Supplementary Information: Ensemble reweighting using Cryo-EM particle images

Wai Shing Tang,[†,‡] David Silva-Sánchez,[¶,†] Julian Giraldo-Barreto,[‡] Bob Carpenter,[†] Sonya M. Hanson,[†,‡] Alex H. Barnett,[†] Erik H. Thiede,[*,†] and Pilar Cossio[*,†,‡]

†*Center for Computational Mathematics, Flatiron Institute, New York, USA*

‡*Center for Computational Biology, Flatiron Institute, New York, USA*

¶*Department of Mathematics, Yale University, New Haven, CT, USA*

E-mail: *ehthiede@flatironinstitute.org; *pcossio@flatironinstitute.org

Supplementary Text

Assessing convergence of Markov chain Monte Carlo sampling

In the MCMC sampling described in the Methods, each of the 8 MCMC chains undergoes 1000 warmup steps, then 10,000 sampling steps to generate a sample of 80,000 draws of α. The convergence of the MCMC posterior samples is assessed by the potential scale reduction statistics $\widehat{R}$ and the effective sample size (ESS) N_{eff}.[1–3] For the chignolin system, we list these convergence parameters in Table S.6. $\widehat{R}$ was less than 1.01 for all MCMC runs, indicating convergence of the MCMC mean.[1,3] For all MCMC runs, the number of effective samples (N_{eff}) for most MC samples are larger than the number of draws (=80,000), except for a few samples for SNR= 10^{-4}, which shows that, except at high data noise, the MC samples are not strongly correlated and are converged.

Cryo-EM imaging model

For each structure extracted from the image-generating trajectory, we represent the protein's conformation x with the Cartesian coordinates of the C_α atoms. To generate a synthetic image of the molecule, we first place the center of mass of the molecule at the origin, then apply a random three-dimensional rotation R_ϕ drawn from the uniform distribution on SO(3). We place a 3D Gaussian of width $\sigma = 1.5$ Å at the position of each C_α atom, then sum the Gaussians over all atoms to obtain the biomolecule's electron density. Using $\vec{x}_n$ to denote the Cartesian position of the n'th C_α atom in the biomolecule prior to rotation, the electron density at any Cartesian position $\vec{r} = (r_i, r_j, r_k)$ is given by

$$\rho_{x,\phi}(\vec{r}) = (2\pi\sigma^2)^{-3/2} \sum_n e^{-||\vec{r} - R_\phi \vec{x}_n||^2/2\sigma^2} \ . \tag{S.1}$$

We then project along the k-direction, resulting in a 2D intensity map $I(r_i, r_j)$

$$I_{x,\phi}(r_i, r_j) = \int_{-\infty}^{\infty} \rho_{x,\phi}(r_i, r_j, r_k) dr_k \ . \tag{S.2}$$

This 2D intensity map is discretized to construct the image I. For the chignolin system presented in the Main Text, we employ a 2D grid of size $N_{\text{pix}} = 256 \times 256$ pixels with pixel width 0.15 Å. We then convolve the real-space point-spread function (PSF) with I to construct our noise-less, PSF-convolved image. The PSF is adopted from refs.[4,5] as

$$\text{PSF}_{\gamma,A,b}(r_i, r_j) = \mathcal{F}^{-1} \left[\text{CTF}_{\gamma,A}(\xi_i, \xi_j) \text{Env}_b(\xi_i, \xi_j) \right] \ . \tag{S.3}$$

where $\mathcal{F}^{-1}$ denotes the inverse of the two-dimensional Fourier-transform. CTF and Env are the contrast transfer and envelope functions, respectively, and they are given by

$$\text{CTF}_{\gamma,A}(\xi) = A \cos(\gamma \xi^2) - \sqrt{1 - A^2} \sin(\gamma \xi^2) \tag{S.4}$$

$$\text{Env}_b(\xi) = e^{-b\xi^2}. \tag{S.5}$$

The arguments ξ_i and ξ_j are the spatial frequencies in the horizontal and vertical direction of the image. A, b, γ are the imaging parameters: $A = 0.1$ is the amplitude-contrast ratio, $b = 1$ is the B-factor, and $\gamma = -\pi\Delta z\lambda_e$ where Δz is the defocus, and λ_e the electron wavelength. The defocus values are sampled randomly from Uniform$(0.027, 0.090)$. To construct the dataset of images, we then add Gaussian i.i.d. noise to each pixel with the appropriate SNR.

Rotational sampling

The rotations are sampled from a uniform quaternion distribution.[6] The initial step generates a four-vector with random values between -1 and 1 to uniformly fill the hypercube,. The hypercube is then culled into a hypersphere by removing any point with norm greater than one. As a precaution, in this step any point with a norm close to 0 is removed. This prevents bad numerics but has no impact on the final distribution. Then, each point in the remaining spherically distributed hypervolume is normalized to unit length to produce a quaternion point on the corresponding hypersphere. The code can be found at `https://github.com/flatironinstitute/Ensemble-reweighting-using-Cryo-EM-particles/blob/main/src/cryoER/imggen\_torch.py#L21-L36`.

Analyzing the Choice of Clustering Algorithm

In the main text, we apply K-medoids clustering method to cluster conformations from the MD trajectory. In this section, we describe an alternative clustering approach based on a Gaussian mixture model (GMM).[7] This clustering method requires as input the number of Gaussians centers M and the C_α positions of the trajectory structures $\{x_t\}$ to be clustered. The algorithm returns the Gaussian cluster centers $\{\chi_m\} \in \mathbb{R}^{3N_{\text{atom}}}$. While these cluster centers are not necessarily a subset of $\{x_t\}$, nor a physical structure, they are taken as representative positions to create the images with the same imaging model as described in Methods. To assess the results for different numbers of clusters, we use $M \in \{10, 20, 50, 100\}$.

