STRUCTURAL BIOLOGY

ISSN 2059-7983

The *ManifoldEM* method for cryo-EM: a step-by-step breakdown accompanied by a modern Python implementation

Anupam Anand Ojha,[a] Robert Blackwell,[b] Eduardo R. Cruz-Chú,[c] Raison Dsouza,[d,e] Miro A. Astore,[a] Peter Schwander[c] and Sonya M. Hanson[a]*

[a]Center for Computational Biology and Center for Computational Mathematics, Flatiron Institute, New York, NY 10010, USA, [b]Scientific Computing Core, Flatiron Institute, New York, NY 10010, USA, [c]Department of Physics, University of Wisconsin-Milwaukee, Milwaukee, WI 53211, USA, [d]Morgridge Institute for Research, Madison, WI 53715, USA, and [e]Department of Biochemistry, University of Wisconsin-Madison, Madison, WI 53706. USA. *Correspondence e-mail: shanson@flatironinstitute.org

Received 25 November 2024
Accepted 17 February 2025

Edited by C. O. Sorzano, National Center of Biotechnology, CSIC, Spain

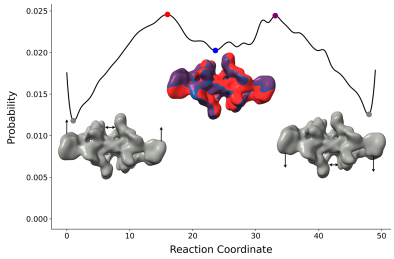
This article is part of a focused issue on Image Processing for CryoEM.

Keywords: cryo-EM; conformational heterogeneity; Python; manifold analysis.

Resolving continuous conformational heterogeneity in single-particle cryo-electron microscopy (cryo-EM) is a field in which new methods are now emerging regularly. Methods range from traditional statistical techniques to state-of-the-art neural network approaches. Such ongoing efforts continue to enhance the ability to explore and understand the continuous conformational variations in cryo-EM data. One of the first methods was the manifold embedding approach or *ManifoldEM*. However, comparing it with more recent methods has been challenging due to software availability and usability issues. In this work, we introduce a modern Python implementation that is user-friendly, orders of magnitude faster than its previous versions and designed with a developer-ready environment. This implementation allows a more thorough evaluation of the strengths and limitations of methods addressing continuous conformational heterogeneity in cryo-EM, paving the way for further community-driven improvements.

1. Introduction

The field of obtaining information about continuous conformational heterogeneity from cryo-electron microscopy (cryo-EM) data sets is rapidly growing. Early efforts with traditional statistical methods such as principal component analysis (PCA; van Heel & Frank, 1981; Stewart, 1990) and maximum-likelihood estimation (MLE; Sigworth, 1998; Scheres & Chen, 2012) enabled the parsing of discrete structural states in cryo-EM data, providing initial insights into structural flexibility. However, their reliance on linearity and discrete-state assumptions limited their ability to capture smooth, continuous changes in structure, making them less effective for systems with complex conformational dynamics. Subsequent advancements saw the adaptation of advanced machine-learning methods, particularly variational autoencoders (VAEs; Kingma & Welling, 2019) and generative adversarial networks (GANs; Goodfellow *et al.*, 2020), which introduced nonlinear dimensionality-reduction approaches for cryo-EM data analysis. VAE-based approaches (Zhong *et al.*, 2021; Punjani & Fleet, 2021; Tang, Zhong *et al.*, 2023) introduced a probabilistic framework capable of modeling complex structural variability and capturing a continuum of conformational states. Such frameworks enable precise mapping of conformational landscapes, allowing the identification of intermediate states and transitions that extend beyond traditional, discrete structural models. *cryoDRGN* employs VAEs to



capture continuous 3D variability in cryo-EM images without requiring manual intervention or pre-specified states, revealing new structural states and large-scale motions in systems such as the ribosome and spliceosome (Zhong *et al.*, 2021). *cryoSPARC* (Punjani *et al.*, 2017) applies stochastic gradient descent (SGD) and branch-and-bound maximum-likelihood optimization, accelerating significant steps in cryo-EM structure determination. Incorporating Bayesian marginalization with SGD allows automated, *ab initio* 3D classification and facilitates the unbiased exploration of structural states. Such methods exemplify recent advancements in deep learning and hardware acceleration, which allow the efficient processing of large cryo-EM data sets and the reconstruction of subtle structural variations and high-resolution conformational mapping. Further advances have also recently been made on this problem using a more classical approach with the *RECOVAR* method, which has seen surprising success analyzing conformational heterogeneity in cryo-EM by regularized covariance estimation and kernel regression (Gilles & Singer, 2024).

One pioneering approach for analyzing conformational heterogeneity is the manifold embedding method, or *ManifoldEM*, originally developed by Abbas Ourmazd, Peter Schwander and Joachim Frank (Dashti *et al.*, 2014; Schwander *et al.*, 2014). Originally implemented as *MATLAB* routines, *ManifoldEM* represents one of the first comprehensive frameworks designed to capture continuous structural variability in single-particle cryo-EM and X-ray free-electron laser (XFEL) data. Since its inception, *ManifoldEM* has evolved from a proof of concept to a multi-step protocol. However, comparing it with more recent methods has been challenging because of software availability and usability.

