Thermodynamic stability versus Kinetic Accessibility: Pareto Fronts for Programmable Self-Assembly

Anthony Trubiano, Miranda Holmes-Cerfon

June 16, 2021

1 Equilibrium Probability Calculations

In equilibrium, the probability of observing a system in a given micro-state, x, is denoted $p(x)$, and is given by the Boltzmann distribution,

$$p(x) = \frac{1}{Z} e^{-\beta U(x)}. \tag{1}$$

Here Z is a normalizing constant called the *partition function*, β is the inverse temperature, and $U(x)$ is the potential energy of the micro-state x. We consider a chain of N particles of unit diameter $d = 1$, in two or three dimensions, with short-ranged, isotropic, pairwise interactions, and we assume the backbone interactions are so strong they do not break. When the interaction range is short enough, a good approximation is to treat this system in the *sticky limit*, in which the range of the interactions goes to zero, i.e. particles interact only when they are in direct contact. Each micro-state x can be assigned and grouped into a macro-state by its adjacency matrix. We will refer to these macro-states as *clusters*.

The probability of observing the system in cluster j in equilibrium, π_j, can then be computed by integrating (1) over the appropriate space. We let C_j denote the set of all micro-states consistent with cluster j; including all rotations, translations, reflections, and deformations along internal degrees of freedom. We find

$$\pi_j = \frac{1}{Z} \int_{C_j} e^{-\beta U(x)} dx = \frac{Z_j}{Z}, \tag{2}$$

where Z_j is the contribution to the total partition function from cluster j. Evaluating the integral (2) in the sticky limit is quite involved; the calculations can be found in [1,2], which can be adapted to consider assembly from a gas or from a polymer. The end result is

$$\pi_j = \frac{z_j}{Z} \kappa^{b_j}, \quad Z = \sum_k z_k \kappa^{b_k}. \tag{3}$$

Here z_j is called the *geometric* partition function for cluster j, and depends only on the geometric properties of the cluster such as the moment of inertia and symmetry number. All the dependence on the interactions is encapsulated by the *sticky parameter*, κ, and b_j denotes the number of bonds in cluster j. The sticky parameter is given by the partition function for a single bond,

$$\kappa = \int_0^{r_c} e^{-\beta U(r)} dr, \tag{4}$$

where r_c is some cutoff distance beyond which the potential goes to zero. Using Laplace asymptotics to evaluate the integral (4), we find

$$\kappa = \sqrt{2\pi} e^{-\beta U(d)} / \sqrt{\beta U''(d)}, \tag{5}$$

where $d = 1$ is the minimum of the potential energy. When we estimate probabilities for an interaction potential that has finite range, such as the Morse potential considered for all the examples in the text, we use (5) to convert the parameters for the potential into a sticky parameter.

According to expression (3) for π_j, the equilibrium probability only depends on κ, since z_j is a constant for a given cluster. Therefore, as long as we know z_j for all j, we can determine π_j for any κ. Alternatively, if we can determine the value of all the π_j for a given κ_0, we can determine equilibrium probabilities for any other κ_1, by performing a simple re-weighting and re-normalization. This reweighting works provided κ_0, κ_1 are close enough that there is sufficient overlap in their associated distributions; it was shown in [3] that the reweighting works over fairly large range of sticky parameters for small clusters.

The reweighting works as follows. Let $K_0^j = \kappa_0^{b_j}$ and $K_1^j = \kappa_1^{b_j}$. The true equilibrium distributions for these sticky parameter choices are

$$\pi_j^0 = \frac{z_j K_0^j}{\sum_n z_n K_0^n}, \quad \pi_j^1 = \frac{z_j K_1^j}{\sum_n z_n K_1^n}. \tag{6}$$

Next, we construct an expression for π_j^1 in terms of π_j^0. We let

$$\Pi_j^1 = \frac{\pi_j^0 \frac{K_1^j}{K_0^j}}{\sum_n \pi_n^0 \frac{K_1^n}{K_0^n}}, \tag{7}$$

and we show that $\Pi_j^1 = \pi_j^1$ by showing that their ratio is 1. The calculation shows

$$\begin{aligned}
\frac{\Pi_j^1}{\pi_j^1} &= \frac{\pi_j^0 K_1^j}{K_0^j \sum_n \pi_n^0 \frac{K_1^n}{K_0^n}} \frac{\sum_n z_n K_1^n}{z_j K_1^j} \\
&= \frac{z_j K_0^j}{\sum_n z_n K_0^n} \frac{K_1^j}{z_j K_1^j} \frac{\sum_n z_n K_1^n}{1} \frac{1}{K_0^j \sum_n K_0^n \frac{z_n}{\sum_m z_m K_0^m} \frac{K_1^n}{K_0^n}} \\
&= \frac{K_0^j}{\sum_n z_n K_0^n} \frac{\sum_n z_n K_1^n}{1} \frac{\sum_m z_m K_0^m}{K_0^j \sum_n z_n K_1^n} \\
&= 1.
\end{aligned}$$

This shows that we can evaluate the equilibrium probabilities at any value κ_1 by knowing the equilibrium probabilities at some reference value, κ_0, and applying Equation (7). All that remains is to compute the reference set of equilibrium probabilities.