Similar to the approach using K-medoid clustering, for each case, the cluster centers $\{\chi_m\}$ are used as input in the reweighting scheme with Eq. 15 to obtain the weight α_m associated with each χ_m. The results for the optimized weights obtained with GMM model are listed in Table S.7. For the high-to-moderate SNRs, clustering using the k-medoids algorithm performs slightly better compared to clustering using the GMM model. This could be because the k-medoids loss function is more closely related to the cluster-center approximation error. The k-medoids algorithm attempts to find clusters that minimize the sum of the distances between the cluster centers and the points in the cluster. The loss for the k-medoids algorithm we employ is given by

$$\mathcal{L} = \operatorname{argmin} \sum_{m=1}^{M} \sum_{x} \mathrm{RMSD}(x, \chi_m)\, \mathbb{1}_m(x)\,, \tag{S.6}$$

where RMSD denotes the root-mean-square deviation between two conformations, minimized over relative rotations is given by

$$\mathrm{RMSD}(x, \chi) = \sqrt{\frac{1}{N_{\mathrm{atom}}} \sum_{n=1}^{N_{\mathrm{atom}}} \|\vec{\chi}_n - g\vec{x}_n\|_2^2}\,. \tag{S.7}$$

Minimizing Eq. S.6 ensures that, on average, each conformation is not too dissimilar from the center of the cluster it is assigned to. Similar conformations produce similar images; in the Appendix of the Supplementary Information, we demonstrate that the RMSD difference between two conformations bounds the difference between their images. Consequently, clustering by minimizing the RMSD distance inside each cluster reduces the error when we approximate the likelihood of a conformation by the likelihood of its cluster center. In contrast, shape-GMM attempts to construct clusters that capture metastability in the protein landscape.[7] Consequently, dissimilar conformations can be assigned to the same cluster as long as they are inside the same metastable basin. Our results suggest that, when choosing a clustering method to use in our algorithm, it is important that all images produced by the

cluster are similar, even if this comes at the cost of having clusters that do a worse job of reflecting the intrinsic features of the protein conformational landscape.

Supplementary Tables

Table S.1: Toy model results with 3D normally-distributed data (shown in Figure 2) with 10 different initial weights randomly drawn from a 3-dimensional Dirichlet distribution. The true cluster population is related to the relative number of data points drawn from the normal distributions, which are centered at positions A, B, and C from Figure 2. The sampled weights $\{\alpha_t\}$ are shown for cases (i), with different ensemble members $\{x_t\}$. The estimated uncertainty is shown as the standard deviation in MCMC samples.

	Initial density			Reweighted density		
	x_t			x_t		
	A	B	C	A	B	C
# run						
0	0.351	0.437	0.212	0.500 ± 0.005	0.299 ± 0.004	0.200 ± 0.006
1	0.468	0.088	0.444	0.501 ± 0.004	0.299 ± 0.005	0.200 ± 0.002
2	0.235	0.206	0.559	0.500 ± 0.005	0.299 ± 0.005	0.200 ± 0.002
3	0.017	0.095	0.887	0.500 ± 0.002	0.299 ± 0.002	0.200 ± 0.000
4	0.180	0.188	0.631	0.500 ± 0.005	0.299 ± 0.005	0.200 ± 0.002
5	0.747	0.085	0.168	0.501 ± 0.002	0.299 ± 0.006	0.200 ± 0.005
6	0.634	0.138	0.228	0.501 ± 0.003	0.299 ± 0.006	0.200 ± 0.005
7	0.065	0.705	0.230	0.500 ± 0.003	0.299 ± 0.001	0.200 ± 0.002
8	0.056	0.187	0.757	0.500 ± 0.003	0.299 ± 0.003	0.200 ± 0.001
9	0.527	0.184	0.289	0.501 ± 0.004	0.299 ± 0.006	0.200 ± 0.004

Table S.2: Number of conformations in each of the metastable states (folded, misfolded, and unfolded) in the image-generating trajectory, structure-generating trajectory. In the rows below, we show the number of cluster centers for each metastable state of the structure-generating trajectory, using K-medoids and GMM clustering, respectively. A structure is classified to be folded or misfolded if it has a C_α root-mean-squared distance (RMSD) less than 1.2 Å from the reference structure of the folded or misfolded state, respectively. *In rare occasions, structures are classified to be both folded and misfolded if the RMSD threshold requirement is satisfied for both folded and misfolded (last column).

	# Total	# folded	# misfolded	# unfolded	# both folded and misfolded*
Image	106,949	82,424	47	24479	1
Structure	120,001	64,200	28,810	27,005	14
	M				
K-medoids[8,9]	10	3	2	5	0
	20	4	3	13	0
	50	10	8	32	0
	100	22	14	64	0
GMM[7]	10	5	3	3	1
	20	8	6	6	0
	50	24	10	16	0
	100	47	26	27	0

Table S.3: Average elapsed time for performing the K-medoids clustering. The K-medoids clustering calculation has two steps: computing the distance matrix (using the C_α RMSD) between all pairs of structures in the 120,000 set, and then partitioning of the 120,000 structures into $M = [10, 20, 50, 100]$ clusters. These calculations were performed using 1 skylake node with 20 tasks per node and 256 GB of RAM on the high-performance computing cluster at the Flatiron Institute. The total time is an average of 10 replicated runs on the same hardware setup. The standard deviation of the elapsed time over the 10 replicas is shown.

			Time
K-medoids	Distance-matrix calculation		$\sim$ 125 min
	Clustering	M	
		10	17.5 $\pm$ 4.1 s
		20	16.2 $\pm$ 3.6 s
		50	16.7 $\pm$ 2.0 s
		100	17.4 $\pm$ 2.3 s

Table S.4: Retrieved populations for the three metastable states of chignolin using K-medoids clustering. For each $M \in \{10, 20, 50, 100\}$, a set of M cluster centers χ_m are extracted from the structure-generating trajectory, then these are reweighted against 106,949 synthetic images generated from random conformations from the image-generating trajectory. Ground truth populations of the folded and misfolded states are determined by counting the number of folded and misfolded conformations among the 106,949 frames drawn from the image-generating trajectory. The retrieved population is the weighted sum of the structures belonging to each state (see Main Text). The error bars shown here are the standard deviation of the estimates calculated using the MCMC samples.