The *ManifoldEM* method has been applied with great success in single-particle analysis in cryo-EM for a range of biomolecular systems, such as an apo ribosome data set (Dashti *et al.*, 2014), the ryanodine receptor (Dashti *et al.*, 2020), the SARS-CoV-2 spike protein (Sztain *et al.*, 2021), and even to show that annealing synchronizes the 70S ribosome conformation (Chu *et al.*, 2022). A distinguishing feature of this method is its initial focus on dimensionality reduction in the conformational space of the biomolecule within 2D projection directions. The analysis within these 2D projection directions follows a systematic progression, beginning with Euclidean distance calculations between images to quantify similarity, followed by diffusion mapping (Coifman *et al.*, 2005) and nonlinear Laplacian spectral analysis (NLSA; Giannakis & Majda, 2012) to transform these patterns into an interpretable conformational space. Recent work has shed light on the advantages and drawbacks of the NLSA method within the context of *ManifoldEM*, including a refined free energy scheme, ESPER, for when it is needed (Seitz *et al.*, 2022, 2023). Another recent advance relates to transitioning from analyzing individual projection directions (PDs) to covering the full orientation space, which was originally a manual, *ad hoc* procedure. This process can now be streamlined by incorporating automated optical flow belief propagation (Maji *et al.*, 2020), providing a structured framework for combining projections. Once belief-propagation decisions are finalized, a final step merges the manifold analyses from all PDs to construct a probability distribution of conformational states and corresponding trajectories of 3D volumes within this conformational space.

The current work presents a step-by-step breakdown of the *ManifoldEM* algorithm along with a description of the updates and speed-ups of the modern Python implementation. Various parameter choices for optimal performance are described, with example data sets for concrete illustrations of the optimization process. These examples include choices for setting optimal aperture parameters, thresholding to filter projection directions, belief-propagation settings for accurate alignment across projection directions and methods for estimating the final energy landscape and reconstructing related 3D volumes. While many of these decisions are automated in this newest version, users can access the relevant parameters if manual adjustments are needed. Examples of real and synthetic cryo-EM data sets are also provided to demonstrate the method and practical considerations when using it.

2. Workflow

As with most other tools for conformational heterogeneity in single-particle cryo-EM, *ManifoldEM* requires input files as produced from traditional 3D refinement algorithms, such as are commonly used in *RELION* (Scheres, 2012) or *cryoSPARC* (Punjani *et al.*, 2017). With the extracted particles in a single image stack and an alignment file with Euler angles and defocus estimates for all particles, the user can then carry out the steps required to perform the *ManifoldEM* analysis with the software presented here. The following sections provide an overview of the different steps in the *ManifoldEM* workflow, with an overview given in Fig. 1. These steps encompass (i) initialization, (ii) distance calculation to assess the structural similarities between particle images, (iii) manifold analysis to capture the intrinsic geometric structure of the data set, (iv) localized conformational mode analysis to facilitate dimensionality reduction, (v) optical flow to estimate the directional conformational transitions, (vi) probability distribution generation, and (vii) 3D volume reconstruction.

This modern Python implementation of *ManifoldEM* now features a command-line interface (CLI) called `manifold-cli`, which allows users to run the entire *ManifoldEM* pipeline directly from the command line, offering substantial advantages over the graphical user interface (GUI). `manifold-cli` enables the efficient handling of large-scale cryo-EM data sets and simplifies complex computations that would be cumbersome in a GUI. Each step in the workflow, such as project setup, noise reduction, distance calculation, manifold embedding and probability distribution estimation, can be executed independently, giving fine-grained control over the entire process. This CLI is particularly useful for large data sets, offering faster, scriptable execution than GUIs. Both the GUI and the CLI support parallel processing, enabling users to allocate multiple CPU cores and speeding up tasks significantly, which is critical when working on high-performance

computing (HPC) clusters. The CLI also enables automation, allowing users to integrate *ManifoldEM* commands into custom workflows and easily repeat analyses across multiple data sets without reconfiguration. Furthermore, this new Python implementation is more developer-friendly, allowing advanced users to run relevant methods outside the GUI or CLI, thus making it easier for new features and algorithms to be added on top of the current implementation of this algorithm.

The following sections describe the step-by-step process in the *ManifoldEM* workflow, with the `manifold-cli` accompanying each section to facilitate the execution of specific steps.

2.1. Data sets

2.1.1. Ryanodine receptor 1

Ryanodine receptor 1 (RyR1) is a calcium-release channel that is crucial for muscle contraction, mediating Ca^{2+} flow from the sarcoplasmic reticulum to the cytoplasm (Van Petegem, 2012; Ogawa, 1994). Known for its complex gating, the conformational states of RyR1 are modulated by ligands such as ATP, caffeine and calcium, making it an ideal system for studying conformational heterogeneity via cryo-EM (McPherson *et al.*, 1991; Kermode *et al.*, 1998; Dashti *et al.*, 2020). The RyR1 data set is featured in this study for two primary reasons. First, it demonstrates the ability of *ManifoldEM* to process large, dynamic macromolecular assemblies, enabling straightforward analysis of conformational states under various physiological conditions. Second, the ligand-modulated gating of the receptor displays the complex structural transitions that *ManifoldEM* can capture, revealing intermediate states essential for understanding the regulation and role of the channel in muscle function.

2.1.2. Thyroglobulin

The thyroglobulin protein, which is essential in thyroid hormone synthesis, serves as a reservoir for intraglandular iodine and plays a major role in metabolism, development and growth (Citterio *et al.*, 2019; Cody, 1984). A synthetic cryo-EM thyroglobulin data set (Astore *et al.*, 2023) was generated with conformational states derived from molecular dynamics (MD) simulations and images created using *cisTEM* (Grant *et al.*, 2018).