While it is possible to evaluate reference probabilities π_j^0 semi-analytically for the rigid ground states[4], this is not possible for floppy states, so instead we estimate them by Monte Carlo sampling. We utilized the stratification sampler developed by Holmes-Cerfon[3] to sample the system exactly in the sticky limit. This sampler proposes moves on a constraint manifold in which bonded particles are exactly unit distance apart, and can also propose to jump to a new constraint manifold, which corresponds to forming or breaking a bond. We only allow non-backbone bonds to break since we assume the backbone bonds are so strong they are unbreakable. We set $\kappa_0 = 2$ to quickly explore the entire state space. During a single run, we generate 5×10^6 points, saving every fifth data point. We then split the data sequentially into ten bins and construct estimators of the equilibrium probability of each state in each bin. We compute error bars as the variance across the bins. If we perform this procedure a few times, we can then combine these estimates, weighting them in such a way as to minimize the variance. We stop when the largest relative error bound does not exceed 3%. This required repeating 4 times for disks and 6 spheres, and 6 times for 7 disks.

2 Brownian Dynamics Simulations

We perform Brownian dynamics simulations of a chain of colloidal particles to study the kinetics of folding. Let $\vec{X}_t$ be a vector denoting the coordinates of the center of N particles of diameter d as a function of time. We choose $\vec{X}_0$ such that the particles are aligned on the x-axis, spaced a distance d apart. The positions are then evolved via the overdamped Langevin equation,

$$\frac{d\vec{X}_t}{dt} = -\frac{D}{kT}\nabla U(\vec{X}_t) + \sqrt{2D}\eta(t), \tag{8}$$

where D is the diffusion coefficient for a particle, k is Boltzmann's constant, T is the temperature, U is a potential energy function, and $\eta(t)$ is a vector of independent white noises. The potential energy is computed as a sum over pair potentials between each particle. To model the short-ranged interactions of the colloids, we use a Morse potential with the form

$$U_M(r) = E\left(e^{-2\rho(r-d)} - 2e^{-\rho(r-d)}\right). \tag{9}$$

Here, r is the inter-particle separation, E is the bond energy, and ρ is a parameter governing the width of the potential. We use a constant $\rho = 40$ in our simulations, which corresponds to an interaction distance of about 6% of the particle diameter[5]. The bond energy is left as a design parameter.

To keep the chain intact, we introduce a stiff harmonic potential between the initially adjacent particles, which replaces the previously described Morse potential. The potential has the form

$$U_H(r) = \frac{k}{2}(r-1)^2, \tag{10}$$

with spring constant $k = 2\rho^2 E_0$, to match the curvature of a Morse potential at the minimum. We set $E_0 = 14kT$ for six particle systems and $E_0 = 12kT$ for seven particle systems. With these choices, we observe that the backbone almost always stays intact, and discard trajectories in the rare cases in which the backbone breaks.

This equation can be non-dimensionalized by scaling positions by the particle diameter, d, scaling time by an unknown parameter, c, and scaling the potential energy by kT. Re-using the same notation, the non-dimensional equation looks like

$$\frac{d\vec{X}_t}{dt} = -\varepsilon\nabla U(X_t) + \sqrt{2\varepsilon}\eta(t), \tag{11}$$

where $\varepsilon = \frac{Dc}{d^2}$ is a dimensionless parameter. We perform our simulations at $\varepsilon = 1$.

We numerically solve Equation 11 using the Euler-Maruyama scheme[6]. Due to the stiff potential, we use a time-step of $\Delta t = 5 \times 10^{-6}$ to ensure stability of the numerical scheme.

To convert our non-dimensional simulation time, t_{sim}, to physical time in a lab, t_{lab}, we determine the value of our scaling parameter, c, using experimental data. Experiments with clusters of $d = 1.3\mu$m colloids above a wall, showed that the diffusion coefficient is approximately $D = 0.1\mu\text{m}^2/\text{s}$ for an isolated particle, and slightly smaller on average for diffusion along an internal degree of freedom, $D = 0.065\mu\text{m}^2/\text{s}$[7]. Using these values, a lower bound for our time scaling is $c \approx 17$ seconds. The estimated lab time is then given by $t_{\text{lab}} = ct_{\text{sim}}$.

