

respective charges. As such, the energy associated with a neutral subunit interacting with any other subunit type is $-\epsilon$, that of oppositely charged subunits is $U = -11\epsilon$, and the energy between like charges is $U = +9\epsilon$. The 25 different primitive particle types studied here and by Troisi et *et al.* are shown in Figure 3.1. These building blocks can translate and rotate on a two-dimensional square lattice, but particle overlaps, inversions, deletions, substitutions, and additions are prohibited. Extensions of the methods investigated here to 3D lattices, other lattice geometries, and off-lattice are possible but beyond the scope of this paper.

3.3 Learning Augmentation and Cluster Monte Carlo

The cMC simulation we develop for the particles in Figure 3.1 utilizes cluster moves similar to those used by Whitelam *et al.* [85] At a given time step i, particles k and l are considered part of the same cluster with probability $p^i_{k,l} = 0.5$ if particles k and l share an edge in time step i and $p^i_{k,l} = 0$ otherwise. To ensure the trial move for the cluster C selected in each time step maintains detailed balance, the trial move acceptance probability is

$$\mathrm{acc}(o \to n) = \min\left(1, e^{-\beta(U_n - U_o)} \frac{\prod_{\substack{k\in C\\ l\notin C}}(1 - p^n_{k,l})}{\prod_{\substack{k\in C\\ l\notin C}}(1 - p^o_{k,l})}\right) \tag{3.1}$$

where β is the inverse temperature $1/k_BT$, k_B is Boltzmann's constant, U_n is the potential energy of the new configuration, and U_o is the potential energy of the old configuration. A detailed derivation of (3.1) is included in Chapter II. The choice of bond probability $p_{k,l}$ for a simulation is arbitrary, but can have a significant impact on equilibration times [83, 85, 105]. Bond-probabilies based on interaction energies are employed in rejection-free cluster moves for spin glasses [81] and can be generalized to fluid systems [83], but are prone to unphysical kinetic traps when particle interactions are stronger than a few k_BT [85]. We choose a constant bond probability specifically because it avoids the unphysical kinetic traps associated with rejection-free cluster moves in the same way as Whitelam and Geissler's cleaving algorithm [85] and for its computational efficiency. Empirically, we find $p_{k,l} = 0.5$ strikes a balance between the growth of clusters (high $p_{k,l}$) and the rate at which their trial moves are accepted (low $p_{k,l}$). The details of our depth-first cluster generation algorithm are included in Chapter II. We find Monte Carlo simulations that include this implementation of cluster moves to be more efficient than those without, as expected. Figure 3.2 shows the trajectories of a cMC and standard MC simulation with identical parameters and initial conditions. For all 25 building blocks, the simulations with cluster moves

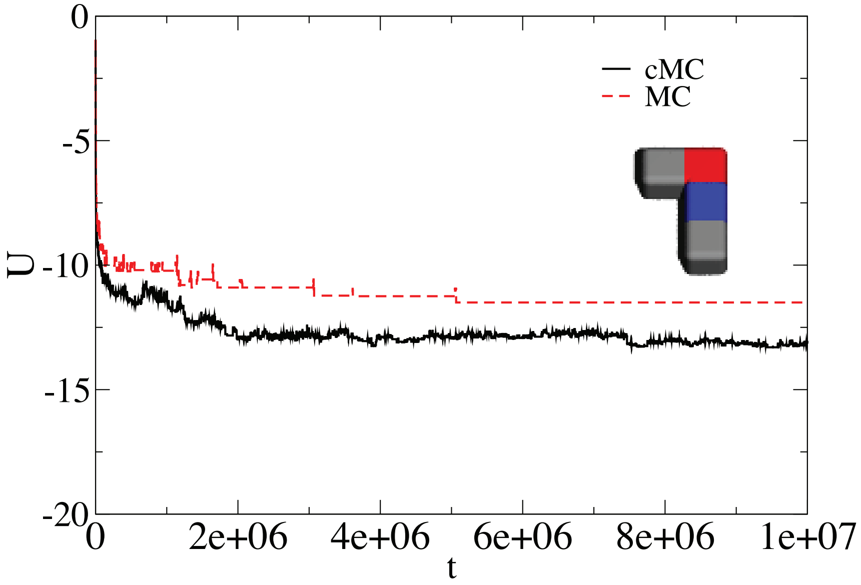


Figure 3.2: Per-particle potential energy profiles for cMC and MC simulations of patchy particle 10, with 40 particles in a 32 × 32 lattice at $k_BT/\epsilon = 1.4$. Reproduced from Reference [97].

sample lower energy configurations, again as expected.

Our LAcMC code is identical to cMC except we do not allow the best clusters in time step t to split apart in time step $t+1$. In LAcMC, we define clusters every time step as in cMC and compare the potential energy of each cluster with the optimal energy recorded for clusters of that size. If a cluster's potential energy is equal to the optimum for its size, we set $p_{k,l}^{t+1} = 1$ for all pairs of particles in the cluster. If the potential energy of a cluster is lower than the recorded optimum, we update the new optimum value. With this algorithm, our LAcMC model is identical to Troisi *et al.*'s LAMC method except that here clusters with less-than-optimal potential energies are allowed to move as whole units. Thus, LAcMC combines the energy-minimizing cluster bias of LAMC and the agglomeration-optimizing cluster moves of cMC.

For each of the 25 patchy particles in Figure 3.1 we perform 500 cMC simulations and 500 LAcMC simulations of 5×10^5 time steps, each with a unique random num-

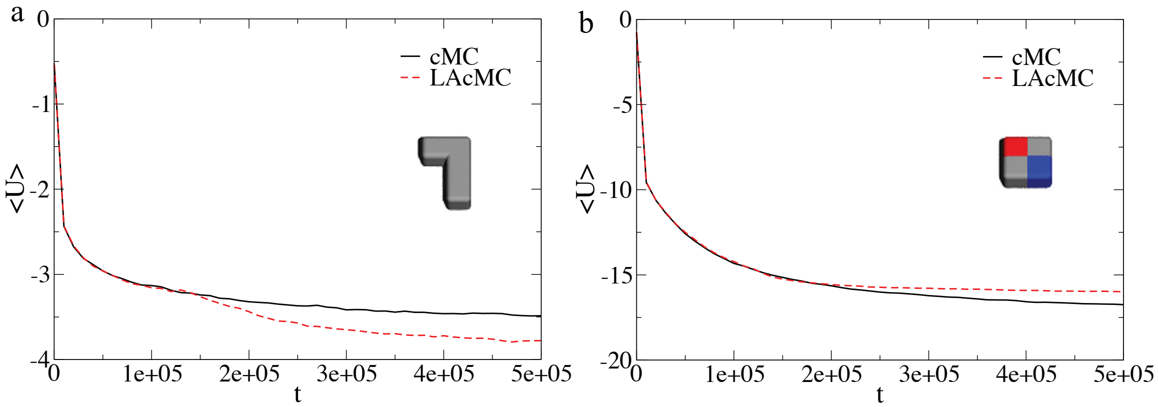


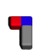
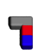
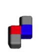
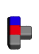
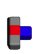
Figure 3.3: Per-particle potential energy histories of (a) patchy particle 4 simulations and (b) patchy particle 3 for cMC and LAcMC. Standard error bars are on the order of the line thickness. Reproduced from Reference [97].

ber generator seed. The simulations are performed on a 32 × 32 lattice with periodic boundary conditions initialized randomly with $N = 40$ identical particles. The temperature is set to $k_BT/\varepsilon = 0.06$ for the patchy particles 1, 4, 11, 16, and 21. For patchy particles 6, 10, 15, 17, and 20, the temperature is set to $k_BT/\varepsilon = 1.4$. The reduced temperature for all other patchy particles is set to $k_BT/\varepsilon = 2.2$. These temperatures are chosen to be low enough to facilitate agglomeration, but high enough to prevent the system from becoming trapped in an arrested state, and were determined by trial and error.

To compare the rates at which cMC and LAcMC converge upon potential energy minima we average the per-particle potential energy histories of the 500 trials for each method. Figure 3.3a presents the histories for patchy particle 4, where LAcMC finds lower energy configurations on average. For patchy particle 3, however, the opposite result is obtained with cMC outperforming LAcMC (Figure 3.3b). Table 3.1 tabulates whether LAcMC performs on average better than (-), worse than (+), or the same as (o) cMC for each patchy particle. In general we find no obvious correlation between particle shape or patterning and whether or not LAcMC will tend to find better-optimized configurations than cMC without learning.

Because LAcMC is an extension of cMC, it is natural to ask how the addition

Table 3.1: Impact of learning augmentation on average final potential energy after 500,000 time steps of cMC. A "+" indicates LAcMC finds higher (worse) U configurations than cMC, "- " indicates lower (better), and "o" indicates no difference.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
																								
-	-	+	-	-	+	+	+	-	o	+	+	-	+	+	-	-	-	o	-	-	-	-	+	-

of an adaptive element alters the ensemble sampled. Figure 3.4a presents a comparison of the cluster size distributions for patchy particle 23 when simulated with learning-augmented and non learning-augmented cluster Monte Carlo. This distribution is representative of those found for all 25 patchy particles. The shift in LAcMC towards larger clusters demonstrates that it is not sampling the same distribution of microstates as cMC, since the cluster size distributions must be a property of the state point. This discrepancy is expected, since the algorithm is designed to bias towards low-energy (that is, large) clusters, but highlights the fact that LAcMC cannot be used to sample a thermodynamic ensemble away from zero temperature.

Furthermore, any algorithm similar to LAMC or LAcMC in which the system "learns" can never be tuned to sample a thermodynamic ensemble. The non-Markovian nature of this type of algorithm, where the transition made from time step t to $t+1$ depends not only on the state at t, but on a history of states, precludes the condition of balance necessary for ergodic sampling [106]. There is no variant of Eqn 3.1 that can rectify the state sampling differences caused by learning augmentation. However, the fact that LAcMC cannot be used at finite temperature is an acceptable shortcoming so long as it outperforms standard methods in the low temperature energy-minimizing case, where standard methods typically fail.

In Table 3.2 we compare the lowest-energy configuration found by LAcMC with those found by cMC, averaged over 500 runs for each method. LAcMC does not reproducibly find lower energy structures than cMC, nor does it find them in fewer

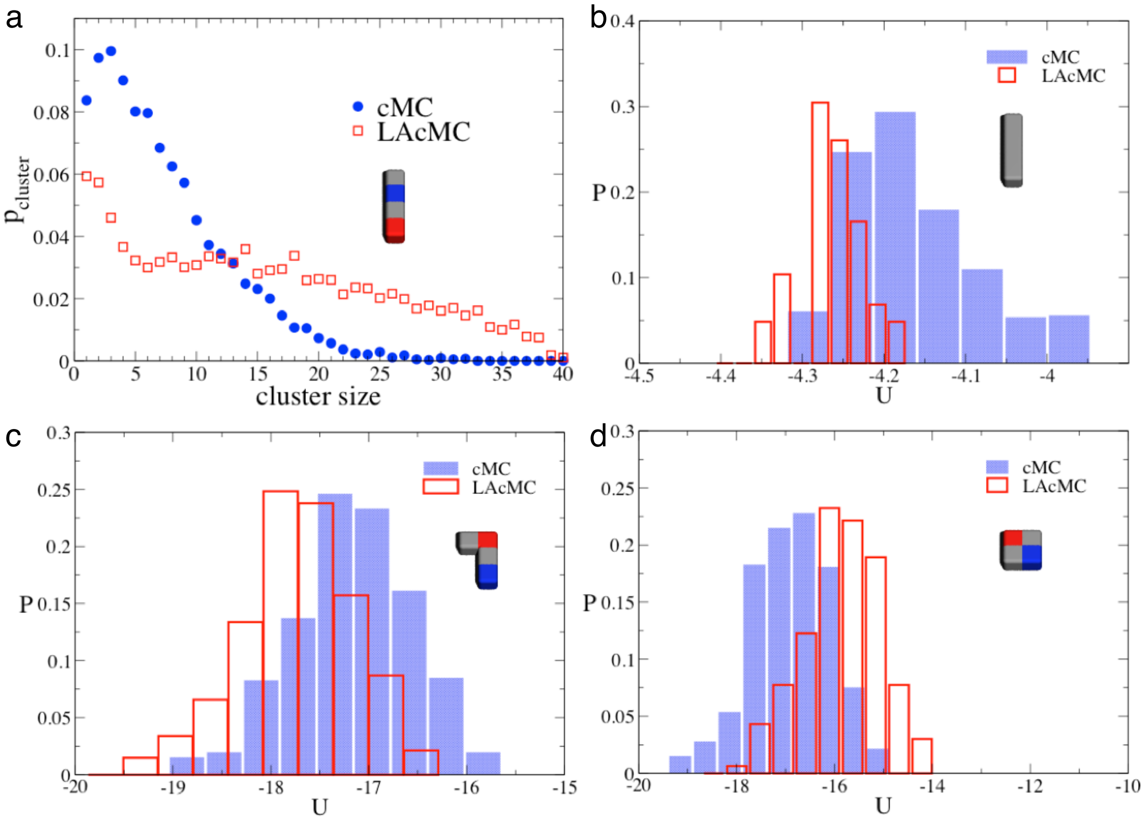


Figure 3.4: Learning augmentation changes the distribution of cluster sizes during a simulation, as well as the distribution of energies sampled. a. Cluster size distribution for patchy particle 23 from simulations with $N = 40$, $k_BT/\varepsilon = 0.1$ on a 32×32 lattice. b. Distributions of the best energy configuration found for patchy particle 21. c. Distributions of the best energy configuration found for patchy particle 9. d. Distributions of the best energy configuration found for patchy particle 3. Reproduced from Reference [97].

timesteps (Table 3.3). Table 3.1 and Table 3.2 are in close agreement, as we would expect: if the average configurations generated by LAcMC for a given patchy particle have lower energies than those generated by cMC, then the lowest-energy configurations generated by LAcMC are also lower than the best structures generated by cMC. However, the averages and standard errors presented in Table 3.2 fail to paint the full picture of how LAcMC changes the distribution of configurations sampled. Figure 3.4b demonstrates that LAcMC samples a much narrower distribution of energies with a lower average for patchy particle 21. In this case, the fact that learning augmentation improves the sampling of lower energy configurations is lost in the averages of Table 3.2, but clearly appears in the side-by-side comparison of the energy distributions sampled by each method.

For some particles (e.g., patchy particle 9), the distribution of best-energy configurations is shifted lower than that for cMC, indicating that the cluster biasing facilitates finding lower energy structures (Figure 3.4c). For other patchy particles (e.g, patchy particle 3), the distribution of best-energy configurations found by LAcMC is worse than that found by cMC (Figure 3.4d). However, regardless of learning augmentation's ability to find lower energy configurations, the cluster size distribution is shifted as in Figure 3.4a for all patchy particles. No general improvement is found for LAcMC over cMC.

While LAcMC and cMC take statistically similar amounts of time steps to find best-energy configurations (Table 4.1), LAcMC offers a small amount of real-time performance improvement. Because energy-minimizing clusters in time step t are maintained in time step $t + 1$, the number of clusters that need to be generated in later time steps are substantially decreased. Because of this, a cMC simulation of 5×10^5 time steps takes on average 52.4 seconds on a 2.8Ghz iMac, and an identical LAcMC simulation takes an average of 39.3 seconds.

Patchy Particle	$\langle U_{cMC} \rangle/\epsilon$	$\langle U_{LAcMC} \rangle/\epsilon$	ΔU
1	-3.27 ± 0.04	-3.29 ± 0.03	-0.02 ± 0.05
2	-18.79 ± 0.72	-18.93 ± 0.68	-0.13 ± 0.99
3	-17.13 ± 0.84	-16.10 ± 0.79	1.03 ± 1.15
4	-3.86 ± 0.13	-4.12 ± 0.08	-0.26 ± 0.15
5	-17.75 ± 0.56	-17.83 ± 0.52	-0.08 ± 0.76
6	-10.51 ± 0.32	-10.13 ± 0.31	0.38 ± 0.45
7	-20.16 ± 0.63	-19.87 ± 0.59	0.30 ± 0.86
8	-18.82 ± 0.83	-18.61 ± 0.77	0.20 ± 1.13
9	-17.34 ± 0.63	-17.90 ± 0.60	-0.56 ± 0.87
10	-11.90 ± 0.36	-11.93 ± 0.35	-0.03 ± 0.50
11	-3.83 ± 0.12	-3.74 ± 0.12	0.09 ± 0.17
12	-19.83 ± 0.56	-19.09 ± 0.60	0.74 ± 0.82
13	-14.35 ± 0.81	-14.19 ± 0.68	0.16 ± 1.05
14	-16.88 ± 0.62	-15.95 ± 0.66	0.93 ± 0.91
15	-11.44 ± 0.32	-11.31 ± 0.32	0.13 ± 0.45
16	-3.84 ± 0.13	-3.74 ± 0.12	0.10 ± 0.18
17	-11.06 ± 0.46	-11.34 ± 0.48	-0.27 ± 0.67
18	-13.58 ± 0.78	-14.08 ± 0.54	-0.50 ± 0.95
19	-18.36 ± 0.73	-18.42 ± 0.75	-0.06 ± 1.05
20	-10.34 ± 0.44	-10.54 ± 0.40	-0.20 ± 0.60
21	-4.19 ± 0.09	-4.28 ± 0.04	-0.09 ± 0.10
22	-16.93 ± 0.61	-17.30 ± 0.58	-0.37 ± 0.84
23	-17.72 ± 0.66	-18.06 ± 0.69	-0.34 ± 0.96
24	-21.25 ± 1.02	-20.57 ± 0.91	0.68 ± 1.37
25	-17.90 ± 0.72	-18.37 ± 0.77	-0.46 ± 1.05

Table 3.2: Average lowest-energy configurations for different patchy particles for cMC and LAcMC. $\Delta U = \langle U_{LAcMC} \rangle/\epsilon - \langle U_{cMC} \rangle/\epsilon$.

Patchy Particle	$\langle ts_{cMC} \rangle$	$\langle ts_{LAcMC} \rangle$	Δts
1	255 ± 130	265 ± 113	10.2 ± 172.2
2	378 ± 97	379 ± 98	1.2 ± 138.0
3	380 ± 97	298 ± 117	-82.6 ± 152.1
4	360 ± 103	398 ± 76	38.3 ± 128.3
5	382 ± 105	374 ± 101	-8.4 ± 145.4
6	380 ± 103	254 ± 126	-125.8 ± 162.7
7	407 ± 85	391 ± 90	-16.4 ± 123.4
8	416 ± 75	419 ± 72	2.7 ± 104.4
9	414 ± 85	428 ± 65	14.0 ± 106.9
10	391 ± 95	400 ± 94	8.0 ± 134.0
11	336 ± 112	324 ± 120	-11.9 ± 164.8
12	404 ± 86	347 ± 111	-57.5 ± 140.9
13	412 ± 88	401 ± 77	-11.8 ± 116.5
14	396 ± 94	348 ± 127	-47.7 ± 158.4
15	399 ± 90	380 ± 103	-19.7 ± 137.1
16	359 ± 109	355 ± 113	-4.3 ± 157.5
17	432 ± 66	437 ± 64	5.4 ± 91.7
18	425 ± 76	423 ± 65	-1.9 ± 99.6
19	413 ± 77	427 ± 69	13.4 ± 103.4
20	403 ± 93	417 ± 74	14.4 ± 119.4
21	338 ± 106	354 ± 94	16.1 ± 141.6
22	410 ± 84	408 ± 74	-2.6 ± 112.0
23	393 ± 91	384 ± 88	-8.5 ± 126.3
24	400 ± 92	372 ± 103	-28.1 ± 138.7
25	416 ± 74	416 ± 70	-0.3 ± 101.6

Table 3.3: Average number of time steps (in thousands) to find the lowest-energy structures in cMC and LAcMC simulations. $\Delta ts = \langle ts_{LAcMC} \rangle - \langle ts_{cMC} \rangle$

3.4 Bottom-Up Building Block Assembly

In both the LAMC method developed by Troisi *et al.* and our LAcMC method, the expectation is that energy minimizing configurations of particles can be found more effectively by biasing the formation of low-energy clusters. However, we observe that adding a learning-augmented cluster bias component does not offer a significant performance increase over traditional methods. This does not imply that learning-augmented methods can not be used or that they should be avoided. In BUBBA, we take this idea of learning and cluster biasing one step further and build energy minimizing clusters from pairs of smaller clusters from the bottom up. To generate an energy minimizing cluster of size N from smaller best-energy clusters, BUBBA employs Algorithm 2. This algorithm generates the lowest energy clusters of size N by checking all configurations for every pair of clusters whose sizes sum to N. Here, we consider a simplification of BUBBA where only pairs of lowest energy clusters are used to build larger clusters. This implementation of BUBBA is efficient but will miss energy-minimizing clusters made from sub-optimal clusters and we return to this point in the Section 3.5.

Table 3.4 presents the per-particle potential energies of the size $N = 40$ clusters generated by BUBBA for each of the 25 tetrominoes alongside the lowest values obtained from cMC and LAcMC. For all patchy particles except 1 and 21, BUBBA finds the lowest energy configurations of 40 particles. For patchy particles 1 and 21, the best-energy configurations of all three methods are identical. BUBBA also offers a substantial performance improvement in terms of computational time. It takes 4 seconds to generate the patchy paticle 8 energy-minimizing cluster of size 40 in BUBBA on a 2.8Ghz iMac. The times for cMC and LAcMC simulations of 5×10^5 time steps are 39.3 and 52.4 seconds, respectively. Additionally, while a single BUBBA run is an order of magnitude faster than a cMC or LAcMC trial run, hundreds of simulations must be run for the Monte Carlo methods in order to reduce

Algorithm 2 Generate clusters of size N

```
Require: generation of clusters from size 1 through $N - 1$
  for all $s$ such that $\frac{N}{2} \leq s \leq N - 1$ do
    for all clusters $y \in clustersOfSize[s]$ do
      for all clusters $m \in clustersOfSize[N - s]$ do
        $B \leftarrow 0$
        calculate perimeter of $y$
        for all cells $cy$ in perimeter of $y$ do
          for all cells $cm$ in cluster $m$ do
            move $m$ such that $cm = cy$
            for all orientations do
              $E \leftarrow energy(m, y)$
              if $E < B$ then
                $B \leftarrow E$
                replace $bestClustersOfSize[N]$ with $cluster(m, y)$
              if $E = B$ and $cluster(m, y) \notin bestClustersOfSize[N]$ then
                add $cluster(m, y)$ to $bestClustersOfSize[N]$
  return $bestClustersOfSize[N]$
```

their standard error. This makes BUBBA computationally much more attractive than either Monte Carlo method for finding potential energy minima, provided that only best-energy, or second-best energy clusters are required to build the best energy structure. This point is discussed in more detail in the next section.

3.5 Discussion

The fundamental differences between Monte Carlo methods and BUBBA account for the large performance gap we have demonstrated. Finding a configuration of particles that minimizes a system's potential energy is an optimization problem. Monte Carlo methods are ill-suited for optimization problems for two reasons. First, at low temperatures the trial move transition probabilities become diminishingly low. Second, the number of energy-minimizing configurations for a given system can be as low as one. For the systems we consider here, the number of energy minimizing configurations depends upon the shape and patterning of the building blocks. For example, there are two ways to put two copies of patchy particle 11 together to create a cluster

Patchy Particle	U_{cMC}/ϵ	U_{LAcMC}/ϵ	U_{BUBBA}/ϵ
1	-3.35	-3.35	-3.35
2	-21.18	-20.80	-22.40
3	-19.80	-18.98	-20.10
4	-4.20	-4.30	-4.35
5	-19.40	-19.40	-21.15
6	-11.38	-11.15	-11.85
7	-22.58	-21.53	-24.70
8	-21.33	-20.78	-23.40
9	-19.50	-19.95	-22.15
10	-12.95	-12.95	-14.13
11	-4.13	-4.08	-4.25
12	-21.73	-21.03	-23.35
13	-16.68	-16.05	-18.05
14	-19.03	-18.53	-20.88
15	-12.43	-12.40	-13.33
16	-4.23	-4.10	-4.35
17	-12.30	-12.63	-14.10
18	-16.10	-15.98	-18.03
19	-20.20	-20.60	-25.68
20	-11.80	-12.15	-12.83
21	-4.33	-4.35	-4.35
22	-18.43	-18.80	-22.38
23	-19.60	-19.95	-22.83
24	-24.95	-23.33	-29.80
25	-20.13	-20.73	-23.40

Table 3.4: Lowest energy configurations generated with cMC, LAcMC and BUBBA for 40 identical copies of each particle in Figure 3.1.

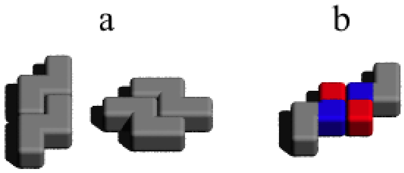


Figure 3.5: a. Best energy clusters of size 2 for patchy particle 11. b. Best energy cluster of size 2 for patchy particle 12. Reproduced from Reference [97].

with the minimum energy (Figure 3.5a). Patchy particle 12, however, only has one energy-minimizing configuration of two particles (Figure 3.5b). As the clusters grow larger, the degeneracy of configurations increases for the homogenous particles (1, 6, 11, 16, and 21), but may remain flat for certain other particles (e.g. patchy particle 8 only has one energy minimizing configuration for every cluster of size $s < 48$).

Optimization problems, such as the one we consider here, can be classified as "easy" or "difficult" depending on the optimal solution degeneracy. Mertens proves that the number of solutions depends upon the complexity of the building blocks [107]. Mertens' proof uses a mapping of the number partitioning problem (NPP) onto an Ising model. Using this mapping, it has been shown that the NPP has two regions: a so-called "easy" region where an instance of the problem has many optimal solutions and a "difficult" region in which there is only one solution [107, 108]. The goal of the NPP is to separate a list of integers into two lists with the smallest difference in sum. If all of the numbers in the list are equal, the problem is trivial: two lists of closest size will have the smallest difference. A list of numbers with widely-varying values, however, is much harder to solve. The transition between "easy" and "difficult" regions depends upon the ratio m/n where m is the number of bits needed to encode a number in the list and n is the size of the list. When $m/n < 1$ there are many optimal solutions to the problem, otherwise there is only one in the limit of large n. While our systems cannot be mapped easily onto the NPP, we may expect similar trends in optimization problem complexity. That is, we should expect the patchy particles requiring the fewest bits to encode (particles 1, 4, 11, 16, and 21) to fall into an “easy” regime relative to the other more complex building blocks. We would therefore expect the Monte Carlo methods to find energies closest to those found by BUBBA for shapes 1, 4, 11, 16 and 21 while performing relatively worse for all other shapes. Table 3.4 shows that the relative energy differences between the Monte Carlo methods and BUBBA are indeed closest for shapes 1, 4, 11, 16, and 21, which is consistent with the “easy”-“difficult” heuristic of Mertens.

BUBBA is effective relative to Monte Carlo methods because it checks a smaller number of configurations. In d dimensions, a cluster made up of n subunits has at most $2dn$ perimeter cells. Therefore, when a cluster of n_1 subunits is combined with a cluster of n_2 subunits to form the best energy cluster with $N = n_1 + n_2$ subunits, the number C of configurations checked is

$$C = 2dn_1n_2 \tag{3.2}$$

$$= 2d(N - n_2)(N - n_1) \tag{3.3}$$

$$= 2d(N^2 - Nn_1 - Nn_2 + n_1n_2) \tag{3.4}$$

giving

$$O(C) = O(N^2) \tag{3.5}$$

There are at most $N/2$ pairs of clusters that must be checked to find the best-energy cluster of size N. Also, all of the clusters up to size $N-1$ must be generated before the best-energy cluster of size N can be found. There are $N-2$ such clusters, with sizes 2 through $N-1$. Performing the $O(N^2)$ cluster pairing $N(N-2)/2$ times results in the total number of configurations on the order $O(N^4)$ regardless of volume or the number of spatial dimensions. While a cMC simulation does not attempt to sample all of the $O(V^{dN})$ possible system configurations (N particles in volume V with d spatial dimensions), the fact that BUBBA generates an energy-minimizing cluster after only $O(N^4)$ potential energy calculations shows how tiny a portion of configuration space it samples. By not checking every possible configuration, however, BUBBA is not guaranteed to find the global energy minimum for a set of particles. BUBBA becomes computationally inefficient when generating clusters larger than $N \cong 150$ on a single-processor machine, but we find the unit cells comprising the energy-minimizing infinite structure can be found when $N < 50$ for the family of particles studied here.

For the implementation of BUBBA we present here, it is guaranteed to fail if the best cluster of size s is formed by two clusters whose energies are not minima for their sizes. At the expense of computational cost, clusters with sub-optimal energies can easily be included in the search for energy minimizing clusters. We find that in most cases, BUBBA performs well when including only energy-minimizing clusters: generating optimal clusters for 24 of the 25 patchy particles in Figure 3.1. Only patchy particle 6 is improved by including second-best clusters. In this case, the best energy $N = 4$ cluster is made from two copies of the second-best energy $N = 2$ cluster (Figure 3.6). By including the second-best clusters for patchy particle 6, BUBBA generates a configuration with $U/\epsilon = -13.3$, lower than that found when only the best-energy clusters are included (Figure 3.6).

Given a system of particles at constant temperature, the stronger the interaction energies between two building blocks, the more likely there are to be trial moves with near-zero acceptance probabilities in any Monte Carlo simulation. For example, in a simulation of shape 12 with $k_BT/\varepsilon = 1.0$, the probability of breaking apart the cluster in Figure 3.5a is $exp(-22/1.0) = 2.9 \times 10^{-10}$. If such a trial move were required to rearrange a non-optimal cluster it would be highly unlikely over the course of a one-billion time step simulation. Furthermore, this low acceptance probability is approaching the precision of the random number generator, 2^{-b}, where $b = 31$ in our simulations [109]. We find that the stronger the interaction energy between two

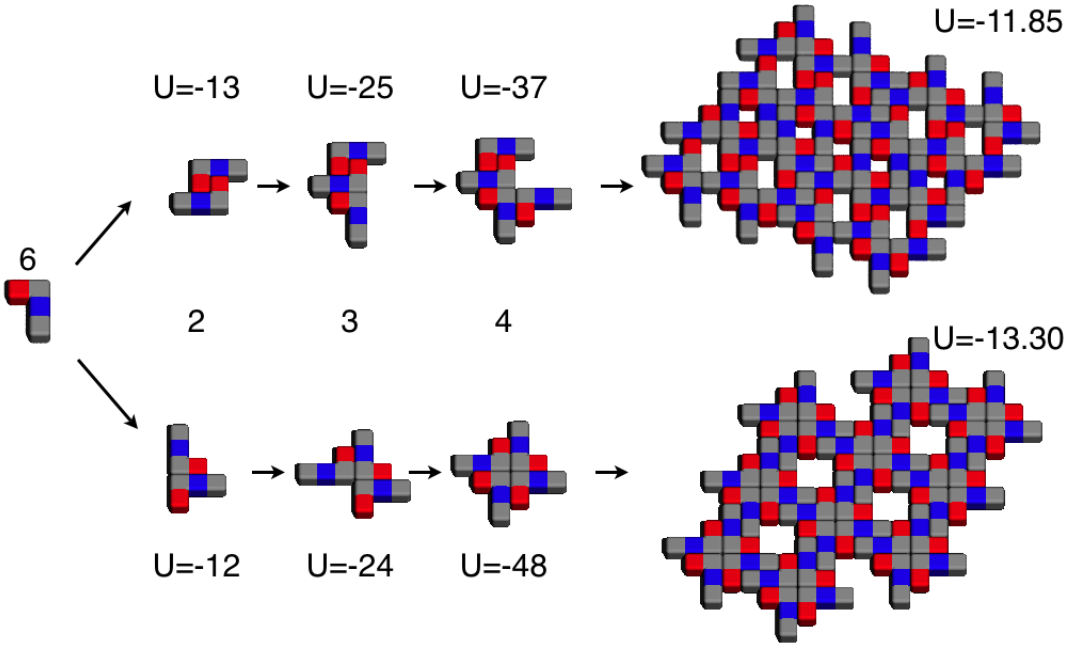


Figure 3.6: The best energy $N = 4$ cluster for patchy particle 6 can be generated from sub optimal clusters of size 2 or 3. Reproduced from Reference [97].

building blocks of a given shape relative to the weakest interaction between them, the more likely they are to become trapped. Here, we quantify a system's susceptibility of becoming trapped by comparing the average minimum potential energies found with cMC in Table 3.2 to those found by BUBBA. Figure 3.7 shows that larger r corresponds to larger performance gaps between BUBBA and cMC. This suggests that another factor contributing to BUBBA's efficacy is the fact that Monte Carlo simulations of patchy particles can easily become trapped in metastable configurations. Further, the ease with which a MC simulation of patchy particles can become trapped as measured by $\Delta = \frac{U_{BUBBA} - \langle U_{cMC} \rangle}{U_{BUBBA}}$ appears to increase as r increases.

3.6 Conclusions

Systems of patchy particles that have a hierarchy of interaction energies are complicated, limiting the extent to which traditional simulation methods can be used to predict ordered configurations arising via self-assembly. In this chapter we have

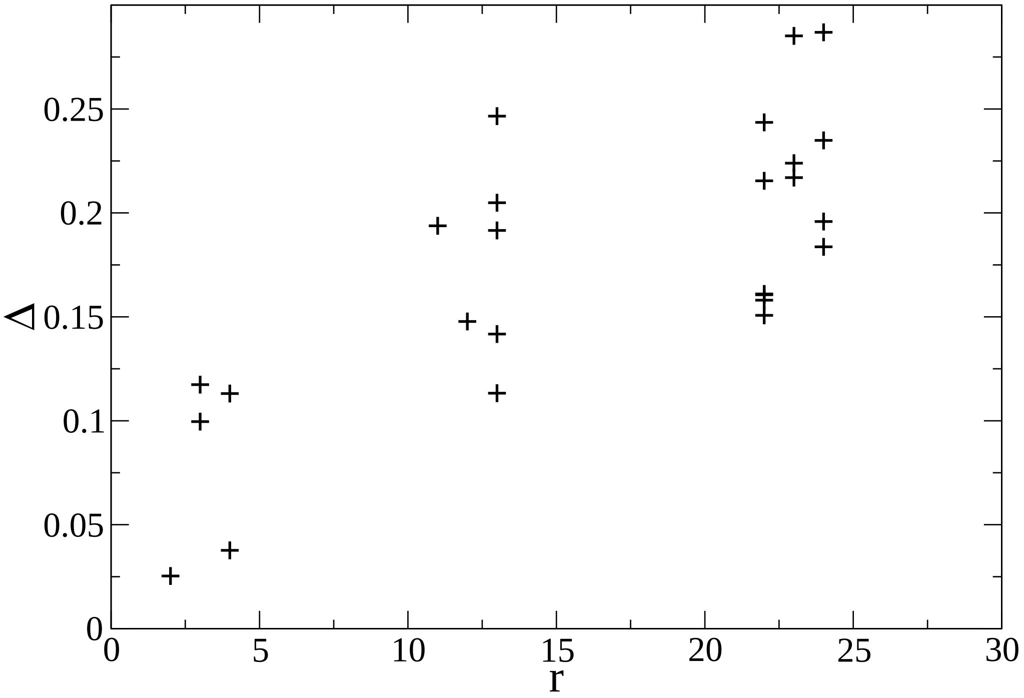


Figure 3.7: Normalized difference in energy, $\Delta = \frac{U_{BUBBA} - \langle U_{cMC} \rangle}{U_{BUBBA}}$, as a function of strongest relative two-particle interaction energy, $r = \frac{U_{strong}}{U_{weak}}$. Multiple shapes can have identical r values. Reproduced from Reference [97].

demonstrated that cluster Monte Carlo and learning-augmented cluster Monte Carlo methods are relatively poor methods for finding potential energy minima formed from patchy particles with disparate interaction energies due to their tendency to become trapped in metastable configurations as well as the low degeneracy of potential energy minimizing configurations. We have shown that learning-augmented cluster Monte Carlo offers insignificant performance improvements over cluster Monte Carlo, but stress that this does not invalidate the potential usefulness of such methods. We have developed a new method - BUBBA - that effectively searches a subset of configuration space for energy-minimizing configurations. While BUBBA, as well as any other heuristic method, is not guaranteed to find optimal solutions, we demonstrate its efficacy relative to cMC and LAcMC. Due to BUBBA's ability to quickly generate energy-minimizing configurations of particles, we expect it to prove useful when many different particles must be evaluated for self-assembly "propensity" [110].

CHAPTER IV

BUBBA at Nonzero Temperature

4.1 Introduction

Self-assembly holds promise as a manufacturing technique for materials with customizable physical properties and for devices with precisely placed nanoscale elements[1, 111]. Building blocks with complicated shapes and finely-tuned directional interactions can now be synthesized from polymers, metals, biomolecules, and combinations of these nanoparticles that have been chemically bound together to create highly anisotropic "patchy" particles[2–4, 13, 38, 111]. The specific interactions and complex shapes of patchy particles can be exploited during self-assembly with particles organizing into complicated patterns as a consequence of free energy minimization[112]. Choosing a building block to assemble a target pattern or assessing if a system of building blocks might assemble any pattern - the two basic problems of self-assembly - are difficult because of the enormous parameter space that is accessible when particle shape, interaction number, interaction strengths, temperature, density, and stoichiometry can each be varied independently. Methods that can predict what new particles might form and why they do or do not reach their thermodynamically preferred structures are therefore important and useful.[1]

[1] This chapter is adapted from Reference [113] E. Jankowski and S.C. Glotzer, Calculation of partition functions for the self-assembly of patchy particles. *Journal of Chemical Physics B*, 115(48):14321-14326, 2011.

Selecting a building block for which a desired structure is thermodynamically stable would be straightforward if, at experimentally realizable conditions, its partition function could be generated[114, 115]. The partition function $\mathcal{Z}$ encodes all the system's thermodynamics, allowing for the calculation of macroscopic properties such as pressure, heat capacity, and packing fraction as a function of thermodynamic variables[116]. Unfortunately, the number of microstates that comprise a particular partition function scales by V^N where V is the volume accessible to each particle and N is the number of particles. This scaling makes exact enumeration of all but the smallest partition functions numerically intractable, and the complicated interactions between real particles renders analytical partition function calculations useless except for Baxter type tricks in special cases[117, 118].

For equilibrium statistical mechanical systems macroscopic properties can be calculated without explicit knowledge of the partition function. This fact follows from the sharp peak in the probability distribution of microstates consistent with equilibrium observables and therefore saddle point approximations, mean field theory, or numerical techniques such as molecular dynamics and Monte Carlo methods can be efficiently used to calculate equilibrium properties[98, 116, 119]. These methods have recently provided integral explanatory and exploratory capabilities for studying self-assembly. Molecular dynamics simulations have been instrumental in explaining the thermodynamics of water[120], predicting ordered arrangements of diblock copolymers[121], tethered nanoparticles[66, 69], and anisotropic virus capsomers[122, 123]. Monte Carlo simulations have elucidated quasicrystalline structure in systems of hard tetrahedra[72], solubility of tetromino mixtures[124], higher-order virial coefficients and clustering in water[125] and explained the dipole-induced structure observed in solutions of stabilized CdTe tetrahedra[80], among many other examples too numerous to list.

When used as an exploratory tool the utility of MC and MD simulations are hindered by a circular problem of time scales: to determine the relaxation time for a slowly evolving system, simulations much longer than the relaxation time must be run[114, 126, 127]. That is, in order to know if a simulation has been run long enough, one must run it longer. This is a particularly troublesome problem when attempting to find the optimal conditions at which a system of particles self-assembles a target structure because the time scale problem is now compounded by each additional state point that must be sampled.

In this work we develop a hierarchical method for calculating partition functions for complex building blocks to arbitrary precision by extending bottom-up building block assembly (BUBBA) to all temperatures[97]. Our method exploits the sharply peaked equilibrium distribution of microstates for $\mathcal{Z}$ at low N to hierarchically generate $\mathcal{Z}$ at successively larger values of N. We accomplish this through three steps at each value of N. First, microstates are enumerated through a cluster pairing procedure. Second, distinguishable microstates are identified. Third, negligible microstates are neglected. We demonstrate the method on a system of model patchy particles in the canonical (N, V, T) ensemble.