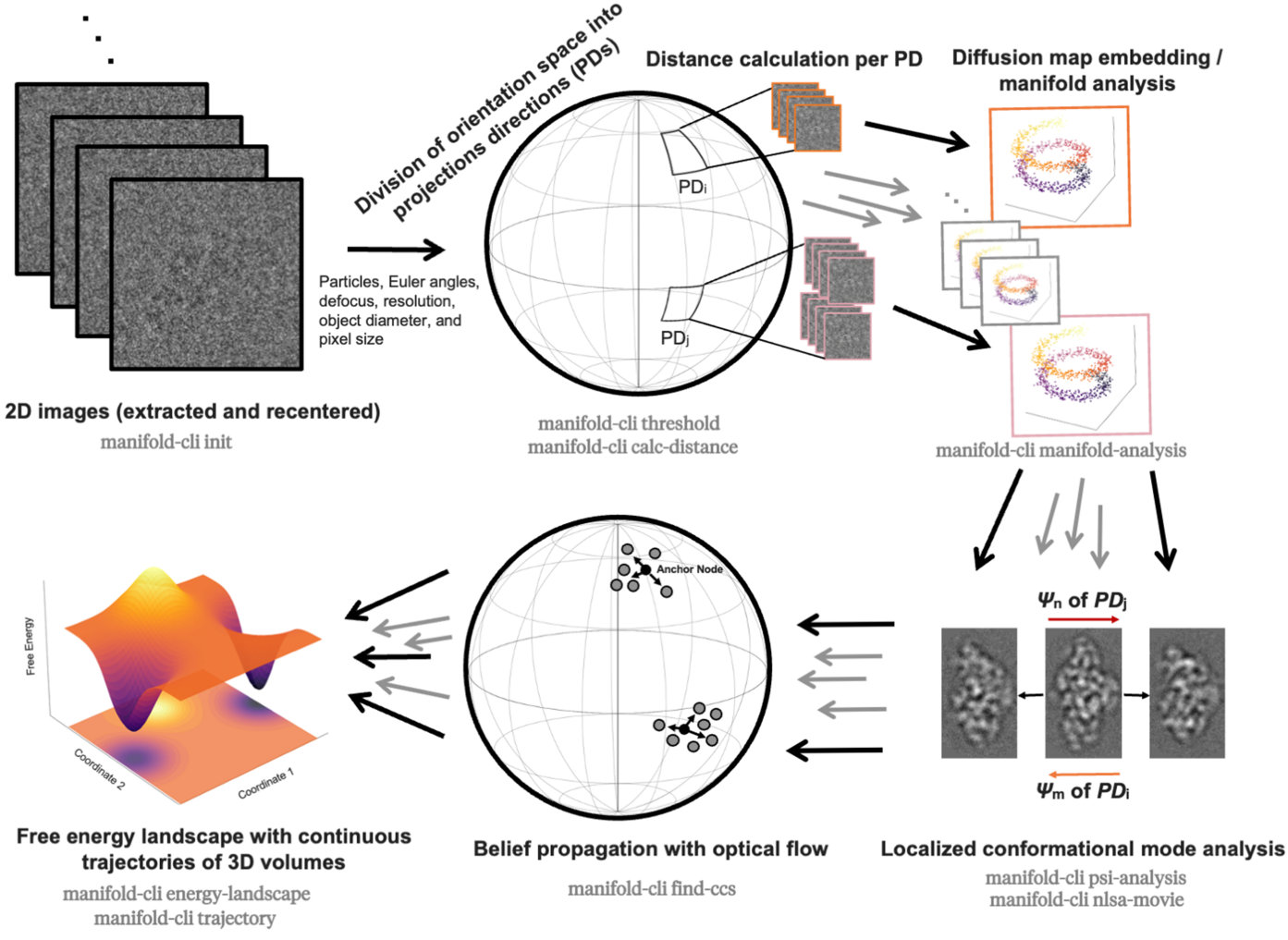


Figure 1
Schematic showing the *ManifoldEM* workflow.

2.2. Initialization

To initialize *ManifoldEM*, three distinct files are required: a single-particle stack (in `.mrcs` format), the relevant alignment file (in `.star` format) and a consensus volume (in `.mrc` format). This last file is strictly used for visualization purposes and is not used by the analysis method itself. While the input format for the alignment file is the *RELION* standard `.star` file, containing metadata about the image stack, including Euler angles, defocus values and particle X and Y shifts. Also provided is a utility to convert *cryoSPARC* output files into this format for convenience.

`manifold-cli init`: Project initialization

The `init` command initializes a new project by constructing a configuration file that defines the input data, such as the average volume, alignment files, image stacks, and parameters such as pixel size and resolution.

```
manifold-cli init -p project_name -v avg_volume.mrc
-a alignment.star -m mask.mrc -i image_stack.mrcs
-s pixel_size -d diameter -r resolution -x
aperture_index -o
```

where
`-p`: Project name
`-v`: Average volume file path (.mrc)
`-a`: Alignment file path (.star)
`-m`: Mask file path (.mrc)
`-i`: Image stack file path (.mrcs)
`-s`: Pixel size in Å/pixel
`-d`: Diameter of the system in Å
`-r`: Data set resolution in Å
`-x`: Aperture index
`-o`: Overwrite existing project

2.3. Preprocessing and distance calculation

Once a project is initialized, *ManifoldEM* starts with several preprocessing steps. The images are first sorted into uniform bins over the 2-sphere, a process that is determined by the input parameters aperture, object diameter and resolution provided by the user. This sorting determines the projection directions (PDs) within which most downstream calculations are performed. Once these PDs have been determined, the first step of *ManifoldEM* is performed: a nearest-neighbor search leading to the distance calculation that employs a defocus-invariant kernel. Each of these steps is detailed below.

