Two-Dimensional Clusters of Colloidal Spheres: Ground States, Excited States, and Structural Rearrangements

Supplemental Material

Rebecca W. Perry,[1] Miranda C. Holmes-Cerfon,[2]
Michael P. Brenner,[1] and Vinothan N. Manoharan[1,3,*]

[1] *School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*
[2] *Courant Institute of Mathematical Sciences, New York University, New York, New York 10012, USA*
[3] *Department of Physics, Harvard University, Cambridge, Massachusetts 02138, USA*

(Dated: May 19, 2015)

I. SAMPLE PREPARATION PROTOCOL

1. Prepare one small (22 mm×22 mm) and one large (24 mm×60 mm) glass coverslip (VWR Micro Cover Glasses, No. 1) by rinsing with deionized water, drying with high-purity compressed nitrogen, and plasma cleaning for 10 minutes in a PDC-32G Plasma Cleaner/Sterilizer (Harrick Plasma) with the RF Level set to High.

2. To make a sample chamber, center the small coverslip on the large coverslip and separate them with narrow strips of 30-µm-thick Mylar® A film parallel to the long edges of the large coverslip. With the two coverslips clamped together (e.g., with binder clips), use UV-curing Norland Optical Adhesive 61 and a UV lamp to seal the two edges of the small coverslip parallel to the spacers. We find that sealing the four corners and then removing the clips and sealing along the two edges works well.

3. Use a pipette to dispense well-dispersed colloidal suspension near one of the unsealed edges of the small coverslip and let capillary action fill the sample chamber. We use a microsphere volume fraction of 7.6×10^{-6}.

4. Use Devcon 5 Minute® Epoxy to seal the last two edges of the small coverslip and to go over the two previously sealed edges for extra protection.

* vnm@seas.harvard.edu

II. IMAGE ACQUISITION, PROCESSING, AND CLUSTER CONFIGURATION IDENTIFICATION

To collect images, we use a Nikon Eclipse TI-E inverted microscope with a Photon Focus camera, a CameraLink cable, and an Epix frame grabber connected to a desktop PC. We use a combination of a 60× water immersion objective (Nikon CFI Plan Apo VC, NA 1.2) and a 1.5× tube lens. We choose a slow frame rate of 3 frames/s to efficiently capture many transitions while still collecting a few frames during each transition. This frame rate is high enough to allow particle tracking as described below.

By establishing four clusters of six particles in the field of view (59 µm×59 µm), we can theoretically capture 4 h of cluster data from a typical hour-long experiment. In reality, 10 of our 44 clusters produced data for the entire duration of the data acquisition. The data series from the other 34 clusters were truncated during post-processing for one of four reasons: the cluster diffused to the edge of the frame (7 of 44); a particle permanently broke away from the cluster (7 of 44); the cluster came less than one particle diameter from merging with another cluster (7 of 44); or the particle locating or tracking algorithm failed because, for example, the optical system drifted out of focus (13 of 44). From 10.2 h of raw video, we were able to obtain 25.6 h of six-particle cluster time series out of a theoretical maximum of 40.7 h, a 63% recovery rate. While we do lose track of many of our clusters over time, this approach to data acquisition requires little supervision and produces twice as much usable data per hour as compared to watching over and tending to a single cluster.

Our post-processing routines are written in Python using the SciPy ecosystem [1]. We locate the particles, identify the clusters they belong to, and track the particles from frame to frame. To locate the particles, we first divide each image by a background image captured with no particles in the field of view to remove static artifacts. We then use the Crocker and Grier centroiding method [2] to locate the particles with better than 20 nm precision, as determined by tracking single particles diffusing in two dimensions at 500 frames/s, and then measuring the deviation from linearity of the mean-square displacements at the smallest lag times. After locating each of the particles, we identify the cluster that each particle belongs to by computing the distance to the four clusters' centers in the preceding frame and selecting the cluster with the shortest distance. We then subtract off the cluster's center of mass from each of the particle locations before linking them into trajectories solely using proximity between locations in consecutive images. Subtracting off the cluster center of mass reduces the apparent distance moved by the particles between frames by removing rigid-body translations. For our close-packed particles that occasionally diffuse distances greater than a full particle radius between frames, subtracting off the cluster center of mass prevents multiple particles from being linked to a single particle in the next frame. Alternative approaches to tracking a collection of close-packed particles include the optimization scheme of [2] and simply using strict proximity at a sufficiently high frame rate, where diffusing more than a particle radius between frames is extremely unlikely.