4.2 Experimental Methods

As a model system, we consider the set of 25 particles introduced by Troisi *et al.* [128] whose shapes resemble Tetris® pieces (tetrominoes, 4.1 [124, 129]). For computational expediency and without loss of generality, we confine the rigid, three-dimensional particles to a two-dimensional surface and describe the surface as a square grid in which the subunits that comprise the particles may each occupy a single grid cell. This dimension reduction is general because the method requires only a discrete search space of configurations, which is independent of dimension. In this model [128], all nearest-neighbor subunits share an attraction (e.g., van der Waals, depletion, solvophobic, etc.) of strength $U = \epsilon$, and the four subunits that comprise

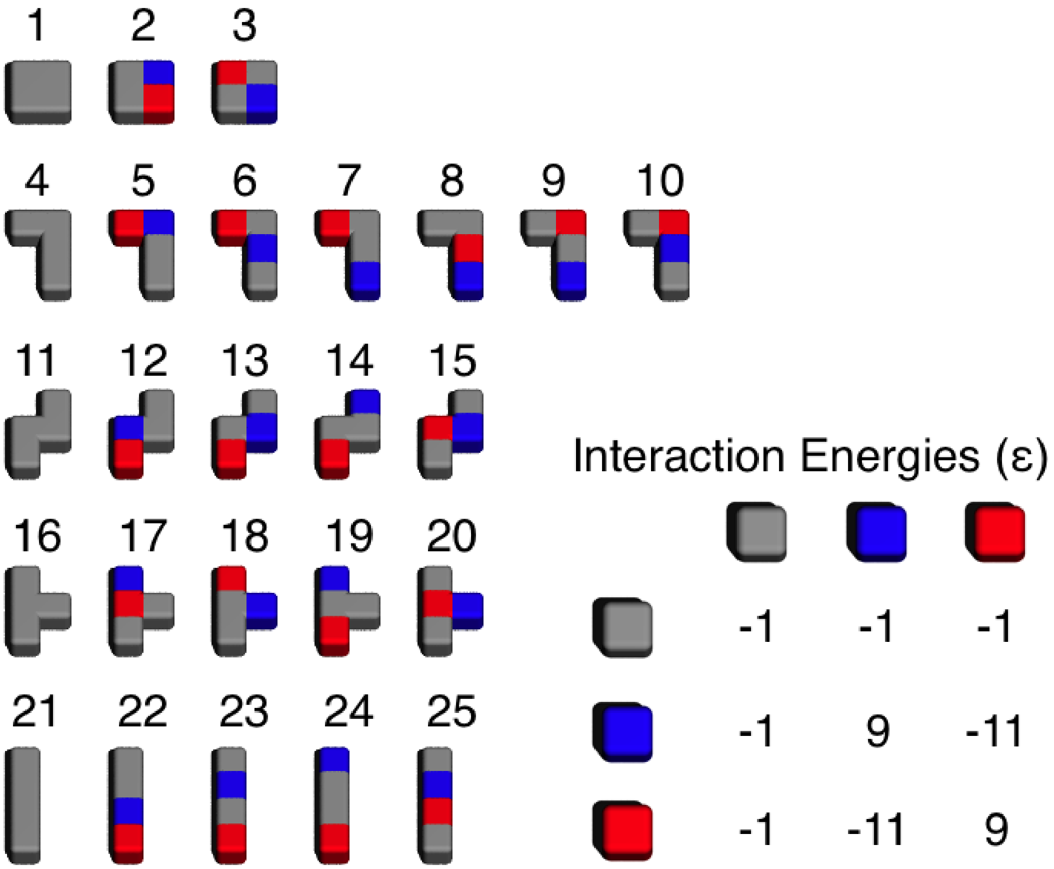


Figure 4.1: Model patchy particles. Reproduced from Reference [113].

each particle can be positively charged, negatively charged, or neutral. When two like-charged subunits share an edge, their resulting potential energy $U = 9\epsilon$, for opposite charges $U = -11\epsilon$, and for a neutral subunit sharing a face with any other subunit type $U = -\epsilon$. The particles can translate and rotate on the grid, but particle overlaps and inversions are prohibited. Since colloids with these geometries and interactions can now be synthesized [31, 34, 130], determining which, if any, of the particles in this family can be used to generate useful structures has immediate applications beyond serving as a proof-of-concept for the methodology presented here.

4.2.1 BUBBA at Finite Temperature

Exact enumeration of a partition function is computationally intractable for large ($N > 5$) clusters because of how rapidly the number of microstates scales with N. We can pare down the number of relevant configurations in the case of attractive particles, considering only the configurations wherein particles are interacting. Taking patchy particle 2 from 4.1 as a concrete example, we enumerate all of the clusters that can be created with two through four copies of the particle. There are 30 different $N = 2$ clusters, 736 different $N = 3$ clusters, and 25234 clusters when $N = 4$. Given any particular $N = 4$ cluster, its potential energy might range anywhere from $U = -68\epsilon$ to $U = 26\epsilon$, and at any of these energy levels there may be as many as 3000 distinct clusters with that energy. In this case, because we can calculate each cluster's degeneracy and energy exactly we can calculate the partition function

$$\mathcal{Z} = \sum_i \Omega_i \exp(-U_i/k_B T) \tag{4.1}$$

exactly, where Ω_i and U_i are the ith cluster's degeneracy and energy, respectively[131, 132]. In 4.2 we show the probability of observing a given cluster (red) and the number of configurations with a particular energy (black). Note that only two energy levels contribute non-negligibly to the partition function at this temperature. This fact may be striking to the uninitiated: less than a thousandth of a percent of the possible configurations for this system matter at this temperature.

To perform BUBBA at a given temperature we rely on the fact that a small fraction of possible microstates comprises the majority of the partition function. This assumption breaks down at high temperatures where all microstates contribute similarly to the partition function, but is valid for the condensed soft-matter systems of interest for self-assembly. We also assume that the microstates that contribute to $\mathcal{Z}(N = n, V, T)$ must be generated by pairings of configurations from $\mathcal{Z}(N < n, V, T)$, and is independent of V (i.e., the system is relatively dilute). This allows rapid generation of partition functions hierarchically because the number of configurations that must be checked scales as a polynomial in N rather than to the power of N. In prac-

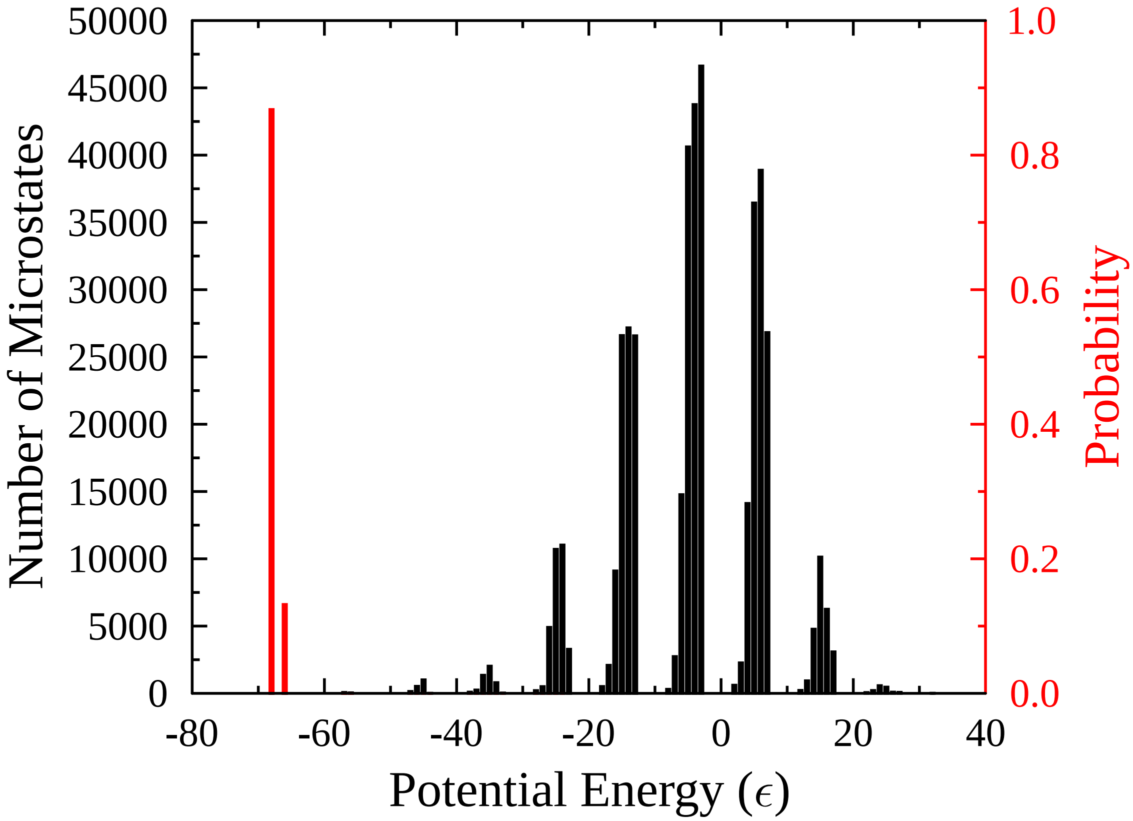


Figure 4.2: Frequencies (black) and relative probabilities (red) of $N = 4$ clusters of patchy particle 2 at $k_BT/\epsilon = 1.0$ as a function of potential energy. Reproduced from Reference [113].

tice, we utilize a "cluster library" to keep track of each cluster generated by BUBBA. Each entry in the cluster library specifies the number, potential energy, relative degeneracy, particle positions, and combinations of smaller clusters that can create a particular cluster. In this framework, using BUBBA to calculate $\mathcal{Z}(N, V, T)$ consists simply of pairing clusters in the library, keeping only the unique microstates, and discarding the negligible microstates.

4.2.2 Cluster Pairing

Given a discretization of configuration space for the clusters of interest, BUBBA combines pairs of clusters to generate larger clusters. When generating size N clusters there are $\lfloor N/2 \rfloor$ combinations of cluster sizes that must be paired. For example, only

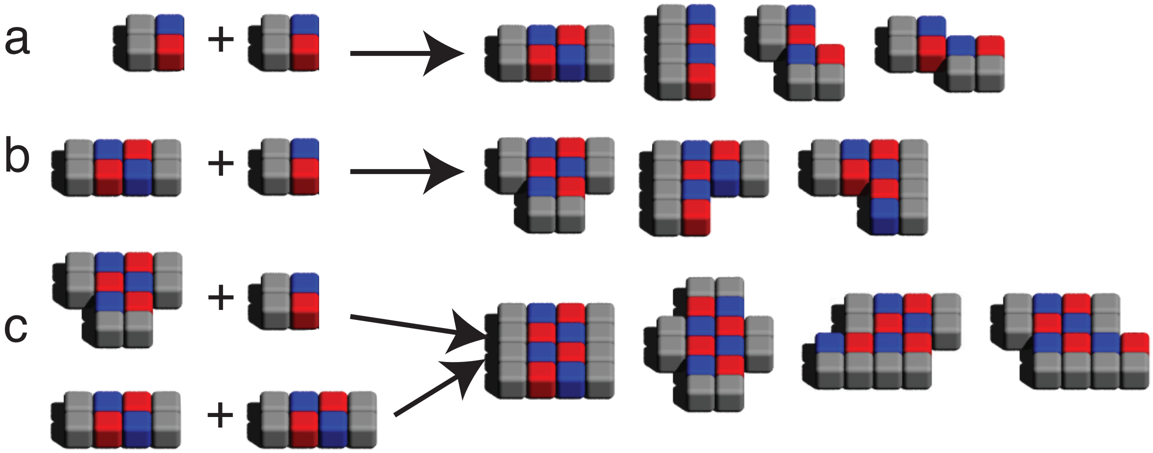


Figure 4.3: Examples of cluster pairing for $N = 2$ (a) $N = 3$ (b) and $N = 4$ (c). Note that for each pairing, not all possible resulting clusters are included. Reproduced from Reference [113].

$N = 1$ clusters need be combined to generate $N = 2$ clusters, but for $N = 4$ clusters we must pair $N = 1$ with $N = 3$ clusters as well as $N = 2$ with $N = 2$ clusters (4.3). Each time a new candidate cluster is created from a cluster pairing we check the cluster library to see if it has already been generated. If the cluster is new, it is added to the cluster library. If it already exists in the library, we update the existing cluster's relative degeneracy as well as the list of cluster combinations that can generate it.

Choosing an appropriate discretization of configuration space for a particular system is crucial and it has important implications for BUBBA's performance. The chosen discretization of search space should depend on the metric by which clusters are distinguished. The clusters depicted in Figure 4.4a and b are two examples of clusters made from five copies of a sphere with a continuous chain of states linking the two structures. Are they the same, or are they different? If they are different, which states count towards the degeneracy of one cluster but not the other? It is conceivable that the statistical mechanics of one system might not need to distinguish between these configurations, but that another might rely upon their difference. Whichever specific metric is chosen, be it Fourier descriptors [133], sum of squared errors[134],

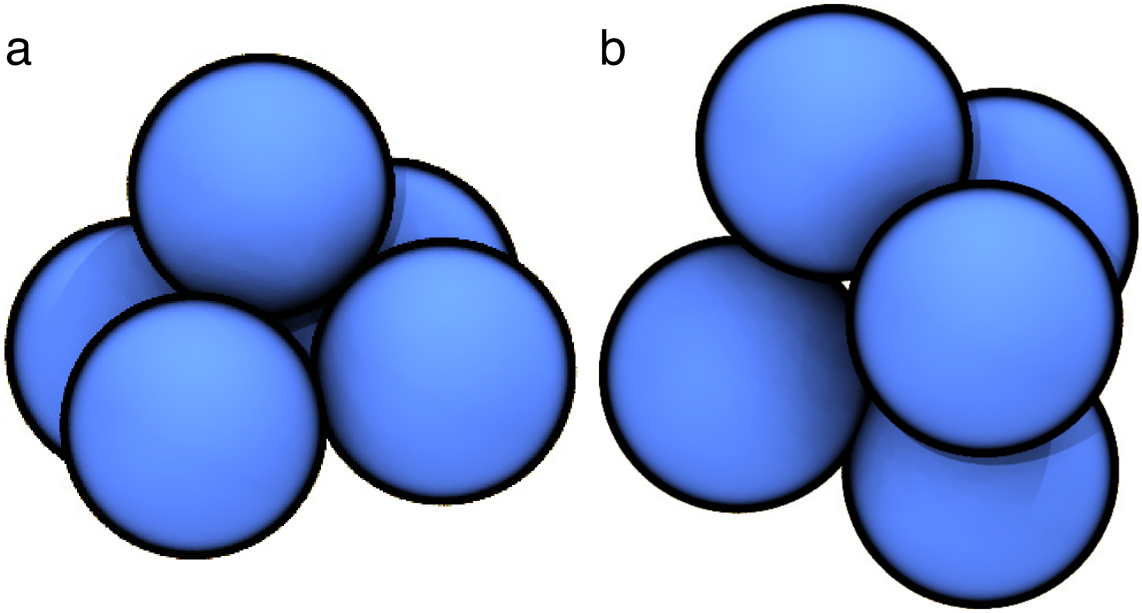


Figure 4.4: a. Example of a cluster of five spheres in a pyramid structure. b. Bipyramid structure made from the same number of spheres as in (a). The partition function generated by BUBBA depends upon the chosen discretization of configuration space. Different discretizations would differ in whether (a) and (b) are distinguishable or not. Reproduced from Reference [113].

contact network analysis[135], or some combination of methods, BUBBA requires that for any pair of clusters they can be classified as being the same or different, and that a finite set of allowed pairings of the two clusters can be generated.

4.2.3 Distinguishing Clusters

To distinguish between configurations we define a graph for each cluster and compare traversals of each graph[136]. Here, each cell in a tetromino defines a node and edges (connections) between nodes are defined by cell adjacencies (4.5). If any one traversal of one graph matches a particular traversal of the second graph, the two configurations are considered identical, otherwise they are different. A traversal is a tree of nodes where each node is visited exactly once. When comparing two traversals, they are equal if the node types and connectivities are identical for both traversals. Here, two nodes are identical if they have the same type (red, blue, or grey), and if they have the same number of connections. Two optimizations that reduce the number of graph traversals that we implement are comparing cluster energies, length, and width before beginning a graph traversal.

4.2.4 Discarding Negligible Clusters

Once a partition function has been generated we discard all the smallest-weighted clusters whose summed weights are less than a chosen cutoff c. Typically, we choose c in the range $0 \leq c \leq 0.01$ because c is the error in the $N = 2$ partition function, and this error will compound roughly as c^N. Next, we sort the clusters by their normalized Boltzmann weights

$$w_i = \Omega_i \exp(-U_i/k_B T)/\mathcal{Z} \tag{4.2}$$

where $\mathcal{Z}$ is the partition function from 4.1[115]. By summing the Boltzmann weights in order, beginning with the largest, we find the clusters for which their sum of weights is less than c, which we then discard. To optimize the calculation of cluster probabilities we sort the clusters by $r_i = -U_i + k_b T \ln(\Omega_i)$. Using r_i we calculate w_i with the formula $w_i = \exp(r_i - r_0)/\sum_j \exp(r_j - r_0)$ where r_0 is for the cluster with the largest Boltzmann weight.

Calculating the degeneracies Ω_i for lattice model clusters can be performed exactly and must be approximated for clusters with continuous degrees of freedom. For a lattice cluster that can be made from p pairings where each pairing is defined by clusters a and b,

$$\Omega_i = \sum_p \Omega_a \Omega_b n_p \tag{4.3}$$

where n_p is the number of distinct ways of combining a and b to make cluster i.

For systems with continuous degrees of freedom it is not straightforward to sum over the same set of cluster combinations because of constraints placed on a cluster's

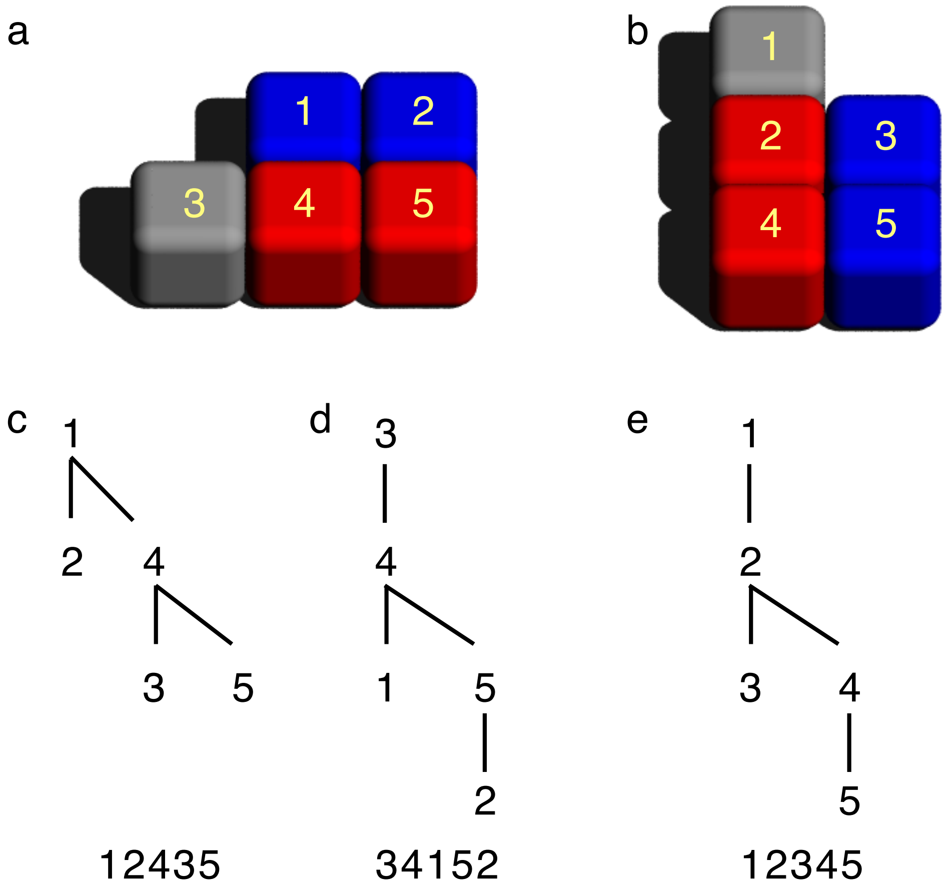


Figure 4.5: a. An example cluster, with numbered node labels. b. Reference cluster with numbered node labels. c. One possible graph traversal of (a) that does not match the particular graph traversal of (b) The sequence of nodes visited during a traversal are shown below the graphs and are generated by first visiting the root (top) node and following branches depth-first, left-to-right, not counting backtraces, until each node has been visited. d. Graph traversal of (a) that matches the particular graph traversal of (b). e. Particular graph traversal of (b) that is used as the reference structure. When traversing a graph, the number of nearest neighbors for the cell and the cell type are compared at each node. Reproduced from Reference [113].

degrees of freedom when the two clusters are combined. The relative degeneracies

$$\Omega_i = q_{i,rot}q_{i,vib} \tag{4.4}$$

for such off-lattice clusters should be calculated with the rotational ($q_{i,rot}$) and vibrational ($q_{i,vib}$) partition functions[115, 137, 138]. The rotational partition function for a general rigid body

$$q_{rot} = \frac{\pi^{1/2}}{\sigma}\left(\frac{8\pi^2 I_x k_B T}{h^2}\right)^{1/2}\left(\frac{8\pi^2 I_y k_B T}{h^2}\right)^{1/2}\left(\frac{8\pi^2 I_z k_B T}{h^2}\right)^{1/2} \tag{4.5}$$

where I_x, I_y, and I_x are the diagonal elements of the rigid body's diagonalized inertial tensor $\mathcal{I}$, σ is the body's symmetry number and h is Planck's constant[115]. The calculation of $\mathcal{I}$ can be performed using

$$\mathcal{I} = \sum_k \mathcal{I}_k + m_k\left[(r_k \cdot r_k)E_3 - r_k \otimes r_k\right] \tag{4.6}$$

where the sum is over all particles in the cluster, $\mathcal{I}_k$ is the inertial tensor of the kth particle about its center of mass, m_k is the kth particle's mass, r_k is a displacement vector from the cluster's center of mass to the kth particle's center of mass, E_3 is the 3×3 identity matrix, and $\otimes$ is the dyadic product. For clusters of the same size at a particular temperature 4.5 simplifies to

$$q_{rot} = \frac{(I_x I_y I_z)^{1/2}}{\sigma} \tag{4.7}$$

because only the relative magnitudes of the rotational partition functions for clusters of the same size matter for their probabilities. To determine the symmetry number σ we count all linearly independent combinations of rotation operations that map a cluster onto itself. For a cluster defined by the types and positions of its particles we try all rotation operations that map one particle i onto all other particles of the same type. For each of these, we attempt a second rotation about an axis passing through i until a second particle j is aligned with a reference particle of the same type. After this alignment we calculate the sum of squared errors

$$\delta = \sum_k |\vec{r_k} - \vec{r_0}|^2 \tag{4.8}$$

where $\vec{r_k}$ is the position of the kth particle and $\vec{r_0}$ is the position of the reference particle of the same type closest to $\vec{r_k}$. If this trial orientation has not been previously generated and if $\delta < 0.001$ we increment σ.

Calculations of the vibrational partition function can also be performed numerically for off-lattice clusters using Einstein oscillators or any analogous method for determining the free volumes accessible to each component of a cluster[115, 137]. Generally

$$q_{vib} = \prod_k V_k \tag{4.9}$$

for clusters with no collective vibrational modes, where V_K is the volume accessible to the kth particle in the cluster. For example, the vibrational degrees of freedom matter substantially for clusters in 4.4 if they are made up of five hard spheres that can move on the surface of a hidden sphere. In this case an Einstein oscillator approximation of q_{vib} would over-predict the entropic contribution from the sphere at the apex of the pyramid in 4.4, but may under-predict the contributions from the "base" spheres. Here, we integrate the accessible volume to each sphere as a fraction of the total center sphere surface area with Monte Carlo integration. This approach generalizes to arbitrary cluster types and interaction types between the subunits of the cluster.

	$c = 0.01$	$c = 0.001$	$c = 0.0001$
k_BT/ϵ=0.1	3	3	4
k_BT/ϵ=1.0	4	35	847
k_BT/ϵ=3.0	4	513	1320

Table 4.1: Minutes required to generate partition functions for all 25 building blocks in 4.1 up to $N = 10$ using BUBBA.

4.3 Results/Discussion

In using BUBBA to generate partition functions of a given size N, the cutoff value c can be chosen to strike a balance between efficiency and accuracy. Small values of c ensure the partition functions built at each N are more complete, at the cost of including more cluster configurations. Of course, as k_BT/ϵ increases, more clusters contribute to the partition function so choosing c too large will cause BUBBA to miss important configurations. 4.1 summarizes the relationship between BUBBA runtime, c, and k_BT/ϵ for the 25 model particles in 4.1. Note that partition functions up to $N = 4$ can be easily generated when $c = 0$, and an extrapolation of the resulting trend suggests that generating an $N = 10$ partition function for $c = 0$ would require $O(10^8)$ years with the current implementation.

To quantify the accuracy of the partition functions generated by BUBBA we compare the $N = 4$ clusters generated at $c > 0$ with the clusters generated when $c = 0$. The error

$$e(c) = \sum_i (w_{c=0,i} - w_{c,i})^2 \tag{4.10}$$

where $w_{c,i}$ is the Boltzmann weight of the ith cluster generated by BUBBA with cutoff c. We find that for any given building block, the required c to generate accurate partition functions will depend strongly on temperature. For example, at $k_BT/\epsilon = 0.1$, we find $e(0.00001) = 0$ for all 25 particles in 4.1, indicating no clusters were discarded at this value of c. For $0.0001 \leq c \leq 0.01$ and $k_BT/\epsilon = 0.1$, $e(c) = 0$ for all particles except for particle 6, for which $e(c) = 2.0$ because the lowest energy

	$e(0.01)$	$e(0.001)$	$e(0.0001)$
k_BT/ϵ=0.1	0.000	0.000	0.000
k_BT/ϵ=1.0	0.007	0.007	0.004
k_BT/ϵ=3.0	0.025	0.016	0.004

Table 4.2: Average partition function error (4.10) for all patchy particles in 4.1, excluding particle 6.

cluster is missed. Considering only the patchy particles in 4.1, we find that $e(c)$ roughly scales with c (4.2) and k_BT/ϵ. We exclude the non-patchy particles from this analysis because the characteristic temperature ranges differ substantially from those for the patchy particles, making comparisons less useful.

BUBBA's efficiency at low temperatures allows for clusters larger than $N = 10$ to be generated easily. As a case study we consider the partition functions for patchy particle 6 at $k_BT/\epsilon = 0.1$, $c = 0.00001$ and $N \leq 50$. 4.6 displays the elapsed runtime for these conditions on a 2.6 GHz Intel processor. For $N > 25$ this data fits a weak exponential with the number of minutes $t = 0.2 \exp(0.12N)$. At large cluster sizes it is apparent that periodic tilings of the patchy particle are the most thermodynamically stable arrangements (4.7). At these conditions, the more porous tiling in 4.7a is only 5.8 times more likely than the tiling in 4.7b, and this difference in entirely entropic in origin. The two tilings have the same number of building blocks, and identical energies, however there are exactly 9,389,090,429,393,174,528 more ways to make the more porous tiling than the 1,972,576,636,788,277,248 ways of generating the tiling in 4.7b, as calculated by BUBBA with $c = 0.00001$. The existence of these two competing tilings has negative implications for this building block's propensity for self-assembly of either pattern at this temperature. Systems of building blocks for which two or more structurally distinct tilings are commensurate in probability will fail to assemble any tiling or pattern robustly because of the enormous energetic penalty to rearrange one competing structure into the other. We confirm that patchy particle 6 does not self-assemble porous arrays robustly via Monte Carlo

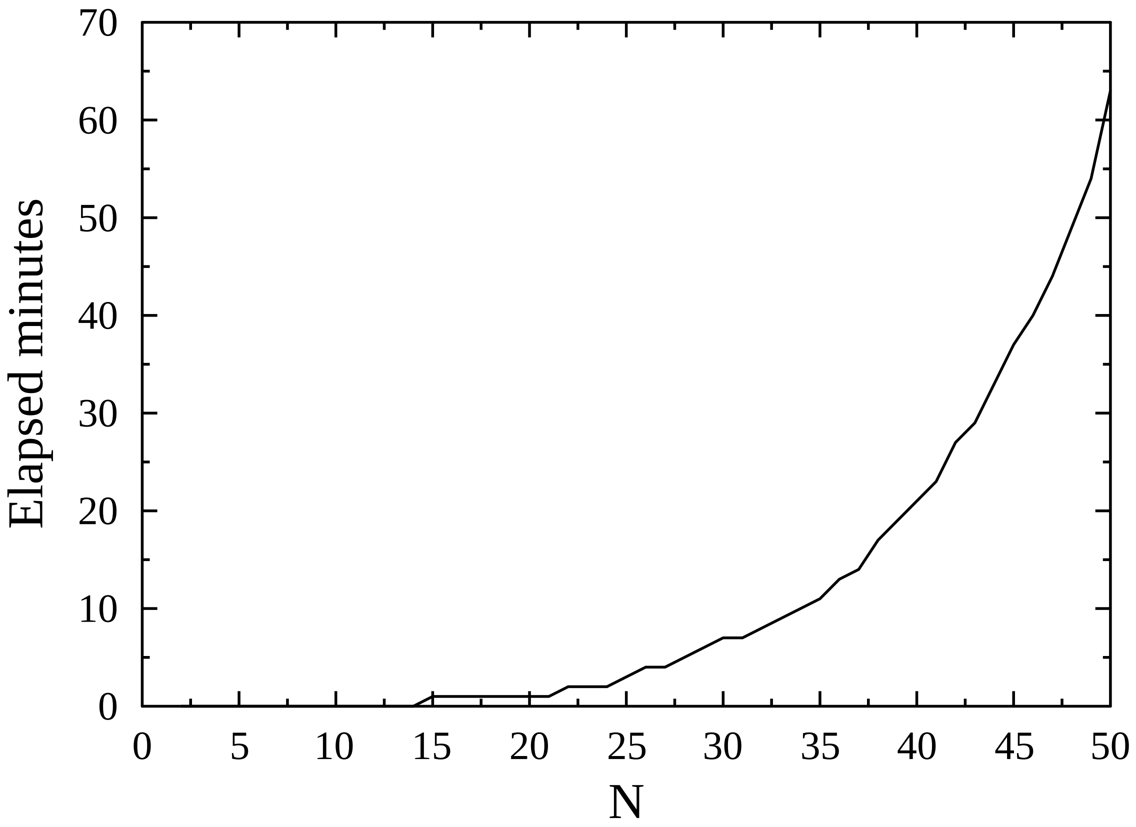


Figure 4.6: Minutes of runtime to generate the Nth partition function for patchy particle 6 at $k_BT/\epsilon = 0.1$ and $c = 0.00001$ using BUBBA. Reproduced from Reference [113].

simulations, and expand upon the identification of robust assembly candidates in another manuscript[110].

We determine thermodynamically stable motifs for all of the 25 particles in 4.1 at $k_BT/\epsilon = 0.1$ in the same way as described above for patchy particle 6. Tilings of the unit cells identified in the most probable $40 \leq N \leq 50$ clusters are presented in 4.8. Although the family of model particles considered here was chosen primarily to provide a proof-of-concept demonstration of our approach, it is easy to envision possible applications of some of the motifs in 4.8a-l. For example, some of these patterns (e.g. 4.8c-i) could be used as nanoscale circuit elements [139, 140]. If the red and blue cells are made of an insulating material while the gray are conductive, these building blocks can be used to assemble arrays of wires separated by insulating

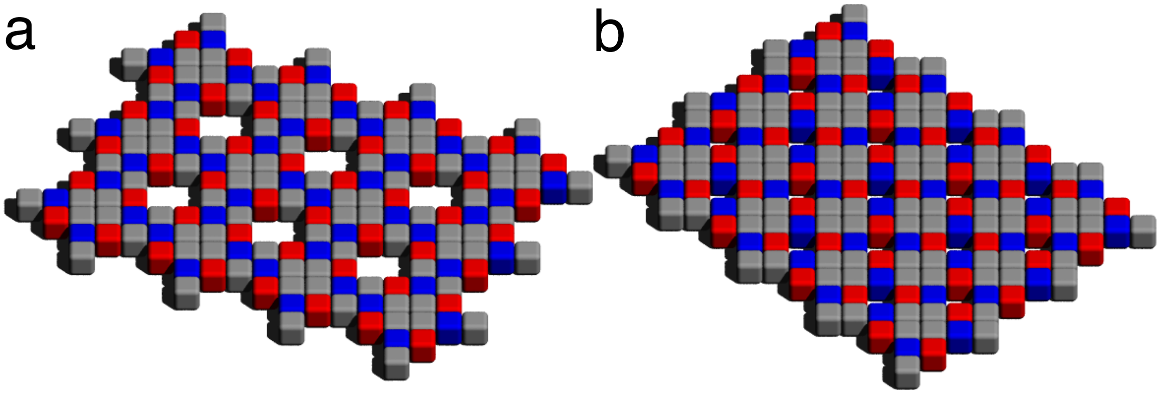


Figure 4.7: a. Most likely $N = 48$ configuration at $k_BT/\epsilon = 0.1$ for patchy particle 6. There are 11,361,667,066,181,451,776 ways to generate this cluster from combinations of distinguishable patchy particles. b. 20th most likely $N = 48$ configuration at $k_BT/\epsilon = 0.1$ for patchy particle 6. This cluster is less likely than (a) despite their identical energies because there are only 1,972,576,636,788,277,248 ways to generate it. Reproduced from Reference [113].

barriers. Patchy particle 19 could also be used to generate a structure with potentially interesting electronic applications: its continuous gray domains (4.8k) could be used in a service that requires a redundant conducting path through an otherwise insulating matrix. The porous arrays of 4.8j-l have potential applications in nanoscale filtration [141] and fuel cells.

4.4 Conclusions

We have demonstrated that the generation of partition functions hierarchically to arbitrary precision is efficient and straightforward, and that it can aid in assessing self-assembly propensity. In the cases where small numbers of distinguishable configurations comprise a majority of a partition function's weight, as is the case for systems at low temperatures and for many anisotropic building blocks with disparate interactions, BUBBA is a particularly effective method for generating partition functions that have been heretofore inaccessible. For the first time, this allows the question of "What structures are thermodynamically favored for this building block at any tem-

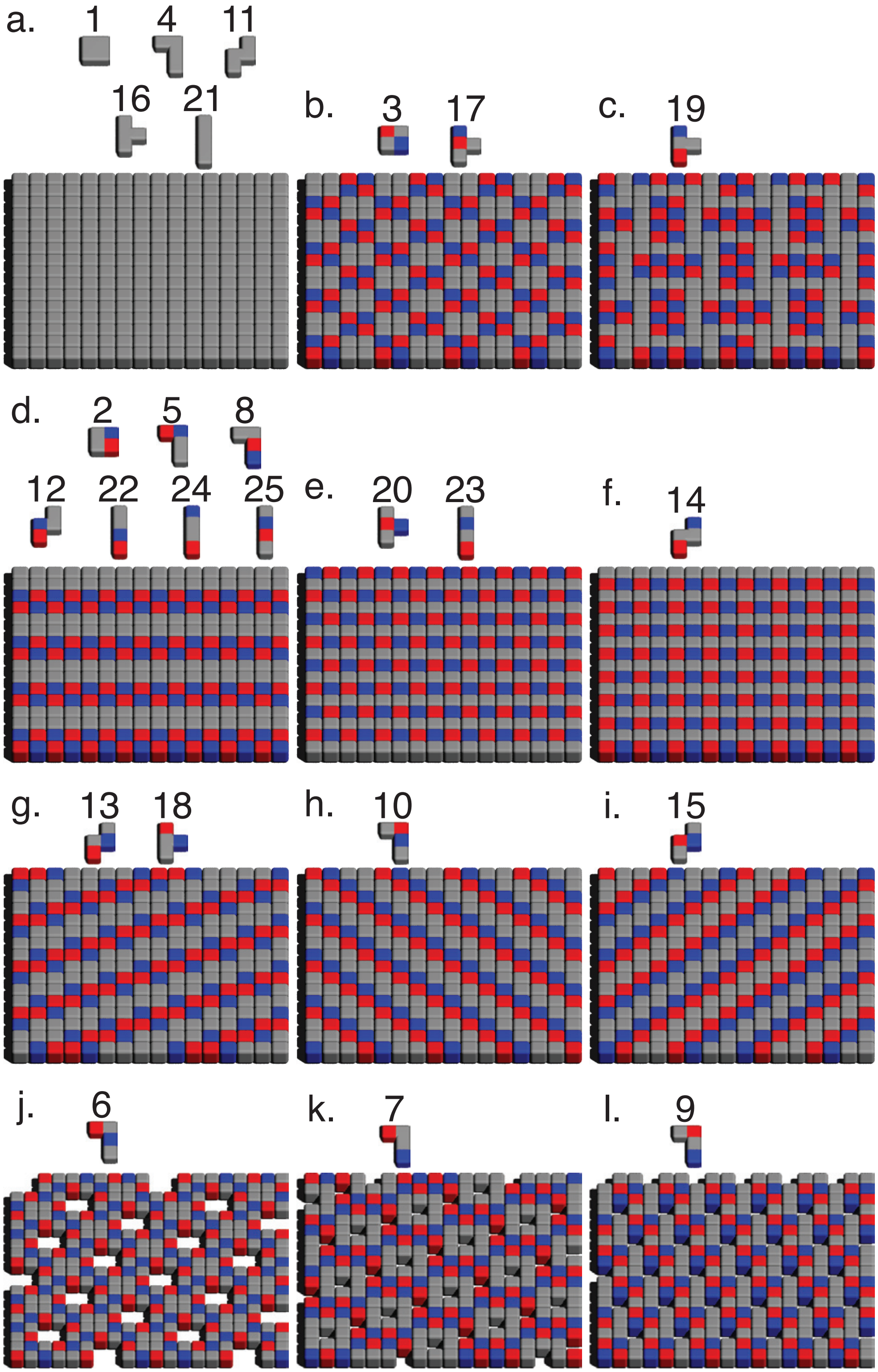


Figure 4.8: Energy minimizing motifs stabilized at $k_BT/\epsilon = 0.1$ for the 25 particles in 4.1, from patterns in $N = 50$ clusters generated with BUBBA. Reproduced from Reference [113].

perature?" to be answered independent of assembly kinetics. Finally, for the family of model patchy particles studied here we discover a diverse set of thermodynamically stable structures that may have practical applications at the nanoscale.

The current implementation of BUBBA is effective and demonstrative, but there are many opportunities for improvement and application. Many elements of the algorithms used stand to benefit substantially from parallelization of inner loops. For example, there is no reason the shape matching of one cluster against thousands of clusters in the cluster library must execute serially, and a GPU implementation of this aspect alone would enable partition functions larger by an order of magnitude to be generated. BUBBA also has promise as an efficient screening tool that would allow for the identification of the parts of a partition function that are consistent or inconsistent with the assembly of target structures. Using BUBBA as a screening tool, implementing kernels on GPU architectures, and extending to other ensembles are the subjects of ongoing work.

CHAPTER V

Assembly Pathway Engineering

5.1 Introduction

Given a system of interacting particles, complex structures on macroscopic length scales can be synthesized via self-assembly when thermodynamics and kinetics favorably conspire[1, 142–145]. As one example, the wires, sheets, helices, and colloidal crystals of supra-particles assembled from CdTe/CdS nano-tetrahedra [12, 14, 80] demonstrate the rich structural diversity accessible for a single type of building block. These nanoparticles hold promise for the assembly of materials with unique photonic, electronic, and mechanical properties[146], as do colloids, DNA, and DNA-functionalized particles[34, 38, 111]. Finding the experimental conditions at which a set of building blocks robustly assembles any one ordered structure can require considerable effort and some amount of luck, and there is no guarantee that the assembled structure will find application. Finding a building block that will self-assemble a prescribed target structure with narrowly specified macroscopic properties is even more difficult because each specification can constrain the building block materials that can be used, possibly precluding self-assembly in any region of experimentally realizable state space. [1]

[1] This chapter is adapted from Reference [110] E. Jankowski and S.C. Glotzer, Screening and designing patchy particles for optimized self-assembly propensity through assembly pathway engineering. *Soft Matter*, 8(10):2852–2859, 2012.