	%folded	%misfolded	%unfolded
Original	0.5350	0.2401	0.2250
Ground	0.7707	0.0004	0.2289
SNR	M = 10		
∞ (No noise)	0.762 ± 0.002	0.0120 ± 0.0004	0.226 ± 0.002
1	0.762 ± 0.002	0.0128 ± 0.0004	0.225 ± 0.002
10^{-1}	0.762 ± 0.002	0.0118 ± 0.0004	0.226 ± 0.002
10^{-2}	0.764 ± 0.002	0.0121 ± 0.0004	0.224 ± 0.002
10^{-3}	0.779 ± 0.002	0.0154 ± 0.0005	0.206 ± 0.002
10^{-4}	0.722 ± 0.003	0.054 ± 0.002	0.225 ± 0.002
SNR	M = 20		
∞ (No noise)	0.764 ± 0.002	0.0067 ± 0.0003	0.230 ± 0.002
1	0.764 ± 0.002	0.0072 ± 0.0003	0.229 ± 0.002
10^{-1}	0.765 ± 0.002	0.0066 ± 0.0003	0.229 ± 0.002
10^{-2}	0.766 ± 0.002	0.0065 ± 0.0003	0.228 ± 0.002
10^{-3}	0.776 ± 0.002	0.0099 ± 0.0004	0.214 ± 0.002
10^{-4}	0.698 ± 0.003	0.038 ± 0.002	0.264 ± 0.003
SNR	M = 50		
∞ (No noise)	0.761 ± 0.002	0.0040 ± 0.0002	0.235 ± 0.002
1	0.761 ± 0.002	0.0041 ± 0.0002	0.235 ± 0.002
10^{-1}	0.761 ± 0.002	0.0040 ± 0.0002	0.235 ± 0.002
10^{-2}	0.762 ± 0.002	0.0041 ± 0.0002	0.234 ± 0.002
10^{-3}	0.770 ± 0.002	0.0069 ± 0.0004	0.223 ± 0.002
10^{-4}	0.685 ± 0.003	0.032 ± 0.002	0.283 ± 0.003
SNR	M = 100		
∞ (No noise)	0.768 ± 0.002	0.0030 ± 0.0002	0.229 ± 0.002
1	0.768 ± 0.002	0.0031 ± 0.0002	0.229 ± 0.002
10^{-1}	0.768 ± 0.002	0.0030 ± 0.0002	0.229 ± 0.002
10^{-2}	0.768 ± 0.002	0.0029 ± 0.0002	0.229 ± 0.002
10^{-3}	0.771 ± 0.002	0.0056 ± 0.0003	0.223 ± 0.002
10^{-4}	0.673 ± 0.003	0.024 ± 0.002	0.303 ± 0.003

Table S.5: Retrieved populations for the three metastable states of chignolin using K-medoids clustering for lower SNR (down to 10^{-10}). A set of $M = 20$ cluster centers χ_m are extracted from the structure-generating trajectory, then these are reweighted against 106,949 synthetic images generated from random conformations from the image-generating trajectory. Ground truth populations of the folded and misfolded states are determined by counting the number of folded and misfolded conformations among the 106,949 frames drawn from the image-generating trajectory. The retrieved population is the weighted sum of the structures belonging to each state (see Main Text). The error bars shown here are the standard deviation of the estimates calculated using the MCMC samples.

	%folded	%misfolded	%unfolded
Original	0.5350	0.2401	0.2250
Ground	0.7707	0.0004	0.2289
SNR	M = 20		
∞ (No noise)	0.764 ± 0.002	0.0067 ± 0.0003	0.230 ± 0.002
1	0.764 ± 0.002	0.0072 ± 0.0003	0.229 ± 0.002
10^{-1}	0.765 ± 0.002	0.0066 ± 0.0003	0.229 ± 0.002
10^{-2}	0.766 ± 0.002	0.0065 ± 0.0003	0.228 ± 0.002
10^{-3}	0.776 ± 0.002	0.0099 ± 0.0004	0.214 ± 0.002
10^{-4}	0.698 ± 0.003	0.038 ± 0.002	0.264 ± 0.003
10^{-5}	0.604 ± 0.009	0.059 ± 0.007	0.337 ± 0.008
10^{-6}	0.52 ± 0.03	0.08 ± 0.02	0.40 ± 0.02
10^{-7}	0.67 ± 0.06	0.09 ± 0.05	0.24 ± 0.05
10^{-8}	0.61 ± 0.12	0.14 ± 0.08	0.24 ± 0.09
10^{-9}	0.44 ± 0.16	0.29 ± 0.13	0.27 ± 0.10
10^{-10}	0.51 ± 0.16	0.25 ± 0.13	0.24 ± 0.09

Table S.6: Convergence diagnostics on the MCMC sampling of weights $\{\alpha_m\}$ for cluster centers $\{\chi_m\}$ obtained by K-medoids clustering. The minimum (min), median, and maximum (max), among clusters $m = \{1, \ldots, M\}$, effective sample size N_{eff} and the potential scale reduction $\widehat{R}$ of MCMC samples of all 8 MCMC chains are shown here.