Although this scaling is for colloids diffusing above a wall, we apply it to both 3-dimensional and 2-dimensional clusters of colloids indiscriminately. We do not expect the scaling to give quantitative agreement with any particular experiment, which would anyways require considering different diffusion coefficients for different internal degrees of freedom[8], but we do expect it to give an estimate of the order of magnitude of the timescales involved. Regardless, the ratio of folding timescales we observe in simulations with different bond energies does not depend on the scaling we choose.

To determine what state the system is in at time t, we construct an adjacency matrix, A_t, using the coordinates $\vec{X}_t$. Let $r_t^{ij} = \|\vec{X}_t^i - \vec{X}_t^j\|$ be the distance between particles i and j at time t. The adjacency matrix is then constructed such that $a_t^{ij} = 1$ if $r_t^{ij} < r_{\text{cut}}$ and $a_t^{ij} = 0$ otherwise. We used a cutoff value $r_{\text{cut}} = 1.04$. This adjacency matrix is then compared against a database of adjacency matrices, that was precomputed when we performed the equilibrium probability calculations as in Section 1.

When reporting cluster yields from our Brownian dynamics simulations, we smooth the yield curves by averaging over a moving window with size equal to 0.5% of the total number of time points.

3 Coarse Grained Dynamical Model

3.1 Overview

Our model approximates the dynamics on the energy landscape as a Continuous Time Markov Chain (CTMC), where each node or state of the Markov chain represents the configurations of a cluster with a particular set of bonds, described by its adjacency matrix. We treat particles as distinguishable in this step, so we do not lump together clusters whose adjacency matrices are equivalent under a permutation of particle labels. We are interested in both the equilibrium probabilities and the dynamics of this CTMC. The equilibrium probability π_i of state i is given by integrating the Boltzmann distribution over the continuous set of configurations that have bonds associated with state i. We estimate this integral for a given set of bond energies, and then reweight the estimate to obtain equilibrium probabilities for other bond energies, as described in Section 1.

The dynamics of the CTMC are fully specified by the rates Q_{ij} of transitioning from state i to state j. We approximate these using the maximium likelihood estimators we would obtain from a long trajectory of the continuous dynamics with infinitely strong bonds, combined with the principle of detailed balance. That is, we set $Q_{ij} = 0$ unless states i, j are related by either adding or breaking a single bond. If a bond is added, i.e. state j has the same bonds as i plus one extra, then we choose $Q_{ij} = P_{ij}/\tau_i$, where τ_i is the mean first-passage time to form a bond from state i, starting with the equilibrium distribution and conditioned on no other bonds breaking, and P_{ij} is the probability that when this first bond forms, it is the one from state j. Notably, Q_{ij} does not depend on the bond energies, it only depends on the diffusion coefficient and the shape of the manifold of configurations corresponding to state i. For the reverse transition, from state j to a state i with one bond broken, we set the rate using detailed balance: $Q_{ji} = Q_{ij}\pi_i/\pi_j$. This choice ensures that the equilibrium probability for the CTMC is still $\pi = (\pi_i)_i$. Details of how we estimate P_{ij}, τ_i are in the following section.

We remark that it might be more appropriate to estimate the exit probabilities and rates P_{ij}, τ_i by starting from the quasi-stationary distribution, not the equilibrium distribution, a distribution that is appropriate for describing rare exits from metastable states; it would be interesting to know whether this give significantly different estimates[9]. We are also neglecting hydrodynamic interactions, which can affect dynamics quite significantly[8].

Given our rate matrix $Q = (Q_{ij})_{i,j}$ and its equilibrium distribution π, we may solve for many kinetic quantities using linear algebra. In this paper we consider the mean first passage time (mfpt) τ from the linear chain (state 0) to our target state. Let S be the indices corresponding to all adjacency matrices that are identified as our target state. Define a vector of mean first passage times, $\vec{\tau} = (\tau_0, \tau_1, \ldots)$ such that

$$\tau_i = E[\inf\{t \geq 0 \text{ such that } X_t \in A\} | X_0 = i]. \tag{12}$$

Each component τ_i gives the mfpt from state i to a state in S. The mfpt from the linear chain (state 0) to S is found by solving the system of linear equations[10]

$$(Q\vec{\tau})_i = -1 \;\; (i \notin \{0\} \cup S), \quad \tau_i = 0 \;\; (i \in S),$$

for the vector $\vec{\tau} = (\tau_0, \tau_1, \ldots)$, and then setting $\tau = \vec{\tau}_0$.