In the case where candidate building blocks for self-assembly have already been identified it is in principle possible to create "phase diagrams" that map out thermodynamically stable equilibrium structures as a function of parameter space. In practice this is not a trivial task and much theoretical work has been devoted to the development of sophisticated computational techniques that allow for the equilibrium simulation of complex building blocks. Standard Monte Carlo (MC) simulation schemes have been extended in numerous ways to include special moves that allow for faster equilibration times of complex building blocks. Volume bias moves, coordinated cluster moves, and convex polyhedra overlap calculations have enabled the efficient simulation of patchy colloids, lattice tetrominoes, and hard tetrahedra[72, 85, 97, 128, 147]. Molecular dynamics (MD) simulations have also played an integral role in the prediction of self-assembled structures, and recent developments in GPU hardware architectures and algorithms have enabled the simulation of block copolymers, tethered nanoparticles, and arbitrary rigid bodies at longer time scales than ever before[70, 75, 76]. Unfortunately, when an equilibrium solution or simulation of patchy particles fails to generate an ordered pattern it is not always obvious whether the culprit is thermodynamics or kinetics. Recently there have been studies that attempt to quantify kinetic trapping through fluctuation-dissipation ratios[148, 149], and through the interplay between specific and nonspecific interactions[143, 145, 150], but these methods do not provide predictive capabilities for thermodynamically stable structures.

The fact that both thermodynamics and kinetics can prevent a system of particles from self-assembling is particularly troublesome for experimentalists that search parameter space via trial-and-error because experiments that fail to assemble do not provide information about how assembly might be improved. In this chapter we propose a methodology (Fig. 5.1) for the rational design of building blocks optimized for self-assembly that focuses on *assembly pathway engineering*: identifying the traps that occur as a system assembles so they may be circumvented. As systems self-assemble we hypothesize that the thermodynamically stable intermediate clusters that arise hold information about their ability to order. These sequences of intermediate clusters are *assembly pathways* and we propose a methodical analysis of them to predict the degree to which a system of building blocks will assemble a target pattern, which we refer to as the building block's *assembly propensity* for the pattern. We foresee assembly pathway engineering proceeding as a collaboration among structural identification, kinetic measurements, and the assembly pathway analysis described here. These components are indicated by the red diamonds in Fig. 5.1.

Our approach begins with the physical properties and structure of a product that we aim to create via self-assembly. The prospective building blocks that could be used are constrained both by the synthesis capabilities of a particular lab and the properties of the target product, e.g., metallic nanoparticles should be avoided if an insulating material is desired. These building blocks are then screened using a thermodynamic method to generate stable structures. From the building blocks whose equilibrium structures are consistent with the target pattern we identify, via assembly pathway analysis, the traps that hinder self-assembly. We then modify the building blocks or the conditions under which they are assembled to optimize assembly, and finally perform experiments to test assembly rates. In principle any of a number of methods including MC or MD simulations could be employed to find thermodynamically stable structures or the intermediates that arise during assembly. In this chapter we use bottom-up building block assembly (BUBBA) for both[97, 113]. Briefly, BUBBA is a computational tool that begins with a single building block and builds successively larger equilibrium structures hierarchically. To make a cluster of size N, BUBBA enumerates all possible combinations of pairs of clusters whose sizes sum to N, where each cluster in the pairing contributes non-negligibly to the ensemble of clusters for its size. In this way, BUBBA efficiently generates free-energy minimizing structures and the stable intermediates that lead to it, which we hypothesize govern assembly propensity.

The structure of this paper is as follows. First, we define and describe the computational methods and measurements we employ for assembly pathway engineering. Second, we consider model systems of patchy colloids and CdTe/CdS tetrahedra for which we test elements of our methodology. Third, we motivate the need for efficient structural screening tools by comparing the assembly propensities of seven model patchy colloids for a target structure. We find that assembly propensity can vary substantially from building block to building block, and show that assembly pathways provide predictive capabilities for assembly propensity. Fourth, we validate our pathway-based approach for a real system of CdTe/CdS tetrahedra. Fifth, we show the utility of BUBBA-informed pathway engineering by demonstrating ways thermodynamic traps can be avoided and by identifying the experimental conditions that maximize assembly. Finally, we discuss the limitations of our approach, highlighting the distinction between steric kinetic traps and interaction-based thermodynamic traps, and conclude with suggestions for further study.

5.2 Methods

The decision points represented by red diamonds in Fig. 5.1 can be, in principle, informed by any of a number of methods including, but not limited to, wet lab synthesis, Monte Carlo, molecular dynamics, simulated annealing, or mean field simulations. In this work we use Monte Carlo computer experiments to determine whether a target structure is kinetically accessible by a system of building blocks and we use BUBBA both for the screening of stable structures and for generating assembly pathways. The intermediate clusters generated with BUBBA are analyzed with shape-matching algorithms for consistency with target motifs, and assembly pathways are compressed into *pathway fingerprints* for clarity. Shape matching is also used to measure the

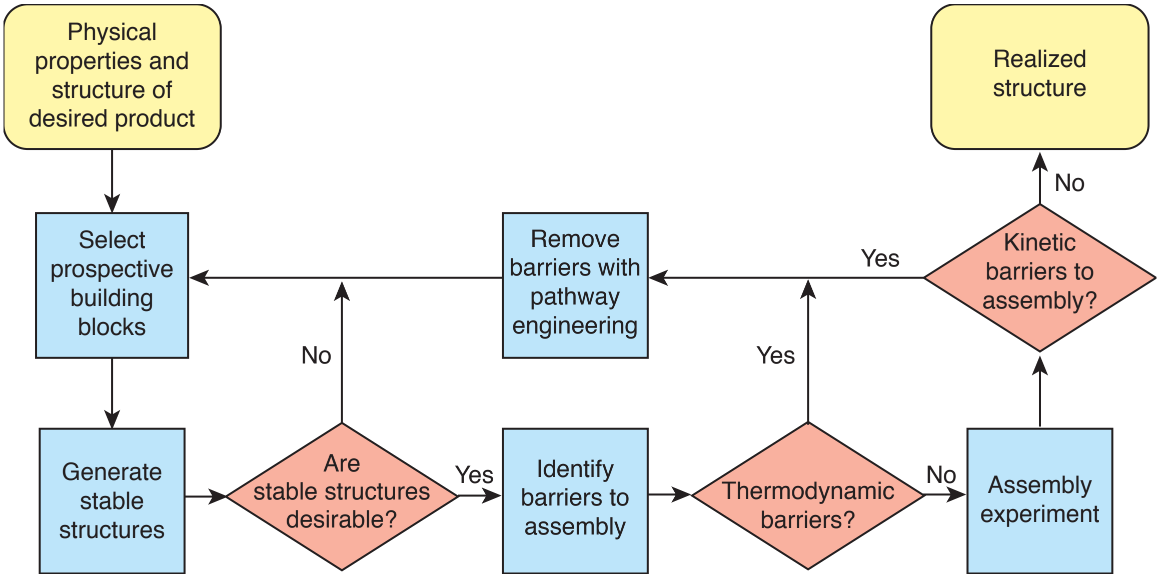

Physical properties and structure of desired product
Realized structure
No
Select prospective building blocks
Remove barriers with pathway engineering
Yes
Kinetic barriers to assembly?
No
Yes
Generate stable structures
Are stable structures desirable?
Yes
Identify barriers to assembly
Thermodynamic barriers?
No
Assembly experiment

Figure 5.1: Assembly pathway engineering algorithm. In this work we use BUBBA with shape matching to identify stable structures and thermodynamic barriers to assembly, and MC simulations as assembly experiments[113]. Perturbation-response methods[148, 149], molecular dynamics, and new shape matching techniques[135] will all play integral roles in assembly pathway engineering. Reproduced from Reference [110]. Reproduced from Reference [110].

degree to which a target structure has been assembled in Monte Carlo simulations.

5.2.1 Assembly Pathways

For building blocks that undergo thermodynamically driven self-organization, an assembly pathway is a sequence of states that leads them from an initially disordered configuration to states that minimize free energy for the system as a whole[150–152]. These pathways can easily be generated for patchy particles with BUBBA, and we demonstrated the tradeoffs between efficiency, accuracy, and temperature in previous work[113]. In this work we consider pathways terminating at $N = 10$ building blocks and use a cutoff $c = 0.00001$ from Ref. [113], ensuring enough clusters are included at each size to represent at least 99.999% of each partition function. The partition functions indicate which clusters are thermodynamically stable and the connectivities between partition functions indicate specific thermodynamically stable assembly pathways. An example assembly pathway generated with BUBBA for one of the patchy particles we study here is shown in Fig. 5.2a. Each blue box and red octagon in Fig. 5.2a is a node representing a cluster configuration, which is drawn near each node. The three numbers in each node represent the cluster's size, energy level (1 for lowest energy, 2 for second-lowest, etc), and proportion of the partition function (out of 1.0) represented by that node at that size. The arrows connecting nodes indicate the cluster at the head can be created by combining the cluster at the tail with another cluster in the network, and the size of the arrowhead is proportional to the number of ways this pairing can be made. The color and shape of each node denotes whether or not the cluster is consistent (blue box) or inconsistent (red octagon) with a chosen target pattern, in this case the wide stripe motif in Fig. 5.3h.

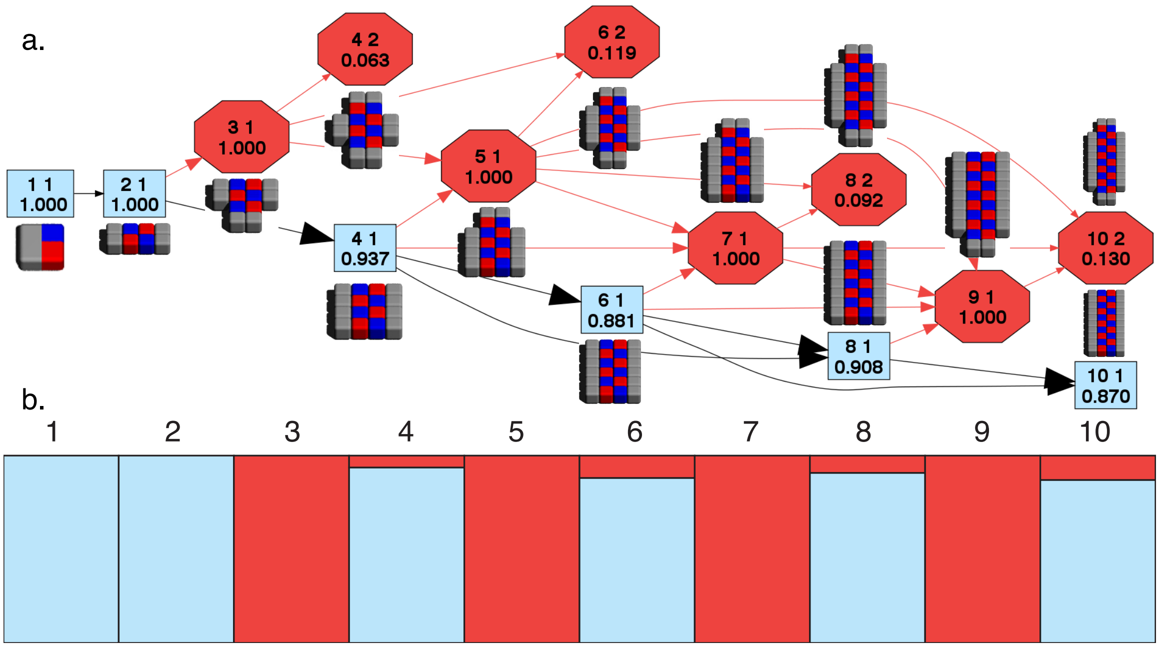


Figure 5.2: a. The self-assembly pathways for patchy particle a from Fig. 5.3 at $k_BT/\epsilon = 0.6$, from $N = 1$ to $N = 10$. Nodes indicate the size of a cluster, its energy level (1 for lowest, 2 for second-lowest, etc), and its probability compared to clusters of the same size. Clusters are depicted near the nodes that represent them. Arrows connecting nodes indicate an assembly pathway, and the size of the arrowhead indicates the degeneracy of the pathway. Red nodes indicate clusters inconsistent with the wide stripe motif (5.3h). b. Assembly fingerprint created from the same data as in a. Each rectangle in a column represents a cluster and its height corresponds to its contribution to an N, V, T partition function. The proportion of red in a column indicates the probability of finding a cluster that is inconsistent with the target motif in an equilibrated N, V, T ensemble. With 100% red columns at $N = 3, 5, 7, 9$ we expect poor assembly of the wide stripe motif because all of these clusters are inconsistent with it. Reproduced from Reference [110].

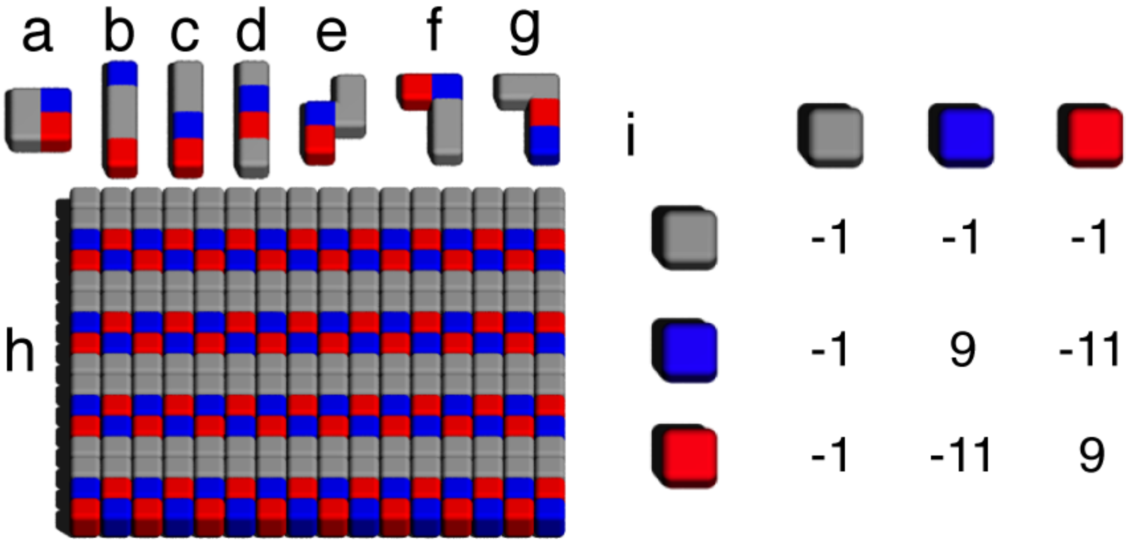


Figure 5.3: a-g. Seven patchy particles from Troisi *et al.*[128], h. Wide stripe motif that can be made by tiling a-g, predicted by BUBBA[113], i. Interaction energies (units of ϵ) defined for neighboring subunits. Reproduced from Reference [110].

5.2.2 Pathway Fingerprints

In general, the assembly pathways for a building block at an arbitrary temperature are not as concise as Fig. 5.2a. The number of clusters that make up the partition function for a given N can grow up to many thousands for even small N, which makes pathway visualization in the style of Fig. 5.2a unwieldy. To visualize complicated assembly pathways in a way that makes them comparable to simple pathways we create assembly pathway “fingerprints” from the pathway data. The pathway fingerprint in Fig. 5.2b is an alternative and compact method of visualizing the cluster weight data from Fig. 5.2a at the cost of losing detailed path information. In a pathway fingerprint, each column represents an approximation of the partition function for clusters of a given size, increasing from one on the left to an arbitrary size on the right. Each cluster from a pathway becomes a rectangle in the fingerprint, whose height is proportional to the cluster’s probability. As in the assembly pathways, target-motif-inconsistent clusters are indicated in red, and target-motif-consistent clusters are indicated in blue. To aid in visualization we omit the black border around a rectangle if the corresponding cluster's probability is less than 0.02.

5.2.3 Monte Carlo

We perform canonical ensemble (constant N, V, T) Monte Carlo (MC) simulations of patchy particles in order to assess the degree to which they self-assemble. Here, $N = 200$, $V = 2500$ (50×50 periodic lattice), and we consider instantaneous quenches to temperatures that are easily accessible in colloidal experiments[153]. All simulation runs are initialized with a random configuration of patchy particles, and we subsequently attempt 2×10^7 trial moves, requiring approximately two minutes of real time on a 2.8GHz Intel Core 2 Duo® processor. We quantify the degree to which a simulation snapshot assembles the wide stripe motif using a 2D Gaussian box filter, a standard technique in image shape matching[135] . We define the "motif match," for the ith subunit in a simulation snapshot as

$$m_i = \frac{1}{A} \sum_{j=-w}^{w} \sum_{k=-w}^{w} \delta(x_i + j, y_i + k) e^{-\frac{j^2+k^2}{2\sigma}} \tag{5.1}$$

where w is the box filter width, x_i and y_i are the coordinates of the ith subunit, and σ controls the width of the Gaussian kernel. We normalize m_i on $[0, 1]$ with

$$A = \sum_{j=-w}^{w} \sum_{k=-w}^{w} e^{-\frac{j^2+k^2}{2\sigma}} \tag{5.2}$$

. The delta function $\delta(x_i + j, y_i + k) = 1$ if the subunit located at $(x_i + j, y_i + k)$ in the simulation snapshot is the same type as the subunit in the wide stripe motif shifted (j, k) away from a reference cell, and $\delta(x_i + j, y_i + k) = 0$ otherwise, and will of course depend upon the orientation of the reference motif relative to the simulation snapshot. The reference cells of a motif are defined by their types, positions, and connectivity to other reference cells of the motif. The motif match for a simulation snapshot is defined as

$$M = \frac{1}{n} \sum_{i=0}^{n-1} \max(m_i \forall o) \tag{5.3}$$

where n is the number of particle subunits, and only orientations o that maximize m_i are included in the sum. Here we use $\sigma = 2$ and $w = 2$ which results in M-values greater than 0.7 having a strong visual similarity to the reference motif, and $M < 0.6$ indicating a poor match.

5.3 Models

We demonstrate the generation and analysis of assembly pathways for a 2D system of patchy tetrominoes[128] and a 3D system of CdTe/CdS tetrahedra[14]. The assembly pathway analysis we present below is general for on-lattice and off-lattice systems in 2D and 3D as detailed in Ref. [113]. The seven patchy tetrominoes (Fig. 5.3a-g) we consider first are composed of two neutral (gray) subunits, one positive (red) subunit, and one negative (blue) subunit, and can rotate and translate on a 2D lattice. These seven tetrominoes are a subset of patchy tetrominoes studied previously, and share a common free energy minimizing motif (Fig. 5.3h) which was determined at $k_BT/\epsilon = 0.1$ with BUBBA[113]. Inter-particle interaction energies are defined to model attractions and repulsions with relative magnitudes reminiscent of van der Waals, depletion, solvophobic, and/or charge-charge interactions. When two like-charged subunits share an edge, their resulting potential energy is $U = 9\epsilon$, for opposite charges $U = -11\epsilon$, and for a neutral subunit sharing a face with any other subunit type $U = -\epsilon$.

We next validate the accuracy and utility of assembly pathway analysis on a system of CdTe/CdS tetrahedra with truncated tips whose surfaces are coated with thioglycolic acid stabilizers. Previously synthesized by Tang and Kotov [12], and studied by Zhang *et al.* [80], Srivastava *et al.* [14], and Xia *et al.*[15] CdTe, CdSe, and

CdS tetrahedra coated with DMAET or TGA stabilizers have been shown to have a rich phase space of self-assembled morphologies including wires, sheets, ribbons, helices, and colloidal crystals of spherical supra-particles. This richness arises from the complicated interactions between building blocks, including their shapes, van der Waals and hydrophobic attractions, hydrogen bonding, and electrostatics. This system admits straightforward analysis with BUBBA because the particle geometry and strong face-face interactions allow for a discretization of configuration space that enables iteration over all possible cluster pairings.

We consider CdTe tetrahedra whose intrinsic dipole moment is normal to one face as in Ref. [14]. We model long-range screened charge-charge interactions as well as charge-dipole and dipole-dipole interactions between tetrahedra using linear Debye-Huckel theory as in Phillies[154], and add a constant surface potential for each pair of tetrahedral faces that are aligned as in Zhang *et. al.*[80]. The cluster degeneracies calculated by BUBBA for continuous systems require vibrational and rotational partition functions to be generated[113]. We assume the contribution of the vibrational partition function is identical for clusters of the same size, a valid assumption for these strongly-interacting particles that have been observed to fuse after assembly. This leaves the rotational partition function $Q_{rot} = B\sqrt{I}/s$ as the relevant contributor to entropy where B is a temperature-dependent constant that is identical for all clusters, I is the determinant of the inertial tensor, and s is the cluster's symmetry number [113, 116, 138].

5.4 Assembly Propensity

A system's ability to self-assemble a thermodynamically stable target pattern depends upon its path through phase space [151, 155]. Experimental conditions such as density, temperature, solvent screening effects, quench rate, etc. all play a crucial role. A primary goal in self-assembly is the maximization of the assembly yield, the

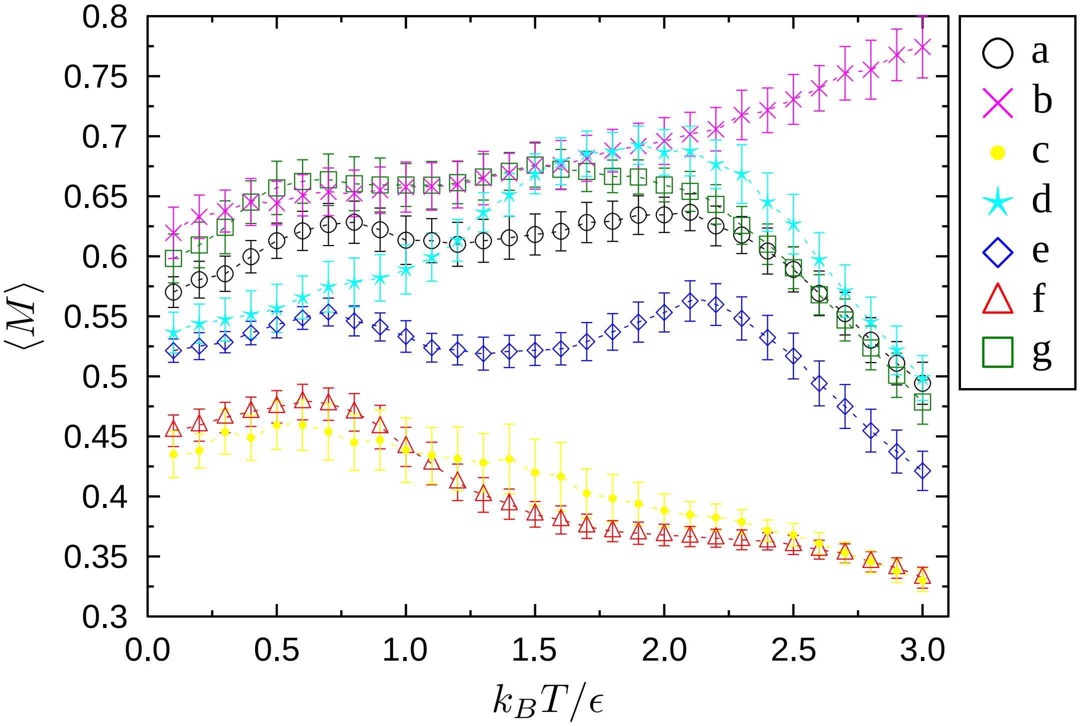


Figure 5.4: Average match to the wide stripe motif (Fig. 5.3h) as a function of temperature for the seven patchy tetrominoes from Fig. 5.3a-g. Error bars are one standard deviations of M averaged over 100 independent simulations for each data point. $\langle M \rangle > 0.7$ corresponds to a strong visual match and $\langle M \rangle < 0.6$ to very poor. Reproduced from Reference [110].

amount of desired product per unit of raw materials. It is therefore useful to define the "assembly propensity" as the degree to which the target pattern is achieved under the most optimal conditions. Given the seven patchy particles *a*-*g* in Fig. 5.3 which has the highest assembly propensity for the wide stripe motif (Fig. 5.3h)? Just by looking at these particles it is not obvious that they share the same structure at low temperatures, nor is it obvious this structure minimizes free energy at higher temperatures. Further, it is not clear why any one of these particles should self-assemble the target motif in Fig. 5.3h more robustly than any other, or which one, if any, is the optimal candidate.

The average motif matches $\langle M \rangle$ for patchy particles *a*-*g* are generated using MC simulations and are shown as a function of temperature in Fig. 5.4. Each data point is the motif match averaged over the last 5×10^6 trial moves of 100 independent simulations, at 10,000 trial-move increments, with error bars denoting one standard deviation of the resulting distribution of M values. We define the self-assembly propensity P as the average value of $\langle M \rangle$ at the experimental conditions with the largest $\langle M \rangle$. It is clear from Fig. 5.4 that the seven particles from Fig. 5.3 have substantially different propensities, ranging from 0.48 for particle c up to 0.78 for particle b. We discern no obvious link between the shape of a particle's motif match profile and the particle's geometry or interaction anisotropy.

To explain the variance in assembly propensity and the difference in $\langle M \rangle$ vs $k_B T/\epsilon$ for the seven patchy particles (Fig. 5.3a-g) studied here we consider assembly pathways which we generate with BUBBA[113]. At each stage in the assembly pathway we use shape matching[135] to identify clusters that are inconsistent with the wide stripe motif. Stages in the assembly pathway that are dominated by clusters inconsistent with a target motif are thermodynamic traps and a warning sign that a building block will not assemble robustly. As a case study we consider the assembly pathway fingerprints for patchy particles a and b, shown in Fig. 5.5 and 5.6, respectively. By visual inspection of the pathway fingerprints for these two building blocks, we expect lower assembly propensity for patchy particle a due to the prevalence of thermodynamic traps in its assembly pathways. Further, we see that for patchy particle a at $k_B T/\epsilon = 3.0$ there are more traps than at $k_B T/\epsilon = 0.8$, which is consistent with the lower average motif match measured at this state point (Fig. 5.5).

5.5 CdTe Tetrahedra

We generate assembly fingerprints for CdTe/CdS tetrahedra to validate the utility of pathway analysis for an experimentally realized system. Keeping temperature

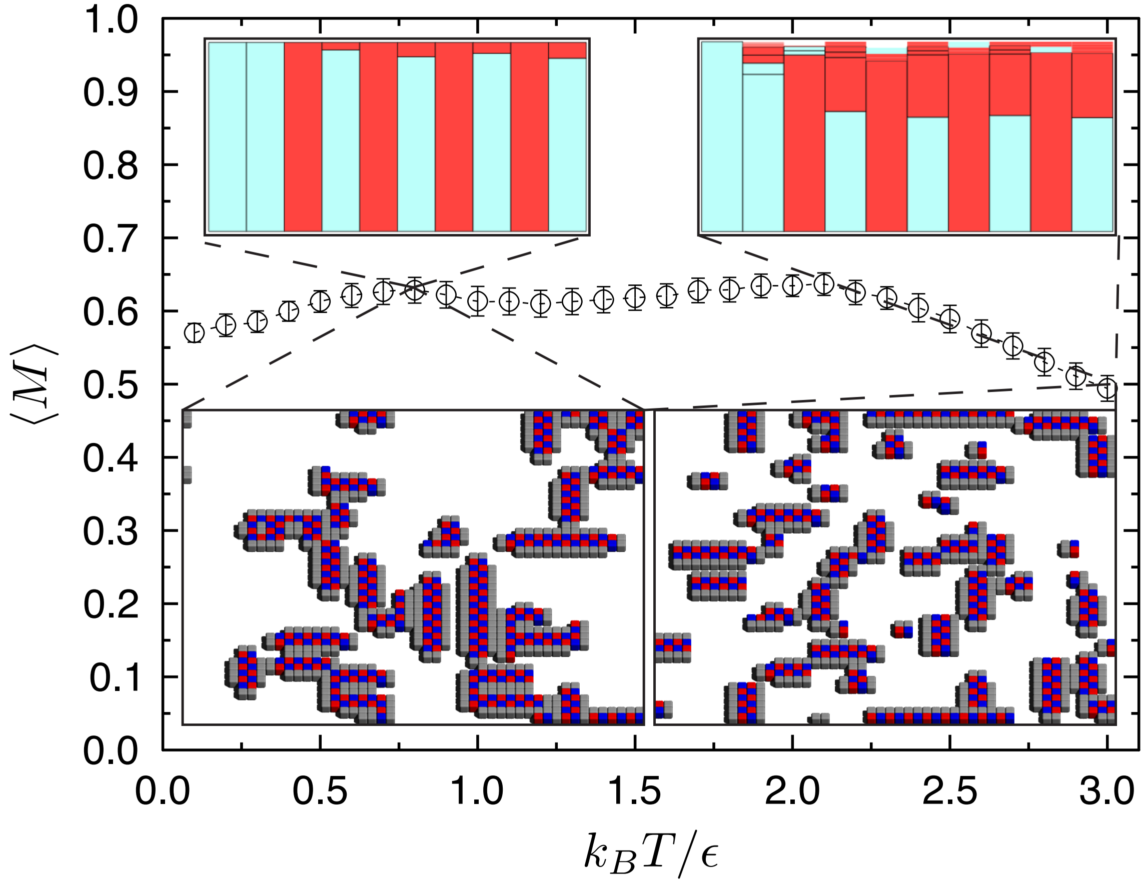


Figure 5.5: Average wide stripe motif match for patchy particle a. Assembly pathway fingerprints and representative simulation snapshots are shown for $k_BT/\epsilon = 0.8$ and $k_BT/\epsilon = 3.0$. Decreased assembly propensity is correlated to increased proportion of red in an assembly fingerprint. Reproduced from Reference [110].

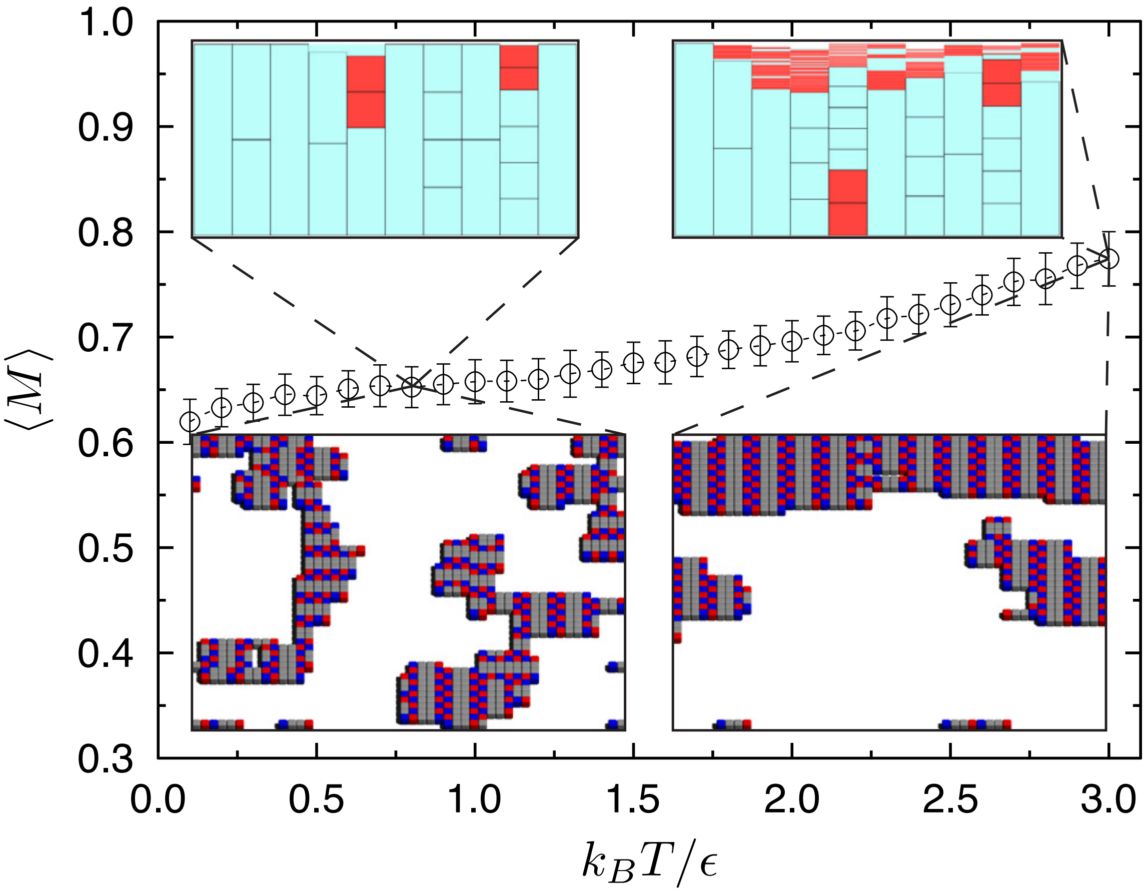


Figure 5.6: Average wide stripe motif match for patchy particle b. Assembly pathway fingerprints and representative simulation snapshots are shown for $k_B T/\epsilon = 0.8$ and $k_B T/\epsilon = 3.0$. While the lower temperature assembly fingerprint appears superior, the shorter relaxation times and predominance of motif-consistent clusters at $k_B T/\epsilon = 3.0$ allow for better assembly. Reproduced from Reference [110].

constant, we use BUBBA to generate assembly pathways for the tetrahedra as functions of charge number q, dipole moment magnitude d (in units of Debye), and surface attraction. Here we constrain the search space to consider only combinations of neighboring tetrahedra whose faces are aligned (Figure 5.7), but allow the dipoles of each tetrahedra to point out any of the four faces. For particle charge of $+1e$ and dipole strength 100 we confirm the stability of single bilayer sheets found in Ref. [12] and the double bilayer sheets found in Ref. [14] (Fig. 5.8b) at $q = +3e$ and $d = 100$. Exploring the case of double-bilayer ribbons in more detail, we generate the assembly pathway fingerprint in Fig. 5.8d in three cpu-hours. For the double bilayer (Fig. 5.8a), we consider motif-inconsistent clusters to be all clusters that have tetrahedra on three or more layers, e.g., Fig. 5.8c.

A full exploration of pathway sensitivity to charge strength, dipole strength, and surface charge is beyond the scope of the present work, but it is worth noting the complexity of the assembly pathways for these building blocks. At $N = 5$, over 6,000 clusters contribute to the partition function, with no single cluster having a weight greater than 2%. For $N = 6$ there is a handful of clusters that comprise a substantial proportion of the partition function, with many thousands of assembly pathways converging to these highest-weighted clusters. It is clear from the pathway fingerprint that not only are there many ways to combine clusters into a double bilayer (96% of the 104,396 clusters of size $N = 10$ are consistent with double bilayers), but also the proportion of out-of motif clusters at all calculated cluster sizes is low. This is expected from the ease with which the double bilayer ribbons are attained in experiments.

5.6 Screening and Designing

The computational efficiency of assembly pathway generation coupled with structure identification is well suited for the screening of patchy particles. It can be used

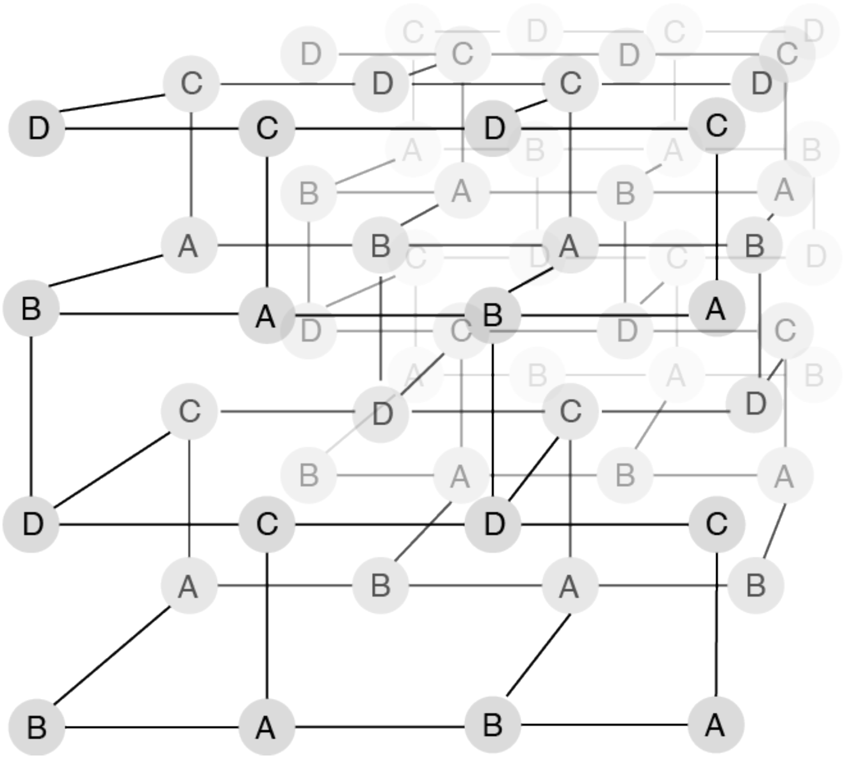


Figure 5.7: Periodic grid used to map an alternating hexagonal lattice onto a 3D square array. Each tetrahedron can have up to four neighbors, whose legal spatial positions are determined by its four faces. There are four types of lattice sites in the hexagonal array, denoted by A, B, C, and D. Each type has a unique nearest-neighborhood. For example, every A has a B-neighbor to the left, right, and out of the page, and a C-neighbor above. Every B has A-neighbors to the left, right, and into the page, and a D-neighbor below. This mapping allows the grid-based shape-matching code from Chapter IV to be used for distinguishing clusters of CdTe/CdS tetrahedra. The grid imposes hexagonal layered structure, but does not prevent multilayers (inconsistent with the double-bilayer) from being assembled.