Once all the particles are found, assigned to clusters, and tracked, we determine the configuration of each cluster in each frame by computing the cluster's adjacency matrix (Fig. S1, [3]). The adjacency matrix uniquely determines the cluster

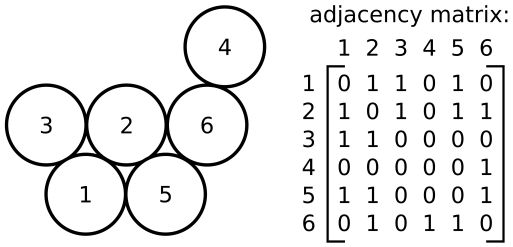

FIG. S1. An adjacency matrix is a representation of the connectivity of a cluster. Each element in the matrix relates a pair of particles identified by the row number and column number; a value of 1 signifies bound, and 0 signifies unbound. The adjacency matrix of the pictured eight-bond excited state is shown as an example.

configuration, including the particular permutation of particles, from our library of configurations with nine bonds, eight bonds, seven bonds, and "other" for clusters with fewer bonds. Such adjacency matrices do not distinguish between chiral enantiomers, which we pair together as single configurations. To determine when particles are bound or unbound, we set a cutoff distance of 1.4 µm, which is determined from the histogram in Fig. S2. We find that the occupation probabilities are insensitive to the choice of cutoff distance.

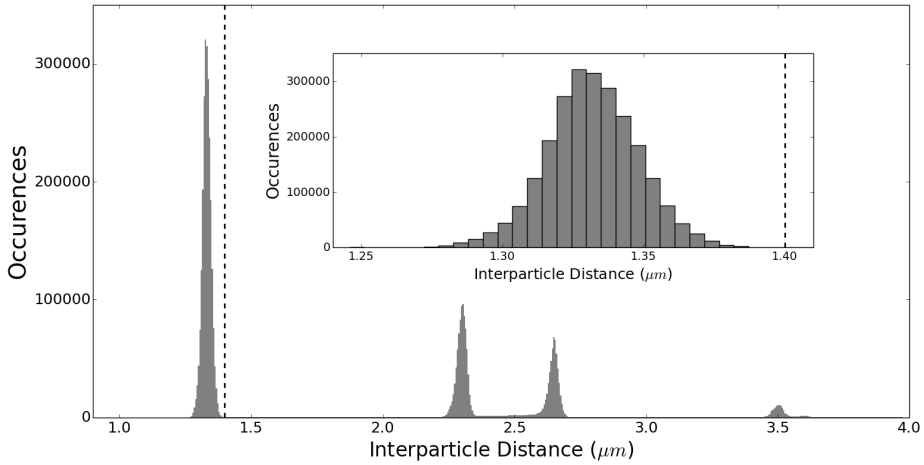

FIG. S2. Distances between all particles within all six-particle clusters at all times. The first peak represents bound particles at distance $a \approx 1.33$ µm. The other peaks are at $\sqrt{3}a$, $2a$, and $\sqrt{7}a$ as expected for close-packed spheres on a plane. The width of the peaks comes from a combination of the particle polydispersity, the width of the interaction potential, and the precision of the particle locating algorithm.

III. GROUND STATE PROBABILITY CALCULATION

Each of the macroscopic ground states—the parallelogram, chevron, and triangle—consists of many microscopic states, so we need to consider entropy in addition to energy in our probability calculations [4]. The probability of a macroscopic ground state s is given by the state's classical configurational integral, Z_s, normalized by the sum over all the ground states:

$$P_s = \frac{Z_s}{\sum_{s'} Z_{s'}}. \tag{S1}$$

Conveniently, Z_s may be split into approximately independent translational, rotational, and vibrational components in addition to the contribution from the potential energy: $Z_s = Z_{t,s} Z_{r,s} Z_{v,s} e^{-\beta U_s}$. The translational component is identical for each ground state because the area of the glass coverslip the clusters can explore is about seven orders of magnitude larger than the area of a cluster. Additionally, each of the ground states has nine identical bonds, so the potential energy contribution is also identical for each ground state. By canceling out these contributions, we arrive at a probability expression that depends only on the rotational and vibrational components:

$$P_s = \frac{Z_{t,s} Z_{r,s} Z_{v,s} e^{-\beta U_s}}{\sum_{s'} Z_{t,s'} Z_{r,s'} Z_{v,s'} e^{-\beta U'_s}} = \frac{Z_t e^{-\beta U} Z_{r,s} Z_{v,s}}{Z_t e^{-\beta U} \sum_{s'} Z_{r,s'} Z_{v,s'}} = \frac{Z_{r,s} Z_{v,s}}{\sum_{s'} Z_{r,s'} Z_{v,s'}}. \tag{S2}$$

The following calculations are for identical microspheres, so we normalize the masses, interparticle distances, and spring constants to unity.