2.3.1. Uniform distribution of projection directions over the 2-sphere (S^2)

In single-particle cryo-EM, three-dimensional structures of molecules are reconstructed from numerous 2D projection images taken at various orientations, where each image represents a unique view of the molecule corresponding to a specific direction in 3D space. In *ManifoldEM*, images are grouped into projection directions (PDs) aligned within a narrow range of orientations, capturing structural heterogeneity while maintaining consistency and sensitivity to conformational changes. *ManifoldEM* analyzes the images within each PD, revealing the conformational state of the system along a specific projection direction, enabling a detailed analysis of dynamic changes within the 3D structure. *ManifoldEM* then applies thresholding to retain only well sampled PDs and eliminate low-occupancy PDs to reduce noise. Pruning low-occupancy PDs in diffusion mapping is an essential step to reduce noise and improve the stability of the analysis. Low-occupancy PDs often correspond to sparsely populated and under-sampled regions of the conformational space, which may amplify noise and lead to unreliable manifold embeddings. By pruning such PDs, the workflow focuses on well-sampled regions of the data set, ensuring that the diffusion map accurately represents the dominant conformational heterogeneity. However, an important consideration is whether low-occupancy PDs, while not capturing distinct conformations, could contribute to volume reconstruction by supporting the overall resolution. This remains an active research area of interest. To reconstruct a particle of diameter D at a resolution d, the angular sampling interval θ should not exceed d/D. This follows from the Crowther criterion (Crowther, 1970), which determines the number of PDs needed for reconstruction by rotation about a single axis. In *ManifoldEM*, the aperture index refines θ by scaling as

$$\theta_{\rm bin} = \frac{\text{aperture index} \times \text{resolution}}{\text{particle diameter}}. \quad (1)$$

Larger aperture indices increase the angular bin size, resulting in fewer PDs, while smaller values provide finer binning with more PDs. This process refines the selection of well-sampled regions while ensuring sufficient coverage of conformational space. The total number of PDs, $N_{\rm PD}$ is estimated by tessellating the unit sphere with bins of approximate area, $\theta_{\rm bin}^2$, given by

$$N_{\rm PD} \approx \frac{2\pi}{\theta_{\rm bin}^2}. \quad (2)$$

In *ManifoldEM* PDs are defined with both a minimum and maximum threshold for the number of particles per PD. By default, the minimum threshold is set to 100 images, while the upper threshold is set to 2000 images to manage computational costs and maintain analytical precision. Users can modify these thresholds based on data-set diversity and available computational resources, with lower thresholds aiding in capturing finer conformational differences and higher thresholds suited for uniform data sets.

For the RyR1 data set containing 14 491 particles, given a resolution of 5 Å, a pixel size of 1.255 Å, an object diameter of 360 Å, and an aperture index of 4, the Shannon angle approximates that 2005 bins would be necessary to cover all unique orientations at this resolution. After thresholding, only 53 PDs met this requirement (Fig. 2*a*), with 1952 PDs containing fewer than 100 particles. A higher threshold of 250 particles per PD is applied for the thyroglobulin data set containing 674 840 particles with a resolution of 4 Å, a pixel size of 1.073 Å, an object diameter of 350 Å, and an aperture index of 4. Based on the Shannon angle, 2957 PDs would ideally represent the structural variability at this resolution, but after applying the 250-particle threshold only 1094 PDs meet this minimum, with 1863 PDs containing fewer than 250

particles (Fig. 2*b*). This reduction helps to maintain only well sampled PDs, balancing structural detail and noise reduction.

`manifold-cli threshold`: Set thresholds for PD selections

The `threshold` command sets upper and lower thresholds for determining the most significant PDs based on the number of particle images, enhancing the analysis by both excluding directions with too few images to reduce noise and focusing on meaningful data, and limiting the maximum number of images included in a projection direction to improve processing time.

```
manifold-cli threshold --prd_thres_low low
--prd_thres_high high input_file.toml
```

where
`--prd_thres_low`: Minimum snapshots per projection direction
`--prd_thres_high`: Maximum snapshots per projection direction
`--tess_hemisphere_vec`: Vector defining the hemisphere for projection directions
`--tess_hemisphere_type`: Sphere tessellation algorithm (Options: "lovisolo_silva", "fibonacci")

The distribution of PDs can be understood as points on the surface of a 2-sphere (S^2), where each direction (or orientation) can be described by a quaternion, a mathematical construct that represents 3D rotations without the ambiguities of Euler angles or other rotation representations (Hu *et al.*, 2020; Hart *et al.*, 1994). Once the projection directions have been uniformly distributed on S^2, they are mapped to one-half of the S^2 by introducing a vector $v_{\rm tess}$ that defines a specific hemisphere by selecting only those points p on S^2 for which the dot product $v_{\rm tess} \cdot p \geq 0$. Here, $v_{\rm tess}$ acts as a reference direction, establishing a consistent boundary that eliminates mirrored, redundant points on the opposite side of the sphere. The half-sphere exploits the mirror symmetry of points on opposite sides of a sphere and increases the number of points within a given PD. This mapping also provides a direct relationship between the 3D orientations of the quaternions and a 2D coordinate system, simplifying the alignment and clustering of 2D cryo-EM images. Currently, the effects of symmetry can be simulated in the preprocessing of the particle stack by duplicating the particle images and applying symmetry expansion before running the manifold embedding pipeline. The coordinates on S^2 are then used to align and cluster similar images, allowing a more effective classification and averaging, and enhancing the signal-to-noise ratio in the final 3D reconstruction. Quaternions, representing 3D orientations, are mapped to S^2 by

$$S^2 = 2\begin{pmatrix} q_1q_3 - q_0q_2 \\ q_0q_1 + q_2q_3 \\ q_0^2 + q_3^2 - 0.5 \end{pmatrix}, \tag{3}$$

where $q = [q_0, q_1, q_2, q_3]$ is the quaternion vector, q_0 is the real part and q_1, q_2 and q_3 are the imaginary components.