In our examples the rate matrices Q are typically quite sparse and structured. This is because each state can only be connected to other states with one more or one fewer bonds, and these connections are further constrained by cluster geometry. We take advantage of this fact by using a sparse solver to invert these linear systems. For $N = 6$ disks, we use a dense-LU factorization based solver. In this case, the system is small enough that using a sparse solver in unnecessary. For $N = 7$ disks and $N = 6$ spheres, we use a sparse-LU factorization based solver. All linear algebra calculations are performed using the Eigen C++ library[11].

3.2 Parameter Estimation

To construct the CTMC model, we need estimators for π_i, the equilibrium probability of state i, τ_i, the mean first exit time out of state i, and P_{ij}, the probability distribution of which state j forms when exiting state i, for all i. We discussed how to estimate π_i in Section (1). We now address how to estimate τ_i and P_{ij}.

The mean first exit time out of state i is defined to be

$$\tau_i = E\left[\min\{n \geq 0 \text{ such that } X_n \neq i\} | X_0 = i\right], \tag{13}$$

the expected value of the first time a new state is reached, conditioned on starting in the equilibrium distribution of state i and having no existing bonds break. One approach to estimating τ_i would be to sample M trajectories, equilibrated in state i and then evolved in time chunks δt, until the next state, j, is formed. Using this method would require significant computation in order to equilibrate each sample independently. Instead, we propose an estimator that only requires a single long trajectory, using all of the data along the way.

Assume an equilibrated trajectory forms a bond at time step n. Then at step 1 of the trajectory, we know it will take n steps to exit state i. At step 2, it takes $n-1$ steps. Continue until at step n we know it takes 0 steps to exit. Each of these samples is drawn from the equilibrium distribution and can be combined into an estimator for the mean first passage time. We get

$$\widehat{\tau_i} = \frac{\delta t}{n}\left(1 + 2 + \cdots + n\right) = \frac{n(n+1)}{2n}\delta t \tag{14}$$

as an estimator after a single transition. We then apply a reflecting boundary condition (i.e. set the state back to what it was at time step $n-1$), reset the timer, and repeat the process, evolving the trajectory independently of the first n steps. To evaluate the total contribution to the estimator after several exits, one only needs to record the exit times, T_k, for $k = 1, \ldots, M$, and the number of times each state j is reached, c_j. After M exits, the estimators can be written as

$$\widehat{\tau_i} = \frac{\delta t}{\sum_{k=1}^{M} T_k} \sum_{k=1}^{M} \frac{T_k(T_k+1)}{2}\delta t, \tag{15}$$

$$\widehat{P_{ij}} = \frac{c_j}{M}, \tag{16}$$

$$\widehat{Q_{ij}} = \frac{\widehat{P_{ij}}}{\widehat{\tau_i}}. \tag{17}$$

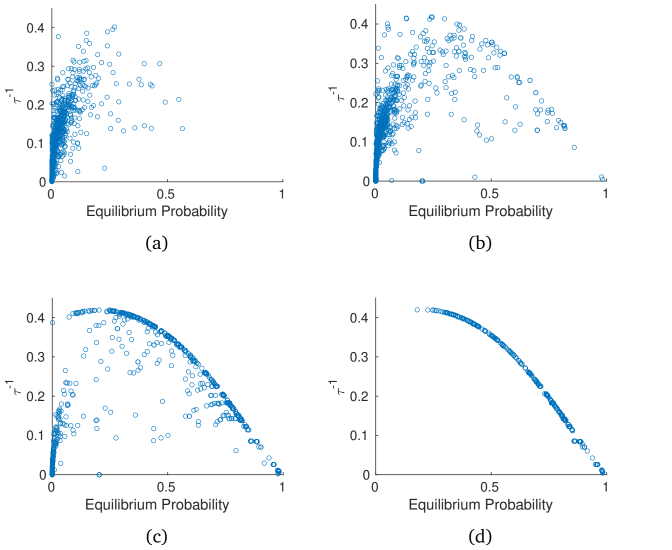
Note that $\sum_{k=1}^{M} T_k$ is simply the total number of time steps sampled. Since the rate of bond formation depends only on the underlying diffusion and not the particle interactions, these estimates only need to be performed once, to sufficient accuracy.

For each state i, we generated six trajectories in parallel using Brownian dynamics simulations, checking for bond formation at time chunks $\delta t = 0.01$. Each trajectory consisted of 16667 exit samples, for a total of about $100,000$ samples for each state. We use the six trajectories to compute error bars on each τ_i estimate. We repeat the sampling procedure until the relative error bars are less than 8% for each state i, which took us three to four applications of the algorithm.