SNR	N_{eff}			$\widehat{R}$		
	$\min_m$	median_m	$\max_m$	$\min_m$	median_m	$\max_m$
M = 10						
∞ (No noise)	104023	137789	151466	1.00002	1.00008	1.00030
1	113011	134873	148202	1.00005	1.00010	1.00020
10^{-1}	110079	130969	142335	1.00002	1.00009	1.00021
10^{-2}	113480	136938	148697	0.99997	1.00014	1.00042
10^{-3}	118112	129900	139821	0.99996	1.00005	1.00016
10^{-4}	50284	90082	138733	1.00001	1.00004	1.00013
M = 20						
∞ (No noise)	100803	145329	171713	0.99999	1.00009	1.00022
1	109791	156537	182861	0.99998	1.00010	1.00027
10^{-1}	113409	155921	188123	0.99998	1.00010	1.00028
10^{-2}	111199	145507	182898	0.99996	1.00010	1.00020
10^{-3}	144589	163038	189415	1.00003	1.00010	1.00016
10^{-4}	36431	122390	146275	1.00000	1.00010	1.00028
M = 50						
∞ (No noise)	82341	110318	121015	0.99997	1.00007	1.00021
1	66881	99234	134445	0.99999	1.00010	1.00035
10^{-1}	61624	100576	124871	0.99997	1.00007	1.00025
10^{-2}	67254	109200	150633	0.99998	1.00008	1.00034
10^{-3}	56298	154431	163432	0.99997	1.00007	1.00025
10^{-4}	29369	123415	146352	0.99998	1.00012	1.00026
M = 100						
∞ (No noise)	81643	184892	205218	0.99998	1.00013	1.00039
1	88270	157726	170779	0.99997	1.00011	1.00049
10^{-1}	83211	180093	198749	0.99997	1.00012	1.00044
10^{-2}	86093	131685	141651	1.00000	1.00009	1.00038
10^{-3}	32584	142694	165413	0.99998	1.00010	1.00064
10^{-4}	13351	69552	133868	0.99997	1.00008	1.00036

Table S.7: Retrieved populations for the three metastable states of chignolin using GMM clustering.[7] For each $M \in \{10, 20, 50, 100\}$, a set of M cluster centers χ_m are extracted from the structure-generating trajectory (described in Supplementary Text), then these are reweighted against 106,949 synthetic images generated from random conformations from the image-generating trajectory. The retrieved population is the weighted sum of the structures belonging to each state (see the Main Text). The standard deviation of the estimate for the MC samples is shown.

	%folded	%misfolded	%unfolded
Original	0.5350	0.2401	0.2250
Ground	0.7707	0.0004	0.2289
SNR	M = 10		
∞ (No noise)	0.805 ± 0.002	0.0210 ± 0.0005	0.174 ± 0.002
1	0.805 ± 0.002	0.0211 ± 0.0005	0.174 ± 0.002
10^{-1}	0.806 ± 0.002	0.0208 ± 0.0005	0.173 ± 0.002
10^{-2}	0.807 ± 0.002	0.0204 ± 0.0005	0.173 ± 0.002
10^{-3}	0.821 ± 0.002	0.0229 ± 0.0006	0.156 ± 0.002
10^{-4}	0.776 ± 0.003	0.058 ± 0.002	0.167 ± 0.003
SNR	M = 20		
∞ (No noise)	0.794 ± 0.002	0.0154 ± 0.0004	0.191 ± 0.002
1	0.794 ± 0.002	0.0152 ± 0.0004	0.191 ± 0.002
10^{-1}	0.794 ± 0.002	0.0153 ± 0.0004	0.191 ± 0.002
10^{-2}	0.796 ± 0.002	0.0153 ± 0.0004	0.189 ± 0.002
10^{-3}	0.812 ± 0.002	0.0170 ± 0.0005	0.171 ± 0.002
10^{-4}	0.759 ± 0.003	0.046 ± 0.002	0.195 ± 0.003
SNR	M = 50		
∞ (No noise)	0.779 ± 0.002	0.0068 ± 0.0003	0.214 ± 0.002
1	0.779 ± 0.002	0.0073 ± 0.0003	0.214 ± 0.002
10^{-1}	0.779 ± 0.002	0.0068 ± 0.0003	0.215 ± 0.002
10^{-2}	0.779 ± 0.002	0.0069 ± 0.0003	0.214 ± 0.002
10^{-3}	0.789 ± 0.002	0.0095 ± 0.0004	0.202 ± 0.002
10^{-4}	0.715 ± 0.004	0.037 ± 0.002	0.248 ± 0.004
SNR	M = 100		
∞ (No noise)	0.778 ± 0.002	0.0052 ± 0.0003	0.217 ± 0.002
1	0.778 ± 0.002	0.0054 ± 0.0003	0.217 ± 0.002
10^{-1}	0.778 ± 0.002	0.0055 ± 0.0003	0.217 ± 0.002
10^{-2}	0.779 ± 0.002	0.0050 ± 0.0003	0.216 ± 0.002
10^{-3}	0.788 ± 0.002	0.0073 ± 0.0004	0.205 ± 0.002
10^{-4}	0.698 ± 0.004	0.032 ± 0.002	0.270 ± 0.004

Table S.8: Average elapsed time for generating the images, performing the structure-image comparison, and the MCMC sampling. The structure-image calculation compares M structures to N images, which are separated into 10 batches such that each batch of images can fully fit into the GPU memory. Image-generation and structure-image comparison are performed on one Nvidia GPU A100 using CUDA version 11.4.4 with 128 GB of RAM. GPU calculations are written with PyTorch (version 1.12.1) with Python 3 (version 3.9.12). The MCMC sampling, described in the Method section, has 8 MCMC chains, 1000 warm-up steps, and 10,000 sampling steps. The MCMC smapling used cmdstanpy (version 1.0.8) and stanc (version 2.30.1) on one CPU node with AMD's high-performance microprocessors, 120 tasks per node (with 15 threads per chain) and 256 GB of RAM. All calculations were performed on the high-performance computing cluster at the Flatiron Institute. The total time is an average of 10 replicated runs on the same hardware setup. The standard deviation of the elapsed time over the 10 replicas is shown.