2.3.2. Nearest-neighbors calculation and orientation refinement

Once the projection directions are uniformly distributed on S^2, the nearest-neighbors search employing the ball-tree algorithm (Liu, 2018) identifies similar orientations for the clustering and alignment of cryo-EM images. S^2 points are then organized into discrete bins, where each bin represents a small region on the sphere, with centers calculated from the uniform distribution of the initial S^2 projection. Let B_i represent a bin centered at C_i with radius r. The inclusion of a point p in the bin is determined by

$$\|p - C_i\| < r, \tag{4}$$

where $\|\cdot\|$ denotes the Euclidean distance.

A thresholding function (equation 5) retains bins with sufficient points, focusing on significant orientations, mini-

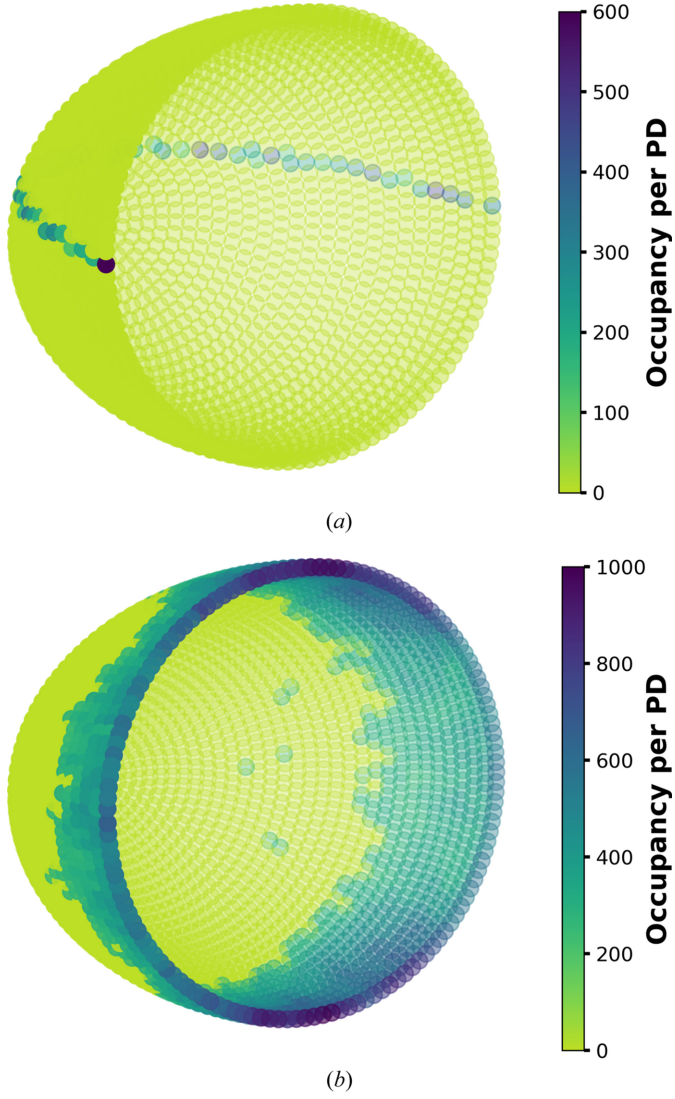


Figure 2
Occupancy distribution across projection directions for the (*a*) RyR1 and (*b*) thyroglobulin data sets. Each bin represents particle occupancy within a specific PD, with the heatmap indicating particle density.

mizing noise and avoiding excessive computational time for sparse bins.

$$N_{\min} \leq \text{count}(B_i) \leq N_{\max}, \tag{5}$$

where $\text{count}(B_i)$ denotes the number of points in bin B_i. The parameters $N_{\min}$ and $N_{\max}$ define the minimum and maximum point counts within each bin. Selecting $N_{\min}$ ensures that bins with very few points, which could represent outliers or noisy data, are excluded. Bins with point counts exceeding $N_{\max}$ are kept, but only the first $N_{\max}$ points in the bin are used in processing to reduce the computational load.

2.3.3. Image processing and contrast transfer function (CTF) correction

Images that had their orientations mirrored are first flipped to reflect their mirrored status. A Gaussian filter is then applied in the Fourier domain to enhance image features and suppress high-frequency noise selectively. The Gaussian filter is defined by

$$G = \exp\left[-\frac{\log(2)}{2}\left(\frac{Q}{f_0}\right)^2\right], \tag{6}$$

where Q represents the spatial frequency grid and f_0 controls the range of frequencies preserved by the filter. If the user does not provide a mask, then a circular mask with a radius of half of the image width is applied to the image. If the user provides a volume mask as input, it is projected into the image plane and applied to the image, setting values outside the mask to zero. Each image is then normalized to remove background noise, adjusting the pixel intensity values and ensuring uniformity by scaling based on background statistics. The normalization is defined as

$$\text{img}_{\text{norm}} = \frac{\text{img} - \mu_{\text{background}}}{\sigma_{\text{background}}}, \tag{7}$$

where $\mu_{\text{background}}$ and $\sigma_{\text{background}}$ are the mean and standard deviation of the background region, respectively.