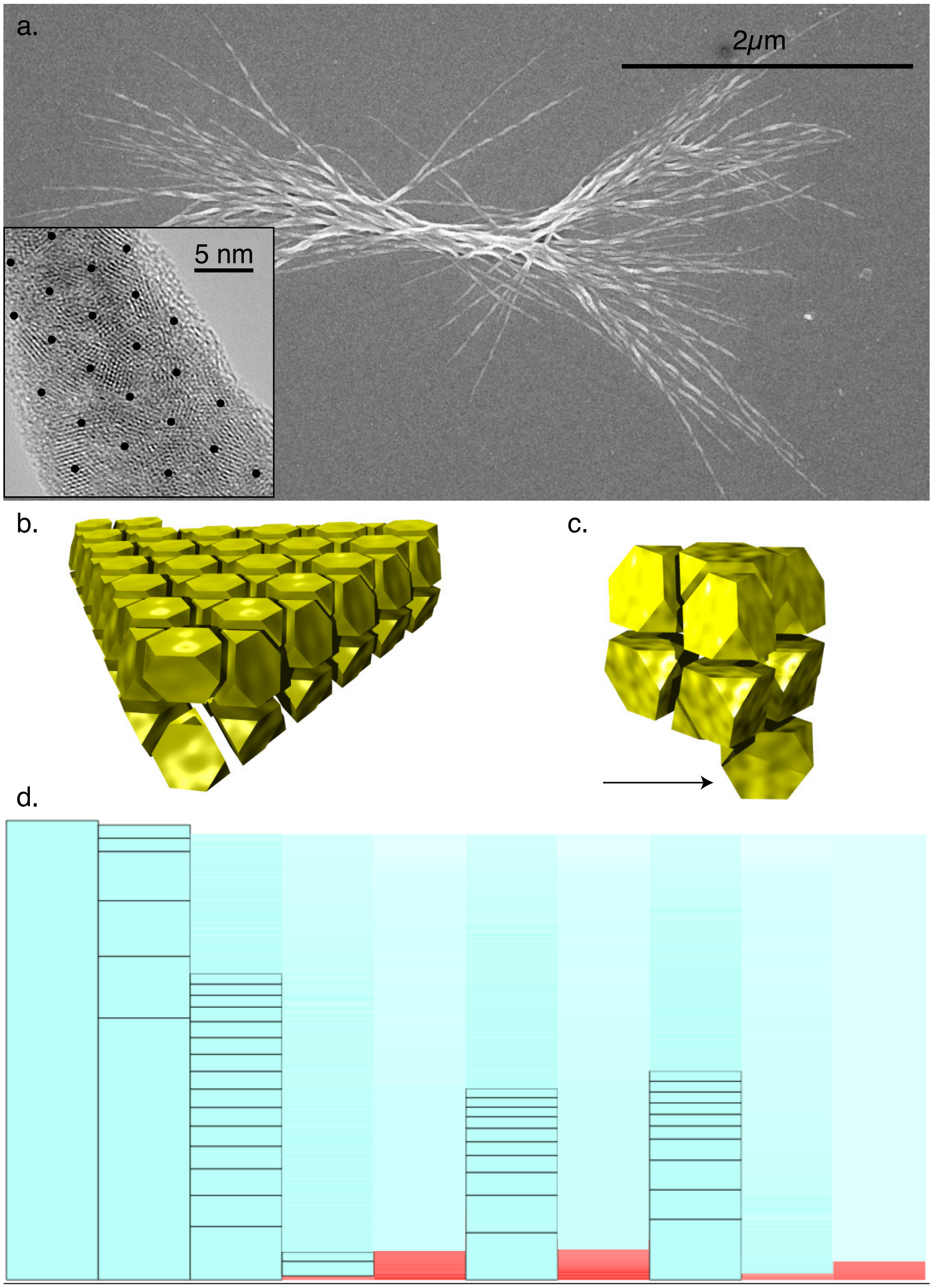


Figure 5.8: a. TEM image of double bilayer ribbons that twist into helices from Ref. [14]. Inset shows a high-resolution image of a section of a ribbon with dots indicating approximate centers of co-planar tetrahedra. b. Double bilayer motif, $N = 100$ cluster predicted by BUBBA. c. Example out-of-motif $N = 10$ cluster, arrow indicates motif-breaking particle. d. Assembly pathway fingerprint for double bilayer-forming tetrahedra from Ref.[14] with a charge of +3 and dipole moment of 100. Out-of-motif clusters have particles on more than two bilayers. Reproduced from Reference [110].

to identify the thermodynamically stable structures for a set of building blocks, as in Ref. [113]. After candidate building blocks for a motif are identified, such as the patchy tetrominoes and wide stripe motif studied here, assembly pathways can be generated at a range of experimental conditions to identify those with the greatest chance for robust assembly. Patchy particle 17 from Troisi *et al.* (shown inset in Fig. 5.9b) exemplifies the utility of screening temperatures for a building block that assembles a checkerboard motif. For reduced temperatures greater than $k_BT/\epsilon = 1.0$, it is clear in Fig. 5.9a that the majority of clusters generated with BUBBA for $N = 3 - 7$ are inconsistent with the energy-minimizing checkerboard motif, implying that optimal self-assembly should occur for $0.1 \leq k_BT/\epsilon \leq 1.0$. The motif match profile generated with MC simulations in Fig. 5.9b confirms this hypothesis. The motif match jumps from $\langle M \rangle = 0.41$ at $k_BT/\epsilon = 0.8$, to 0.57 at $k_BT/\epsilon = 0.7$, is a maximum with $\langle M \rangle = 0.70$ at $k_BT/\epsilon = 0.6$, and then drops to $\langle M \rangle = 0.60$ for $k_BT/\epsilon = 0.5$. Thus, the quick generation (90 minutes for the 30 temperatures generated serially here, compared to 6000 cpu-hours for the MC simulations optimized with cluster moves) of pathway fingerprints with BUBBA permits the identification of state space where self-assembly is optimized. Efficient screening in this way is essential, as a naïve extrapolation of the data with $k_BT/\epsilon > 1.0$ to low temperature would miss conditions with acceptable assembly.

Using shape-matching to identify motif-inconsistent clusters in assembly pathways we can determine patterns of traps that inform our pathway engineering strategies. One common pattern of trap shared across many of the patchy tetrominoes is the presence of motif-inconsistent clusters with odd-numbered sizes (e.g., $N = 3, 5, 7, 9, ...$). Patchy particle *a* exemplifies this pattern of traps (Fig. 5.5). These traps are caused by a single particle attaching to and breaking the symmetry of a desired cluster. In the case of patchy particle *a*, single-particle “caps” prevent further addition of particles to obtain wide stripes (Fig. 5.3h). This suggests that the $N = 2$ cluster may be

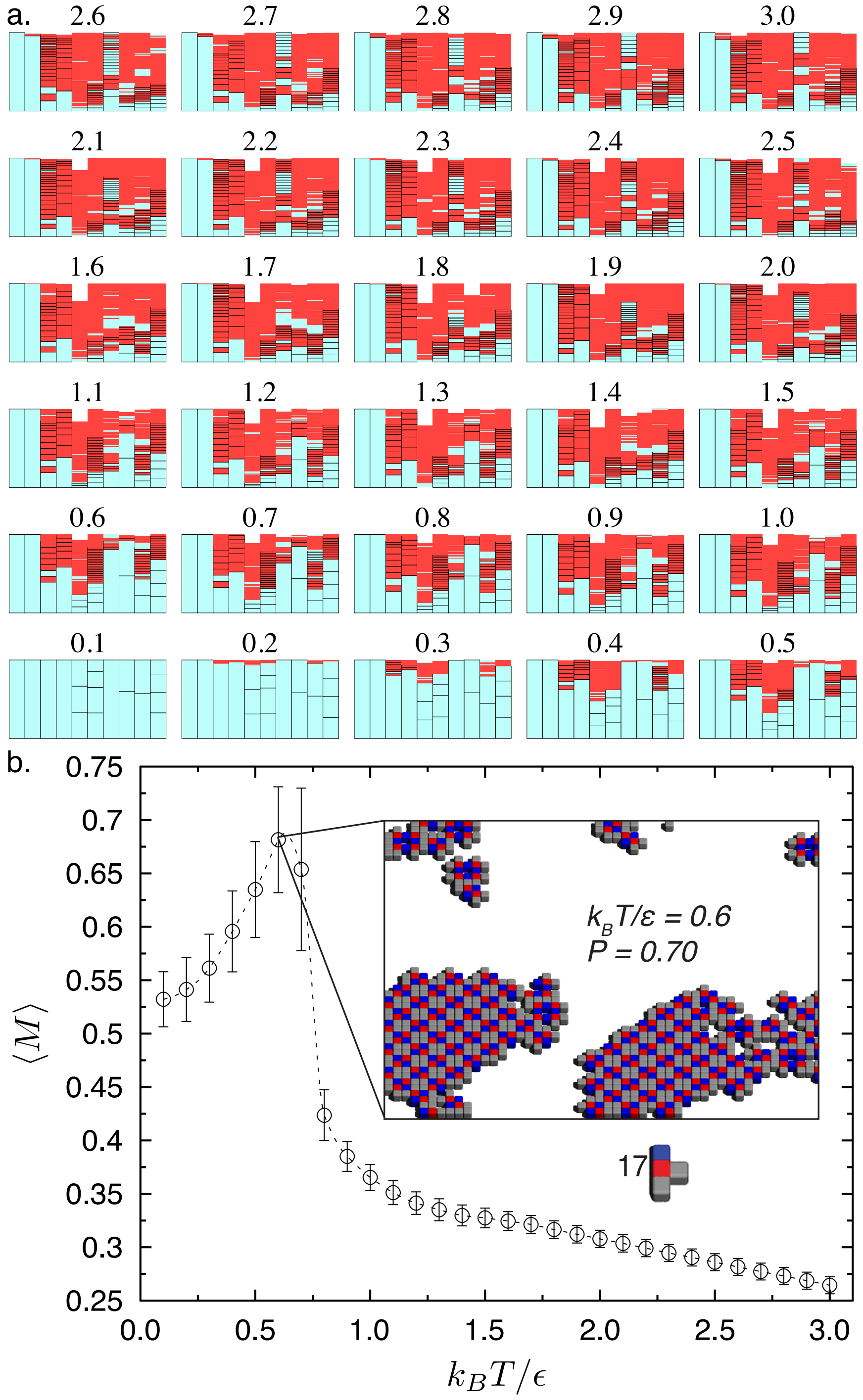


Figure 5.9: a. Pathway fingerprints for patchy particle 17 for $0.1 \leq k_BT/\epsilon \leq 3.0$. b. Motif match profile for patchy particle 17 with target motif and best-assembled snapshot inset. The motif match for this patchy particle is measured against a checkerboard reference structure that can be seen in the ordered central regions of the two large clusters in the inset snapshot. Reproduced from Reference [110].

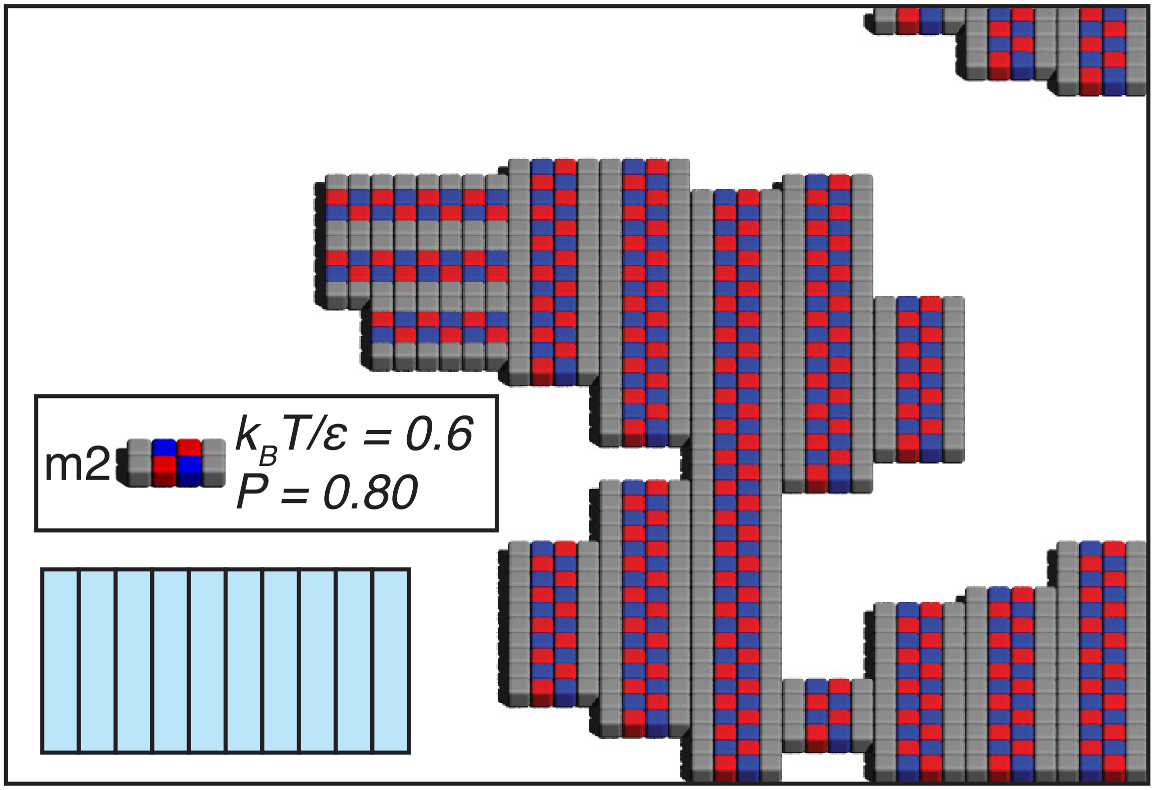


Figure 5.10: Representative MC simulation snapshot for the $N = 2$ mesoblock made from patchy particle a at $k_BT/\epsilon = 0.6$, the temperature with the maximum motif match as determined by MC simulations. The assembly pathway fingerprint at this temperature is inset, showing no thermodynamic traps for clusters with 10 or fewer building blocks. Reproduced from Reference [110].

a more effective building block candidate for self-assembly than the original particle because the same symmetry-breaking traps will be impossible to form. We perform this computer experiment by conducting Monte Carlo simulations at the same conditions as in Fig. 5.5, but where the 200 copies of particle a have been replaced by 100 copies of the $N = 2$ "mesoblock" which we denote $m2$. At the same temperature where patchy particle a assembles best with $\langle M \rangle = 0.66$, the mesoblock achieves $\langle M \rangle = 0.80$.

5.7 Discussion

We have performed assembly pathway fingerprint analysis for systems of patchy tetrominoes and truncated tetrahedra and demonstrated that there is a strong correlation between in-motif clusters in the assembly pathways of these fingerprints with favorable assembly propensity. There are three related ways in which comparisons between fingerprints presented here have not been perfectly correlated with assembly propensity measured by MC simulations at the same conditions. The first instance is the higher assembly propensity for patchy particle b at higher temperatures (Fig. 5.6), despite the presence of more motif-inconsistent clusters. The second is the drop in assembly propensity for particle 17 of Ref. [113] when $k_BT/\epsilon < 0.6$ (Fig. 5.9) despite the favorable-looking fingerprints. The third case is the imperfect assembly in Fig. 5.10 despite the perfect assembly fingerprint. All three cases are explained by assembly kinetics. For patchy particles b and 17, the thermodynamically preferred clusters generated by BUBBA show that larger proportions of the partition function are represented by in-motif clusters at low temperatures. At these low temperatures, however, the relaxation timescales are too long for robust assembly to occur. As BUBBA is a thermodynamic method, it does not predict regions of kinetic trapping, which highlights the important complementary contribution of the methods developed by Jack, Klotsa, Hagan, and Chandler[148, 149]. The imperfect assembly of the mesoblock in Fig. 5.10 is also due to assembly kinetics. In this case the timescale of motif-consistent clusters agglomerating end-on rather than perpendicular diverges as clusters grow in length. While this is unfortunate for this particular building block, it also provides a path forward for improved assembly, as it suggests that increasing the attraction between grey subunits might facilitate the wide stripe formation.

5.8 Conclusion

Through the use of assembly pathways, we have demonstrated that both model and real patchy particles can be efficiently screened for assembly propensity. We showed how building blocks designed to avoid particular barriers might assemble target patterns with higher propensity. We also showed how successive steps revealed in the assembly pathways could provide a blueprint for directed bottom-up assembly. Further, we showed that the fingerprint visualization of pathways is a useful tool in identifying thermodynamic conditions (such as temperature) that maximize self-assembly propensity. The computational efficiency of generating pathway fingerprints compared to experiments with unknown relaxation timescales makes it ideal for screening candidate building blocks and experimental conditions. Combining pathway-based screening techniques with assembly kinetics analysis we proposed a methodology for assembly pathway engineering which proceeds as in Fig. 5.1. We expect this method, its extensions, and alternative implementations to play a central role in the focused development of assembly engineering strategies.

CHAPTER VI

Applications of BUBBA

6.1 Introduction

In this chapter we provide an overview of additional systems where implementations of BUBBA have been used to successfully predict structure and screen for assembly candidates. This chapter is organized as follows. In Section 6.2 we detail the calculation of vibrational partition functions for DNA-tethered spheres and demonstrate the accurate prediction of small terminal clusters over a wide range of experimental conditions. In Section 6.3 we describe the off-lattice implementation of BUBBA for 2D disks used to study continuous patchy tetromino analogues and a model system of shape-shifting colloids. The work in this chapter is adapted from Reference [156] D. Ortiz, E. Jankowski, and S.C. Glotzer, Self-assembly of colloidal tetrominoes, *in prep*, Reference [157] T.D. Nguyen, E. Jankowski, and S.C. Glotzer, Self-assembly and reconfigurability of shape-shifting particles *ACS Nano*, 5, 11, 2011, and Reference [158] C.L. Phillips, E. Jankowski, M. Marval, and S.C. Glotzer, Self-assembling clusters inspired by mathematical extremal points on the surface of a sphere. *arxiv:1201.5131* 2012.

6.2 Spherical Codes

Anisotropic particles are compelling building blocks for self-assembled materials because their directional interactions can be exploited to create complicated and useful patterns, which has been the focus of this and previous work[3, 113, 147, 159–162]. One way to create anisotropic building blocks is to self-assemble them from simpler particles, where the building block represents a free-energy minimizing structure. Recently a number of papers have been published synthesizing and simulating compound building blocks that are clusters of spheres[22–27, 73, 74, 147, 163–165]. Colloidal spheres are attractive candidates for assembly because they can be made from a wide variety of polymers and metals, and their interaction potentials can be tuned with organic ligands, solvents, and salts. [1]

Two of the most ubiquitous nano-scale building blocks available today are colloidal spheres and DNA[166, 166–172], both of which can be synthesized in bulk quantities and with a high degree of homogeneity. Colloids are attractive candidates for assembly because they can be made from a wide variety of polymers and metals, whose interaction potentials can be tuned with polymer surfactants, solvents, and salts. Combined with DNA, whose specific interactions can be tuned to different length scales and energy scales, there exists much potential for the assembly of colloidal clusters that can be used at the building blocks for more complex structures.

Consider a colloidal sphere that is uniformly coated with single-stranded DNA and exposed to a bath of larger colloidal spheres grafted with complementary strands of DNA. Given sufficient assembly time and no other attractions between particles, the larger "halo" spheres will bind to the smaller "central" sphere until no more halo spheres can fit. The self-limiting "terminal" cluster that results can be used as an anisotropic building block [80, 173, 174], as a template for creating different

[1] This section is adapted from Reference [158] C.L. Phillips, E. Jankowski, M. Marval, and S.C. Glotzer Self-assembling clusters inspired by mathematical extremal points on the surface of a sphere. *arxiv:1201.5131* 2012.

anisotropic building blocks, or as a container for the central particle cargo inside (Fig. 6.1). Using terminal clusters as anisotropic building blocks requires that the clusters are themselves uniform in size and shape.

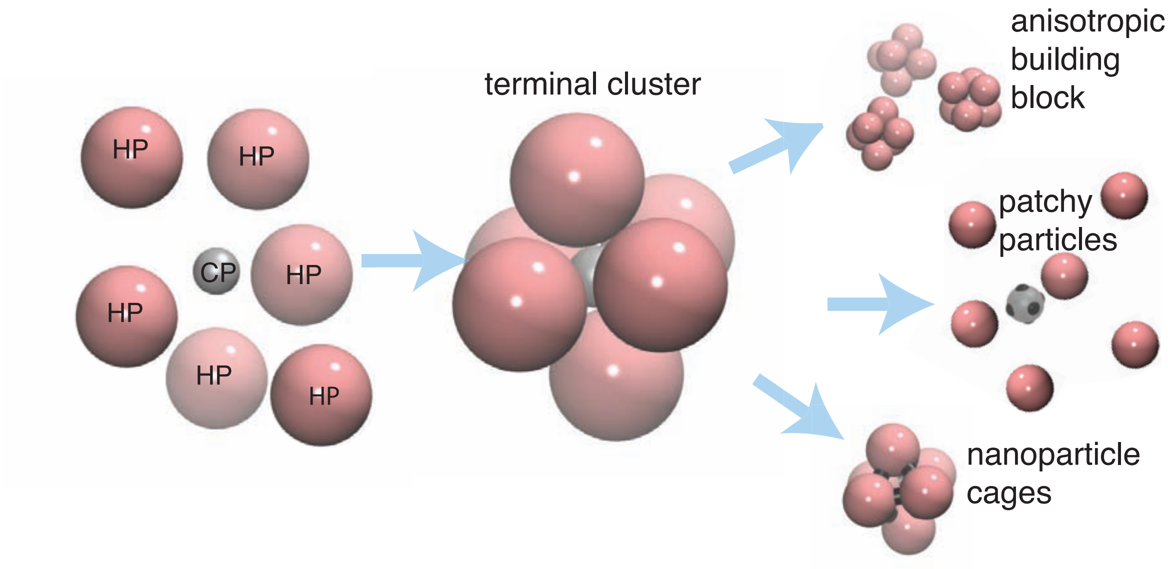


Figure 6.1: A terminal N-cluster with an octahedral structure ($N = 6$) is self-assembled from a bath of halo particle (pink) and a central particle (grey). This cluster has applications as an anisotropic building block, could be used to manufacture a "patchy particle" by imparting patches on the CP at the contact points, or could be locked into a nanocolloidal cage structure. Reproduced from Reference [158].

Arrangements of halo particles (HPs) on the surface of a central particle (CP) have been studied extensively by mathematicians in the context of optimal arrangements of points on a sphere[175–177]. The solutions provide a library of anisotropic clusters that can in principle be created with properly designed interactions among the constituent particles. In this work we study hard sphere HPs that are attractive only to dilute CPs and not to other HPs, thereby producing clusters of HPs around a single CP. The arrangements of these HPs bear comparison to a particular set of solutions, the *spherical codes*, for certain ratios of particle diameters (Figure 6.2). In this work we detail the free energy calculations used with BUBBA to predict terminal clusters of HPs around a CP as a function of temperature and diameter ratio. We show that the energetic and vibrational portions of the partition function provide leading order predictive capabilities.

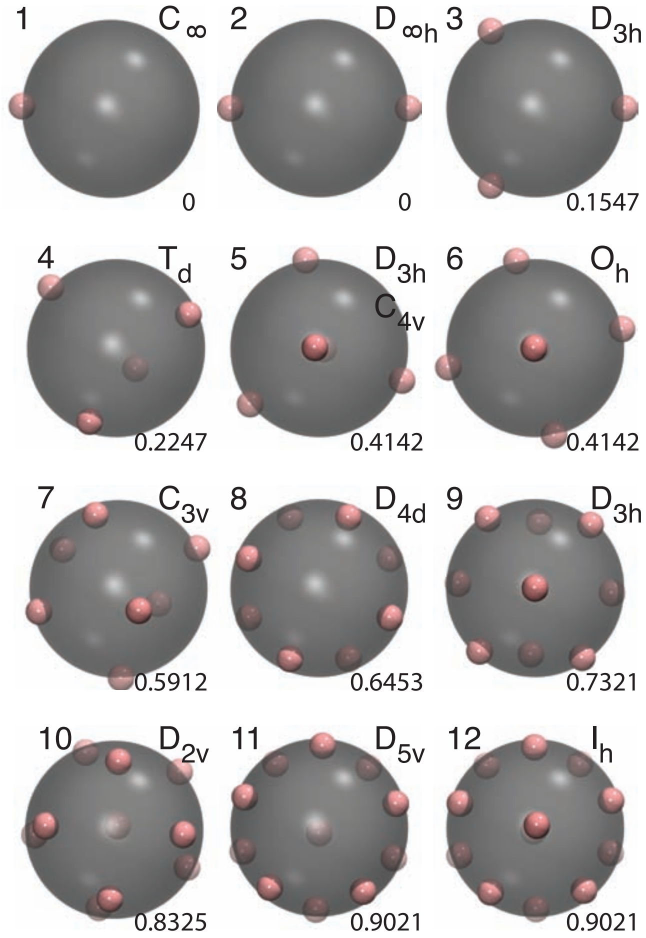


Figure 6.2: The arrangement of points (pink) that correspond to each spherical code solution for $1 \leq N \leq 12$. The point group of each arrangement is shown to the upper right of each arrangement, and the densest packing diameter ratio $D_c/D_h = \Lambda_N$ is shown to the lower right. For $N = 5$, the triangular bipyramid configuration is shown. Reproduced from Reference [158].

Fig. 6.2 depicts the spherical code solutions for $1 \leq N \leq 12$. The arrangement of points for $N = 4$ corresponds to the vertices of a regular tetrahedron, $N = 6$ an octahedron, $N = 8$ a square anti-prism, and $N = 12$ an icosahedron. The point arrangement of $N = 11$ is equal to the $N = 12$ solution minus a single point, or an icosahedron with one truncated pentagonal face. For each N, the point group – the group of isometries that keeps one point fixed – of the arrangement[176] is shown in the upper right corner. Each optimal arrangement of N points on the surface of the sphere is unique except for $N = 5$ which has a continuum of solutions ranging from a triangular bipyramid (point group D_{3h}, shown in Fig. 6.2, to a square pyramid (point group C_{4v})). All solutions in the $N = 5$ continuum have two points at opposite poles of the central sphere and differ by the positions of the three remaining points on the equator. The square pyramid arrangement is equal to the $N = 6$ solution minus a single point.

If the N points represent sphere centers, the spherical code solution corresponds to the densest packing of N hard halo spheres that all "kiss" a central sphere. For any packing of spheres around a central sphere, we define Λ to be the ratio of the central sphere diameter, $D_c = 1$, to the halo sphere diameter, D_h. We denote the minimal possible diameter ratio for N spheres, which corresponds to the spherical code solution, as Λ_N. In Fig. 6.2, Λ_N of each arrangement is shown to four significant digits in the bottom right corner. Notably, $\Lambda_{N=5} = \Lambda_{N=6}$ and $\Lambda_{N=11} = \Lambda_{N=12}$.

6.2.1 Free Energy Calculations

We predict the probability of observing a cluster of N HPs around a CP by calculating the grand canonical (μ, Λ, T) partition function for all allowable HP-CP clusters. The grand canonical partition function is

$$\mathcal{Z}(\mu, \Lambda, T) = \sum_{\gamma} \Omega_{\gamma} e^{-\beta(U_{\gamma} - \mu N)} \tag{6.1}$$

where μ is the chemical potential of an HP, U_γ is the potential energy of cluster γ, Ω_γ is the entropy of cluster γ, and $\beta = 1/k_B T$. Here, we define a hard sphere interaction between HPs

$$U_{HP-HP}(r) = \begin{cases} \infty & r < D_h \\ 0 & r \geq D_h \end{cases} \tag{6.2}$$

which prevents HPs from overlapping and gives all clusters the same potential energy $U = 0$. We define $\mu = \epsilon_0$ and express our dimensionless temperatures $T^* = k_B T/\epsilon_0$ in units of ϵ_0. The Boltzmann weight of a single cluster γ is therefore

$$w(\gamma) = \Omega_\gamma e^{\beta\mu N} \tag{6.3}$$

and the probability of observing cluster γ is

$$P(\gamma) = w(\gamma)/\mathcal{Z}. \tag{6.4}$$

For a particular $\Lambda \leq \Lambda_N$ clusters with up to N HPs can be assembled and the likelihood of one cluster over any other depends upon the balance between chemical potential obtained with larger N and the vibrational freedom obtained for smaller N. To calculate the vibrational partition functions for an N-cluster, we first assume an equilibrium configuration defined by N HPs in a spherical code configuration. The cluster's entropy here is determined by the product of the vibrational degrees of its HPs. The vibrational freedom of each HP can be measured as the fractional area of the surface of the CP it has access to, subject to the restrictions imposed by its neighboring spheres (Figure 6.3). We approximate the vibrational area available to a given HP in a particular configuration by using a Monte Carlo numerical approach whereby new positions for the HP are randomly generated and accepted if the HP does not overlap another HP. The accessible vibrational area is proportional to the total number of accepted positions that are part of a contiguous area that includes the HP's original position divided by the total number of random trials.

6.2.2 Results

We calculate $\mathcal{Z}(\mu, \Lambda, T^*)$ as the sum over spherical code clusters with $1 \leq N \leq 12$ for $0.1 \leq \Lambda \leq 1.4$ and $0.02 \leq T^* \leq 0.2$. In the "phase diagram" shown in the lower

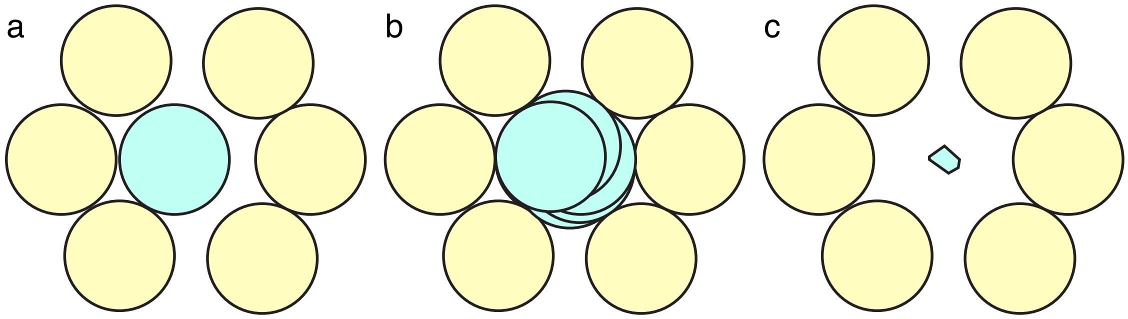


Figure 6.3: Monte Carlo integration of vibrational degrees of freedom. a. Given a configuration of particles, here 2D discs, but equivalently 3D HPs on the surface of a CP, we aim to calculate the volume accessible by the blue disk, which is constrained by the yellow disks. b. We generate random vectors specifying the center of the blue disk and check to see if they overlap with any yellow disks. c. The total accessible volume is approximated by the ratio of randomly drawn points that fall within the small blue area over the total number of vectors generated.

left of Figure 6.4 we indicate the regions in (T^*, Λ) space in which one spherical code cluster is more likely than all others. Colored lines in the phase diagram are coexistence curves that indicate the clusters on either side of the curve are equally likely. We compare our predicted cluster probabilities against cluster likelihood observations from Brownian dynamics simulations performed in [158]. The plots in the upper left and lower right of Figure 6.4 compare our predictions with the Brownian dynamics experiments at $T^* = 0.1$ and $T^* = 0.02$, respectively. Here, Λ^m refers to the diameter ratio specified in the Brownian dynamics simulations and Λ^f refers to the diameter ratio from our free energy calculations.

The qualitative agreement between the free energy calculations and Brownian dynamics simulations is striking and indicates that the competition between chemical potential and single-HP vibrational degrees of freedom is the primary factor in determining cluster likelihood. The two notable discrepancies between our free energy calculations and the Brownian dynamics observations are 1) that it appears the cluster probability curves are shifted to higher Λ in the free energy calculations, and 2) the

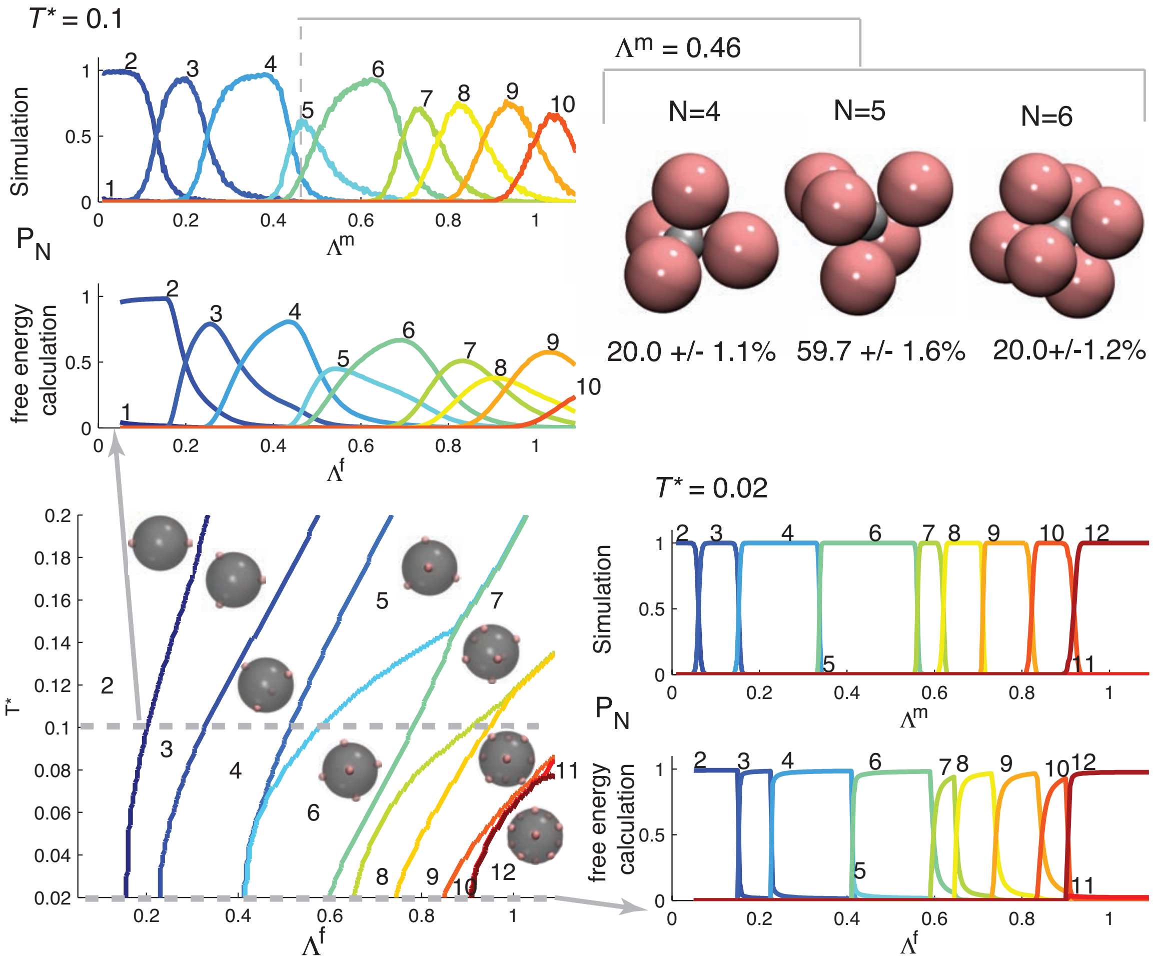


Figure 6.4: The distributions of cluster sizes as a function of temperature and Λ as given by the free energy calculation and the BD simulations are compared. Bottom left corner: phase diagram of the free energy prediction of the most probable cluster size. Lower right and upper left corners: in-page slices of the probability of finding each cluster size P_N as predicted by the free energy calculation and BD simulation at the high and low temperature. Upper right corner: the three most common clusters found in the BD simulation at the high temperature and $\Lambda^m = 0.46$. . Reproduced from Reference [158].

peaks in the Brownian dynamics experiments are slightly sharper (less broad) than for the free energy calculations. The diameter discrepancy is due to the difficulty in precisely modeling hard spheres using integrable potentials. The shape discrepancy is due to collective modes of particle motion contributing to the vibrational degrees of freedom, and are characterized in detail in [158].

6.3 Off-Lattice BUBBA

In this section we describe our spatial decomposition scheme used to study off-lattice analogues of the tetrominoes studied in previous chapters and report on energy-minimizing tilings generated using BUBBA. Extending BUBBA to off-lattice systems, as previously discussed in Chapter IV, depends upon how accurately a system with continuous degrees of freedom can be discretized to allow cluster pairing and shape-matching algorithms to be used in the hierarchical generation of clusters.

The off-lattice discretization scheme used here is described in Figure 6.5. For each disk that comprises a patchy particle we generate a set of neighbor points (pink circles in Figure 6.5b) at regular intervals around a circle centered at the disk center (diameter σ), with radius σ. The example shown in Figure 6.5b is for 12 points at intervals of $\pi/6$ radians, but in practice we use 72 points at intervals of $\pi/36$ radians to ensure both hexagonal and cubic tilings (and their $\pi/36$ rotations) are considered. When two patchy particles are paired, BUBBA selects one subunit (here, the orange disk in Figure 6.5c) to be placed nearby a subunit (here, the green disk in Figure 6.5c) of the stationary (blue) cluster. For each proposed pairing, BUBBA places the center of the orange disk at the center of a neighbor point of the green disk and checks all 72 orientations as it generates new clusters. Once two clusters have been combined into a larger cluster, the neighbor points for the larger cluster is the union of the neighbor points of its subunits that do not overlap with any subunits (pink circles in Figure 6.5d).

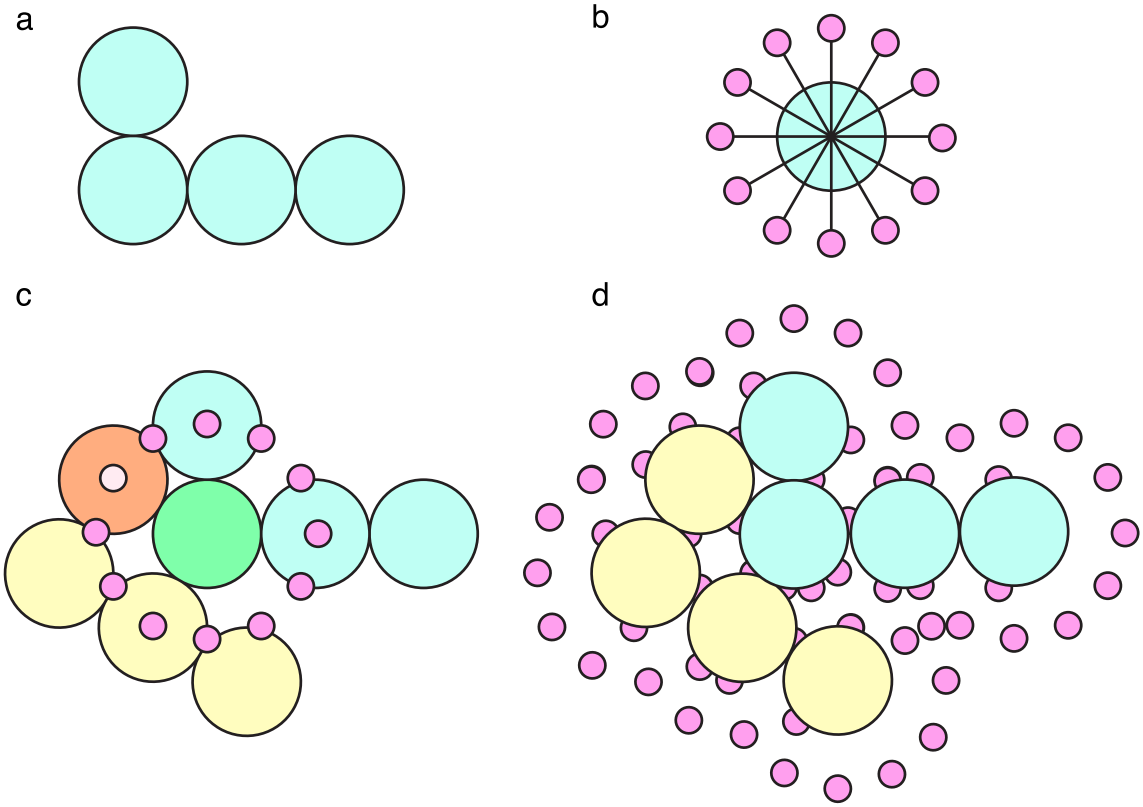


Figure 6.5: Patchy tetromino spatial discretization. a. Example building block composed of four disks. b. Neighbor sites are indicated with pink circles and are determined by uniformly dividing the unit circle. Here, 12 neighbor sites are shown. For systems of particles with long-range potentials, minima in the interaction potential would determine the radial distance of neighbor sites, otherwise neighbor areas/volumes must be defined. c. Example cluster pairing performed by BUBBA. Here, the red subunit of the yellow cluster is placed on a neighbor site of the blue cluster's green subunit. d. The neighbor set for a composite cluster is determined by the union of the neighbor points of its constituent subunits that are not contained in any subunit (pink).