	Time	
Image generation	5.3 ± 0.3 ms (per image)	
Structure-image comparison	M	N = 100000
	10	154 ± 2
	20	301 ± 1
	50	749 ± 2
	100	1505 ± 2
	N	M = 20
	1000	3 ± 1
	2000	7 ± 1
	5000	16 ± 2
	10000	29 ± 2
	20000	62 ± 2
	50000	155 ± 4
	100000	301 ± 1
MCMC sampling	10	369 ± 2 s
	20	649 ± 9 s
	50	1980 ± 97 s
	100	5313 ± 74 s

Supplementary Figures

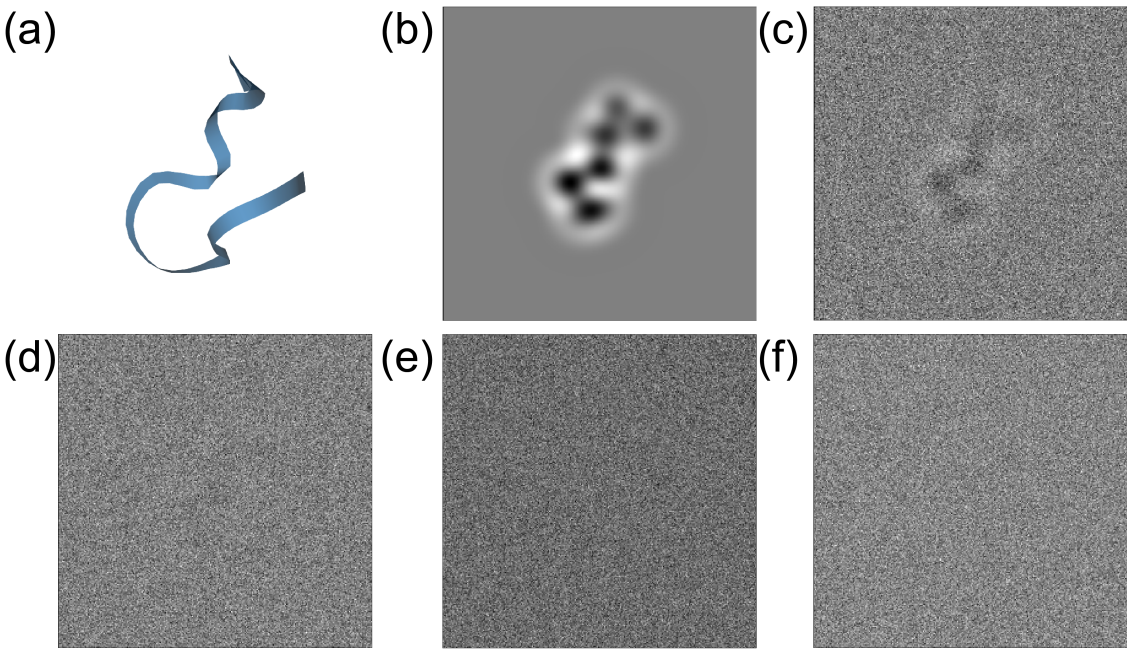

Figure S.1: Examples of synthetic cryo-EM images at different levels of noise. (a) Ribbon representation of the chignolin structure used to generate the example images shown here. (b)-(e) synthetic cryo-EM images generated with the imaging model described in the Methods with different levels of noise, *i.e.*, for image (b)-(e), SNR $= \infty$ (No noise), 1, 0.1, 10^{-2}, 10^{-3}, respectively.

Figure S.2: Example of the cryo-EM likelihood (Main Text Eq. 17), for one image, visualized in ortographic projection over the rotation sphere (ϕ) using 4608 quaternions uniformly distributed in $SO(3)$[10] and plotted using healpy.[11] The optimal pose (*i.e.*, projection direction) has a highly peaked likelihood (yellow point). The likelihood shown here is normalized over all angular bins.

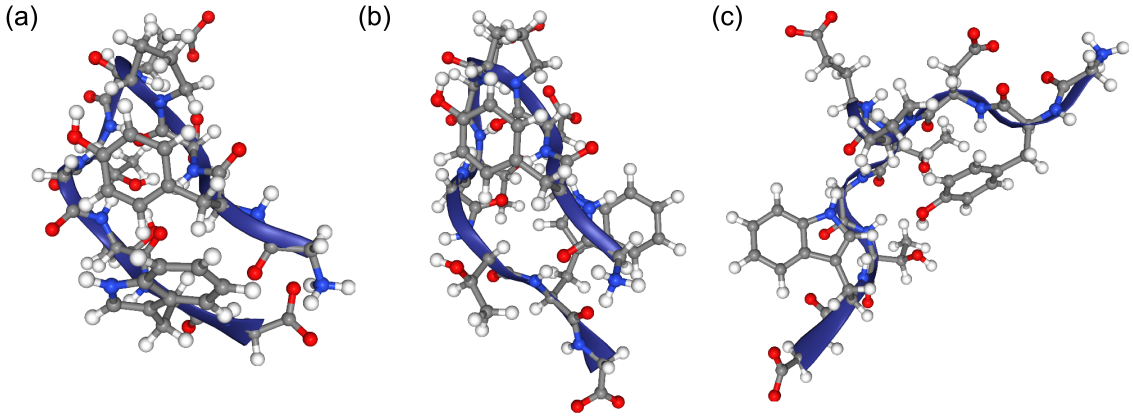

Figure S.3: Representative structures used for defining the (a) folded and (b) misfolded states, respectively, the unfolded center is also shown here but not used in any of the analyses. These representative structures are obtained by clustering the 120,000 structures from the structure-generating trajectory into 3 clusters using K-medoids. (see Methods for details).