2.3.4. Distance calculation

The squared Euclidean distance between every pair of images, i and j, in a PD is the first substantive calculation in the *ManifoldEM* algorithm. The defocus-corrected distance between any pair of images (Schwander *et al.*, 2014) is defined by

$$D^2(i,j) = \|F(\text{img}_i)\cdot \text{CTF}_j - F(\text{img}_j)\cdot \text{CTF}_i\|^2, \tag{8}$$

where $F(\text{img}_i)$ represents the Fourier transform of image i (here already normalized, but not notated for simplicity of presentation) and the products represent the sum of the element-wise multiplication of each Fourier-transformed image i with the contrast transfer function CTF_j of the image j. The CTF characterizes how the electron microscope modulates the phase and amplitude of different spatial frequencies in the image (Zhu *et al.*, 1997; Bhamre *et al.*, 2016), here calculated by

$$\text{CTF}(k) = \sin\left(\frac{1}{2}\pi C_s\lambda^3 k^4 - \pi\lambda\Delta f k^2\right) - \alpha\cdot\left[\cos\left(\frac{1}{2}\pi C_s\lambda^3 k^4 - \pi\lambda\Delta f k^2\right)\right], \tag{9}$$

where C_s denotes spherical aberration, λ is the electron wavelength, Δf is the defocus, k is the spatial frequency and α represents the amplitude contrast, *i.e.* the proportion of phase shift caused by scattering in the imaging process, affecting the visibility of low-frequency components in the resulting image.

In the initial MATLAB implementation, the distance calculation was the most time consuming step. The current optimized distance-calculation pipeline enhances the efficiency of squared Euclidean distance computations between image pairs by incorporating defocus-invariant CTF correction directly in memory. By avoiding redundant I/O operations and integrating the CTF correction step within the distance calculations, the approach eliminates the previous bottlenecks associated with disk writes. This in-memory, defocus-invariant implementation, combined with parallelized processing of images in bins and multiprocessing, efficiently handles large data sets, meeting the computational demands of modern cryo-EM analysis.

`manifold-cli calc-distance`: Calculate S^2 distances

The `calc-distance` command calculates the squared distances between particle images to determine structural relationships among conformational states.

```
manifold-cli calc-distance --num_psi num_psi
input_file.toml
```

where
`--num_psi`: Number of eigenfunctions for conformational analysis
`--distance_filter_type`: Filter type for image preprocessing (Options: "Butter", "Gaussian")
`--distance_filter_cutoff_freq`: Cut-off frequency for the filter (Nyquist threshold)
`--distance_filter_order`: Order of the Butterworth filter (controls response steepness)

2.4. Diffusion mapping and manifold analysis

The manifold analysis protocol in the *ManifoldEM* framework captures the intrinsic geometric structure of high-dimensional cryo-EM data sets. From estimating the optimal Gaussian kernel width to embedding data in a lower-dimensional space, this approach provides a systematic workflow to analyze conformational heterogeneity in cryo-EM data. Diffusion maps (Coifman & Lafon, 2006; Lafon, 2004), which are a core component of this protocol, capture both local and global structural variations by constructing a weighted graph based on local similarities with a Gaussian kernel. Key highlights of the manifold analysis protocol within the *ManifoldEM* framework are outlined below.

2.4.1. Weight matrix and optimal σ estimation

The distance matrix $D(i,j)$ represents pairwise distances between projection directions, defining the local and global

geometry of the data set. A Gaussian kernel that measures the distance-based similarity between two points is used to construct the weight matrix W, represented by

$$W_{ij} = \exp\left(-\frac{D_{ij}^2}{2\sigma^2}\right), \tag{10}$$

where D_{ij} is the distance between the data points i and j. W is crucial for defining the local and global geometry of the data and is used in constructing graph-based representations of the data set, such as in diffusion maps and spectral clustering (Ng *et al.*, 2001; Coifman & Lafon, 2006). Selecting an appropriate Gaussian kernel width (σ) is essential for accurately capturing