6.3.1 Assembly of Shape-Shifting Particles

We use BUBBA to study a system of anisotropic particles from [157] that are composed of 2D disks and which can shift shape from one geometry to another. In this work each building blocks is composed of four or five Lennard-Jones (LJ) disks. We use BUBBA to predict the energy-minimizing configurations of these building blocks, which are inset in Figures 6.6-6.8. In companion Brownian dynamics simula-

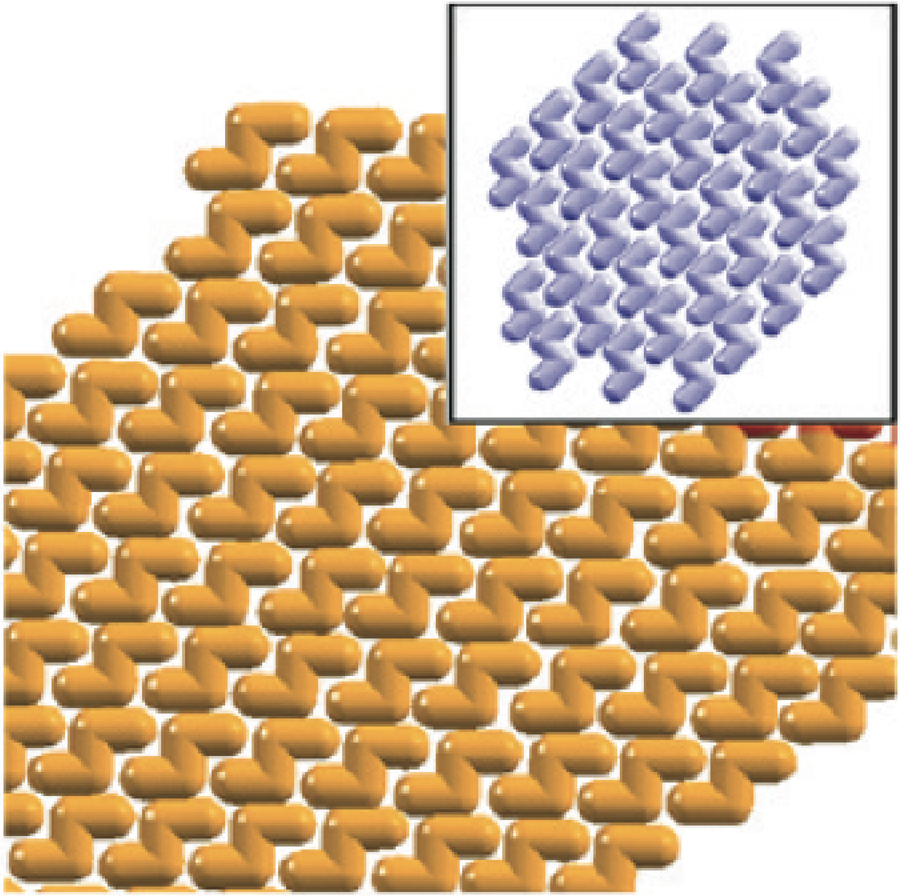


Figure 6.6: Final Brownian dynamics snapshot of zigzag particles composed of four LJ disks that shape-shift from self-assembled rods. The energy-minimizing structure for the zigzags is inset and predicted by BUBBA. Reproduced from Reference [157].

tions where particles switch shape, it is demonstrated that easy-to-assemble shapes such as rods and disks can be used to template structures which can are then transformed. Slowly shifting particle shape from the rods and disks into more complicated structures, such as zigzags (Figures 6.6, 6.7) or “Y”-shapes allows the complicated energy-minimizing structures to be achieved where direct assembly could not.[2]

[2] This subsection is adapted from Reference [157] T.D. Nguyen, E. Jankowski, and S.C. Glotzer. Self-assembly and reconfigurability of shape-shifting particles. *ACS Nano*, 5, 11, 2011.

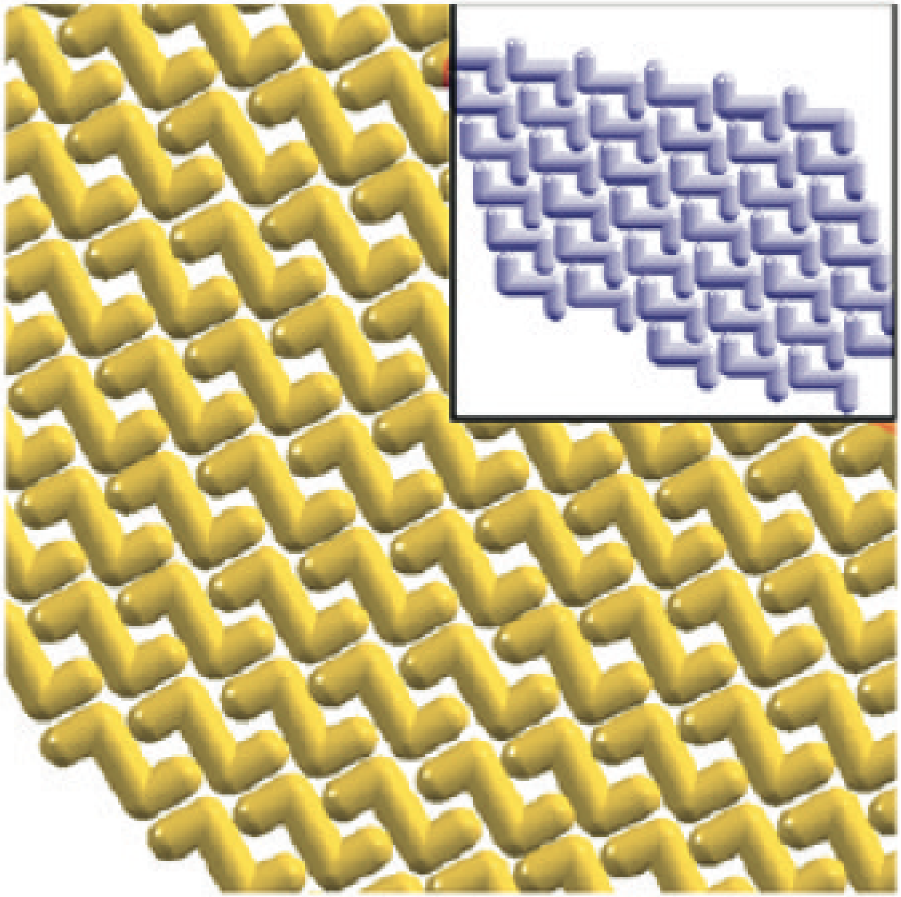


Figure 6.7: Final Brownian dynamics snapshot of zigzags composed of five LJ disks that shape-shift from self-assembled rods. The energy-minimizing structure for the zigzags is predicted by BUBBA and inset. Reproduced from Reference [157].

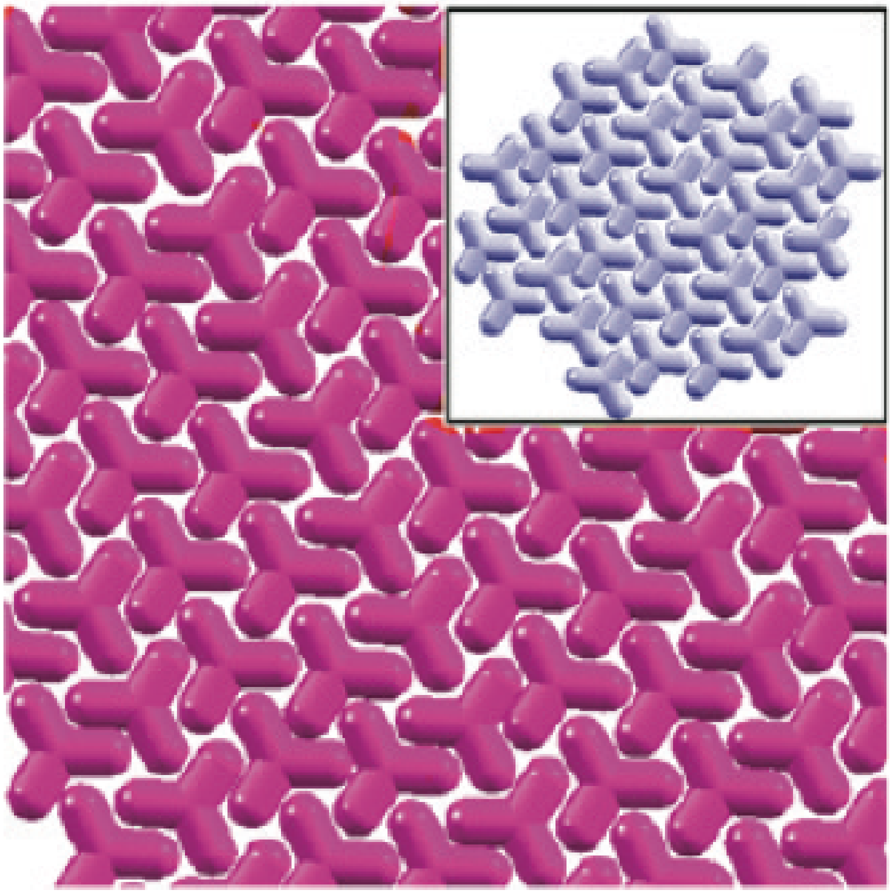


Figure 6.8: Final Brownian dynamics snapshot of “Y”-particles composed of four LJ disks that shape-shift from self-assembled disks. The energy-minimizing structure is predicted by BUBBA and inset. Reproduced from Reference [157].

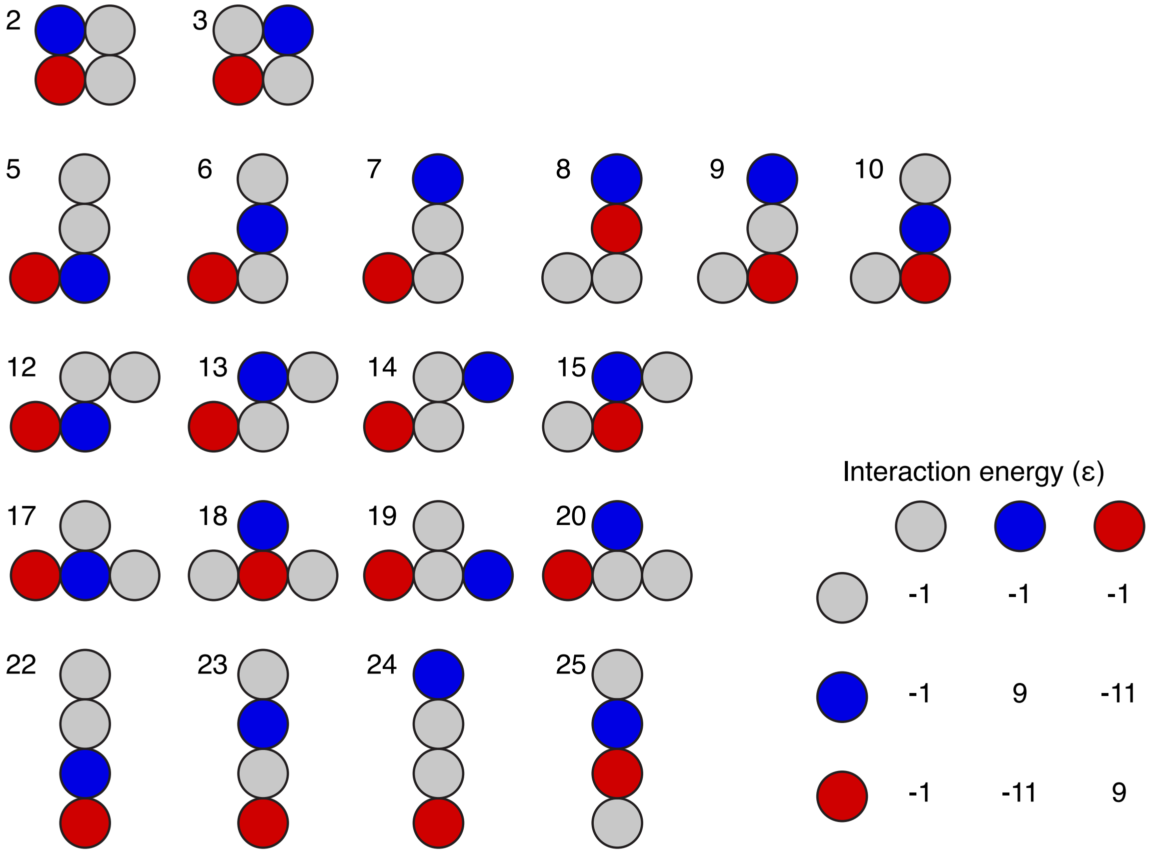
6.3.2 Patchy Tetromino Results

In this work we consider a set of twenty patchy particles that could in principle be synthesized from colloidal spheres (Figure 6.9)[31]. These particles are the off-lattice analogues of the patchy tetrominoes studied in Chapters III-V. For exploratory purposes, we consider only energy-minimizing clusters with BUBBA and generate clusters up to $N = 55$ for the building blocks in Figure 6.9. Similar to the on-lattice case we find that there exist energy-minimizing motifs that are preferred by multiple building blocks (e.g. Figures 6.10 and 6.11). We also see packings that are generated for only a single building block (Figures 6.12). The additional rotational degrees of freedom and the ability for particle subunit disks to "interdigitate" expand the complexity of the periodic structures stabilized by these particles relative to their on-lattice analogues. All of the packings generated are included in Appendix B for completion.[3]

The off-lattice analogue of the wide stripe packing observed in Chapter IV has parallel rows of neutral and charged species, but the neutral subunits are able to interdigitate to achieve lower energy. Like its on-lattice counterpart, this wide stripe motif structure is favored by more building blocks than any other.

The big hexagonal rings in Figure 6.11 are stabilized by building blocks 18 and 15. Hexagonal packings such as this are not possible on the square lattice. The case of particle 19 is particularly interesting because the large hexagonal rings in Figure 6.12 are composed of 108 building blocks. The tiling in Figure 6.12 was not generated by BUBBA, but was constructed from the cluster of 54 building blocks (Figure 6.13). Six copies of particle 19 make a small cluster, three copies of which make the small three-fold symmetric ring. Six of the small rings comprise the large ring observed in the Figure 6.12 tiling. Of course we cannot say that this tiling minimizes energy,

[3] This subsection is adapted from Reference [156] D. Ortiz, E. Jankowski, and S.C. Glotzer Self-assembly of colloidal tetrominoes, *in prep.*



Interaction energy (ε)

	⚪	🔵	🔴
⚪	-1	-1	-1
🔵	-1	9	-11
🔴	-1	-11	9

Figure 6.9: Twenty patchy tetrominoes, each comprised of two grey “neutral” disks, one red “positive” disk and one blue “negative” disk. Interaction energies describe the depth of the square-well potential in units of $k_B T$ when two disks centers are $\sigma + 0.05\sigma$ apart or closer.

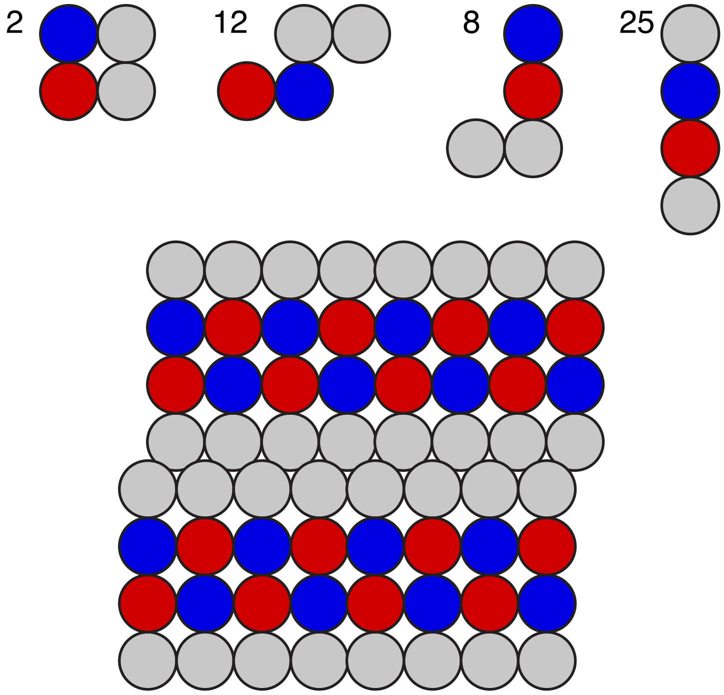


Figure 6.10: Interdigitated wide stripes

as surely particles can be added within the gaps to lower potential energy, but this example serves to highlight the richness in structure that can arise from a simple building block. Whether this tiling minimizes energy at this packing fraction is an open question that is the subject of future work.

6.4 Conclusions

The partition function calculations and cluster pairing routines which enable BUBBA's success for on-lattice patchy particles show promise for extension to more complicated systems. We show that self-assembly is a feasible fabrication technique for making anisotropic building blocks from easier-to-fabricate isotropic particles. This expands our ability to select particle properties which could give rise to more complicated self-assembled structures. We demonstrate the prediction of energy-minimizing patterns using off-lattice implementations of BUBBA and quickly screen the intricate packings and periodic motifs that are stabilized by composite colloids

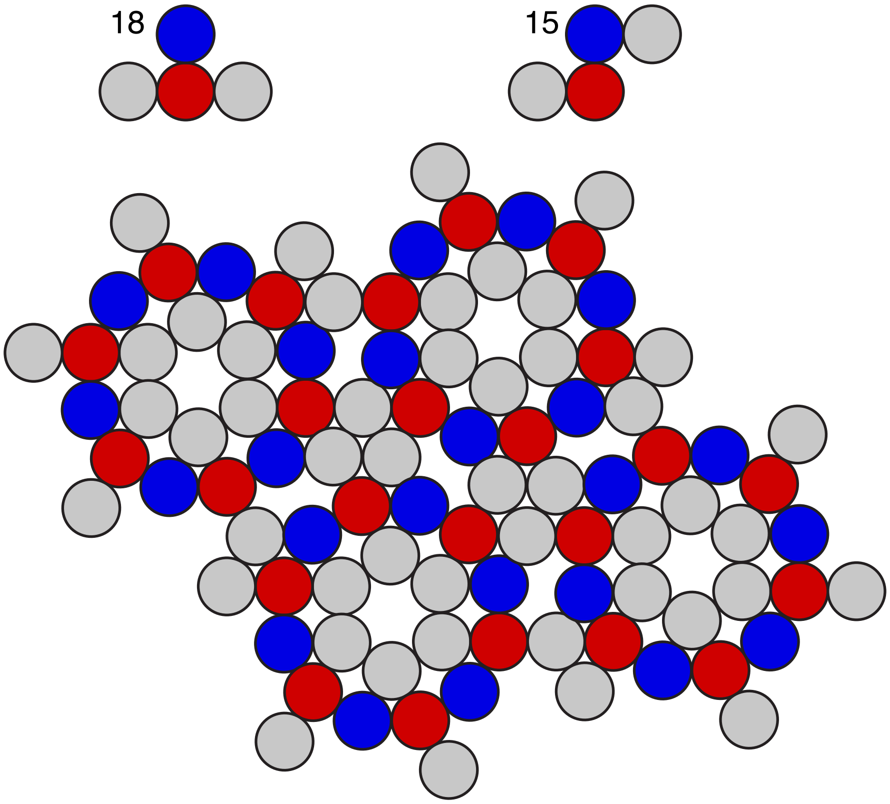


Figure 6.11: Hexagonal rings

in 2D. The thermodynamic stability of zigzags and “Y” particles are confirmed by Brownian dynamics simulations and we find a wide variety of stable structures can be created from collections of disk-based patchy tetrominoes.

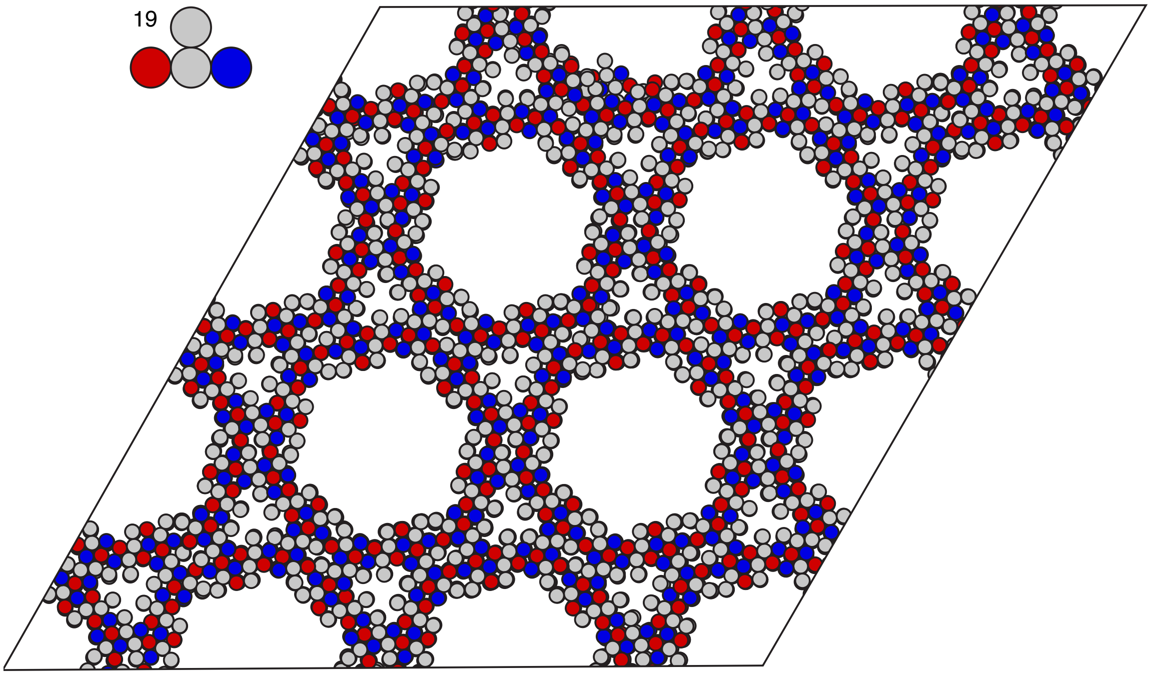


Figure 6.12: Hexagonal packing of particle 19 into a pattern with 108 particles making up one ring.

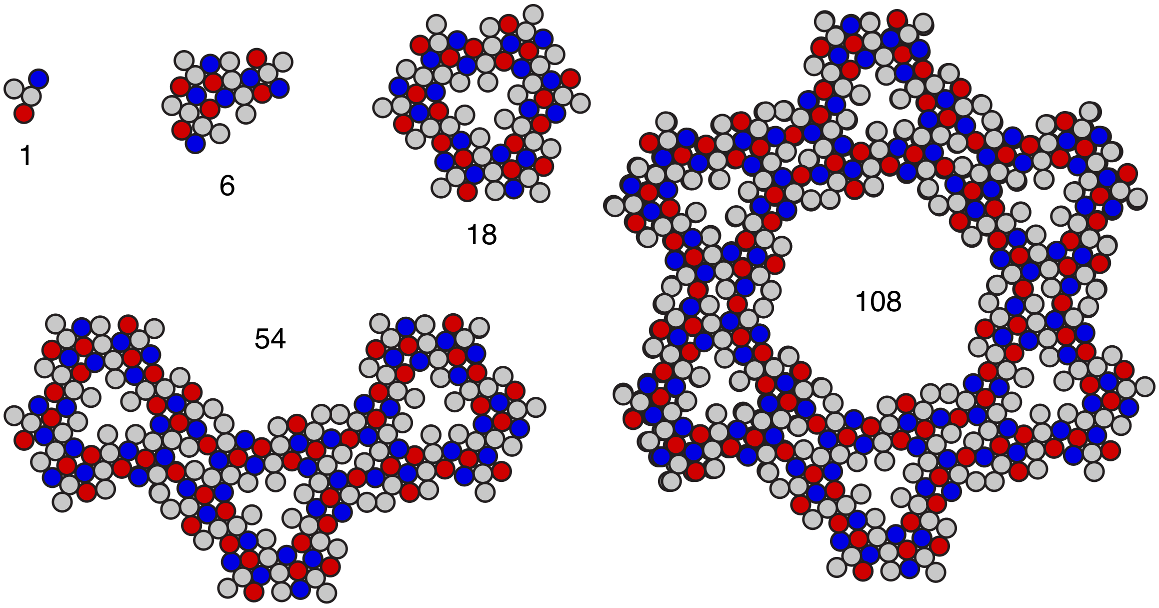


Figure 6.13: Hierarchical assembly of particle 19 into a cluster of size 6, 18, and 108. The cluster of size 54 generated with BUBBA was used to identify the hexagonal cluster of 108 particles which tiles the lattice in Figure 6.12.

CHAPTER VII

Conclusions

7.1 Summary

In this thesis we have developed new computational tools and theory for studying the self-assembly of patchy particles. Our goal was to understand how to choose building blocks that self-assemble into useful structures. Broadly speaking this problem remains unsolved as we still lack a prescription that states, for a given material, whether it can be fabricated via self-assembly and if so what building blocks to use. We have, however, made substantial progress towards this aim. The tools we have developed enable increasingly complex building blocks to be studied more efficiency and demonstrate how experimental costs and design time can be reduced by connecting theory to fabrication through computation.

The patchy tetrominoes we consider in Chapter III exemplify how complex behaviors can arise for seemingly simple agents. It was not possible *a priori* to know the diversity of energy-minimizing structures that were stabilized for these particles, or that the difficulty in assembling them varied nonmonatonically with temperature. The problem of determining their energy-minimizing configurations falls under the umbrella of NP-complete problems, the class of problems for which it is famously unknown if there are efficient computational solutions. From this perspective, the challenge of understanding self-assembly in general may seem overwhelming, because if we cannot prove solutions for simplified models of particles, how can we expect to understand the behavior of real molecules? The varying degrees of success with which Monte Carlo and heuristic approaches have predicted patchy particle properties underscores the need for improved tools in this arena. Our cluster Monte Carlo code was one such advancement that enabled assembly simulations to be performed more quickly, but is still kinetically limited from providing predictive capabilities at low temperatures.

Our work developing bottom-up building block assembly (BUBBA) is informed and inspired by the barriers to property prediction we encountered with Monte Carlo methods. By divorcing the kinetic problem from the thermodynamic problem, we show that easily calculated partition functions for small clusters can be used effectively to understand the thermodynamics of larger clusters. The philosophy behind BUBBA is a consistent thread throughout our study of self-assembly: solve the solvable problems first and build towards larger solutions. The predictive capabilities of BUBBA for patchy tetrominoes shows that developing new heuristics for tackling NP-complete problems is not a challenge that should be avoided out of fiat, and can result in useful tools that complement materials design.

In addition to providing predictive capabilities for collections of patchy particles, we develop the idea of assembly pathway engineering that is enabled by BUBBA. The ability to identify thermodynamic traps which prevent robust self-assembly is a new capability that informs the rational design of particles used to fabricate new materials. The assembly pathway fingerprints generated by BUBBA provide a convenient way to visualize the pathways and their proportion of traps. We demonstrated this utility for CdTe/CdS tetrahedra whose rich morphology arises from complicated long range interactions and local packing constraints. Our work extending BUBBA to systems with continuous degrees of freedom has shown promise for predicting assembly propensity of more complicated building blocks, including systems of DNA-tethered spheres and patchy colloids.

7.2 Outlook

Moving forward there exist many exciting challenges and opportunities for extending the work presented here, application of these techniques to real-life engineering problems, and the development of complementary tools and theory. Much of this thesis focused on using statistical mechanics to lay the foundation for BUBBA and parallelized Monte Carlo simulations, but future work combining these capabilities should allow screening of systems that are orders of magnitude larger in size or complexity. Many of the kernels, including potential energy evaluation, cluster combinatorics, and entropy calculation would benefit from the massively multithreaded parallelism of GPU platforms. Other elements of BUBBA, such as the pruning of cluster libraries, are inherently serial. Developing new software that benchmarks the predictive capabilities of BUBBA on GPUs and on asynchronous frameworks such as Google's MapReduce will inform the next generation of thermodynamic screening tools.

Another BUBBA-specific challenge is its formulation for general systems with continuous degrees of freedom in three dimensions. Current implementations have benefitted from system-specific information that have enabled on-lattice approximations that facilitate shape matching and cluster combinations. It is not yet clear all the challenges and bottlenecks that exist in the general case, where the challenge of distinguishing between "different" clusters is not necessarily well defined. Preliminary advances towards this goal in our spherical codes work are promising, but stop short of the complicated cluster combination routines that were used extensively in our on-lattice work. Continued developments in this area hold promise for the study of particles with non-rigid degrees of freedom and that have long-range interactions.

Partnerships between experimentalists synthesizing and assembling nanoparticles with computational scientists using BUBBA, MC, and MD should be emphasized for solving next-generation manufacturing problems. Currently the ecosystem of industrial-computational interaction is limited and there are few example successes to show the utility of computation in materials manufacturing. New emphasis on specific product development will help focus the computational capabilities on the relevant physics, and help guide experimentalists towards more efficient use of their resources. It seems as though we are on the verge of designing particles and agglomerates that could be used as advanced MRI image contrasts or gene delivery vectors. Focusing on improving these technologies is at least as important as developing the computational tools which enable it.

Other interesting questions that were opened during this work revolve around system complexity. It seems intuitive that a particle composed of many subunits is more complicated than a particle composed of few. Similarly, a particle with multiple subunit types seems more complex that a geometrically identical particle composed of one subunit. Do there exist measures for these characteristics that provide guidance for how easy or difficult it is for them to self-assemble? Can mixtures of building blocks be used to enhance the assembly of complicated patterns? It seems that there is no lack of work for the computational chemical engineer and the advances that should be enabled by improved computing power, chemical synthesis, and numerical tools will be exciting to follow into the future.

APPENDICES

APPENDIX A

Tetromino pathway fingerprints

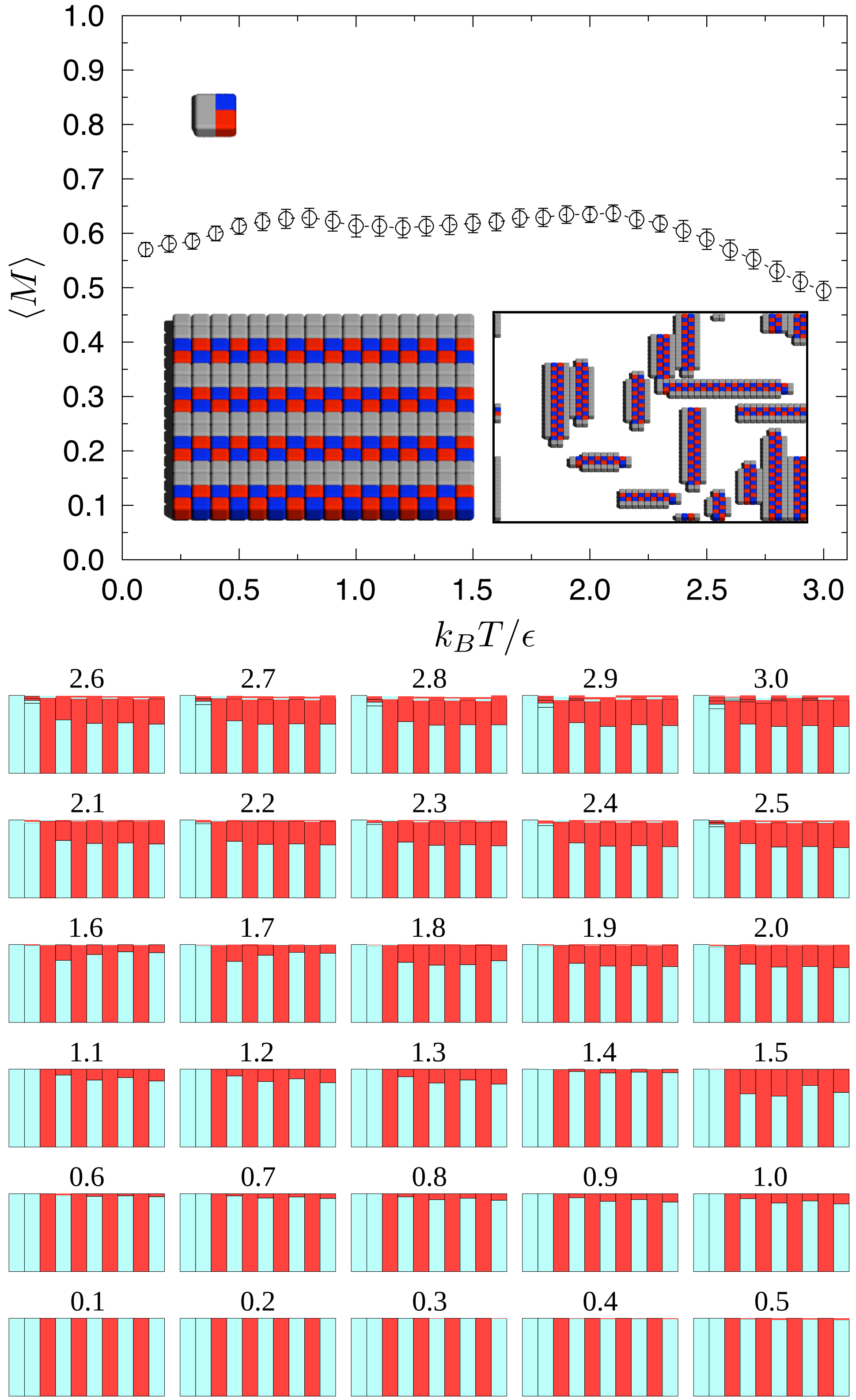


Figure A.1: Patchy particle 2: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

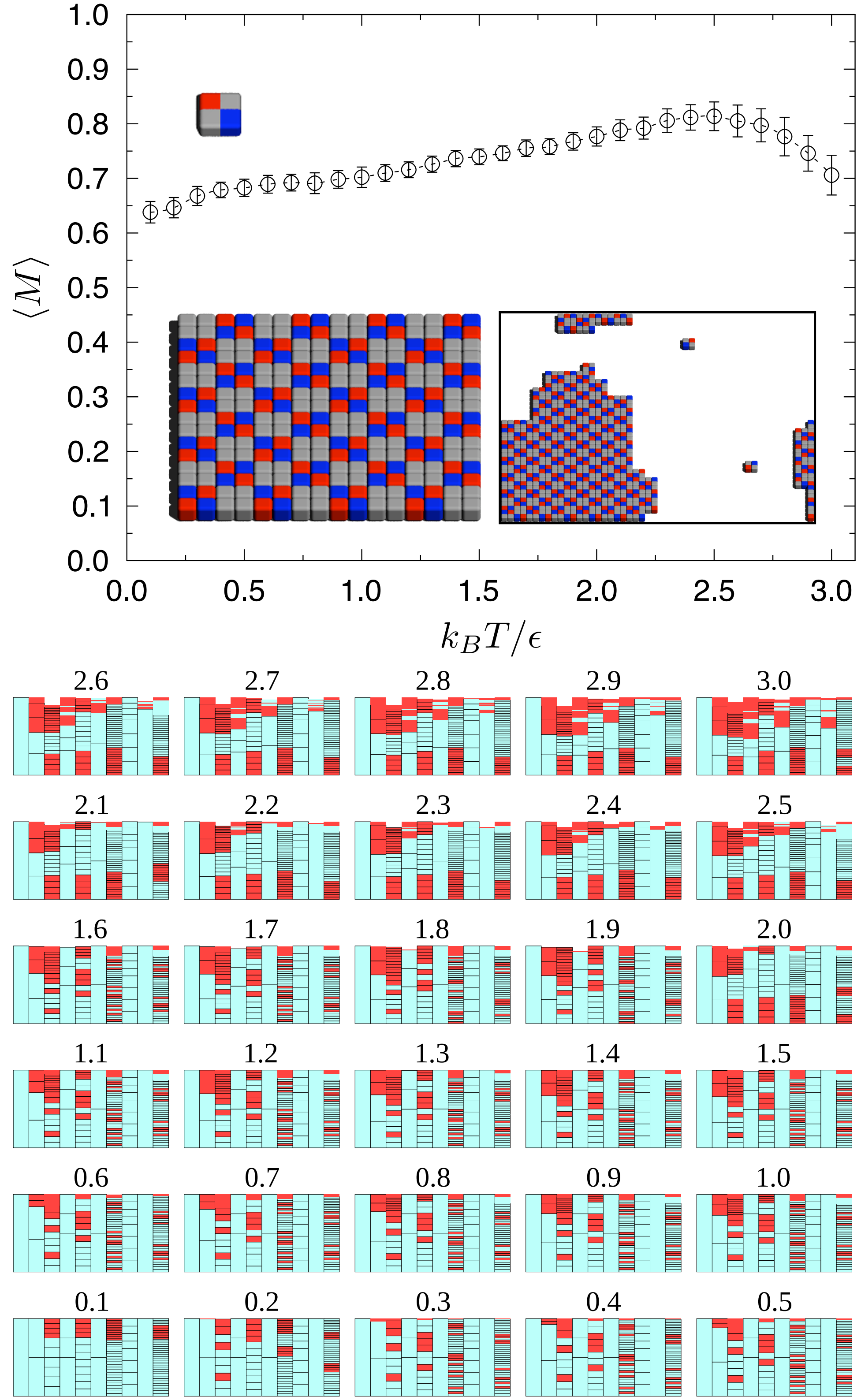


Figure A.2: Patchy particle 3: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

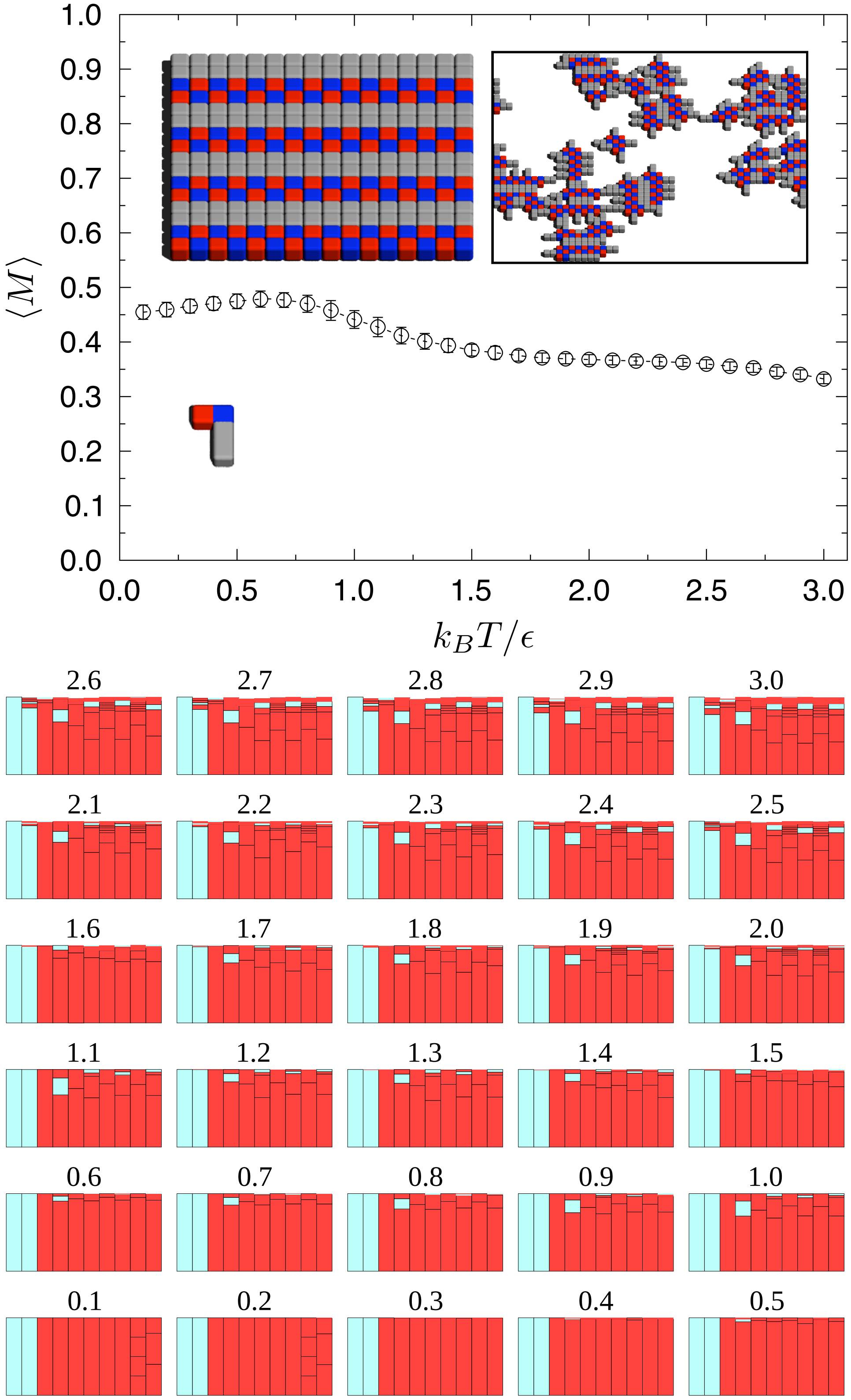


Figure A.3: Patchy particle 5: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