The rotational component of the classical configurational integral in systems of identical colloidal clusters depends on the state's moment of inertia I_s, chirality χ_s, and symmetry number σ_s, which accounts for the effects of permutations [5]:

$$Z_{r,s} \propto \frac{\chi_s \sqrt{I_s}}{\sigma_s}. \tag{S3}$$

The moment of inertia is more generally the determinant of the moment of inertia tensor, but here the cluster has only one rotational axis. The chirality χ_s is 1 if the configuration is achiral and 2 if the configuration is a pair of chiral enantiomers.

To compute the vibrational contribution to the ground state probabilities, we use the harmonic approximation for the interparticle interactions. The vibrational contribution is inversely proportional to the product of the frequencies of the normal modes. There are $2N-3$ normal modes, since there are $2N$ degrees of freedom, and we have already removed two translational degrees of freedom and accounted for one rotational degree of freedom. The vibrational frequencies are given by the square root of the non-zero eigenvalues of the matrix $\boldsymbol{H_s}$, constructed from $N \times N$ super-elements. Each super-element is a 2×2 Hessian matrix describing the interactions between particles i and j [6]:

$$H_{ij} = \begin{bmatrix} \frac{\partial^2 U}{\partial x_i \partial x_j} & \frac{\partial^2 U}{\partial x_i \partial y_j} \\ \frac{\partial^2 U}{\partial x_j \partial y_i} & \frac{\partial^2 U}{\partial y_i \partial y_j} \end{bmatrix}. \tag{S4}$$

TABLE SI. Comparison of the components factoring into the probabilities of the three ground states for two-dimensional clusters of six particles.

	Parallelogram	Chevron	Triangle
$\sqrt{I_s}$	$\sqrt{5\frac{1}{2}}$	$\sqrt{4\frac{5}{6}}$	$\sqrt{5}$
χ_s	2	1	1
σ_s	2	1	3
$Z_{r,s}$	$\sqrt{5\frac{1}{2}}$	$\sqrt{4\frac{5}{6}}$	$\frac{1}{3}\sqrt{5}$
$Z_{v,s}$	$\frac{8}{27}\sqrt{\frac{2}{11}}$	$\frac{8}{27}\sqrt{\frac{6}{29}}$	$\frac{8}{27}\sqrt{\frac{1}{5}}$
Probability	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{1}{7}$

The eigenvalues $k_{\alpha,s}$ of $\boldsymbol{H_s}$ are the squares of the normal mode frequencies, which allow us to compute the vibrational contribution to the classical configuration integral:

$$Z_{v,s} \propto \prod_{\alpha=1}^{2N-3} \sqrt{\frac{1}{k_{\alpha,s}}}. \tag{S5}$$

This expression for the vibrational contribution is the last piece we need in order to use Eq. (S2) to calculate the probabilities of the parallelogram, chevron, and triangle. The results are presented in Table SI.

IV. OCCUPATION PROBABILITY ERROR BARS

The empirical occupation probability of each excited state is computed by taking the total amount of time we observe its adjacency matrix, and dividing by the total time spent in all configurations with identical energy. To estimate the error bar on this statistic we need to know the number of effectively independent samples. In general this is not the same as the number of data points, since the data are correlated in time: if a cluster has a particular adjacency matrix during one time step, it is more likely to remain in that adjacency matrix in subsequent time steps. After enough time steps, however, the data become decorrelated, and only then can new data be treated as independent. Roughly, the number of effectively independent samples is the length of the data, divided by the "correlation time" of the data.

A cluster can be thought of as a stochastic process $X_t \in \mathbb{R}^{2N}$, where X_t lists the positions of the particles. An adjacency matrix corresponds to a subset $A \subset \mathbb{R}^{2N}$ of configuration space. We would like to know the average amount of time the system spends in set A, which we write as $p_A = \mathbb{E}1_{(X_t \in A)}$.