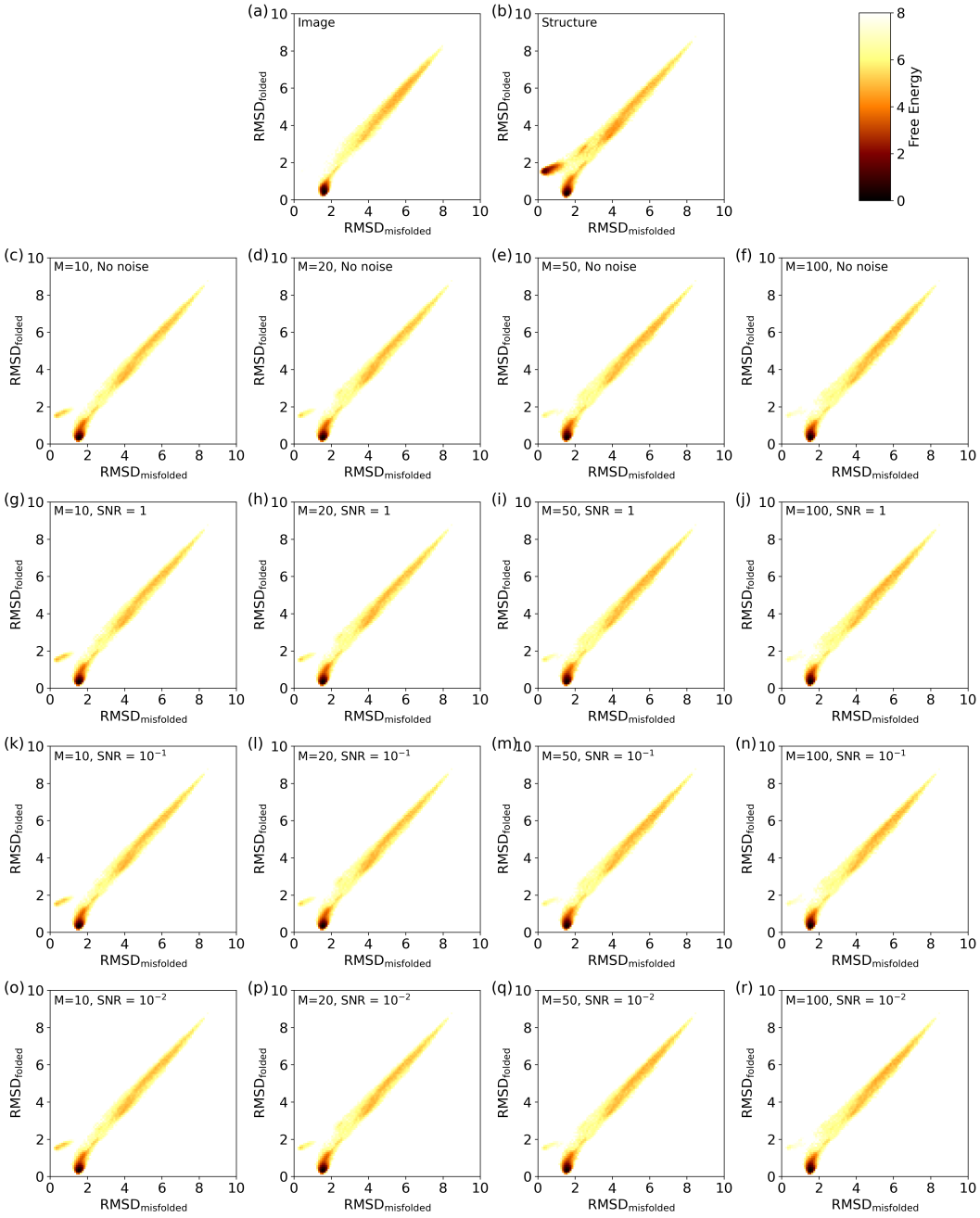

Figure S.4: 2D free-energy surface as a function of the C_α RMSD with respect to the misfolded structure (x-axis), and of the C_α RMSD with respect to the folded structure (y-axis) for (a) 106,000 frames from the image-generating trajectory, (b) 120,000 frames from the structure-generating trajectory, (c)-(r) 120,000 frames from the structure-generating trajectory reweighted by approximating the ensemble density on clusters (section 2.2.2). The ensemble weights α extracted for $M = [10, 20, 50, 100]$ K-medoids cluster centers (from left to right) are distributed to their respective cluster members from the 120,000 structure-generating trajectories. Reweighting is performed using the image set at different noise levels, *i.e.*, SNR$= [\infty, 1, 0.1, 10^{-2}]$ (from top to bottom).

Appendix

Bounding the difference between two images with the minimum RMSD between their conformations

Our intuition regarding the success of the RMSD clustering is based on the intuition that conformations that are close in RMSD produce similar images. Here, we formalize this intuition by showing that the maximum pixel-wise square difference between two images, minimized over relative rotations, can be bounded by the minimum RMSD between the two conformations.

To start, we examine our imaging model more closely. Recall that under our imaging model, the image of our molecule in conformation x imaged at an angle ϕ and with microscope parameters θ is given by Main Text Eq. 16. We will assume that electron density ρ is written as a sum of contributions from each atom, represented by a kernel function k whose value depends only on the distance from the atom center,

$$\rho(\vec{r}) = \sum_{n=1}^{N_{\text{atom}}} k(\vec{r} - \vec{x}_n) \ , \tag{S.8}$$

where $\vec{r} = (r_i, r_j, r_k)$ is the position vector of the pixel, and $\vec{x}_n$ is the position vector of the nth atom. In this work, we use a Gaussian kernel function centered at the C_α atoms. However, more general kernel functions can be used[12] as long as they obey $\int \left(k(\|\vec{r}\|)\right)^2 d\vec{r} < \infty$ and are isotropic to rotations.

First, without loss of generality we can set $\phi = 0$ (*i.e.* we assume that the molecule is imaged from above). We define the projected kernel,

$$\tilde{k}(\bar{r}) = P_z k(\|\vec{r}\|) = \int k(\|\vec{r}\|) dr_k \ . \tag{S.9}$$

where $\bar{r} = (r_i, r_j)$ is the vector in $\mathbb{R}^2$ that represents the projection of the vector $\vec{r}$ onto the

horizontal plane. Hereafter, we will use $\vec{\ }$ to denote vectors in $\mathbb{R}^3$ and $\bar{\ }$ to denote vectors in $\mathbb{R}^2$. For the Gaussian kernel used in our work, the projected kernel is a two-dimensional Gaussian, with the same variance as the Gaussian in three-dimensional space.

Substituting Eq. S.8 into Main Text Eq. 16, using the fact that convolution is linear, and applying the projections to each kernel function, we obtain

$$\mathrm{Img}_{\theta,\phi,x}(\bar{r}) = \sum_{n=1}^{N_{\mathrm{atom}}} \mathrm{PSF}_\theta * \tilde{k}(\|\bar{r} - P_z(\vec{x}_n)\|) \ . \tag{S.10}$$

Equation S.10 suggests a proof strategy. First, we show that if the PSF changes sufficiently slowly, we can bound the distance between the two images by the distance between the positions of the atoms projected onto the plane of the image. Then, we show that the total distance between the projected atoms is bounded by the minimum RMSD between the two conformations.