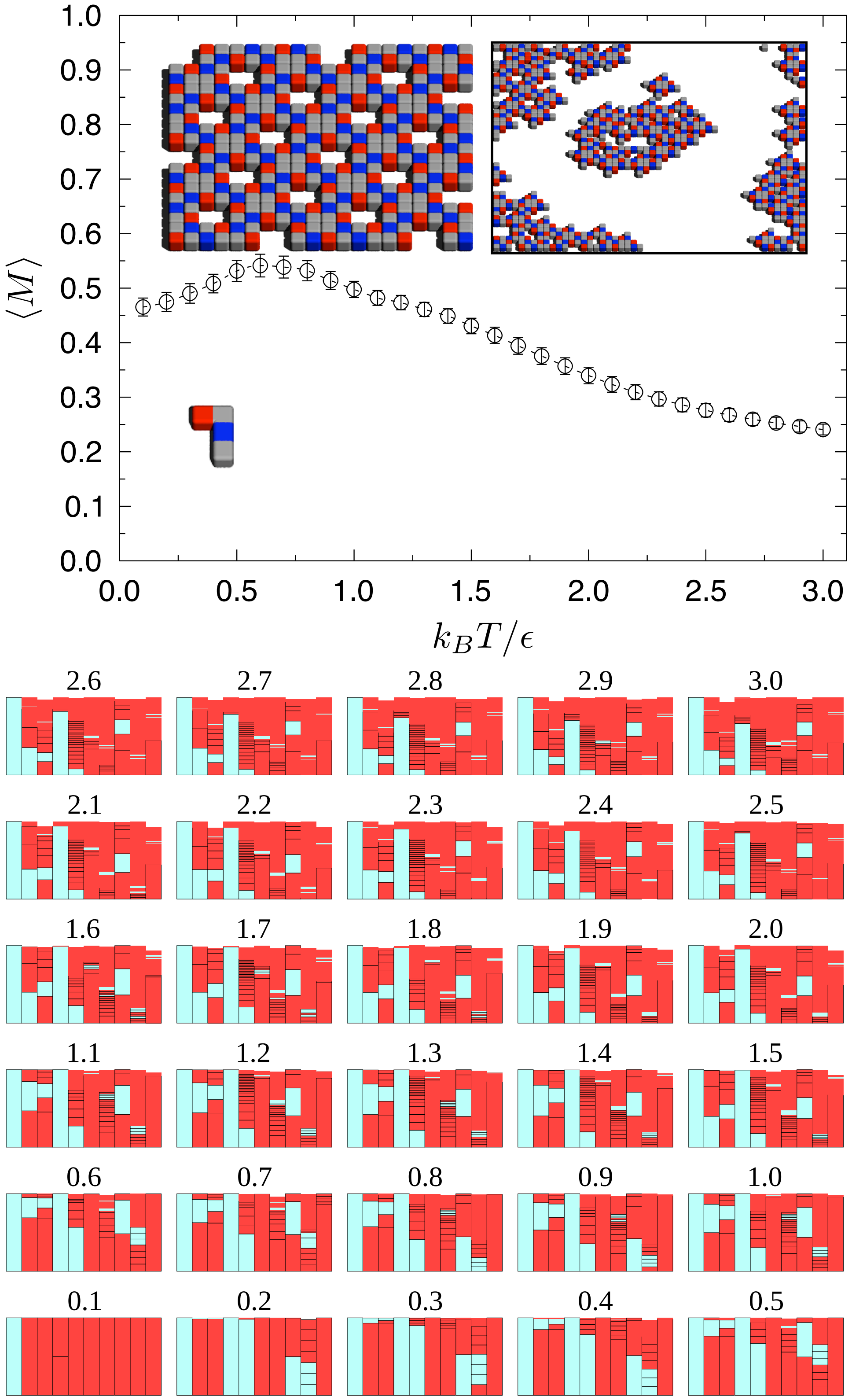


Figure A.4: Patchy particle 6: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

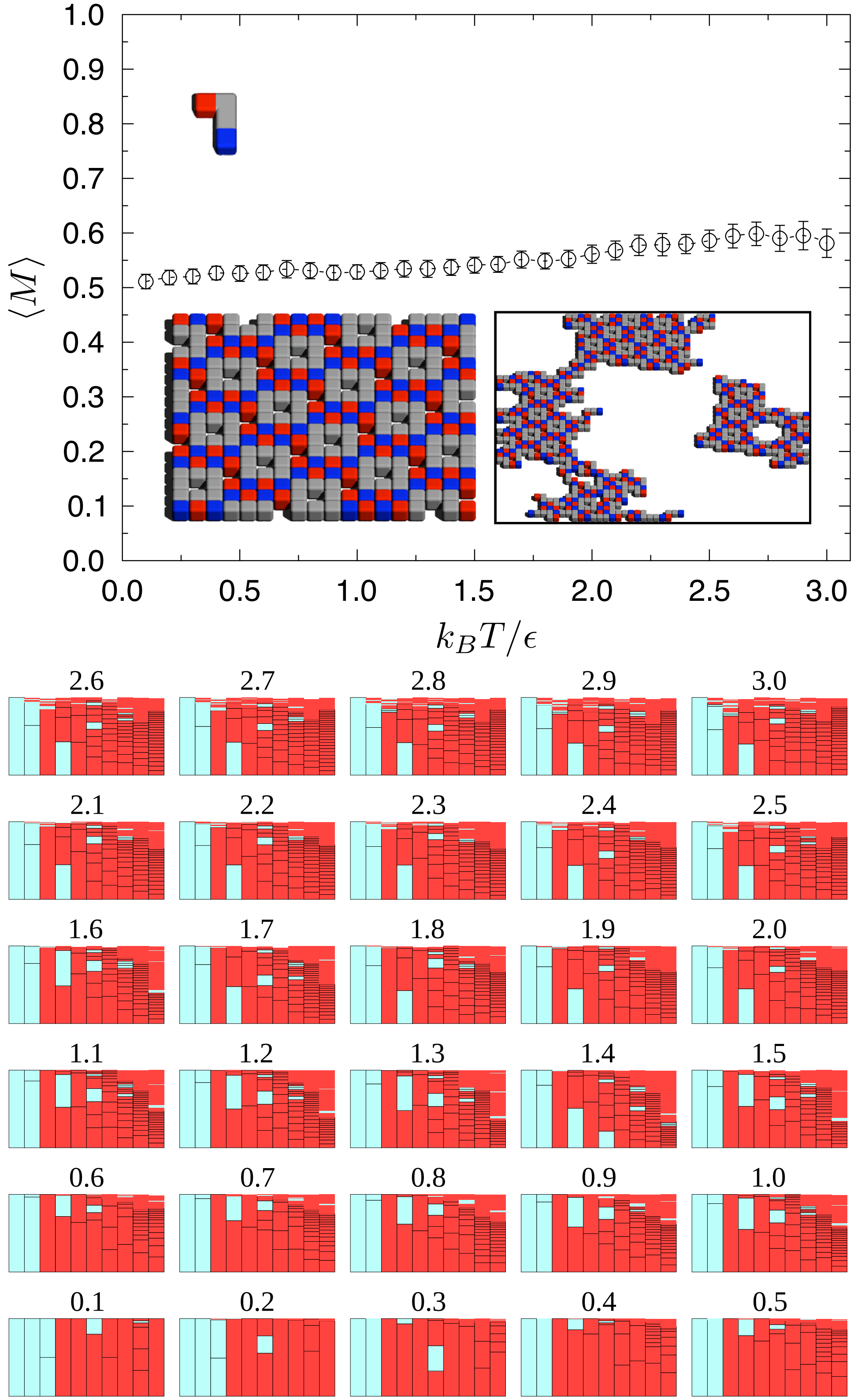


Figure A.5: Patchy particle 7: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

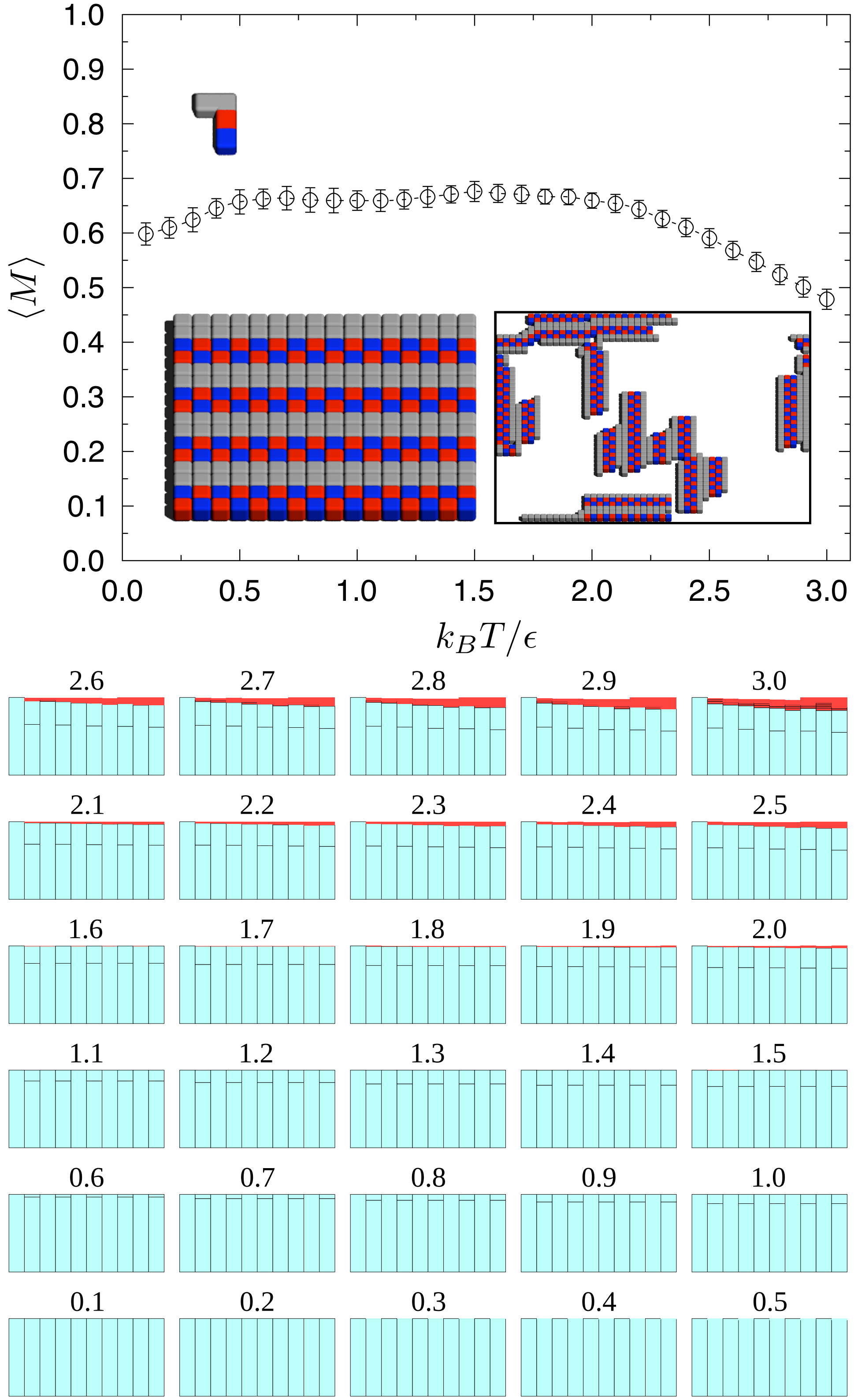


Figure A.6: Patchy particle 8: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

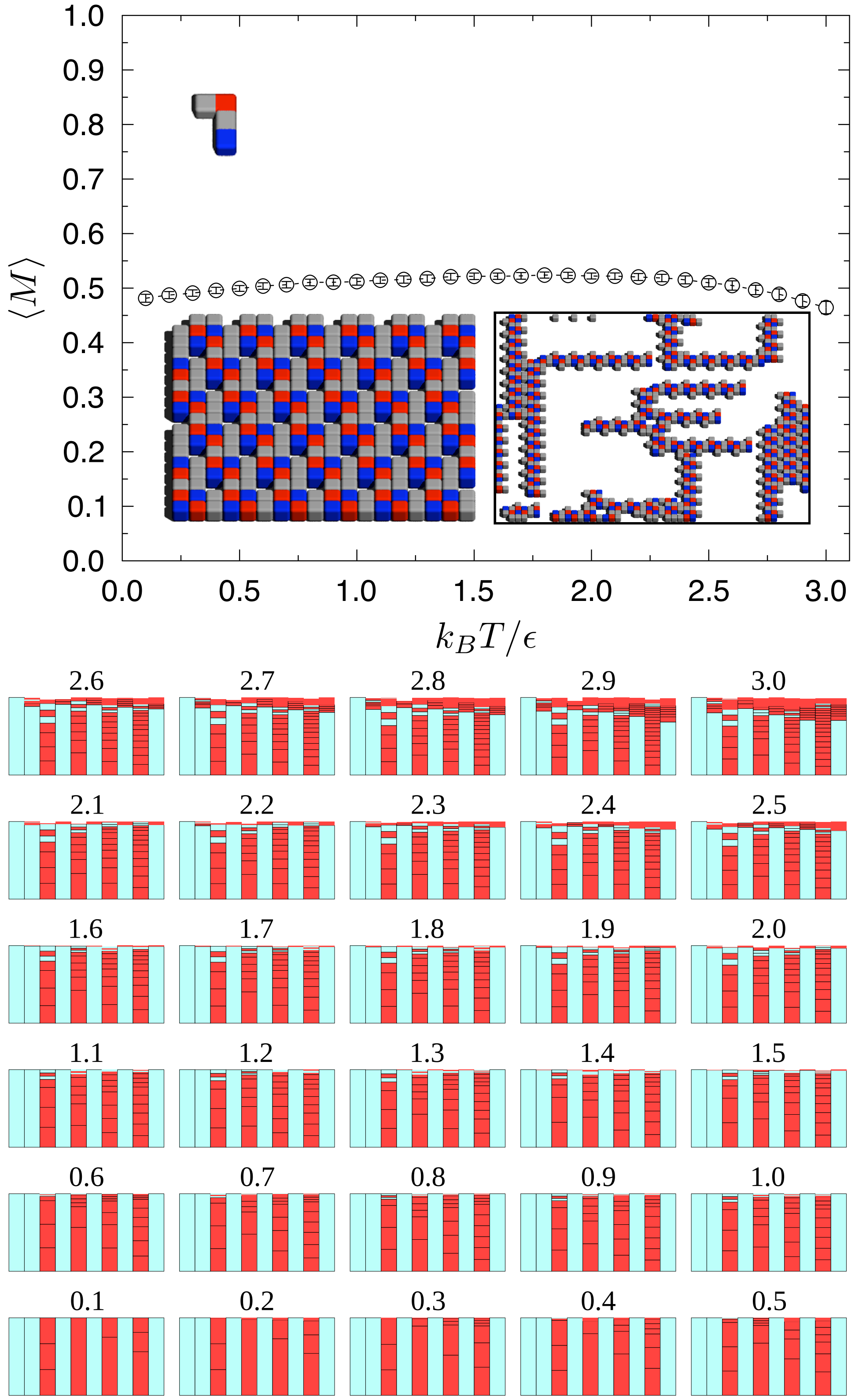


Figure A.7: Patchy particle 9: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

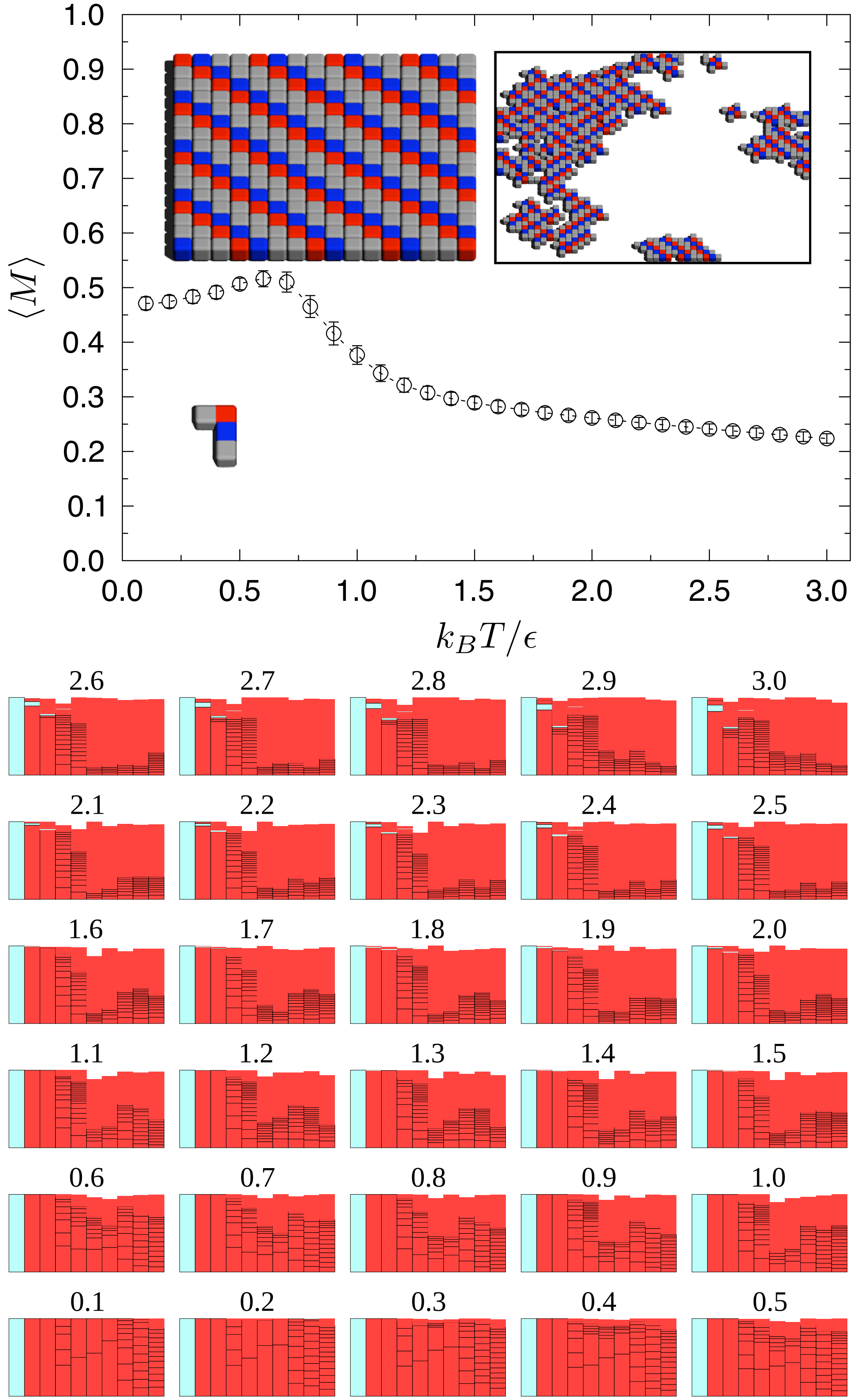


Figure A.8: Patchy particle 10: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

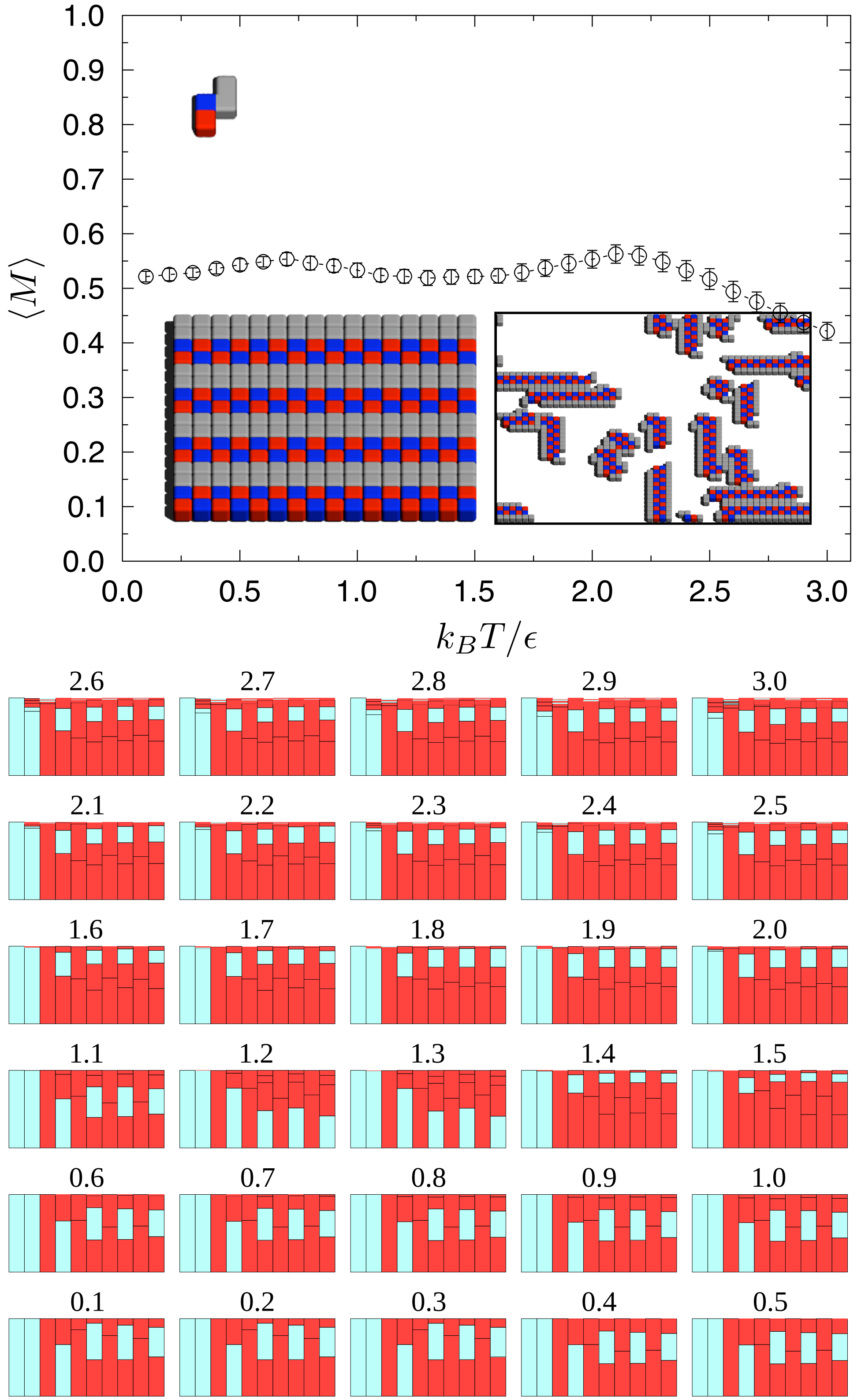


Figure A.9: Patchy particle 12: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

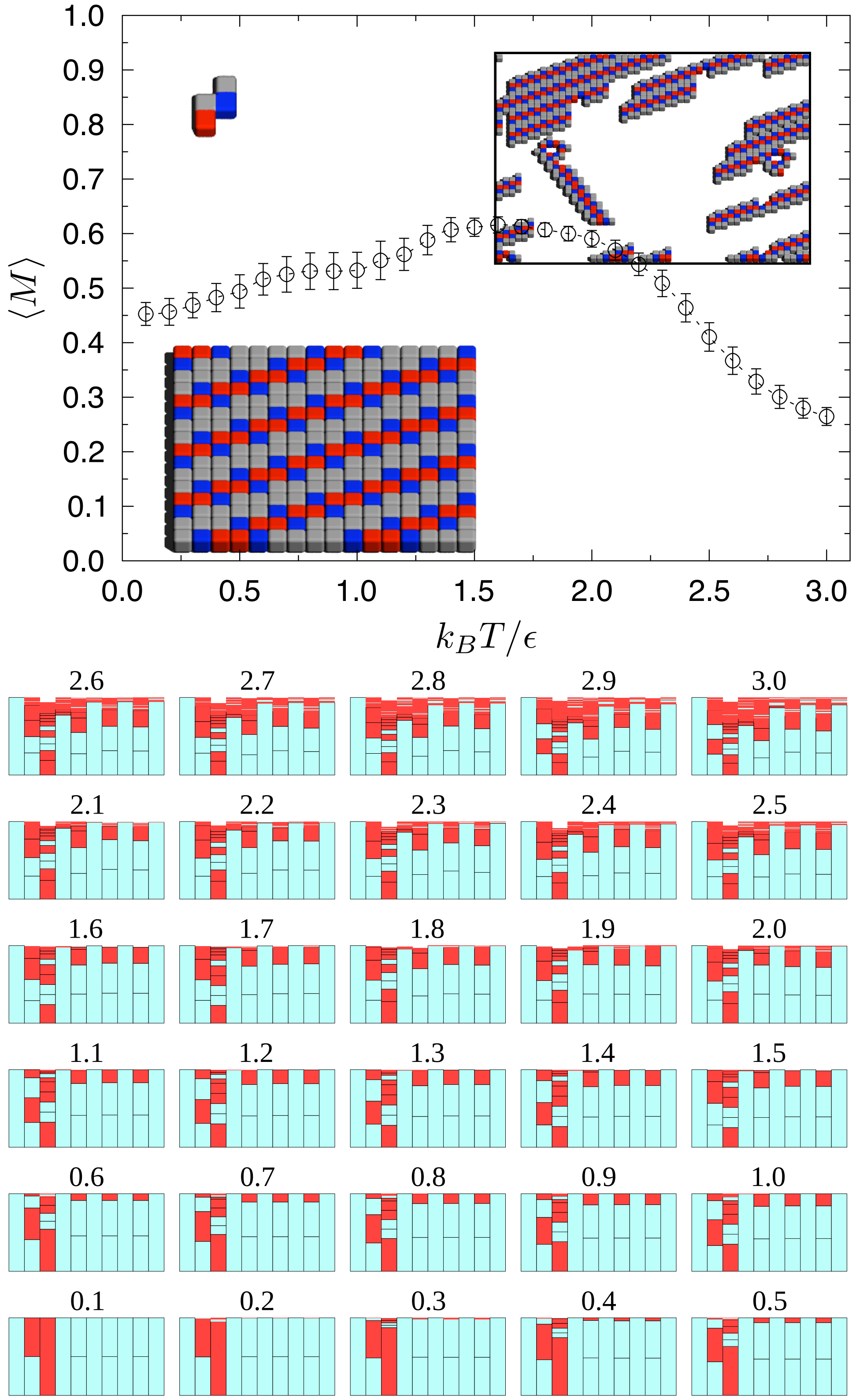


Figure A.10: Patchy particle 13: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

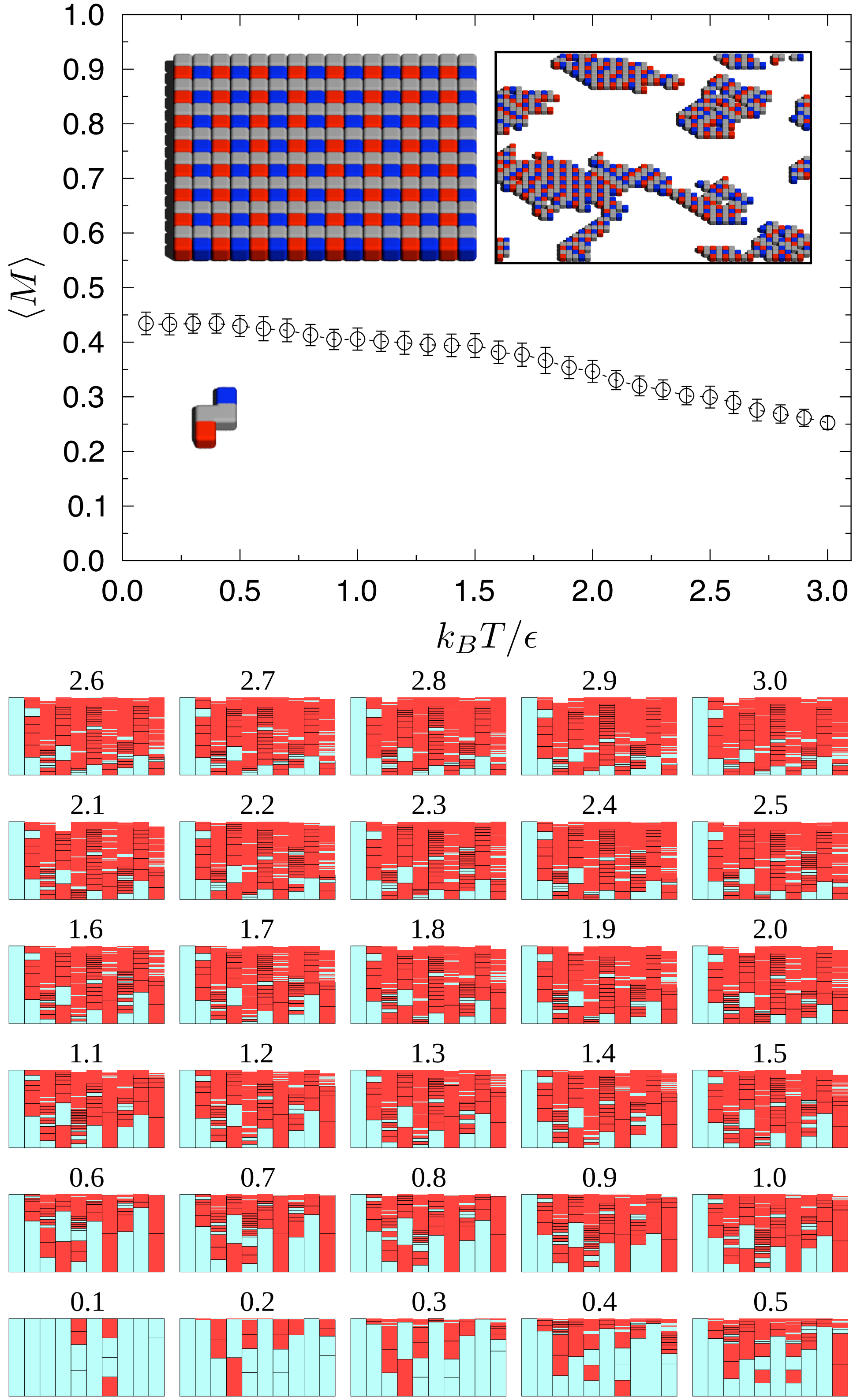


Figure A.11: Patchy particle 14: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

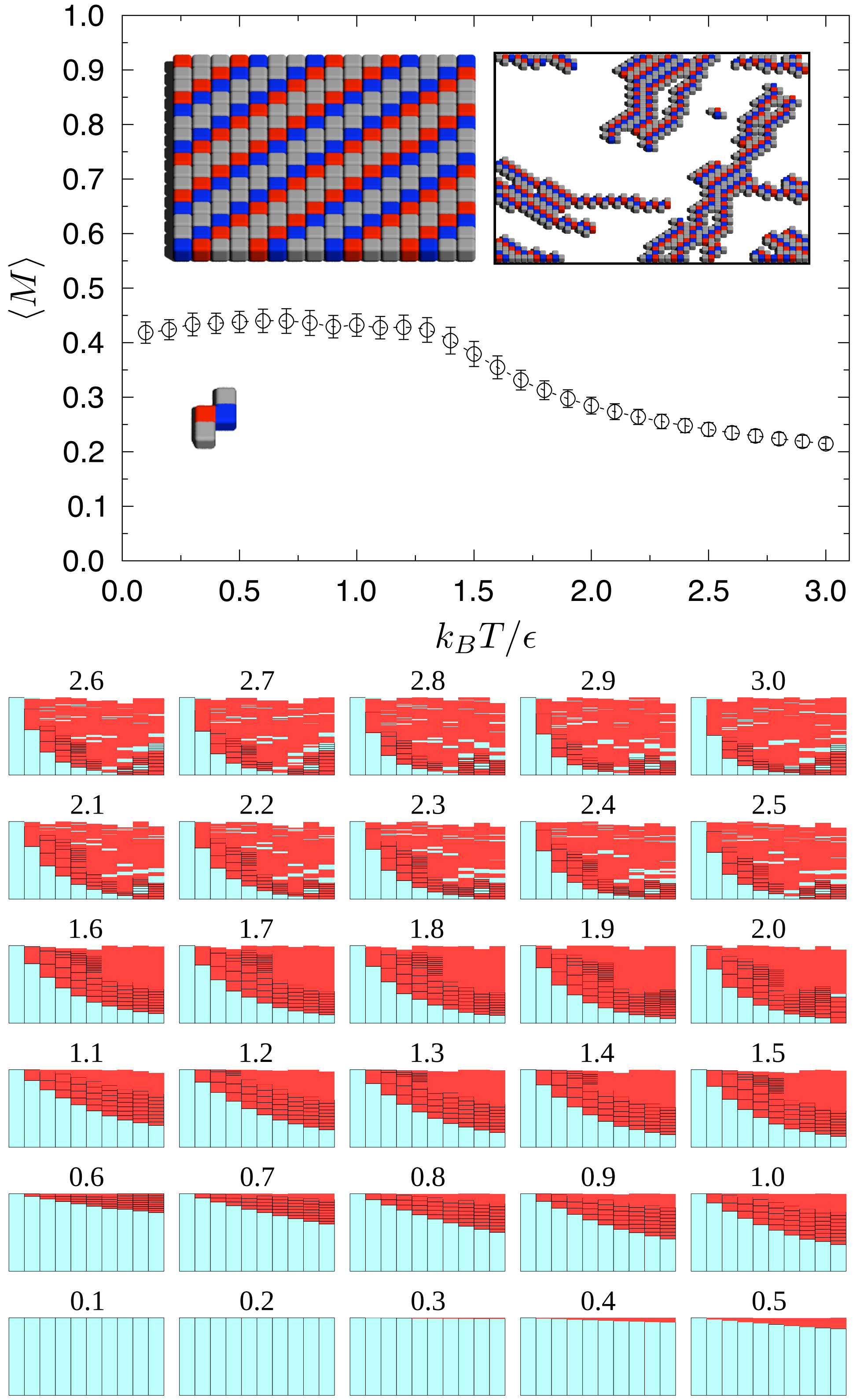


Figure A.12: Patchy particle 15: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

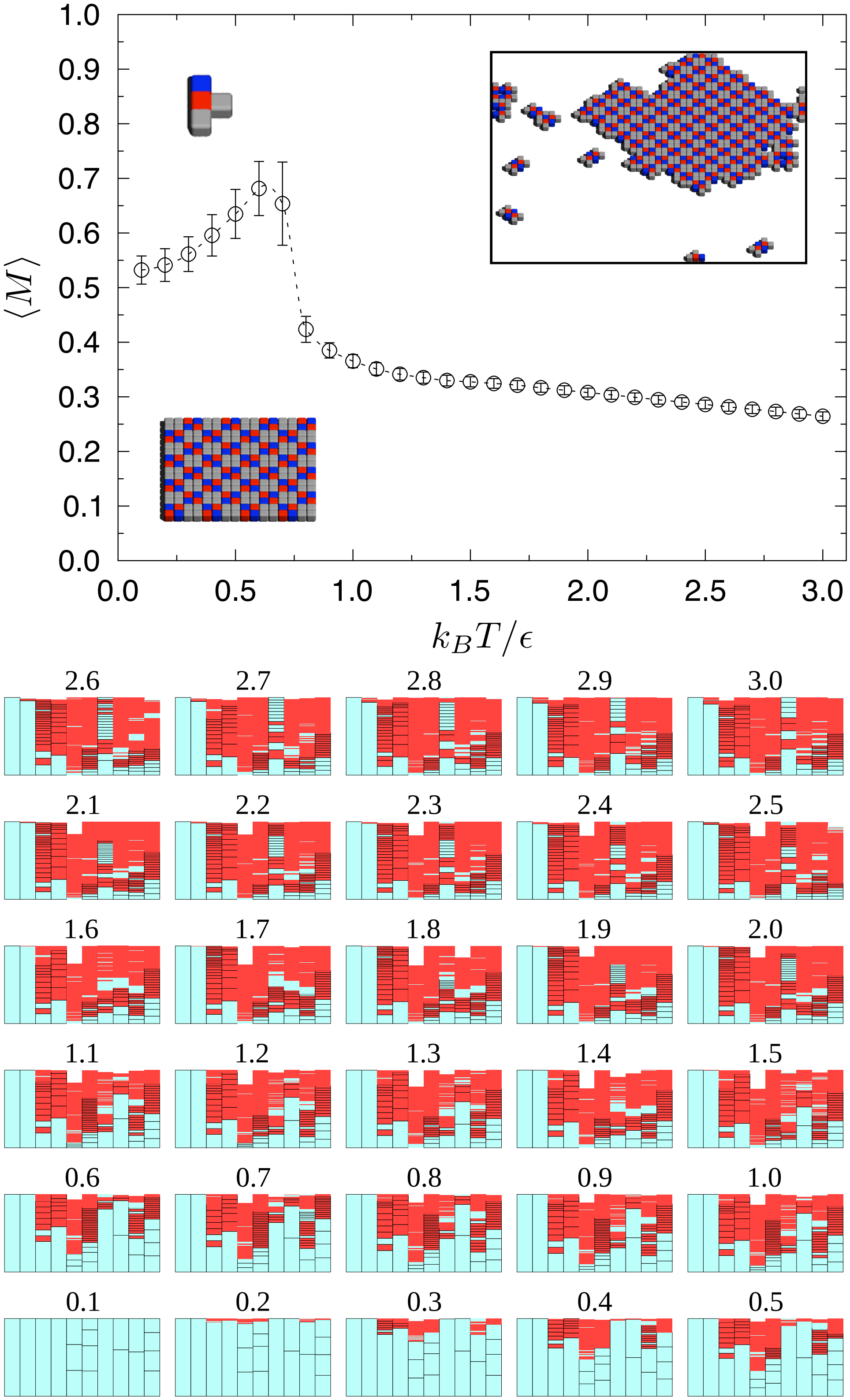


Figure A.13: Patchy particle 17: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

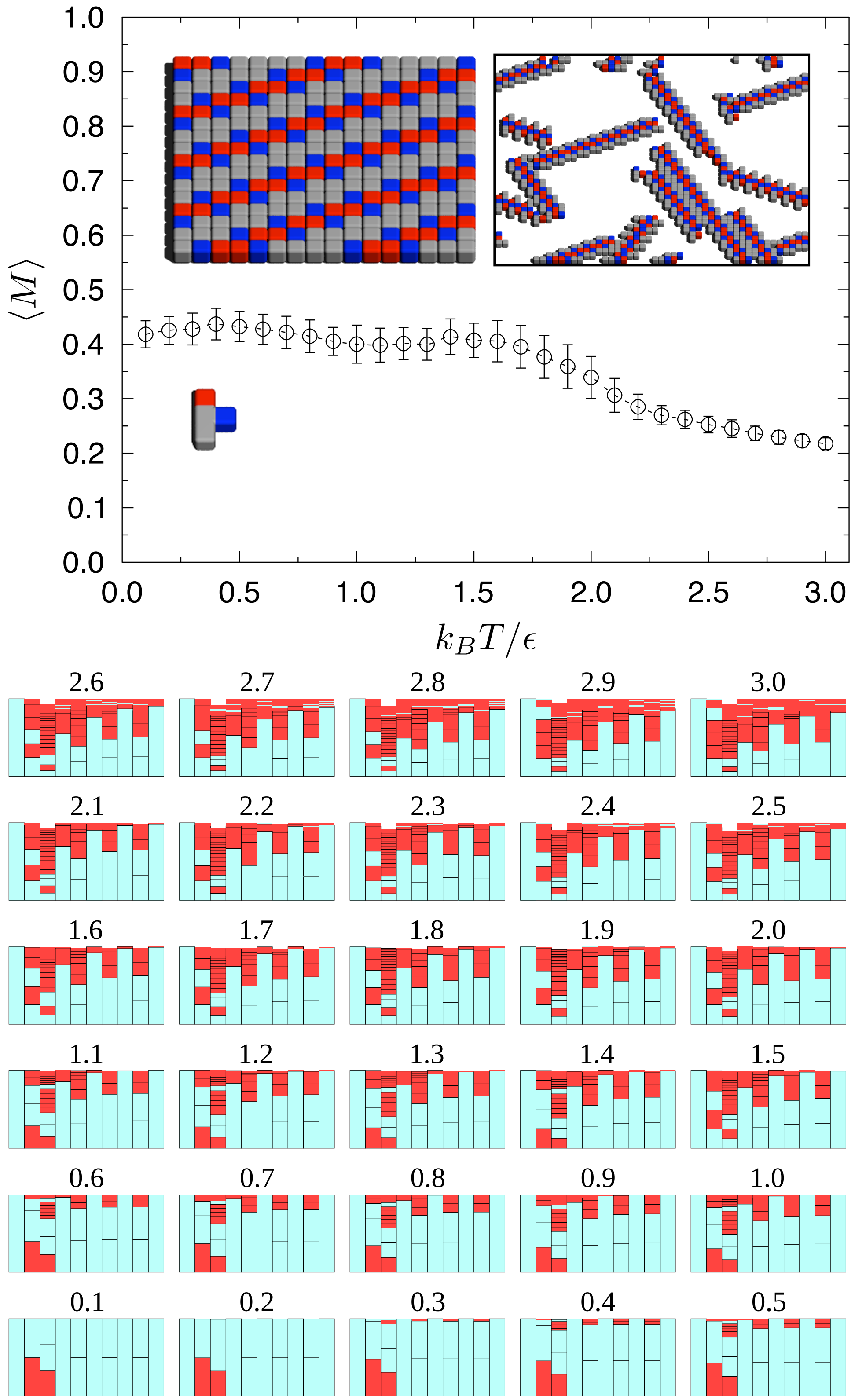


Figure A.14: Patchy particle 18: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