Let's define a process $X_A(t) \equiv 1_{(X_t \in A)}$ to be the process that is 1 if $X(t) \in A$, and 0 otherwise. Then $p_A = \mathbb{E}X_A(t)$. Let $\hat{p}_A = \frac{1}{T}\int_0^T X_A(t)dt$ be an estimator for p_A. Let's suppose this estimator is Gaussian, i.e. $\hat{p}_A = p_A + \sigma_A z_A$, where σ_A is the standard deviation of the estimator, and $z_A \sim N(0,1)$ is a copy of the standard normal. Then, we can construct construct 95% error bars as $e = 1.96\sigma_A$.

How can we determine the standard deviation σ_A? If each observation were independent, then we would have $\sigma_A^2 = \frac{\sigma_{A,0}^2}{n}$, where $\sigma_{A,0}$ is the standard deviation of $X_A(t)$ at a single point in time (equal to $p_A(1-p_A)$ for our process since it's an indicator function), and n is the number of independent observations.

For a process that is correlated in time, a similar result holds provided we replace n with the number of "effectively" independent samples [7]. This is given by $n_{\mathit{eff}} = T/\tau$, where T is the total length of time of the sample, and τ is the correlation time. The correlation time is defined (for a stationary process) from the correlation function $C_A(t) \equiv \mathbb{E}X_A(s)X_A(s+t)$ to be

$$\tau = \frac{1}{C_A(0)} \int_{-\infty}^{\infty} C(t)dt. \tag{S6}$$

Geometrically, this comes from taking all the area under the correlation function and forming it into a rectangle with the same height as the covariance function at $t = 0$, so the width is τ. Note that $C_A(0) = \sigma_{A,0}^2$.

The estimate for σ_A^2 is then

$$\hat{\sigma}_A^2 = \frac{\sigma_{A,0}^2}{n_{\mathit{eff}}} = \frac{1}{T} \int_{-\infty}^{\infty} C_A(t)dt. \tag{S7}$$

We have used the fact that $\sigma_{A,0} = C_A(0)$ to rewrite the integral. This integral is calculated numerically from the data following the algorithm described in section B.

A. Conditional probabilities

The numbers we report in manuscript Figs. 1 and 2 are conditional probabilities: the probability of the cluster having a particular adjacency matrix, conditional on it having a given number of bonds. Calculating the variance of these conditional probabilities requires extra considerations.

Suppose we want to estimate the relative probability of being in set A, conditional on also being in a set B. That is, we would like to estimate $p_{A|B} = P(X(t) \in A | X(t) \in B) = \frac{P(X(t)\in A)}{P(X(t)\in B)} = \frac{\mathbb{E}1_{(X(t)\in A)}}{\mathbb{E}1_{(X(t)\in B)}}$. Let $X_B(t) = 1_{(X(t)\in B)}$. Then an estimator for $p_{A|B}$ is $\hat{p}_{A|B} = \frac{\hat{p}_A}{\hat{p}_B}$. When σ_i is small, this can be expanded as:

$$\frac{\hat{p}_A}{\hat{p}_B} = \frac{p_A + \sigma_A z_A}{p_B + \sigma_B z_B} = \frac{p_A}{p_B} + \frac{\sigma_A z_A}{p_B} - \frac{p_A \sigma_B z_B}{p_B^2} + O(\sigma_i^2).$$

The variance of this estimator for small σ_i is approximately

$$\mathrm{var}\left(\frac{\hat{p}_A}{\hat{p}_B}\right) = \frac{\sigma_A^2}{p_B^2} + \frac{p_A^2\sigma_B^2}{p_B^2} - \frac{2p_A\sigma_A\sigma_B\mathbb{E}z_A z_B}{p_B^2} = \frac{\sigma_A^2}{p_B^2} + \frac{p_A^2\sigma_B^2}{p_B^2} - \frac{2p_A\sigma_{AB}^2}{p_B^2}. \tag{S8}$$

We can estimate σ_A, σ_B as in the previous section. To compute the cross-correlation term $\sigma_A\sigma_B\mathbb{E}z_A z_B = \sigma_{AB}^2$, we compute the cross-correlation function $C_{AB}(t) = \mathbb{E}X_A(s)X_B(s+t)$ and determine the variance from this, as in the previous section.

B. How to compute the correlation time τ

The correlation function is very noisy at late times, so the integral to compute τ will also be very noisy. In fact, the bias as $n \to \infty$ is 0, but the variance is $O(1)$. Therefore that integral is not a consistent estimator of τ [7].