This estimation can be excessively time consuming when using Brownian dynamics simulations, if there are a lot of clusters. We use Brownian dynamics simulations to generate trajectories in the case of $N = 6$ disks. For the larger systems, we further approximate the dynamics with the stratification sampler described in Section 1. Although this is a Monte Carlo sampler, it proposes small, local moves, which can move along any of a cluster's internal degrees of freedom, and therefore we expect it approximates overdamped dynamics for a small enough timestep (though this has yet to be shown rigorously.) This sampler can be modified to account for reflecting boundary conditions by removing proposal moves that jump to new constraint manifolds. Doing this will sample the equilibrium distribution of state i, and we denote a bond formation event when two particles are within a cutoff distance. The result is a sequence of configurations, but with no temporal information. To get approximate dynamical information from this scheme, we make use of the mean-square displacement property of a Brownian particle. For a one dimensional Brownian motion, the mean-square displacement is related to time via the formula $\langle (X(t) - X(0))^2 \rangle = 2Dt$, where D is the diffusion coefficient. For our proposal moves, we use an isotropic Gaussian over the tangent space of the current point on the manifold. If we choose the standard deviation, σ, of this proposal to be small, most moves will be accepted, and the mean-square displacement will be proportional to σ^2. We use $\sigma = 0.15$ in our simulations. Based on this, we introduce an approximate artificial time for each MC step as $\Delta t_{MC} = \sigma^2/2$. Any constant multiplicative factors that may be missing will multiply each estimated rate in the same way, meaning this method should approximate the rate matrix up to a multiplicative constant. We again generate six trajectories in parallel, each consisting of 16667 exit samples, and compute error bars across the trajectories.

4 Genetic Algorithm Details

The basics of our genetic algorithm were outlined in the main text. In addition to the cross-over operation, and re-sampling from mutation, we also implement a 'maximizing' mutation. The idea behind this is that the optimal design parameters typically consist of at least one interaction being as strong as possible. Thus, to improve the quality of solutions and convergence speed, we build this into a mutation. If a given energy, E is greater than some threshold



(a)

(b)

(c)

(d)

Figure 1 Time evolution of the population from an application of the genetic algorithm in identifying the Pareto front for the self assembly of a triangle from an ABABAB chain. The generations shown are 1, 7, 15, and 25, in (a)-(d), respectively. This run used $P = 1000$ members, $p = 50\%$ as the cutoff for the mating pool, $r = 10\%$ mutation chance, and $p_M = 40\%$ maximization chance with a threshold $E^* = 8$.

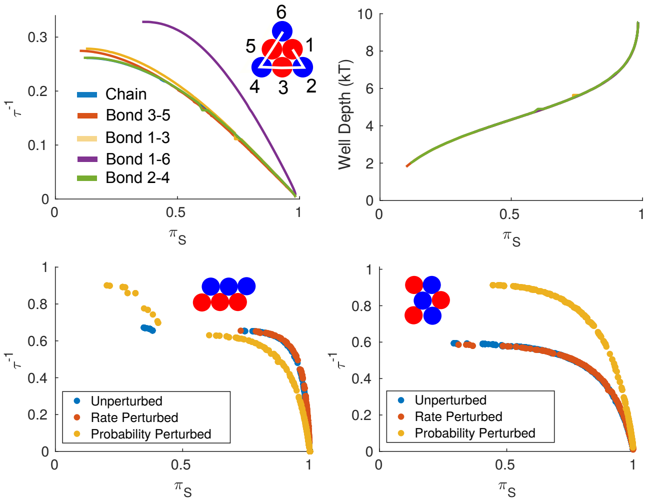


Figure 2 Top Row: Pareto fronts and non-trivial bond energy parameterizations for the triangle state with two particle types, for various 1-bond starting states. With the exception of the linear chain, all starting states are sampled from the equilibrium measure in that state. Bottom Row: Pareto fronts for the other six-disk clusters after perturbing model parameters. Exit rates out of coarse grained states are perturbed by adding Gaussian noise with $\sigma = 0.2/\tau_i$. Probabilities are perturbed by making each transition out of a state equally likely.

E^*, then with probability p_M, we set the energy to the maximum value, E_M. An example showing the convergence of the population to the Pareto front is shown in Figure 1.

Next, we discuss how we compute non-dominated points, and our dominated points metric. There are divide-and-conquer approaches, but we simply perform an $O(P^2)$ brute force method. For each point, i, we compare its objective values to each other point, j, and count how many points j have greater values in both objective functions, which we call D_i. We do this for each point, i, and then sort the population from least to greatest D_i. If a point has $D_i = 0$, this point is Pareto optimal for the current generation, and is carried over to the next generation with probability 1. The remainder of the population is filled by mating.