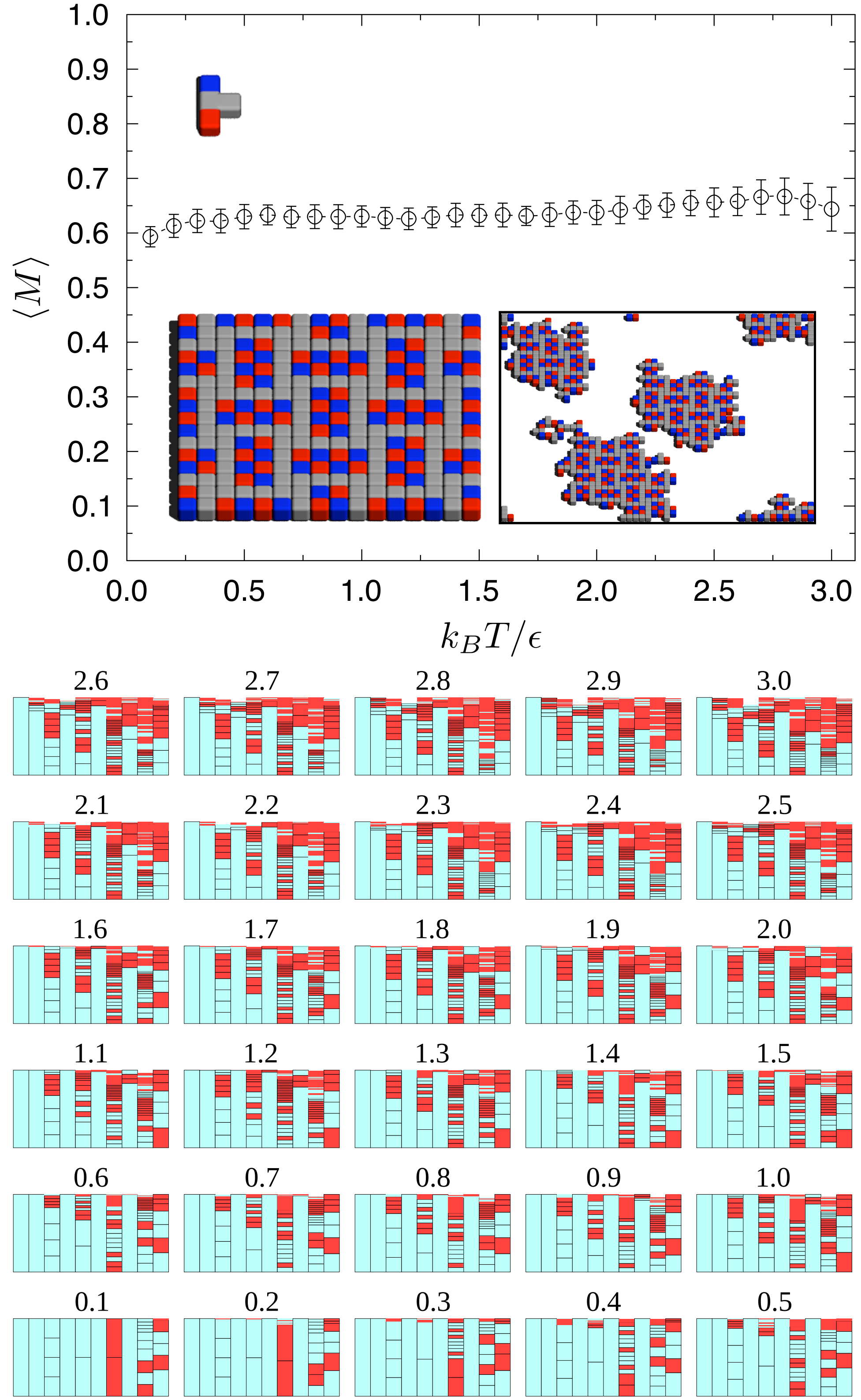


Figure A.15: Patchy particle 19: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

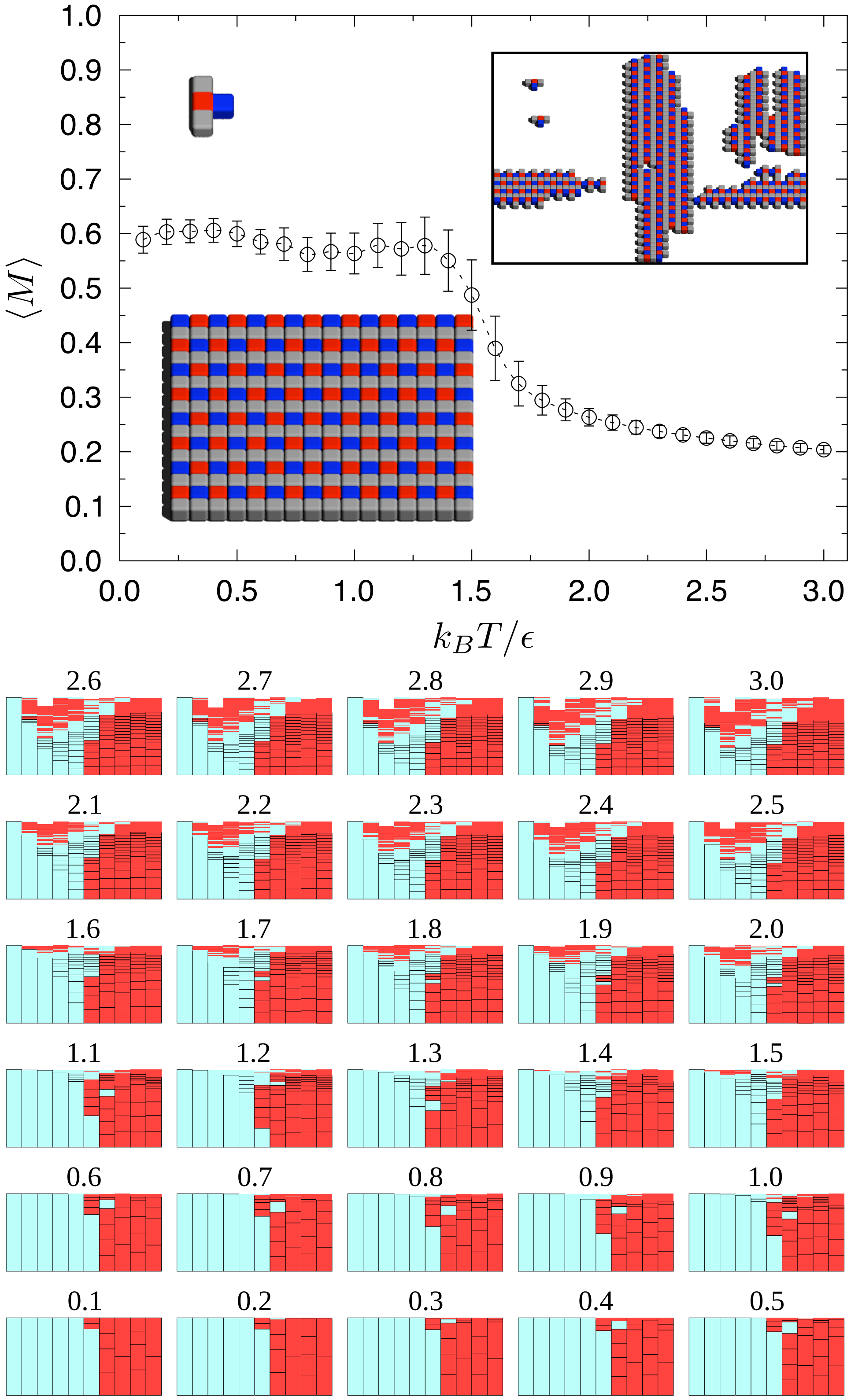


Figure A.16: Patchy particle 20: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

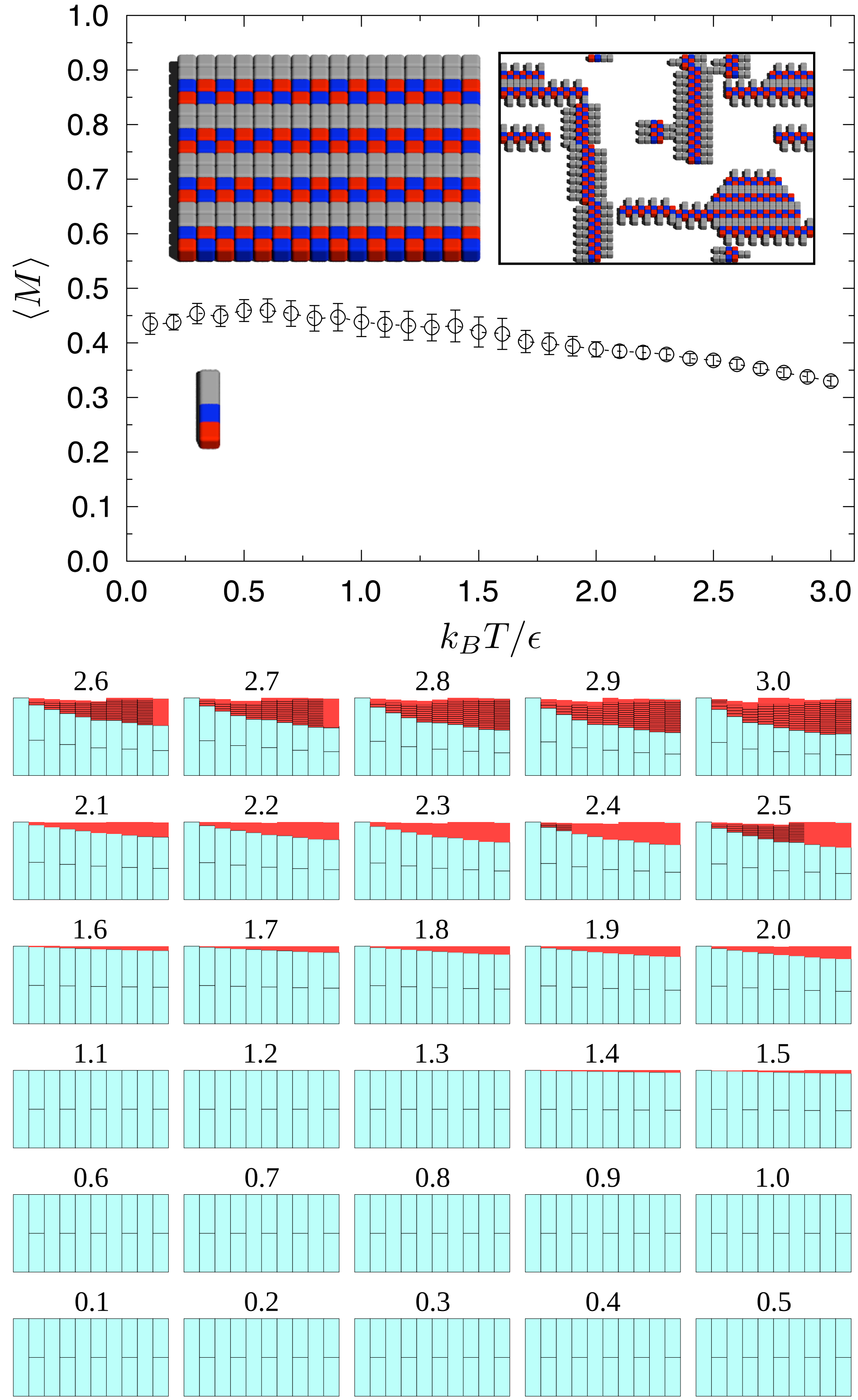


Figure A.17: Patchy particle 22: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

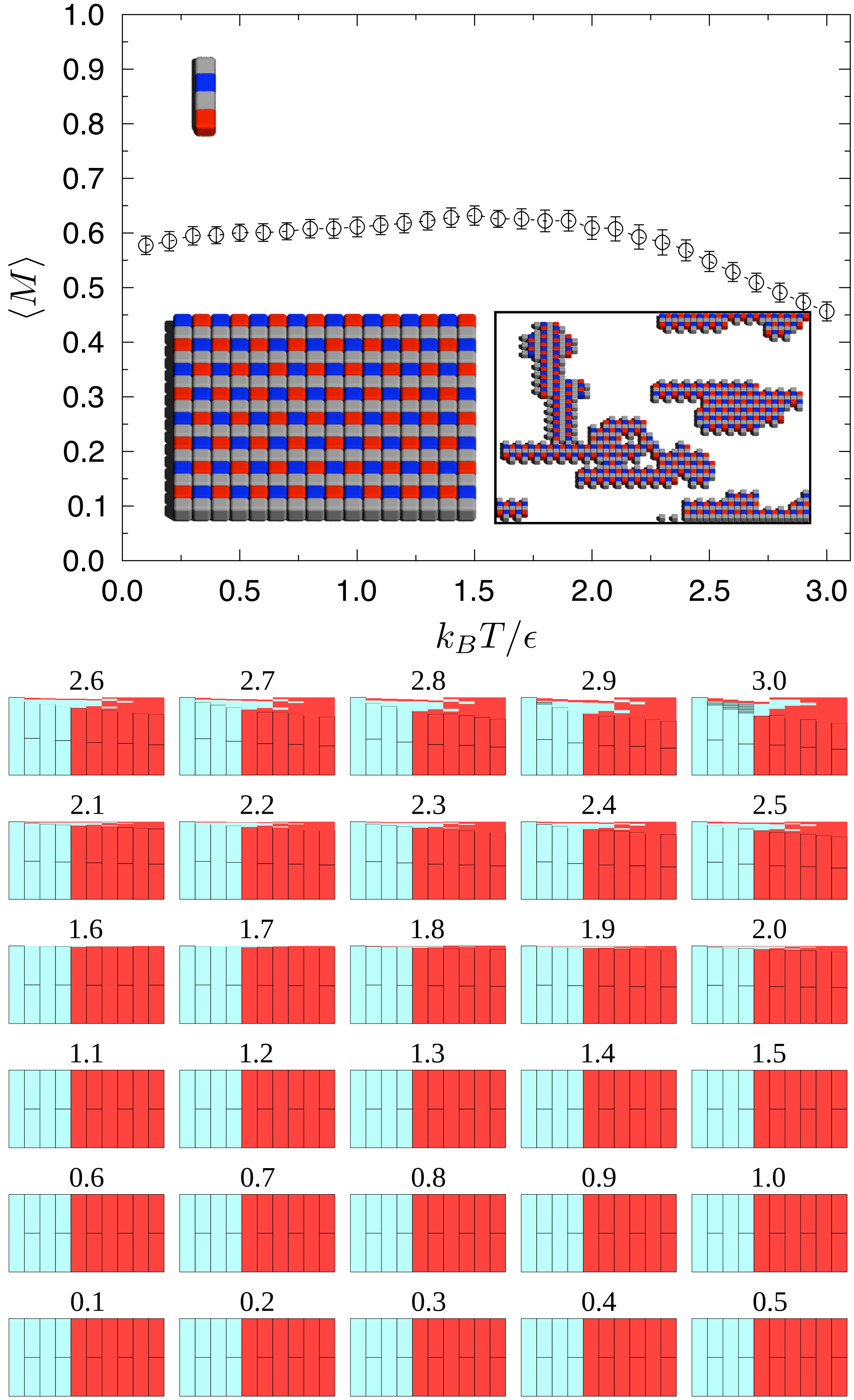


Figure A.18: Patchy particle 23: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

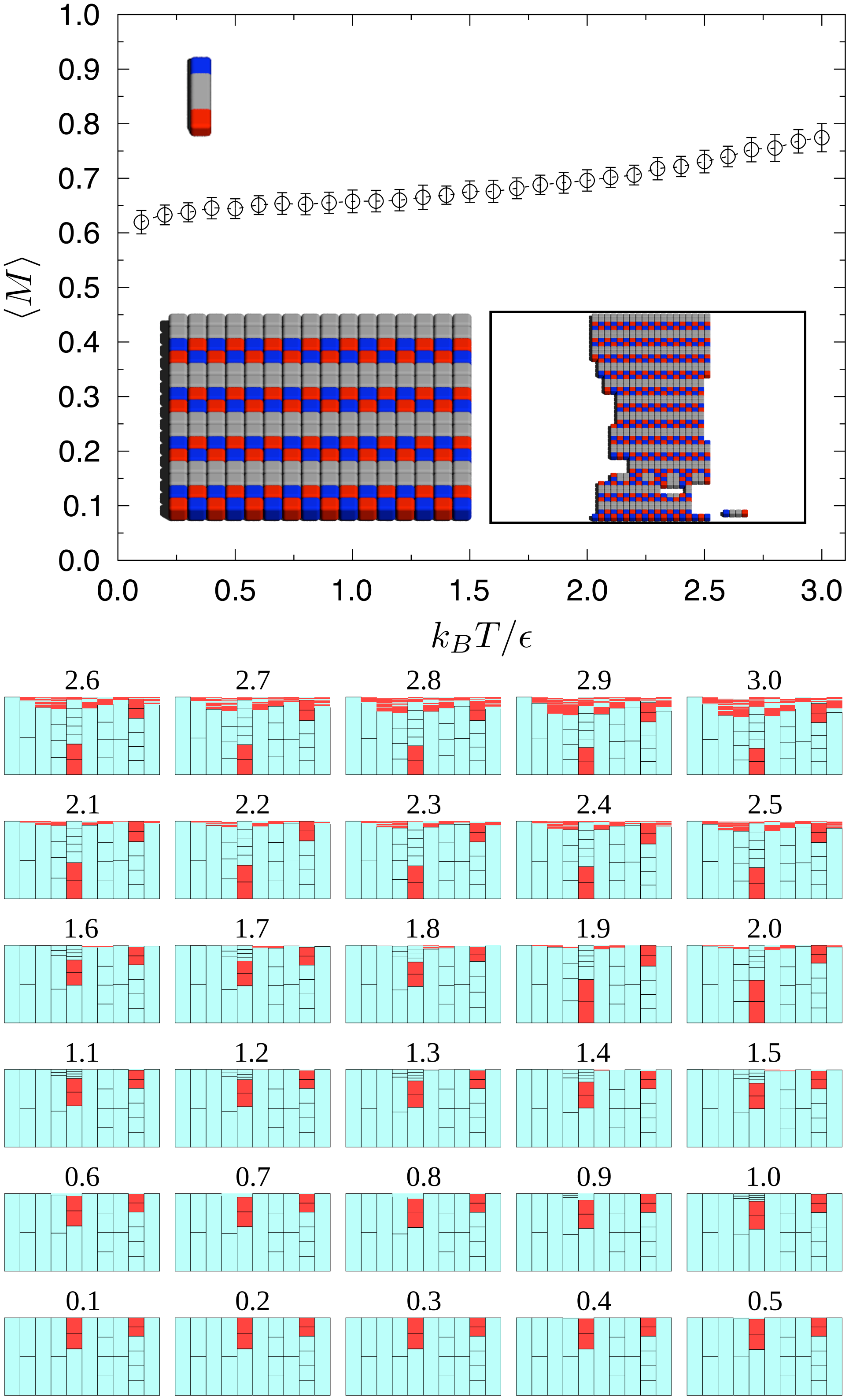


Figure A.19: Patchy particle 24: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

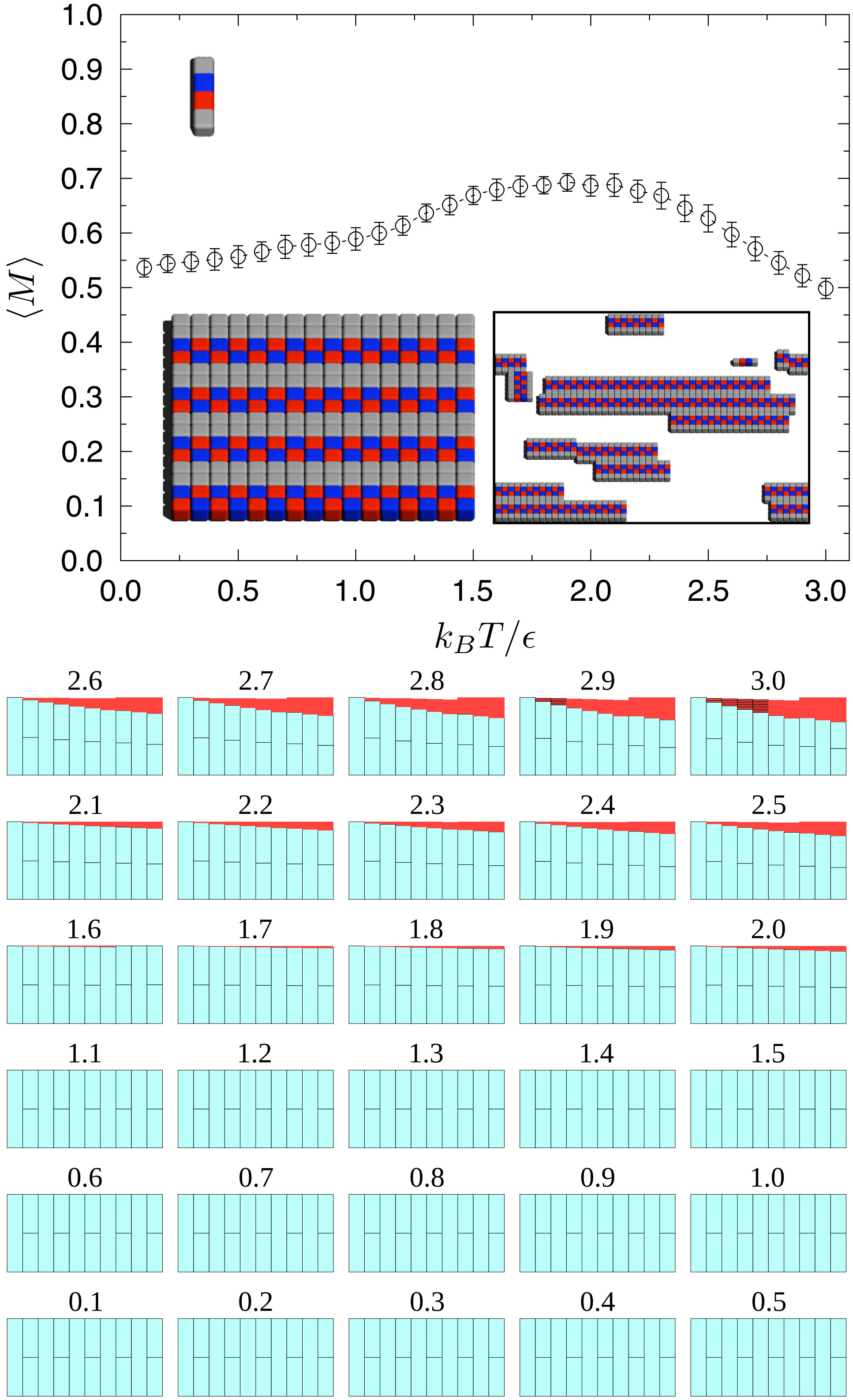


Figure A.20: Patchy particle 25: (a) Motif match profile with building block, motif, and best-match snapshot inset. (b) fingerprint landscape

APPENDIX B

Off-lattice BUBBA clusters

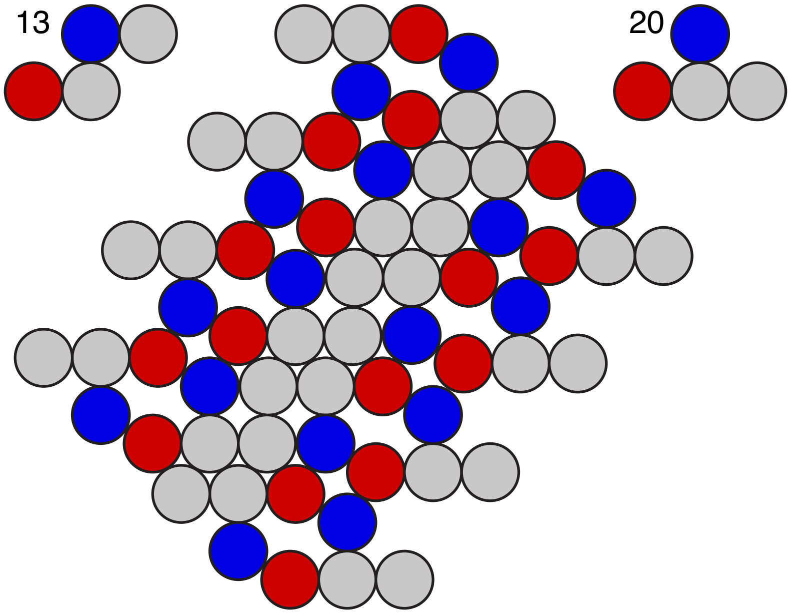


Figure B.1: Slanted stripes

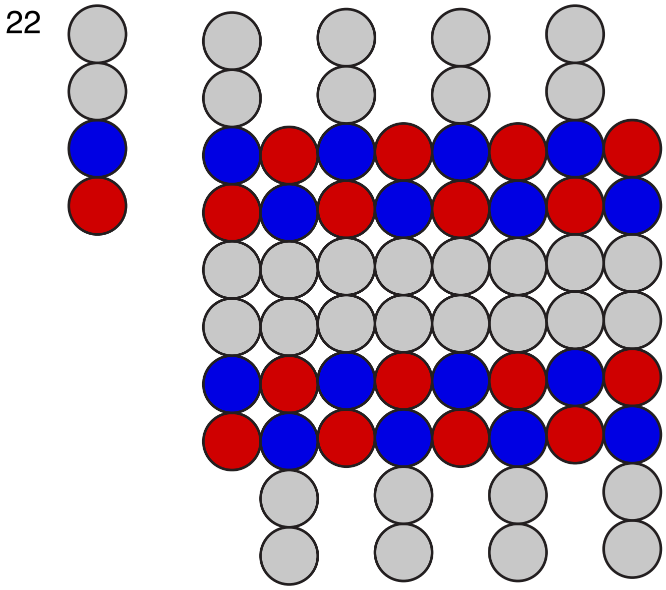


Figure B.2: Wide stripes

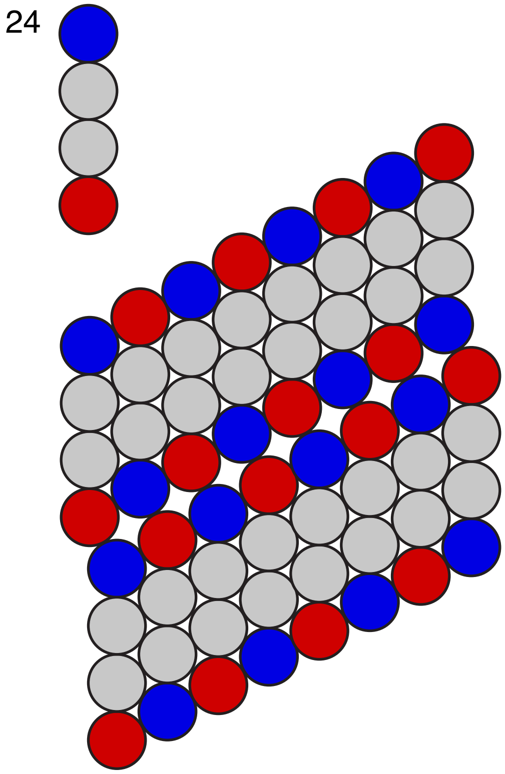


Figure B.3: Slanted interdigitated stripes

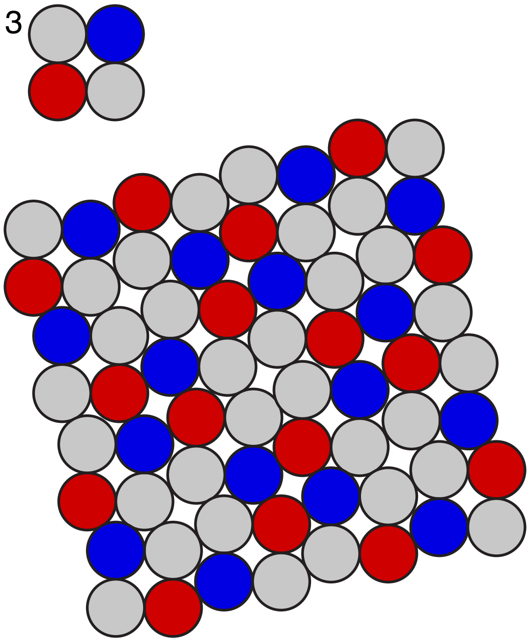


Figure B.4: Interdigitated checkerboard

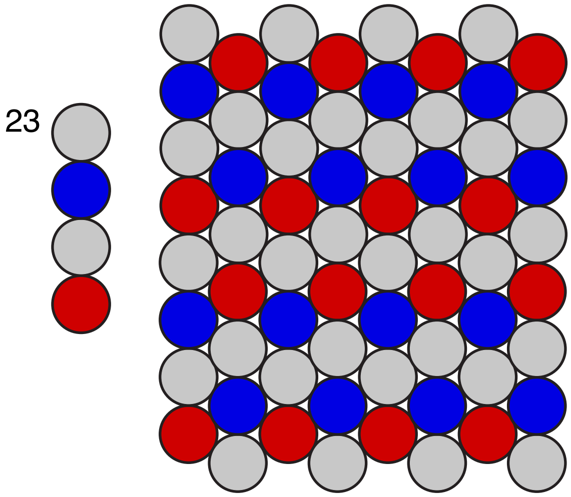


Figure B.5: Close-packed stripes

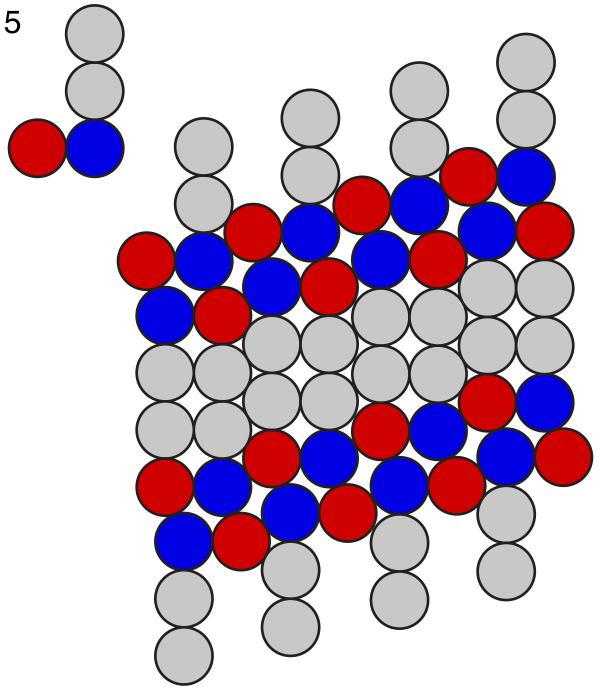


Figure B.6: Two copies of building block 5 make up the unit cell of the shifted stripes in this packing, though the best energy cluster of two building blocks is not the unit cell.

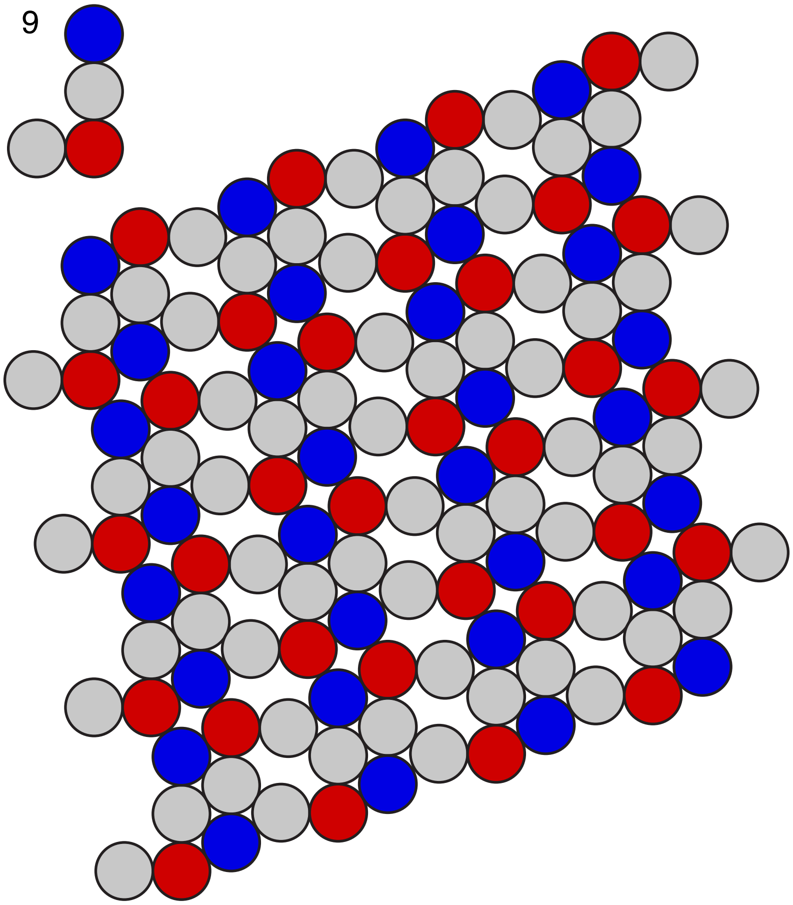


Figure B.7: Two copies of building block 9 make up the unit cell of this twofold symmetric packing.

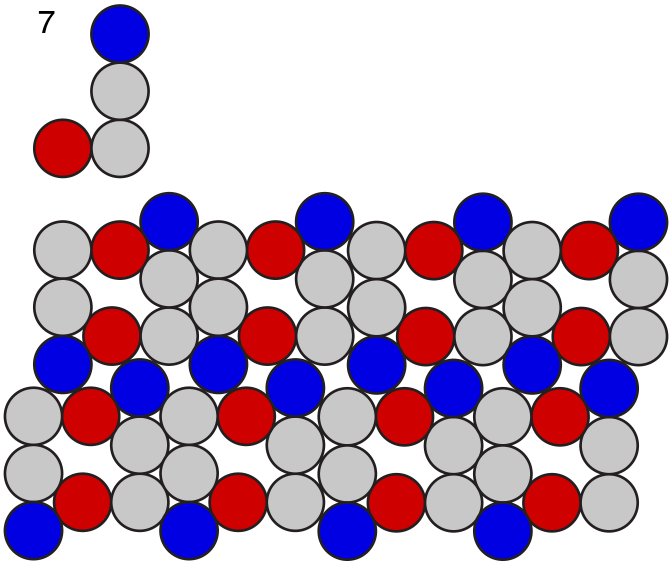


Figure B.8: Two copies of building block 7 make up the unit cell of this twofold symmetric packing.

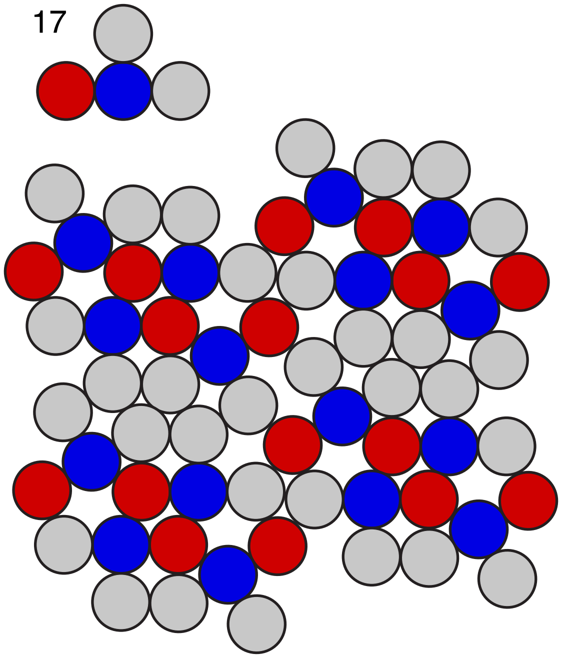


Figure B.9: This twofold symmetric packing of building block 17 has four building blocks in its unit cell.

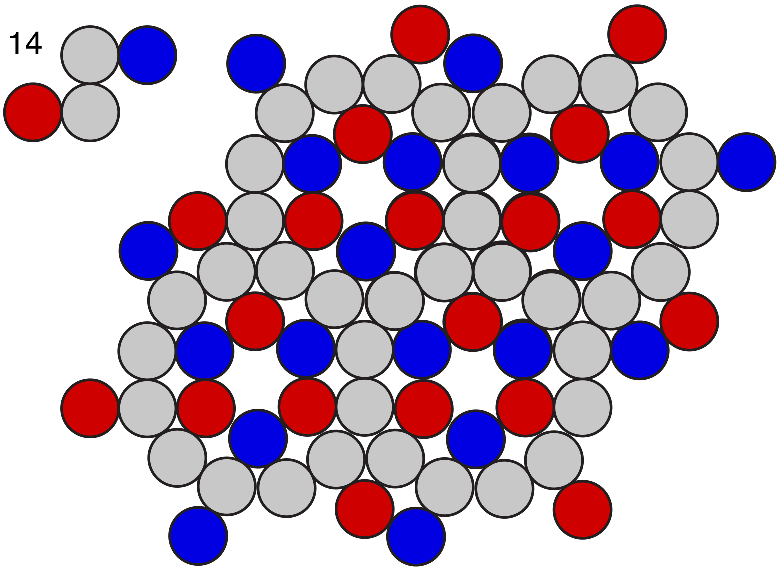


Figure B.10: Six copies of building block 14 make up one ring, which tiles space hexagonally.

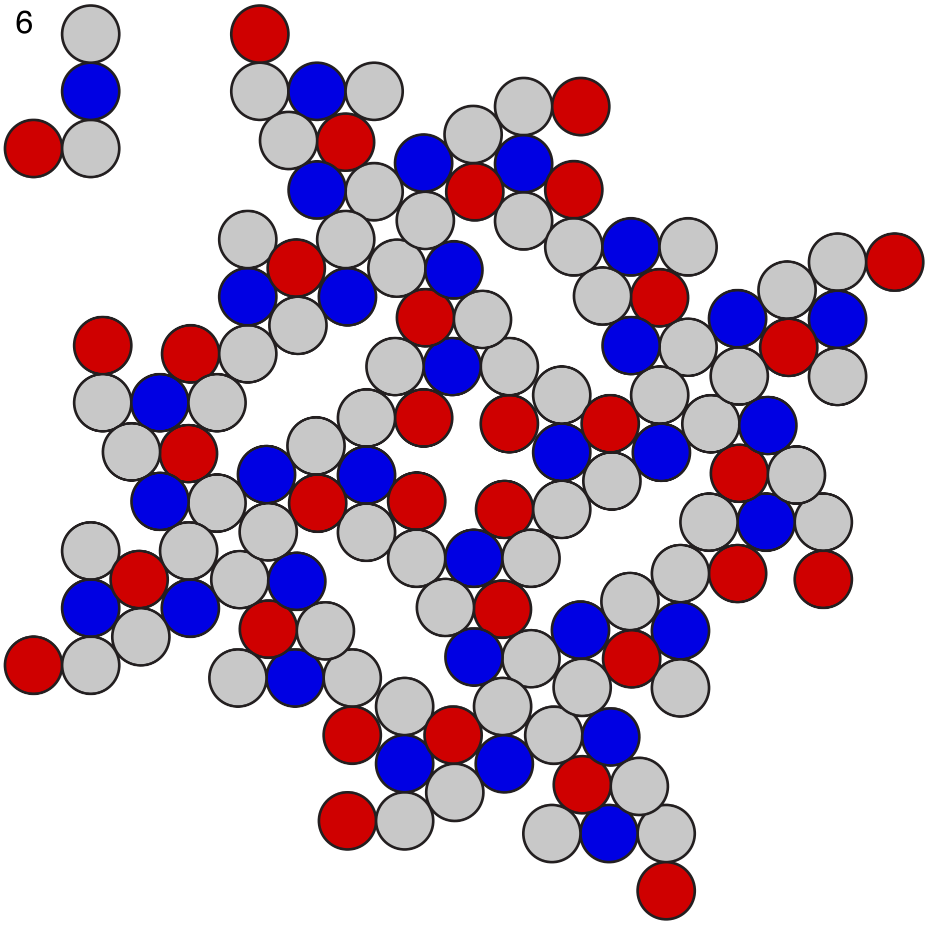


Figure B.11: Fourfold symmetric packing with eight building blocks in the unit cell.