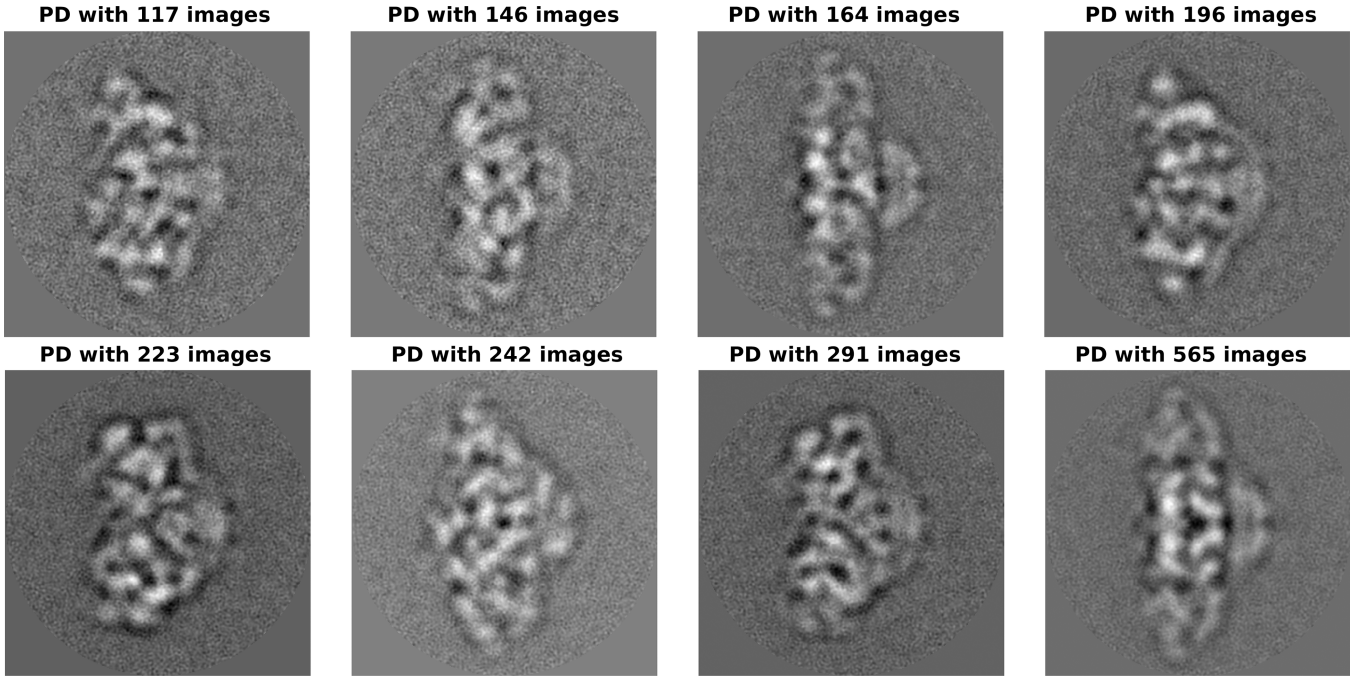


Figure 3
2D projection images from selected projection directions (PDs) for (*a*) the RyR1 data set and (*b*) the thyroglobulin data set. Each image represents an average from all of the images in its corresponding projection direction, labeled with the total number of contributing images, arranged from the lowest to the highest.

the data structure. The choice of σ directly influences the sensitivity of the kernel to variations in D_{ij}. If σ is too small the kernel becomes too localized, resulting in a sparse weight matrix that only captures very close relationships. Conversely, a large σ makes the kernel insensitive to distance, causing it to lose local geometric details. The optimal σ is identified through a curve-fitting method maximizing the log-determinant of the kernel matrix, ensuring that the weight matrix, W, captures both local and global data structures, thereby facilitating accurate manifold learning and representation of the cryo-EM data. The `nlsa_tune` parameter, specified in the configuration TOML file, governs the Gaussian kernel width by scaling the variance term used in the kernel. This parameter is optional and defaults to a value of 3 if not explicitly defined. Smaller values of `nlsa_tune` broaden the kernel, emphasizing global patterns, while larger values focus on local relationships. Users can adjust this parameter based on the characteristics of the data set to achieve an appropriate balance between capturing local and global features.

To examine the structural diversity and conformation distribution across PDs for the RyR1 and thyroglobulin data sets, average images were generated for all PDs, ranging from the lowest to highest occupancy. Note that average images for each PD are not used in the algorithm itself but are useful for the user to visualize PDs. To produce these average images, a Wiener filter is applied to the Fourier-transformed images to enhance signal quality and suppress noise by

$$\mathcal{W}(k) = \frac{\mathrm{CTF}(k)}{\mathrm{CTF}^2(k) + (1/\mathrm{SNR})}, \tag{11}$$

where SNR is the signal-to-noise ratio. Lower-occupancy PDs may exhibit higher noise levels due to limited data points, whereas higher-occupancy PDs benefit from an improved signal-to-noise ratio (SNR), which enhances structural clarity and the representation of finer details. For the RyR1 data set, circular masks (Fig. 3*a*) were employed that focus on central regions and reduced edge noise, ensuring uniform processing. The thyroglobulin data set employed adaptive masks (Fig. 3*b*) that adjust to particle features, removing image pixels that do not include the molecule. Figs. 3(*a*) and 3(*b*) show selected averaged PD images for the RyR1 and thyroglobulin data sets, respectively, arranged in an ascending occupancy order from lowest to highest. Intermediate-occupancy PDs are included to represent a range of orientations and image densities, providing a detailed view of conformational changes and subtle structural variations across the conformational landscape in each data set.

2.4.2. Laplacian matrix construction and spectral decomposition

Once the weight matrix has been constructed, spectral analysis can be performed. This process begins by constructing the Laplacian matrix, L, a representation of the graph structure of the data that captures the relationships between data points based on the weight matrix, W. The Laplacian matrix, defined by equation (12), is central to manifold-learning approaches, such as diffusion maps and spectral clustering, as it encodes the connectivity and geometric structure of the data set, enabling effective dimensionality reduction and clustering.

$$L = \mathcal{D} - W, \tag{12}$$

where $\mathcal{D}$ is the degree matrix, a diagonal matrix where each diagonal entry $\mathcal{D}_{ii}$ represents the sum of the weights of the edges connected to node i (equation 13).

$$\mathcal{D}_{ii} = \sum_j W_{ij}. \tag{13}$$

The spectral decomposition of the Laplacian matrix is defined as

$$L = U \Lambda U^{\mathrm{T}}, \tag{14}$$

where U is a matrix whose columns are the eigenvectors of L and Λ is a diagonal matrix of the corresponding eigenvalues. The leading eigenvectors, corresponding to the smallest nonzero eigenvalues, represent lower-dimensional data. This representation is then used for clustering and visualizing the manifold structure of the cryo-EM data.

2.4.3. Manifold trimming and embedding refinement

Radius-based trimming refines the spectral embedding by iteratively removing points beyond a specified radius (rad) from the origin of the embedded space. Identifying and excluding outliers or noise ensures a robust manifold representation. An initial spectral embedding is performed using the distance matrix, $D(i, j)$ (equation 15).

$$\psi_{\mathrm{dist}} = \psi[:, 0]^2 + \psi[:, 1]^2 + \psi[:, 2]^2)^{1/2}, \tag{15}$$

where ψ represents the embedding coordinates. Points with $\psi_{\mathrm{dist}} <$ rad are retained for further analysis. The value of rad is set to 5 by default, but users can access and modify this parameter if needed.

The trimmed distance matrix is recalculated for the refined set of points. A diffusion map embedding is then performed, optimizing the Gaussian kernel width (σ) and computing the eigenvalues (λ) and eigenvectors (ψ). This iterative process continues until convergence.