To show the first half of the proof, we will first require the following lemma.

Lemma 1 (Convolving Shifted Functions Against a Lipschitz Filter)**.** *Let $f : \mathbb{R}^d \to \mathbb{R}$ be a square-integrable function obeying the Lipschitz condition*

$$|f(a) - f(b)| \leq L\|a - b\|_2 \tag{S.11}$$

for all $a, b \in \mathbb{R}^d$ and for a constant $L \in \mathbb{R}$. Here $\|\cdot\|_2$ denotes the L2 norm, and consequently $\|a - b\|_2$ is the distance between points a and b. Then, for all square integrable functions $g : \mathbb{R}^d \to \mathbb{R}$,

$$|(f * g)(a) - (f * g)(b)| \leq L'\|a - b\|_2 \tag{S.12}$$

where $*$ denotes n-dimensional convolution and $L' = L \int |g(r)|dr$.

Proof. Substituting in the definition of the convolution, we can rewrite the convolution as

$$\begin{aligned}\left| \int f(a-s)g(s)ds - \int f(b-s)g(s)ds \right| &= \left| \int \left(f(a-s) - f(b-s)\right) g(s)ds \right| \\ &\leq \int \left|f(a-s) - f(b-s)\right| \left|g(s)\right| ds \\ &\leq \int L \left\|(a-s) - (b-s)\right\|_2 \left|g(s)\right| ds \\ &= L' \left\|a - b\right\|_2 \ .\end{aligned}$$

completing the proof.

□

Equipped with this lemma, we can proceed with our main theorem.

Theorem 2 (Bounding the difference in images with the minimum RMSD distance between configurations)**.** *Assume that the point-spread function* PSF_θ *obeys the Lipschitz condition specified in Lemma 1 with constant* L*. Then, for all possible pairs of configurations* $x, \chi \in \mathbb{R}^{3N_{\mathrm{atom}}}$*, and viewing angles*

$$\min_{g \in SO(3)} \max_{\bar{r}} \left| \mathrm{Img}_{\theta,\phi,\chi}(\bar{r}) - \mathrm{Img}_{\theta,\phi,x}(\bar{r}) \right| \leq C \, \mathrm{RMSD}(\chi, x) \ . \tag{S.13}$$

where C *is a fixed constant and* RMSD *denotes the root-mean-square distance between the atoms in two conformations, minimized over relative rotations.*

Proof. We begin this proof by replacing the rotation g with the rotation than minimizes the

RMSD, which we denote g^{RMSD},

$$\begin{aligned}
&\min_{g\in SO(3)} \max_{\bar{r}} \left|\mathrm{Img}_{\theta,\phi,\chi}(\bar{r}) - \mathrm{Img}_{\theta,\phi,x}(\bar{r})\right| \\
&\qquad = \min_{g\in SO(3)} \max_{\bar{r}} \left| \sum_{n=1}^{N_{\mathrm{atom}}} \Big(\mathrm{PSF}_\theta * \tilde{k}(\bar{r} - P_z(\vec{\chi}_n)) - \mathrm{PSF}_\theta * \tilde{k}(\bar{r} - P_z(g\vec{x}_n)) \Big) \right| \\
&\qquad \leq \max_{\bar{r}} \left| \sum_{n=1}^{N_{\mathrm{atom}}} \Big(\mathrm{PSF}_\theta * \tilde{k}(\bar{r} - P_z(\vec{\chi}_n)) - \mathrm{PSF}_\theta * \tilde{k}\big(\bar{r} - P_z(g^{\mathrm{RMSD}}\vec{x}_n)\big) \Big) \right| \\
&\qquad \leq \max_{\bar{r}} \sum_{n=1}^{N_{\mathrm{atom}}} \left| \mathrm{PSF}_\theta * \tilde{k}(\bar{r} - P_z(\vec{\chi}_n)) - \mathrm{PSF}_\theta * \tilde{k}\big(\bar{r} - P_z(g^{\mathrm{RMSD}}\vec{x}_n)\big) \right| \ ,
\end{aligned}$$

where the last line follows from the triangle inequality. Here $\vec{x}_n$ and $\vec{\chi}_n$ denote the nth atoms in the configurations x and χ, respectively. Next, we assume that the PSF obeys the Lipschitz condition specified in Lemma 1 with constant L (see Lemma 3). We can then write

$$\min_{g\in SO(3)} \max_{\bar{r}} \left|\mathrm{Img}_{\theta,\phi,\chi}(\bar{r}) - \mathrm{Img}_{\theta,\phi,x}(\bar{r})\right| \leq \sum_{n=1}^{N_{\mathrm{atom}}} \left(L \int |\tilde{k}(r)| dr \right) \left\| P_z(\vec{\chi}_n) - P_z(g^{\mathrm{RMSD}}\vec{x}_n) \right\|_2 \ .$$

Moreover, projecting points onto a plane always brings them closer together: $\|P_z a - P_z b\|_2 \leq \|a - b\|_2$. Consequently,

$$\min_{g\in SO(3)} \max_{\bar{r}} \left|\mathrm{Img}_{\theta,\phi,\chi}(\bar{r}) - \mathrm{Img}_{\theta,\phi,x}(\bar{r})\right| \leq \left(L N_{\mathrm{atom}} \int |\tilde{k}(r)| dr \right) \sum_{n=1}^{N_{\mathrm{atom}}} \left\| \vec{\chi}_n - g^{\mathrm{RMSD}}\vec{x}_n \right\|_2 \ .$$

Finally, we observe that the right-hand-side is the l_1 norm of the vector of distances between atoms, whereas the RMSD distance is the l_2 norm of the same vector, scaled by ${N_{\mathrm{atom}}}^{-1}$. Using the inequality that for a vector v, of length n, $\|v\|_1 \leq \sqrt{n}\|v\|_2$ and setting $C = L{N_{\mathrm{atom}}}^2 \int |\tilde{k}(r)| dr$, we have that

$$\min_{g\in SO(3)} \max_{\bar{r}} \left|\mathrm{Img}_{\theta,\phi,\chi}(\bar{r}) - \mathrm{Img}_{\theta,\phi,x}(\bar{r})\right| \leq C \min_{g\in SO(3)} \sqrt{\frac{1}{N_{\mathrm{atom}}} \sum_{n=1}^{N_{\mathrm{atom}}} \left\| \vec{\chi}_n - g\vec{x}_n \right\|_2^2} \ . \tag{S.14}$$

and recalling the definition of the $RMSD$ distance from Eq. S.7 completes the proof. □

The final thing that remains to show is that the specific point-spread function used by our model is globally Lipschitz.