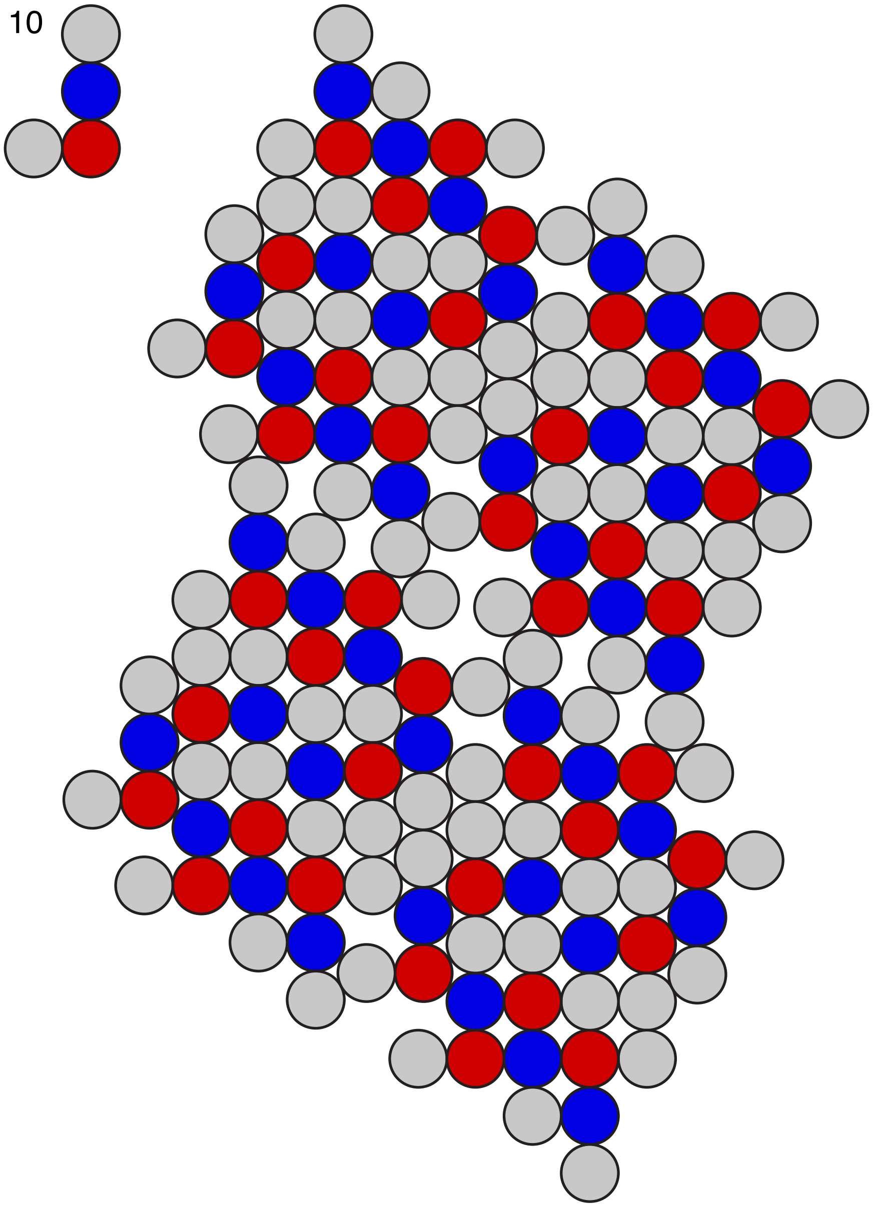


Figure B.12: The complicated unit cell of this packing is composed of ten building blocks.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] GM Whitesides and B. Grzybowski. Self-assembly at all scales. *Science*, 295(5564):2418–2421, 2002.

[2] M Tirrell. Modular materials by self-assembly. *Aiche Journal*, 51(9):2386–2390, 2005.

[3] Sharon C. Glotzer and Michael J. Solomon. Anisotropy of building blocks and their assembly into complex structures. *Nature Materials*, 6(8):557–562, 2007.

[4] Frank S. Bates and Glenn H. Fredrickson. Block copolymer thermodynamics: Theory and experiment. *Annual Review of Physical Chemistry*, 41(1):525–557, 1990.

[5] Rongchao Jin, YunWei Cao, Chad A. Mirkin, K. L. Kelly, George C. Schatz, and J. G. Zheng. Photoinduced conversion of silver nanospheres to nanoprisms. *Science*, 294(5548):1901–1903, 2001.

[6] Nikhil R. Jana, Latha Gearheart, and Catherine J. Murphy. Wet chemical synthesis of high aspect ratio cylindrical gold nanorods. *The Journal of Physical Chemistry B*, 105(19):4065–4067, 2001.

[7] Temer S. Ahmadi, Zhong L. Wang, Travis C. Green, Arnim Henglein, and Mostafa A. El-Sayed. Shape-controlled synthesis of colloidal platinum nanoparticles. *Science*, 272(5270):1924–1925, 1996.

[8] Sihai Chen, Zhong Lin Wang, John Ballato, Stephen H. Foulger, and David L. Carroll. Monopod, bipod, tripod, and tetrapod gold nanocrystals. *Journal of the American Chemical Society*, 125(52):16186–16187, 2003.

[9] Sang-Min Lee, Young-wook Jun, Sung-Nam Cho, and Jinwoo Cheon. Single-crystalline star-shaped nanocrystals and their evolution: programming the geometry of nano-building blocks. *Journal of the American Chemical Society*, 124(38):11244–11245, 2002. PMID: 12236719.

[10] Yugang Sun and Younan Xia. Shape-controlled synthesis of gold and silver nanoparticles. *Science*, 298(5601):2176–2179, 2002.

[11] Z. Tang, Y. Wang, S. Shanbhag, M. Giersig, and N.A. Kotov. Spontaneous transformation of cdte nanoparticles into angled te nanocrystals: From particles

and rods to checkmarks, x-marks, and other unusual shapes. *Journal of the American Chemical Society*, 128(20):6730–6736, 2006.

[12] Zhiyong Tang, Zhenli Zhang, Ying Wang, Sharon C. Glotzer, and Nicholas A. Kotov. Self-Assembly of CdTe Nanocrystals into Free-Floating Sheets. *Science*, 314(5797):274–278, 2006.

[13] Heung-Shik Park, Ashish Agarwal, Nicholas A. Kotov, and Oleg D. Lavrentovich. Controllable side-by-side and end-to-end assembly of au nanorods by lyotropic chromonic materials. *Langmuir*, 24(24):13833–13837, 2008.

[14] Sudhanshu Srivastava, Aaron Santos, Kevin Critchley, Ki-Sub Kim, Paul Podsiadlo, Kai Sun, Jaebeom Lee, Chuanlai Xu, G Daniel Lilly, Sharon C Glotzer, and Nicholas A Kotov. Light-controlled self-assembly of semiconductor nanoparticles into twisted ribbons. *Science*, 327(5971):1355–1359, Mar 2010.

[15] Yunsheng Xia, Trung Dac Nguyen, Ming Yang, Byeongdu Lee, Aaron Santos, Paul Podsiadlo, Xhiyong Tang, Sharon C. Glotzer, and Nicholas A. Kotov. Self-assembly of self-limiting monodisperse supraparticles from polydisperse nanoparticles. *Nature Nanotechnology*, 6(9):580–587, 2011.

[16] Jason P. Rolland, Benjamin W. Maynor, Larken E. Euliss, Ansley E. Exner, Ginger M. Denison, and Joseph M. DeSimone. Direct fabrication and harvesting of monodisperse, shape-specific nanobiomaterials. *Journal of the American Chemical Society*, 127(28):10096–10100, 2005.

[17] Dhananjay Dendukuri, Daniel C. Pregibon, Jesse Collins, T. Alan Hatton, and Patrick S. Doyle. Continuous-flow lithography for high-throughput microparticle synthesis. *Nature Materials*, 5(5):365–369, 2006.

[18] Luis M. Liz-Marzán, Michael Giersig, and Paul Mulvaney. Synthesis of nanosized goldsilica coreshell particles. *Langmuir*, 12(18):4329–4335, 1996.

[19] Jenifer Blacklock, Ye-Zi You, Qing-Hui Zhou, Guangzhao Mao, and David Oupicky. Gene delivery in vitro and in vivo from bioreducible multilayered polyelectrolyte films of plasmid dna. *Biomaterials*, 30(5):939 – 950, 2009.

[20] Julien Polleux, Nicola Pinna, Markus Antonietti, Christian Hess, Ute Wild, Robert Schlögl, and Markus Niederberger. Ligand functionality as a versatile tool to control the assembly behavior of preformed titania nanocrystals. *Chemistry*, 11(12):3541–51, Jun 2005.

[21] Chad A. Mirkin, Robert L. Letsinger, Robert C. Mucic, and James J. Storhoff. A dna-based method for rationally assembling nanoparticles into macroscopic materials. *Nature*, 382(6592):607–609, 1996.

[22] Vinothan N. Manoharan, Mark T. Elsesser, and David J. Pine. Dense packing and symmetry in small clusters of microspheres. *Science*, 301(5632):483–487, 2003.

[23] Eric Lauga and Michael P. Brenner. Evaporation-driven assembly of colloidal particles. *Phys. Rev. Lett.*, 93:238301, Dec 2004.

[24] Young-Sang Cho, Gi-Ra Yi, Shin-Hyun Kim, David J. Pine, and Seung-Man Yang. Colloidal clusters of microspheres from water-in-oil emulsions. *Chemistry of Materials*, 17(20):5006–5013, 2005.

[25] Young-Sang Cho, Gi-Ra Yi, Jong-Min Lim, Shin-Hyun Kim, Vinothan N. Manoharan, David J. Pine, and Seung-Man Yang. Self-organization of bidisperse colloids in water droplets. *Journal of the American Chemical Society*, 127(45):15968–15975, 2005.

[26] Young-Sang Cho, Gi-Ra Yi, Shin-Hyun Kim, Mark T. Elsesser, Dana R. Breed, and Seung-Man Yang. Homogeneous and heterogeneous binary colloidal clusters formed by evaporation-induced self-assembly inside droplets. *Journal of Colloid and Interface Science*, 318(1):124 – 133, 2008.

[27] Liang Hong, Shan Jiang, and Steve Granick. Simple method to produce janus colloidal particles in large quantity. *Langmuir*, 22(23):9495–9499, 2006.

[28] K.M Keville, E.I Franses, and J.M Caruthers. Preparation and characterization of monodisperse polymer microspheroids. *Journal of Colloid and Interface Science*, 144(1):103 – 126, 1991.

[29] Qian Chen, Jonathan K. Whitmer, Shan Jiang, Sung Chul Bae, Erik Luijten, and Steve Granick. Supracolloidal reaction kinetics of janus spheres. *Science*, 331(6014):199–202, 2011.

[30] Ali Mohraz and Michael J. Solomon. Gelation and internal dynamics of colloidal rod aggregates. *Journal of Colloid and Interface Science*, 300(1):155–162, 2006.

[31] Kyung Eun Sung, Siva A. Vanapalli, Deshpremy Mukhija, Hugh A. McKay, Joanna Mirecki Millunchick, Mark A. Burns, and Michael J. Solomon. Programmable fluidic production of microparticles with configurable anisotropy. *Journal of the American Chemical Society*, 130(4):1335–1340, 2008.

[32] Sho Asakura and Fumio Oosawa. Interaction between particles suspended in solutions of macromolecules. *Journal of Polymer Science*, 33(126):183–192, 1958.

[33] Marie Adams, Zvonimir Dogic, Sarah L. Keller, and Seth Fraden. Entropically driven microphase transitions in mixtures of colloidal rods and spheres. *Nature*, 393(6683):349–352, 05 1998.

[34] Fan Li, David P Josephson, and Andreas Stein. Colloidal assembly: the road from particles to colloidal molecules and crystals. *Angew Chem Int Ed Engl*, 50(2):360–388, Jan 2011.

[35] Colin J. Loweth, W. Brett Caldwell, Xiaogang Peng, and A. Paul Alivisatos. Dna-based assembly of gold nanocrystals. *Angewandte Chemie International Edition*, 38(12):1808–1812, 1999.

[36] SG Zhang. Fabrication of novel biomaterials through molecular self-assembly. *Nat Biotechnol*, 21(10):1171–1178, 2003.

[37] Hao Yan, Sung Ha Park, Gleb Finkelstein, John H. Reif, and Thomas H. LaBean. DNA-Templated Self-Assembly of Protein Arrays and Highly Conductive Nanowires. *Science*, 301(5641):1882–1884, 2003.

[38] Paul W. K. Rothemund. Folding dna to create nanoscale shapes and patterns. *Nature*, 440(7082):297–302, 03 2006.

[39] Dmytro Nykypanchuk, Mathew M. Maye, Daniel van der Lelie, and Oleg Gang. Dna-guided crystallization of colloidal nanoparticles. *Nature*, 451(31):549–552, January 2008.

[40] Sung Yong Park, Abigail K. R. Lytton-Jean, Byeongdu Lee, Steven Weigand, George C. Schatz, and Chad A. Mirkin. Dna-programmable nanoparticle crystallization. *Nature*, 451(7178):553–556, 2008.

[41] Gary Zabow, Stephen Dodd, John Moreland, and Alan Koretsky. Microengineered local field control for high-sensitivity multispectral mri. *Nature*, 453(19):1058–1063, 2008.

[42] Armen R. Kherlopian, Ting Song, Qi Duan, Mathew A Neimark, Ming J Po, John K Gohagan, and Andrew F Laine. A review of imaging techniques for systems biology. *BMC Systems Biology*, 2(74), 2008.

[43] Dong-Hyun Kim, Elena A. Rozhkova, Ilya V. Ulasov, Samuel D. Bader, Tijana Ragh, Maciej S. Lesniak, and Valentyn Novosad. Biofunctionalized magnetic-vortex microdiscs for targeted cancer-cell destruction. *Nature Materials*, 9:165–171, 2009.

[44] Angela K. Pannier and Lonnie D. Shea. Controlled release systems for dna delivery. *Mol Ther*, 10(1):19–26, 07 2004.

[45] Liyu Liu, Xixiang Huang, Cai Shen, Zhengyou Liu, Jing Shi, Weijia Wen, and Ping Sheng. Parallel-field electrorheological clutch: Enhanced high shear rate performance. *Applied Physics Letters*, 87(10):104106, 2005.

[46] Wenzhen Li, Xin Wang, Zhongwei Chen, Mahesh Waje, and Yan. Carbon nanotube film by filtration as cathode catalyst support for proton-exchange membrane fuel cell. *Langmuir*, 21(21):9386–9389, 2005.

[47] M.J. Decker, C.J. Halbach, C.H. Nam, N.J. Wagner, and E.D. Wetzel. Stab resistance of shear thickening fluid (stf)-treated fabrics. *Composites Science and Technology*, 67(3–4):565 – 578, 2007.

[48] Xi Jiang, Min Wang, David Y. Graham, and Mary K. Estes. Expression, self-assembly, and antigenicity of the norwalk virus capsid protein. *Journal of Virology*, 66(11), 1992.

[49] W. W. Newcomb, F. L. Homa, D. R. Thomsen, F. P. Booy, B. L. Trus, A. C. Steven, J. V. Spencer, and J. C. Brown. Assembly of the herpes simplex virus capsid: Characterization of intermediates observed during cell-free capsid formation. *Journal Of Molecular Biology*, 263(3):432–446, Nov 1996.

[50] Daniel Luque, Irene Saugar, J.F. Rodriguez, N. Verdaguer, D. Garriga, C. San Martin, J.A. Velazquez-Muriel, B.L. Trus, J.L. Carrascosa, and J.R. Caston. Infectious bursal disease virus capsid assembly and maturation by structural rearrangements of a transient molecular switch. *Journal of Virology*, 81(13), 2007.

[51] Marc Kirschner and Tim Mitchison. Beyond self-assembly: from microtubules to morphogenesis. *Cell*, 45:392–342, 1986.

[52] Nenad Ban, Poul Nissen, Jeffrey Hansen, Peter B. Moore, and Thomas A. Steitz. The complete atomic structure of the large ribosomal subunit at 2.4 å resolution. *Science*, 289(5481):905–920, 2000.

[53] R. Clausius. Xi. on the nature of the motion which we call heat. *Philosophical Magazine Series 4*, 14(91):108–127, 1857.

[54] James Clerck Maxkwell. Illustrations of the dynamical theory of gases. *Philosophical Magazine Series 4*, 20(130):21–37, 1860.

[55] Ludwig Boltzmann. Sitzungsber. kais. akad. wiss. wien. *Math. Naturwiss. Classe*, 76:373—435, 1877.

[56] J Willard Gibbs. *Elementary Principles in Statistical Mechanics. Developed with Especial Reference to the Foundation of Thermodynamics.* Yale University Press, 1902.

[57] Robert M. Ziff, Erdagon Gulari, and Yoav Barshad. Kinetic phase transitions in an irreversible surface-reaction model. *Phys. Rev. Lett.*, 56:2553–2556, Jun 1986.

[58] R. G. Larson. Self-assembly of surfactant liquid crystalline phases by monte carlo simulation. *The Journal of Chemical Physics*, 91(4):2479–2488, 1989.

[59] John H. Holland. Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, 2(2):88–105, 1973.

[60] L.B. Booker, D.E. Goldberg, and J.H. Holland. Classifier systems and genetic algorithms. *Artificial Intelligence*, 40(1–3):235 – 282, 1989.

[61] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[62] Marjorie J Vold. Computer simulation of floc formation in a colloidal suspension. *Journal of Colloid Science*, 18(7):684 – 695, 1963.

[63] Moti Lal. 'monte carlo' computer simulation of chain molecules. i. *Molecular Physics*, 17(1):57–64, 1969.

[64] B. J. Alder and T. E. Wainwright. Studies in molecular dynamics. i. general method. *The Journal of Chemical Physics*, 31(2):459–466, 1959.

[65] Michael Levitt and Ariel Warshel. Computer simulation of protein folding. *Nature*, 253(5494):694–698, 1975.

[66] Zhenli Zhang, Mark A. Horsch, Monica H. Lamm, and Sharon C. Glotzer. Tethered nano building blocks: toward a conceptual framework for nanoparticle self-assembly. *Nano Letters*, 3(10):1341–1346, 2003.

[67] Christopher R. Iacovella, Mark A. Horsch, Zhenli Zhang, and Sharon C. Glotzer. Phase diagrams of self-assembled mono-tethered nanospheres from molecular simulation and comparison to surfactants. *Langmuir*, 21(21):9488–9494, 2005.

[68] Christopher R. Iacovella, Aaron S. Keys, Mark A. Horsch, and Sharon C. Glotzer. Icosahedral packing of polymer-tethered nanospheres and stabilization of the gyroid phase. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 75(4):040801, 2007.

[69] Trung Dac Nguyen, Zhenli Zhang, and Sharon C. Glotzer. Molecular simulation study of self-assembly of tethered v-shaped nanoparticles. *The Journal of Chemical Physics*, 129(24):244903, 2008.

[70] Carolyn L. Phillips, Christopher R. Iacovella, and Sharon C. Glotzer. Stability of the double gyroid phase to nanoparticle polydispersity in polymer-tethered nanosphere systems. *Soft Matter*, 6:1693–1703, 2010.

[71] Aaron S. Keys and Sharon C. Glotzer. How do quasicrystals grow? *Phys. Rev. Lett.*, 99:235503, Dec 2007.

[72] Amir Haji-Akbari, Michael Engel, Aaron S. Keys, Xiaoyu Zheng, Rolfe G. Petschek, Peter Palffy-Muhoray, and Sharon C. Glotzer. Disordered, quasicrystalline and crystalline phases of densely packed tetrahedra. *Nature*, 462(7274):773–777, 12 2009.

[73] Ting Chen, Zhenli Zhang, and Sharon C. Glotzer. A precise packing sequence for self-assembled convex structures. *Proc Natl Acad Sci*, 104(3):717–722, January 2007.

[74] Ting Chen, Zhenli Zhang, and Sharon C. Glotzer. Simulation studies of the self-assembly of cone-shaped particles. *Langmuir*, 23(12):6598–6605, 2007.

[75] Joshua A. Anderson, Chris D. Lorenz, and A. Travesset. General purpose molecular dynamics simulations fully implemented on graphics processing units. *Journal of Computational Physics*, 227(10):5342 – 5359, 2008.

[76] Trung Dac Nguyen, Carolyn L. Phillips, Joshua A. Anderson, and Sharon C. Glotzer. Rigid body constraints realized in massively-parallel molecular dynamics on graphics processing units. *Computer Physics Communications*, 182(11):2307 – 2313, 2011.

[77] Carolyn L. Phillips, Joshua A. Anderson, and Sharon C. Glotzer. Pseudo-random number generation for brownian dynamics and dissipative particle dynamics simulations on gpu devices. *Journal of Computational Physics*, 230(19):7191 – 7201, 2011.

[78] J.A. Barker and R.O. Watts. Structure of water; a monte carlo calculation. *Chemical Physics Letters*, 3(3):144 – 145, 1969.

[79] R. Eppenga and D. Frenkel. Monte carlo study of the isotropic and nematic phases of infinitely thin hard platelets. *Molecular Physics*, 52:1303–1334, 1984.

[80] Zhenli Zhang, Zhiyong Tang, Nicholas A Kotov, and Sharon C Glotzer. Simulations and analysis of self-assembly of cdte nanoparticles into wires and sheets. *Nano Letters*, 7(6):1670–1675, Jun 2007.

[81] R.H. Swendsen and J.S. Wang. Replica monte carlo simulation of spin-glasses. *Phys. Rev. Lett.*, 57(21), 1986.

[82] Ulli Wolff. Collective monte carlo updating for spin systems. *Phys. Rev. Lett.*, 62(4):361–364, Jan 1989.

[83] Jiwen. Liu and Erik Luijten. Rejection-free geometric cluster algorithm for complex fluids. *Phys. Rev. Lett.*, 92(3), 2004.

[84] Jiwen Liu and Erik Luijten. Generalized geometric cluster algorithm for fluid simulation. *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics)*, 71(6):066701, 2005.

[85] Stephen Whitelam and Phillip L. Geissler. Avoiding unphysical kinetic traps in monte carlo simulations of strongly attractive particles. *The Journal of Chemical Physics*, 127(15):154101, 2007.

[86] John E. Stone, James C. Phillips, Peter L. Freddolino, David J. Hardy, Leonardo G. Trabuco, and Klaus Schulten. Accelerating molecular modeling applications with graphics processors. *Journal of Computational Chemistry*, 28(16):2618–2640, 2007.

[87] Boris D. Lobachevsky. Efficient parallel simulations of asynchronous cellular arrays. *Complex Systems*, 1(6):1099–1123, 1987.

[88] G Korniss, M.A Novotny, and P.A Rikvold. Parallelization of a dynamic monte carlo algorithm: A partially rejection-free conservative approach. *Journal of Computational Physics*, 153(2):488 – 508, 1999.

[89] Tobias Preis, Peter Virnau, Wolfgang Paul, and Johannes J. Schneider. Gpu accelerated monte carlo simulation of the 2d and 3d ising model. *Journal of Computational Physics*, 228(12):4468 – 4477, 2009.

[90] Tal Levy, Guy Cohen, and Eran Rabanai. Simulating lattice spin models on graphics processing units. *Journal of Chemical Theory and Computation*, 6(11):3293—3301, 2010.

[91] Ruichao Ren and G. Orkoulas. Acceleration of markov chain monte carlo simulations through sequential updating. *The Journal of Chemical Physics*, 124(6):064109, 2006.

[92] Ruichao Ren and G. Orkoulas. Parallel markov chain monte carlo simulations. *The Journal of Chemical Physics*, 126(21):211102, 2007.

[93] C. J. O'Keeffe and G. Orkoulas. Parallel canonical monte carlo simulations through sequential updating of particles. *The Journal of Chemical Physics*, 130(13):134109, 2009.

[94] Joshua A. Anderson, Eric Jankowski, Thomas Grubb, Michael Engel, and Sharon C. Glotzer. Calculation of the hard sphere equation of state on graphics hardware. *in prep*, 2012.

[95] Richard Durstenfeld. Algorithm 235: Random permutation. *Commun. ACM*, 7(7):420–, 1964.

[96] Etienne P. Bernard and Werner Krauth. Two-step melting in two dimensions: First-order liquid-hexatic transition. *Phys. Rev. Lett.*, 107:155704, Oct 2011.

[97] Eric Jankowski and Sharon C. Glotzer. A comparison of new methods for generating energy-minimizing configurations of patchy particles. *Journal of Chemical Physics*, 131(10):104104, 2009.

[98] Daan Frenkel and Berend Smit. *Understanding Molecular Simulations: From Algorithms to Applications*. Elsevier, 2002.

[99] M. P. Allen and D. J. Tildesley. *Computer simulation of liquids*. Clarendon Press, 1987.

[100] Marshall N. Rosenbluth and Arianna W. Rosenbluth. Monte carlo calculation of the average extension of molecular chains. *The Journal of Chemical Physics*, 23(2):356–359, 1955.

[101] Bin Chen and J. Ilja Siepmann. Improving the efficiency of the aggregation-volume-bias monte carlo algorithm. *Journal of Physical Chemistry B*, 105(45):11275–11282, 2001.

[102] M Mura, N Martsinovich, and L Kantorovich. Theoretical study of melamine superstructures and their interaction with the au(111) surface. *Nanotechnology*, 19(46):465704 (14pp), 2008.

[103] Alessandro. Troisi, Vince. Wong, and Mark A. Ratner. An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Sciences of the United States of America*, 102(2):255–260, Jan 2005.

[104] C. Domb. Lattice animals and percolation. *Journal of Physics A*, 9(L141), 1976.

[105] A. Troisi, V. Wong, and MA V. Ratner. Self-assembly on multiple length scales: A monte carlo algorithm with data augmentation. *Journal of Chemical Physics*, 122(024102):0, 2005.

[106] Vasilios I. Manousiouthakis and Michael W. Deem. Strict detailed balance is unnecessary in monte carlo simulation. *The Journal of Chemical Physics*, 110(6):2753–2756, 1999.

[107] Stephan. Mertens. A physicist's approach to number partitioning. *Theoretical Computer Science*, 265:79–108, 2001.

[108] C. Borgs, J.T. Chayes, and B. Pittel. Sharp threshold and scaling window for the integer partitioning problem. *Proceedings of the 2001 ACM Symposium on the Theory of Computing*, pages 330–336, 2001.

[109] Robert M. Ziff. Four-tap shift-register-sequence random-number generators. *Computers in Physics*, 12(4):385–392, 1998.

[110] Eric Jankowski and Sharon C. Glotzer. Screening and designing patchy particles for optimized self-assembly propensity through assembly pathway engineering. *Soft Matter*, 8(10):2852–2859, 2012.

[111] Zhihong Nie, Alla Petukhova, and Eugenia Kumacheva. Properties and emerging applications of self-assembled structures made from inorganic nanoparticles. *Nature Nanotechnology*, 5(1):15–25, 01 2010.

[112] Zhang and Sharon C. Glotzer. Self-assembly of patchy particles. *Nano Letters*, 4(8):1407–1413, 2004.

[113] Eric Jankowski and Sharon C. Glotzer. Calculation of partition functions for the self-assembly of patchy particles. *The Journal of Physical Chemistry B*, 115(48):14321–14326, 2011.

[114] David J. Wales and Tetyana V. Bogdan. Potential energy and free energy landscapes. *J Phys Chem B*, 110:20765–20776, 2006.

[115] David J. McGinty. Vapor phase homogeneous nucleation and the thermodynamic properties of small clusters of argon atoms. *The Journal of Chemical Physics*, 55(2):580–588, 1971.

[116] Donald A McQuarrie. *Statistical Mechanics.* University Science Books: Sausalito, CA, 2000.

[117] R. J. Baxter. Partition function of the three-dimensional zamolodchikov model. *Phys. Rev. Lett.*, 53(19):1795–1798, Nov 1984.

[118] Stephan Mertens. A physicist's approach to number partitioning. *Theoretical Computer Science*, 265(1-2):79 – 108, 2001.

[119] Antoine Georges, Gabriel Kotliar, Werner Krauth, and Marcelo J. Rozenberg. Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions. *Rev. Mod. Phys.*, 68(1):13, Jan 1996.

[120] Peter H. Poole, Francesco Sciortino, Ulrich Essmann, and H. Eugene Stanley. Phase behavior of metastable water. *Nature*, 360:324–328, 1992.

[121] Goundla Srinivas, Dennis E. Discher, and Michael L. Klein. Self-assembly and properties of diblock copolymers by coarse-grain molecular dynamics. *Nature Materials*, 3(9):638–644, 09 2004.

[122] D. Rapaport. Self-assembly of polyhedral shells: A molecular dynamics study. *Phys. Rev. E.*, 70(5):051905, 2004.

[123] Hung D. Nguyen, Vijay S. Reddy, and Charles L. Brooks. Deciphering the kinetic mechanism of spontaneous self-assembly of icosahedral capsids. *Nano Letters*, 7(2):338–344, 2007.

[124] Brian C Barnes, Daniel W Siderius, and Lev D Gelb. Structure, thermodynamics, and solubility in tetromino fluids. *Langmuir*, 25(12):6702–16, Jun 2009.

[125] Kenneth M. Benjamin, Andrew J. Schultz, and David A. Kofke. Virial coefficients of polarizable water: applications to thermodynamic properties and molecular clustering. *The Journal of Physical Chemistry C*, 111(43):16021–16027, 2007.

[126] J D Bryngelson and P G Wolynes. Spin glasses and the statistical mechanics of protein folding. *Proceedings of the National Academy of Sciences*, 84(21):7524–7528, 1987.

[127] Martin Karplus and Andrej Sali. Theoretical studies of protein folding and unfolding. *Current Opinion in Structural Biology*, 5(1):58 – 73, 1995.

[128] Alessandro Troisi, Vance Wong, and Mark A Ratner. An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Sciences of the United States of America*, 102(2):255–260, Jan 2005.

[129] Fabio Cicoira and Federico Rosei. Playing tetris at the nanoscale. *Surface Science*, 600(1):1 – 5, 2006.

[130] Tobias M. Schneider, Shreyas Mandre, and Michael P. Brenner. Algorithm for a microfluidic assembly line. *Phys. Rev. Lett.*, 106(9):094503, Feb 2011.

[131] Frank H. Stillinger and Thomas A. Weber. Packing structures and transitions in liquids and solids. *Science*, 225(4666):983–989, 1984.

[132] Birgit Strodel and David J. Wales. Free energy surfaces from an extended harmonic superposition approach and kinetics for alanine dipeptide. *Chemical Physics Letters*, 466(4-6):105 – 115, 2008.

[133] Charles T. Zahn and Ralph Z. Roskies. Fourier descriptors for plane closed curves. *Computers, IEEE Transactions on*, C-21(3):269 –281, march 1972.

[134] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:239–256, 1992.

[135] Aaron S. Keys, Christopher R. Iacovella, and Sharon C. Glotzer. Characterizing structure through shape matching and applications to self-assembly. *Annual Review of Condensed Matter Physics*, 2(1):263–285, 2011.

[136] Nicos Christofides. The optimum traversal of a graph. *Omega*, 1(6):719 – 732, 1973.

[137] Giuseppe Foffi and Francesco Sciortino. Extended law of corresponding states in short-range square wells: A potential energy landscape study. *Phys. Rev. E*, 74(5):050401, Nov 2006.

[138] Guangnan Meng, Natalie Arkus, Michael P Brenner, and Vinothan N Manoharan. The free-energy landscape of clusters of attractive hard spheres. *Science*, 327(5965):560–563, Jan 2010.

[139] Mark S. Gudiksen, Lincoln J. Lauhon, Jianfang Wang, David C. Smith, and Charles M. Lieber. Growth of nanowire superlattice structures for nanoscale photonics and electronics. *Nature*, 415(6872):617–620, 02 2002.

[140] Michael C. McAlpine, Robin S. Friedman, Song Jin, Keng-hui Lin, Wayne U. Wang, and Charles M. Lieber. High-performance nanowire electronics and photonics on glass and plastic substrates. *Nano Letters*, 3(11):1531–1535, 2003.

[141] Renuga Gopal, Satinderpal Kaur, Zuwei Ma, Casey Chan, Seeram Ramakrishna, and Takeshi Matsuura. Electrospun nanofibrous filtration membrane. *Journal of Membrane Science*, 281(1-2):581 – 586, 2006.

[142] Stephen Whitelam, Edward H. Feng, Michael F. Hagan, and Phillip L. Geissler. The role of collective motion in examples of coarsening and self-assembly. *Soft Matter*, 5:1251–1262, 2009.

[143] Stephen Whitelam. Control of pathways and yields of protein crystallization through the interplay of nonspecific and specific attractions. *Physical Review Letters*, 105(8):088102, Aug 2010.

[144] Thomas K. Haxton and Stephen Whitelam. Design rules for the self-assembly of a protein crystal. *arXiv:1110.5610v1 [cond-mat.soft]*, 2011.

[145] James Grant, Robert L. Jack, and Stephen Whitelam. Analyzing mechanisms and microscopic reversibility of self-assembly. *arXiv:1108.4542v1 [cond-mat.soft]*, 2011.

[146] Weijia Wen, Xianxiang Huang, and Ping Sheng. Electrorheological fluids: structures and mechanisms. *Soft Matter*, 4:200–210, 2008.

[147] Alexander J. Williamson, Alex W. Wilber, Jonathan P. K. Doye, and Ard A. Louis. Templated self-assembly of patchy particles. *Soft Matter*, 7:3423–3431, 2011.

[148] Robert Jack, Michael F. Hagan, and David Chandler. Fluctuation-dissipation ratios in the dynamics of self-assembly. *Physical Review E*, 76(2):021119, 2007.

[149] Daphne Klotsa and Robert L. Jack. Predicting the self-assembly of a model colloidal crystal. *Soft Matter*, pages 6294–6303, 2011.

[150] Stephen Whitelam. Nonclassical assembly pathways of anisotropic particles. *Journal of Chemical Physics*, 132(19):194901, May 2010.

[151] Michael F. Hagan and David Chandler. Dynamic pathways for viral capsid assembly. *Biophysical Journal*, 91(1):42 – 54, 2006.

[152] Jonathan P K Doye and Claire P Massen. Characterizing the network topology of the energy landscapes of atomic clusters. *Journal of Chemical Physics*, 122(8):84105, Feb 2005.

[153] W.B. Russel, D. A. Saville, and W. R. Schowalter. *Colloidal Dispersions*. Cambridge University Press: Cambridge, U.K., 1989.

[154] George D. J. Phillies. Excess chemical potential of dilute solutions of spherical polyelectrolytes. *The Journal of Chemical Physics*, 60(7):2721–2731, 1974.

[155] Alex W. Wilber, Jonathan P. K. Doye, Ard A. Louis, Eva G. Noya, Mark A. Miller, and Pauline Wong. Reversible self-assembly of patchy particles into monodisperse icosahedral clusters. *The Journal of Chemical Physics*, 127(8):085106, 2007.

[156] Daniel Ortiz, Eric Jankowski, and Sharon C. Glotzer. Self-assembly of colloidal tetrominoes. *in prep*, 2012.

[157] Trung Dac Nguyen, Eric Jankowski, and Sharon C. Glotzer. Self-assembly and reconfigurability of shape-shifting particles. *ACS Nano*, 5(11):8892—8903, 2011.

[158] Carolyn L. Phillips, Eric Jankowski, Michelle Marval, and Sharon C Glotzer. Self-assembling clusters inspired by mathematical extremal points on the surface of a sphere. *arxiv:1201:5131*, 2012.

[159] Stefano Sacanna and David J. Pine. Shape-anisotropic colloids: Building blocks for complex assemblies. *Current Opinion in Colloid & Interface Science*, January 2011.

[160] Matthew R. Jones, Robert J. Macfarlane, Byeongdu Lee, Jian Zhang, Kaylie L. Young, Andrew J. Senesi, and Chad A. Mirkin. Dna-nanoparticle superlattices formed from anisotropic building blocks. *Nature Materials*, 9:913–917, 2010.

[161] Szilard N. Fejer, Dwaipayan Chakrabarti, and David J. Wales. Emergent complexity from simple anisotropic building blocks: Shells, tubes, and spirals. *ACS Nano*, 4(1):219–228, 2010.

[162] Szilard N. Fejer, Dwaipayan Chakrabarti, and David J. Wales. Self-assembly of anisotropic particles. *Soft Matter*, 7:3553–3564, 2011.

[163] DJ Wales and JPK Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

[164] Francesco Sciortino, Achille Giacometti, and Giorgio Pastore. A numerical study of one-patch colloidal particles: from square-well to janus. *Phys. Chem. Chem. Phys.*, 12:11869–11877, 2010.

[165] S. Mossa, F. Sciortino, P. Tartaglia, and E. Zaccarelli. Ground-state clusters for short-range attractive and long-range repulsive potentials. *Langmuir*, 20(24):10756–10763, 2004.

[166] Marie-Pierre Valignat, Olivier Theodoly, John C. Crocker, William B. Russel, and Paul M. Chaikin. Reversible self-assembly and directed assembly of dna-linked micrometer-sized colloids. *Proceedings of the National Academy of Sciences of the United States of America*, 102(12):4225–4229, 2005.

[167] Dmytro Nykypanchuk, Mathew M. Maye, Daniel van der Lelie, and Oleg Gang. Dna-based approach for interparticle interaction control. *Langmuir*, 23(11):6305–6314, 2007.

[168] Mirjam E. Leunissen and Daan Frenkel. Numerical study of dna-functionalized microparticles and nanoparticles: Explicit pair potentials and their implications for phase behavior. *The Journal of Chemical Physics*, 134(8):084702, 2011.

[169] Nicholas A. Licata and Alexei V. Tkachenko. Self-assembly of dna-coded nanoclusters. *Phys. Rev. E*, 74(4):040401, Oct 2006.

[170] Nicholas A. Licata and Alexei V. Tkachenko. Colloids with key-lock interactions: Nonexponential relaxation, aging, and anomalous diffusion. *Phys. Rev. E*, 76(4):041405, Oct 2007.

[171] Paul L. Biancaniello, Anthony J. Kim, and John C. Crocker. Colloidal interactions and self-assembly using dna hybridization. *Phys. Rev. Lett.*, 94(5):058302, Feb 2005.

[172] Dmytro Nykypanchuk, Mathew M. Maye, Daniel van der Lelie, and Oleg Gang. DNA-guided crystallization of colloidal nanoparticles. *Nature*, 451(7178):549–552, January 2008.

[173] Emanuela Bianchi, Ronald Blaak, and Christos N. Likos. Patchy colloids: state of the art and perspectives. *Phys. Chem. Chem. Phys.*, 13:6397–6410, 2011.

[174] Amar B. Pawar and Ilona Kretzschmar. Fabrication, assembly, and application of patchy particles. *Macromolecular Rapid Communications*, 31(2):150–168, 2010.

[175] Fejes Tóth. *Regular Figures.* The Macmillan Company, 1964.

[176] Theodor William Melnyk, Osvald Knop, and William Robert Smith. Extremal arrangements of points and unit charges on a sphere: equilibrium configurations revisited. *Canadian Journal of Chemistry*, 55(10):1745–1761, 1977.

[177] L. L. Whyte. Unique arrangements of points on a sphere. *The American Mathematical Monthly*, 59(9):pp. 606–611, 1952.