5 Sensitivity Analysis

In our analysis, we perturb the model kinetics in two ways to probe how sensitive the Pareto fronts are to the exact parameters of the model. One test is to add random noise to the exit rate of each state. To do this, for each state i, we set the perturbed exit rate to be $\hat{r}_i = \frac{1}{\tau_i} + \sigma N$, where τ_i is the mean first exit time out of state i, N is a standard normal random number, and σ is set to $0.2/\tau_i$, so typical perturbations in the rate are within about 20% of the estimated value. Our second procedure involves perturbing the transition probabilities. Instead of using the measured values, we impose a uniform distribution over the set of possible transitions, which is a drastic change.

In addition to perturbing model parameters, we also gauge the Pareto front's sensitivity to the assumed initial state. Instead of using the linear chain as the initial state, we repeat our calculations assuming the system starts in various 1-bonded configurations, drawn from the equilibrium distribution in that connectivity state. We chose some example configurations that were consistent with the target state, and others with mis-folded bonds.

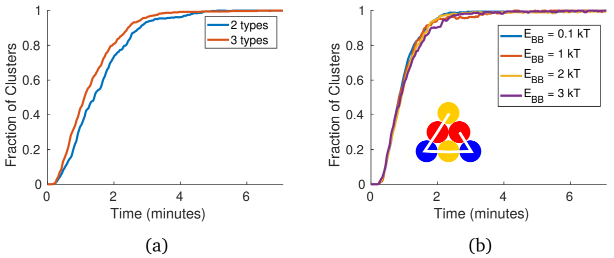
Figure 2 shows the results of these perturbation on the Pareto fronts for the system of six disks using an ABABAB chain. We find that perturbing the rates in the way we do has a marginal effect on the Pareto fronts. Perturbing the transition probabilities can have a large effect on τ^{-1} but the Pareto fronts still have the same qualitative behavior. The same is true of the choice of initial state; most of the Pareto fronts are very similar with the exception of starting in a loop, which assembles at a higher rate but with a qualitatively similar Pareto front. The bond energy parameterizations along the Pareto front are remarkably insensitive to all of the perturbations we tested.

6 Confirming Observations With Brownian Dynamics Simulations

We ran Brownian dynamics simulations to test some of the more interesting findings of our coarse-grained model, to see if they are indeed true or an artifact of our model assumptions or numerical algorithms.

Figure 6(b) shows that a vertical Pareto front is possible with both two and three particle types, but a higher rate is attainable with three types. To test this, we simulated the assembly of the chevron under the optimal interactions found for both two and three particle types. Figure 3(a) shows the fraction of clusters in the chevron state as a function of estimated physical time, and we see that there is indeed a slightly higher formation rate with three types. We estimate the mean first passage time to the chevron to be about 1.57 minutes for two types, and 1.31 minutes for three types, which is about 17% shorter when using three types. This is likely because the two type chain usually folds by wrapping around the lone B-type particle, proceeding one bond at a time, whereas the three type chain can form several sub-units simultaneously, speeding up the process.

Another interesting prediction was that the inclusion of some weak, auxiliary bonds not present in the target state could speed up formation of the target. As an example, we found a near vertical Pareto front for the six-disk triangle state using $m = 3$ par-



(a)

(b)

Figure 3 (a) Results of a Brownian dynamics simulation of 400 AAAAAB (2 particle types) and AAABCB (3 particle types) chains of 6 colloidal disks, initialized in a linear chain. The interactions were chosen in accordance with the optimal solution for the chevron computed by the genetic algorithm. (b) Results of a Brownian dynamics simulation of 400 ABCBAC chains of 6 colloidal disks, initialized in a linear chain. Interaction energies were chosen at various points along the Pareto front, with $E_{AA} = E_{AC} = 12$, and $E_{BB} \in \{0.1, 1, 2, 3\}$, with all others being set to 0.1. Result is conditioned on ending in the triangle, in order to compare rates. The fraction of clusters in each state is plotted as a function of physical time, computed by scaling the non-dimensional simulation time by an appropriate dimensional constant.

ticle types in the configuration ABCBAC. The optimal bond energies were found to be a large E_{AA} and E_{AC}, while the rest remain weak. All but one of these weak bonds remained constant along the front, with the one exception being E_{BB}, which parameterized the near vertical front, increasing in rate as E_{BB} was increased. We performed Brownian dynamics simulations using the optimal parameters and $E_{BB} \in \{0.1, 1, 2, 3\}$. As stated before, the yield of the triangle is only about 0.5, so in 3(b) we plot the fraction of clusters in the triangle state as a function of time, conditioned on ending up in the triangle state. We see that the value of E_{BB} from our test set has little to no effect on the rate of formation of the triangle, so we are left to conclude that in this case the vertical Pareto front is probably an artifact of our model.