We use a windowing method to estimate τ, which integrates the correlation function up to a multiple W of the current estimate of τ. As is commonly done we set $W = 5$. Here is the method in pseudo-code:

$\hat{\rho}_t = C(t)/C(0)$
$\tau = 1$
$t = 1$
`while(`$\tau < Wt$`)` {
$\tau = \tau + 2\hat{\rho}_t$
$t = t + 1$
}

This produces an estimator whose variance goes to zero as the number of samples increases, but with a small bias of size $O(e^{-W})$ (if the covariance function has exponential tails.)

C. Why this works

Here is a brief explanation for Eq. (S7). The variance of $\hat{p}_A$ is

$$\begin{aligned}\left(\frac{1}{T}\int_0^T X_A(t)dt\right)\left(\frac{1}{T}\int_0^T X_A(s)ds\right) - p_A^2 &= \frac{1}{T^2}\int_0^T\int_0^T C_A(t-s)dt\,ds \\ &= \frac{1}{2T^2}\int_{-T}^{T}\int_{u}^{2T-u} C_A(u)dv\,du \\ &= \frac{1}{T}\int_{-T}^{T} C_A(u)(1-\frac{u}{T})du \\ &\approx \frac{1}{T}\int_{-\infty}^{\infty} C_A(u)du.\end{aligned}$$

The last approximation is valid when T is large enough that $C_A(u)$ has decayed.

V. MEASURING THE DIFFUSION COEFFICIENTS

To measure diffusion coefficients, we must first parameterize each of the one-dimensional transition paths between rigid clusters. A cluster can be written as a vector $x \in \mathbb{R}^{2N}$ listing the centers of each sphere in two dimensions. We find a path $x(s)$ depending on parameter s, such that

1. $\frac{dx}{ds}$ is perpendicular to infinitesimal rotations, infinitesimal translations, and motions that change the bond lengths

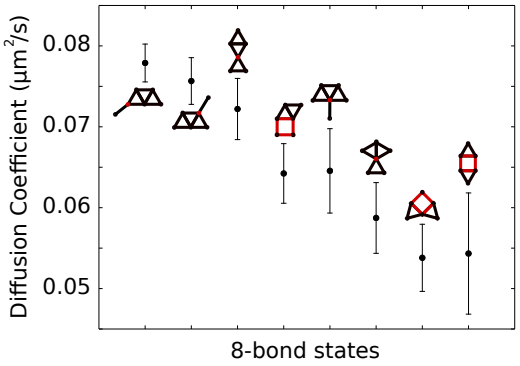

FIG. S3. Measured diffusion coefficients for the one-dimensional soft modes of the eight-bond states. Hinge-like joints and non-rigid squares are labeled in red. The error bars are 95% confidence intervals.

2. $|\frac{dx}{ds}| = 1$.

The first is possible because the space of rotations, translations, and bond lengths is $(2N-1)$-dimensional since there is exactly one bond "missing," so at each point along the path there is a one-dimensional tangent space spanned by unit vector t_s. The second is possible because the space we are parameterizing is one-dimensional, so we can always find an arc-length parameterization.

We store the path as a discrete set of clusters $x_{s_0}, x_{s_1}, \ldots, x_{s_m}$, where $s_k = k\Delta s$ for fixed step size Δs. Each $x_{s_{i+1}}$ is found from x_{s_i} by taking a step of size Δs in the direction of the unit tangent t_{s_i}, and then orthogonally projecting back to the manifold of constraints: $x_{s_{i+1}} = P(x_{s_i} + t_{s_i}\Delta s)$, where P is an orthogonal projection operator. The details of P are provided in Ref. [8].

We next analyze our data to obtain a time series of s-values along each transition path. For each data point with eight bonds, we find its corresponding s-value by first performing an orthogonal projection onto the transition path to remove the vibrational degrees of freedom. This projection step was crucial to obtaining good statistics. Then, we identify the closest cluster in the list $\{x_{s_0}, \ldots, x_{s_m}\}$, using a Euclidean metric on the space of sorted bond distances. Finally, for each pair of consecutive points that lie on the same transition path with s-values $\hat{s}_1, \hat{s}_2$, we compute the change in s-values $\Delta = \hat{s}_2 - \hat{s}_1$.

The result is a sequence of increments $\Delta_1, \Delta_2, \ldots, \Delta_M$ associated with each transition path. Close to the ends of the manifolds, the allowed sizes of steps taken towards the end become more and more restricted by the end of the manifold. To avoid biasing due to the non-Gaussian distributions of Δ near the ends of the manifolds, we only analyze steps towards the center of the manifold. Since the velocity correlation time is much shorter than the time between measurements, the cluster performs Brownian motion along the transition path, so the average diffusion coefficient along a path can be estimated as $D = a^2 \frac{1}{2\Delta t} \frac{1}{M} \sum_{i=1}^{M} (\Delta_i^2)$. Here Δt is the time between measurements, and the average is with respect to the stationary distribution along each path. The square of the interparticle spacing, a^2, is the conversion factor between diffusion in the parameterized space and in real-space.