Lemma 3 (PSF is Lipschitz)**.** *The PSF used in our imaging model is globally Lipschitz.*

Proof. It suffices to show that the Jacobian of the PSF is globally bounded. We first consider the partial derivative of the PSF with respect to r_i. From the properties of the Fourier Transform, we have that

$$\mathcal{F}(\partial_{r_i}\mathrm{PSF}_{\gamma,A,b})(\xi_i,\xi_j) = 2\pi\mathrm{i}\left(A\cos(\gamma\xi^2) - \sqrt{1-A^2}\sin(\gamma\xi^2)\right)\left(\xi_i e^{-b\xi^2}\right) \tag{S.15}$$

$$\begin{aligned}
|\partial_{r_i}\mathrm{PSF}_{\gamma,A,b}(r_i,r_j)| &= \left|\int e^{\mathrm{i}2\pi\xi_i r_i} e^{\mathrm{i}2\pi\xi_j r_j} 2\pi\mathrm{i}\left(A\cos(\gamma\xi^2) - \sqrt{1-A^2}\sin(\gamma\xi^2)\right)\left(\xi_i e^{-b\xi^2}\right) d\xi_i d\xi_j\right| \\
&\leq \int \left|2\pi\mathrm{i}\left(A\cos(\gamma\xi^2) - \sqrt{1-A^2}\sin(\gamma\xi^2)\right)\left(\xi_i e^{-b\xi^2}\right) e^{\mathrm{i}2\pi\xi_i r_i} e^{\mathrm{i}2\pi\xi_j r_j}\right| d\xi_i d\xi_j \\
&= \int \left|2\pi\mathrm{i}\left(A\cos(\gamma\xi^2) - \sqrt{1-A^2}\sin(\gamma\xi^2)\right)\right| \left|e^{\mathrm{i}2\pi\xi_i r_i} e^{\mathrm{i}2\pi\xi_j r_j}\right| \left|\xi_i e^{-b\xi^2}\right| d\xi_i d\xi_j
\end{aligned}$$

Applying the triangle inequality and the fact that the magnitude of trigonemetric functions and complex exponentials are bounded by 1, we have that

$$\begin{aligned}
|\partial_{r_i}\mathrm{PSF}_{\gamma,A,b}(r_i,r_j)| &\leq 2\pi\left(|A| + |\sqrt{1-A^2}|\right)\int \left|\xi_i e^{-b(\xi_i+\xi_j)^2}\right| d\xi_i d\xi_j &\text{(S.16)} \\
&= 2\left(|A| + |\sqrt{1-A^2}|\right)\left(\frac{\pi}{b}\right)^{3/2} &\text{(S.17)}
\end{aligned}$$

By similar arguments, we have the same bound for $|\partial_{r_j}\mathrm{PSF}_{\gamma,A,b}(r_i,r_j)|$. Consequently, the Jacobian is bounded globally, completing the proof.

□

References

(1) Gelman, A.; Rubin, D. B. Inference from iterative simulation using multiple sequences. *Statistical Science* **1992**, 457–472.

(2) Geyer, C. J. Practical Markov chain monte carlo. *Statistical Science* **1992**, 473–483.

(3) Vehtari, A.; Gelman, A.; Simpson, D.; Carpenter, B.; Bürkner, P.-C. Rank-normalization, folding, and localization: an improved R for assessing convergence of MCMC (with discussion). *Bayesian Analysis* **2021**, *16*, 667–718.

(4) Penczek, P. A. In *Cryo-EM, Part B: 3-D Reconstruction*; Jensen, G. J., Ed.; Methods in Enzymology; Academic Press, 2010; Vol. 482; pp 35–72.

(5) Cossio, P.; Hummer, G. Bayesian analysis of individual electron microscopy images: Towards structures of dynamic and heterogeneous biomolecular assemblies. *Journal of Structural Biology* **2013**, *184*, 427–437.

(6) Hanson, A. J. *ACM SIGGRAPH 2005 Courses*; 2005; pp 1–es.

(7) Klem, H.; Hocky, G. M.; McCullagh, M. Size-and-Shape Space Gaussian Mixture Models for Structural Clustering of Molecular Dynamics Trajectories. *Journal of Chemical Theory and Computation* **2022**,

(8) Schubert, E.; Lenssen, L. Fast k-medoids Clustering in Rust and Python. *Journal of Open Source Software* **2022**, *7*, 4183.

(9) Schubert, E.; Rousseeuw, P. J. Fast and eager k-medoids clustering: O (k) runtime improvement of the PAM, CLARA, and CLARANS algorithms. *Information Systems* **2021**, *101*, 101804.

(10) Yershova, A.; Jain, S.; Lavalle, S. M.; Mitchell, J. C. Generating uniform incremental grids on SO (3) using the Hopf fibration. *The International journal of robotics research* **2010**, *29*, 801–812.

(11) Zonca, A.; Singer, L.; Lenz, D.; Reinecke, M.; Rosset, C.; Hivon, E.; Gorski, K. healpy: equal area pixelization and spherical harmonics transforms for data on the sphere in Python. *Journal of Open Source Software* **2019**, *4*, 1298.

(12) Madsen, J.; Susi, T. The abTEM code: transmission electron microscopy from first principles [version 2; peer review: 2 approved]. *Open Research Europe* **2021**, *1*.