7 Lattice Polymer

We consider a two-dimensional, square lattice model, where particles only interact if they are adjacent on the lattice. Each pair of particles, (i, j), in contact contribute an energy E_{ij}, depending on the types of the particles and model input parameters. The system is initialized in a linear chain on the x-axis, and an MCMC method is used to update the configuration. The proposal moves consist of an end move, rotating a particle at the beginning or end of the chain about its neighbor, and a corner move, in which a particle in a corner flips to the opposite corner, if possible. During each MCMC step, all valid moves for the chain are listed, and a proposal is selected uniformly at random from these possibilities. Proposal moves are accepted in accordance with the Metropolis-Hastings acceptance probability. Dynamical time is measured in MCMC steps.

We again form our coarse-grained model for this lattice system, lumping by adjacency matrix, and use the model to evaluate equilibrium probabilities and mean first passage times. Since these systems are highly degenerate, and often one is interested in a particular permutation, we choose individual permutations as the target state, instead of lumping them into a target set. We study the system with $N = 8$ particles on the lattice; the smallest system that can form geometrically distinct ground states. We choose one permutation of each of the two ground states, and apply our genetic algorithm to characterize their Pareto fronts.

8 Sampling for Larger Systems

Our goal will be to compute the parameterization of the previous Pareto fronts without relying on the coarse-grained model. We do this by taking a sampling approach to evaluate the competing measures for the genetic algorithm. An immediate issue presents itself; equilibrium simulations will take exceedingly long in the presence of kinetic traps, making both of our measures infeasible to compute in a reasonable amount of time.

One way to make the sampling computationally tractable is to introduce different functions for the objectives. If we can find functions that are correlated with the objectives, preserve the structure of the Pareto front, and are efficient to sample, then we can extract the optimal transformed parameterizations. There are many such functions that can be used, but here we consider just two that have shown promising results.

We replace π_S, the equilibrium probability of the target set, by a probability related to staying in the target state once it has formed, p_s. Each bond in the target state has an energy, E_i, for $i = 1, \cdots, b$, where b is the number of bonds. We compute the harmonic sum of these energies, $h(\vec{E}) = \left(\sum_{i=1}^{b} E_i^{-1}\right)^{-1}$, which gives a notion of an average energy barrier out of the target state. We then say the probability to stay in the target state is $p_s(\vec{E}) = 1 - \exp(-h(\vec{E}))$. Note that this quantity does not have to be sampled, it can be directly evaluated given the target state and input parameters. If all the bonds are weak, $p_s \to 0$, and if all the bonds are strong, $p_s \to 1$.

We replace our rate, τ^{-1}, with a measure related to the kinetic accessibility of a given target state, called k_A. To do so, we define two times, t_{trap} is the maximum amount of time to spend in one state until the system is considered "trapped", and $T > t_{\text{trap}}$ is the total simulation time. During a simulation, if the system spends longer than t_{trap} in a state, without breaking or forming any additional bonds, we stop and compute the *misfolded energy*, E_{misfold}. That is, we add up the energy of all bonds not consistent with the target state using the trapped configuration. If no trap states form, we set $E_{\text{misfold}} = 0$. We average this energy over many trajectories to compute $k_A = \exp(-\langle E_{\text{misfold}} \rangle)$. With this measure, if the system is able to easily escape from misfolded states, $k_A \to 1$, and if it gets stuck in deep kinetic traps, $k_A \to 0$.

Using the coarse-grained Markov model, we are able to evaluate these measures semi-analytically. There is a slight ambiguity in how we define a "trapped" state. Given a set of bond energies, $\vec{E}$, we construct the corresponding rate matrix and sort the diagonal entries in absolute value from least to greatest, which give the rates of exiting each state. Typically there is a large gap in this sorted list, so we consider all states before this gap "trapped", on a case-by-case basis. We can then evaluate our rate measure by taking a dot product of the misfolded energy vector with a hitting probability vector, which can be computed by solving a matrix equation similar to the linear system we solve for τ.

We revisit the $N = 8$ particle, 2-type, rectangular lattice protein from Figure 7(a), and study its behavior under the new measures. Figure 10(a) shows the Pareto front in the original measures (solid blue) and how it maps to the new measures (solid red). We see a large region where $k_A = 1$, which is due to a lack of kinetic traps, according to our above definition. There is a sharp transition to a region in which k_A continuously decreases as p_S increases, meaning the Pareto front is preserved. We then apply our genetic algorithm to the new measures, where we estimate k_A by sampling, using the optimal particle ordering AABABABB. We use a population of size 100, a limit of 200 generations, final time $T = 500$, a trap time $t_{\text{trap}} = 300$, a mating cutoff of the top $p = 25\%$, and 500 samples to estimate k_A for each population member. We do not observe convergence of the whole population, for reasons we will discuss, so we report only the non-dominated portion of the population, shown as unfilled red circles.