The values we arrive at are between 0.05 and 0.08 $\mu m^2/s$ as shown in Fig. S3.

VI. TABLE OF Z_n FOR CLUSTERS WITH $N \leq 6$

To compute the sticky parameter, κ, we need to know the total geometrical partition function, Z_n, for manifolds with n bonds, for at least two different values of n. The "geometrical" partition function is the part which comes from integrating the rotational and vibrational partition functions; this is geometrical because it depends only on the locations, shapes, and sizes of the particles, and not on the potential energy or temperature.

The total geometrical partition function is

$$Z_n = \sum_i z_i^{(n)}, \tag{S9}$$

where $z_i^{(n)}$ is the geometrical partition function for a single manifold with n bonds, and the index i runs over all manifolds with n bonds. The geometrical partition function for a single manifold $\Omega_i^{(n)}$ is

$$z_i^{(n)} = \int_{\Omega_i^{(n)}} h_i^{(n)}(y) I_i^{(n)}(y) d\sigma_{\Omega_i^{(n)}}(y), \tag{S10}$$

where $d\sigma_{\Omega_i^{(n)}}(y)$ is the volume element on the manifold, $I_i^{(n)}(y)$ is the rotational partition function, and $h_i^{(n)}(y)$ is the "geometrical" part of the vibrational partition function. The latter equals $\prod_j \lambda_j^{-1/2}$, where λ_j are the non-zero eigenvalues of the Hessian of the potential energy, in the harmonic approximation with the spring constant set to 1.

To compute Eq. (S10) numerically, we parameterize each manifold and use a finite-element method to compute the integral. The supplemental information of Ref. [8] contains more details on how to compute the parameterization and volume element.

Table SII lists the numerically computed values of the total geometric partition function for the zero, one, and two-dimensional manifolds.

TABLE SII. The following geometrical partition functions are generated by applying the methods from Ref. [8] to 2D clusters. Note: clusters with a single disconnected sphere are not included in these calculations.

N	Z_{2N-3}	Z_{2N-4}	Z_{2N-5}
3	0.770	4.19	–
4	4.00	23.4	60.2
5	37.0	231	763
6	498	3320	11900

VII. REALTIME_TRANSITIONS.AVI

Video segments show the eight-bond transitions between ground states. The clusters transition from the ground state pictured on the left to the ground state pictured above. Connectivity diagrams label the excited state shown in each movie. The micrographs were divided by a background to remove static artifacts and scaled to create identical background intensities. We created this compilation using the Matplotlib library [9]. Video segments are played back at the recording rate of 3 frames/s.

VIII. 10XFAST_FOURCLUSTERS.AVI

This clip of 11 min (2000 frames) of raw data shows our experimental arrangement for simultaneously observing four clusters of six spheres while they rotate, translate, and rearrange. The clusters rearrange frequently, but rarely break apart. Playback is 10 times faster than recorded.

---

[1] E. Jones, T. Oliphant, P. Peterson, *et al.*, SciPy: Open source scientific tools for Python `http://www.scipy.org`.

[2] J. C. Crocker and D. G. Grier, J. Colloid Interface Sci. **179**, 298 (1996).

[3] N. Arkus, V. N. Manoharan, and M. P. Brenner, SIAM J. Discrete Math. **25**, 1860 (2011).

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

[5] M. K. Gilson and K. K. Irikura, J. Phys. Chem. B **114**, 16304 (2010).

[6] E. Eyal, L.-W. Yang, and I. Bahar, Bioinformatics **22**, 2619 (2006).

[7] A. Sokal, "Monte Carlo methods in statistical mechanics: Foundations and new algorithms," in *Functional Integration*, NATO ASI Series, Vol. 361, edited by C. DeWitt-Morette, P. Cartier, and A. Folacci (Springer, New York, 1997) pp. 131–192.

[8] M. Holmes-Cerfon, S. J. Gortler, and M. P. Brenner, Proc. Natl. Acad. Sci. U.S.A. **110**, E5 (2013).

[9] J. D. Hunter, Comput. Sci. Eng. **9**, 90 (2007).