The sampled Pareto front is close to the analytically computed curve, but seems to give overestimates across most of the range. This is because of how we create the next generation. A non-dominated member of generation n is carried to generation $n+1$, without re-estimating its objective values. This has the effect of only keeping the maximum estimates of k_A for a given range of p_S values, since values of k_A closer to the true average will be dominated by the larger estimates.

Despite the over-estimate of the position of the actual Pareto fronts, the parameterizations of the front are unaffected by the statistical bias towards larger k_A. We extracted the parameters for each of the sampled points along the (p_S, k_A) Pareto front and used the coarse grained model to evaluate the corresponding values of (π_S, τ^{-1}), which are plotted as blue, unfilled circles. We find that these points mostly lay on the original Pareto front, confirming that we find the same parameterization. Some of the points lay on the π_S axis, with vanishingly small values of τ^{-1}. This is an issue with our choice of p_S; we can maximize p_S by making each of the bonds energies as large as possible, but doing so reduces π_S because of entropic considerations. We handle this issue by discarding population members that are trying to maximize all bond energies.

The above tests were performed using the already known optimal particle ordering. We also performed tests where the particle types are also selected by the genetic algorithm. In this case, the performance was significantly worse; we see no convergence and only a handful of points become non-dominated. We do note however, that the non-dominated points are the ones that find the optimal particle ordering. Based on this observation, we propose a two step procedure. An initial run can be used to determine possible candidates for the optimal ordering, and a secondary run can use these candidate orderings to search for a Pareto front.

As a final test, we apply the sampling approach directly to a larger problem in which enumeration would be expensive. For an $N = 16$ lattice protein, we set the target state to be a square and choose the permutation in which the backbone zig-zags up four units and then down four units repeatedly. An optimal configuration using $m = 8$ particle types is easy to identify by hand; group particles such that every two have the same type, i.e. AABBCCDDEEFFGGHH. Using the genetic algorithm, the minimum number of types required to achieve a value $k_A > 10^{-3}$ seems to be $m = 5$, with the configuration CDDBABBEBDABCBBD. We can again identify Pareto fronts, but they are less informative for this example; the presence of chiral traps begins to skew our k_A measure for this larger system. There are so many chiral traps in this case that even if distinct particles are used, more than 90% of trajectories will get stuck in a state that is not the target. To test whether the Pareto optimal parameters we've computed actually result in efficient assembly, we estimate the yield of the target states compared to the most common kinetic traps for each of the lattice structures we have studied. Figure 10(b) shows this yield comparison, using the parameters found by the genetic algorithm that give assembly comparable to distinct particles. We see that the presence of chiral traps result in a non-negligible reduction of the yields for $N = 8$, and the effect gets worse as the system size increases, highlighting the importance of specifying bond angles even more.

Notes and references

[1] M. Holmes-Cerfon, S. J. Gortler and M. P. Brenner, *Proceedings of the National Academy of Sciences*, 2013, **110**, E5–E14.

[2] Y. Kallus and M. Holmes-Cerfon, *Phys. Rev. E*, 2017, **95**, 022130.

[3] M. Holmes-Cerfon, *Journal of Chemical Physics*, 2020, **153**, 164112.

[4] G. Meng, N. Arkus, M. P. Brenner and V. N. Manoharan, *Science*, 2010, **327**, 560–563.

[5] A. Trubiano and M. Holmes-Cerfon, *Phys. Rev. E*, 2020, **101**, 042608.

[6] C. W. Gardiner, *Handbook of stochastic methods for physics, chemistry and the natural sciences*, Springer-Verlag, Berlin, 3rd edn, 2004, vol. 13, pp. xviii+415.

[7] R. W. Perry, M. C. Holmes-Cerfon, M. P. Brenner and V. N. Manoharan, *Phys. Rev. Lett.*, 2015, **114**, 228301.

[8] S. Barkley, A. Neophytou, A. Trubiano, E. D. Klein, D. Chakrabarti, M. Holmes-Cerfon and V. N. Manoharan, *In preparation*, 2021.

[9] G. Di Gesù, T. Lelièvre, D. Le Peutrec and B. Nectoux, *Faraday Discuss.*, 2016, **195**, 469–495.

[10] J. R. Norris, *Markov Chains*, Cambridge University Press, 1997.

[11] G. Guennebaud, B. Jacob *et al.*, *Eigen v3*, http://eigen.tuxfamily.org, 2010.