`manifold-cli manifold-analysis`: Manifold embedding

The `manifold-analysis` command performs the initial manifold embedding, reducing the high-dimensional data into lower dimensions to reveal the underlying conformational states and to capture the structural heterogeneity of the system.

```
manifold-cli manifold-analysis input_file.toml
```

where
`--nlsa_tune`: Tuning parameter for diffusion map embedding

2.5. Localized conformational mode analysis

The localized conformational mode-analysis step provides a localized data-set exploration focusing on identifying dominant conformational modes within each projection direction. This step translates the manifold analysis results into interpretable low-dimensional representations by constructing neighborhood-based graphs, making it ideal for dissecting

subtle conformational changes in specific data regimes. This step enables a detailed exploration of the conformational landscape by constructing a weight matrix and subsequent spectral decomposition. Key highlights of the localized conformational mode-analysis protocol within the *ManifoldEM* framework are outlined below.

2.5.1. Weight-matrix construction using *k*-nearest neighbors (KNN)

Unlike manifold analysis, localized conformational mode analysis employs a local k-nearest neighbors (KNN) graph, where each point represents an embedded coordinate in the reduced-dimensional conformational space derived from the preceding manifold analysis. These points correspond to localized representations of particle conformations, capturing their structural variations within specific projection directions. The KNN graph connects each point to its k closest neighbors based on the Euclidean distance in the reduced-dimensional space. k is computed internally by the *ManifoldEM* framework but is influenced by the `con_order_range` parameter specified in the configuration file. The parameter determines how the neighbors are grouped, and the resulting value of k reflects the number of connected neighbors used for analysis. Smaller values of `con_order_range` result in higher connectivity (more neighbors), while larger values reduce connectivity (fewer neighbors).

The Euclidean distance between two points x_i and x_j in the data set is given by

$$D(x_i, x_j) = \|x_i - x_j\|. \tag{16}$$

For each data point x_i, the k-nearest neighbors are identified by finding the k points with the smallest Euclidean distances to x_i.

Once the nearest neighbors have been determined, edges are formed between each point x_i and its neighbors x_j to form a local connectivity network. A Gaussian kernel is then applied to these neighbors, assigning weights, W_{ij}, to each edge in the local KNN graph (equation 17).

$$W_{ij} = \exp\left(-\frac{D(x_i, x_j)^2}{2\sigma^2}\right), \tag{17}$$

where σ is the kernel width controlling the locality scale in graph construction. This weighted KNN graph captures local relationships and the global structure of the data, allowing spectral decomposition and manifold embedding to represent the high-dimensional data in lower-dimensional space more effectively.

2.5.2. Laplacian matrix construction and eigen decomposition

A Laplacian matrix, L, constructed from the local weight matrix, W_{ij}, is defined similarly as in manifold analysis (equation 12). The eigen decomposition of the Laplacian matrix L yields eigenvectors, U, and corresponding eigenvalues, Λ (equation 14). The eigenvectors associated with the smallest nonzero eigenvalues are particularly important for representing the data in a lower-dimensional space.

2.5.3. Dimensionality reduction and embedding

After eigenvalue decomposition, dimensionality reduction is achieved by selecting eigenvectors corresponding to the smallest nonzero eigenvalues. Let ψ_k represent the eigenvectors corresponding to the smallest k nonzero eigenvalues,

$$\psi_k = [\psi_1, \psi_2, \ldots, \psi_k]. \tag{18}$$

Once the eigenvectors ψ_k corresponding to the smallest nonzero eigenvalues have been identified, dimensionality reduction is performed. These eigenvectors provide a lower-dimensional representation of the original high-dimensional data while retaining the essential structure of the data and discarding noise and irrelevant variations. The data points are then embedded into a k-dimensional space using the eigenvectors

$$\mathbf{y}_i = \psi_k^{\mathrm{T}} \mathbf{x}_i, \tag{19}$$

where $\mathbf{x}_i$ represents the original data points and $\mathbf{y}_i$ are the embedded points in the reduced-dimensional space.

2.5.4. Nonlinear Laplacian spectral analysis (NLSA)

Nonlinear Laplacian spectral analysis (NLSA; Giannakis & Majda, 2012; Jordanger & Tjøstheim, 2022) constructs temporal or spatial embeddings that capture the nonlinear geometry of the data by applying nonlinear dimensionality-reduction methods. It involves time-delayed embeddings, identifying recurrent patterns and analyzing dynamics to reveal complex relationships across different temporal scales, providing a comprehensive understanding of the underlying dynamics. In the case of time-delayed embeddings, a trajectory matrix is created where each row represents a time-shifted version of the original data set, allowing the method to capture the temporal evolution of the system. This matrix is represented as

$$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T], \tag{20}$$

where each column $\mathbf{x}_t$ represents data points at time t. This enables NLSA to identify recurrent patterns and dynamics for detailed temporal analysis. The reduced-dimensional representation is then analyzed to identify patterns, clusters or anomalies, providing insights into the dynamics or structure of the original data set.

`manifold-cli find-ccs`: Find conformational coordinates

The `find-ccs` command identifies the conformational coordinates within the data set, highlighting the distinct states present and helping to differentiate between various structural conformations.

```
manifold-cli find-ccs input_file.toml
```

Once the Laplacian matrix has been constructed from the distance matrix, spectral decomposition extracts eigenvalues and eigenfunctions that capture the primary modes of struc-

tural variation within the data set. The first mode represents the dominant conformational change, while subsequent modes reveal additional significant variations. Figs. 4(*a*) and 4(*b*) display scatter plots of the first three conformational modes for the two leading eigenfunctions derived from projection directions within the RyR1 and thyroglobulin data sets, respectively. Each data set includes PDs representing the lowest and highest image counts, highlighting how well sampled versus sparsely sampled orientations capture the conformational landscape. In Figs. 4(*a*) and 4(*b*), scatter plots of mode 1 versus mode 2, mode 2 versus mode 3, and mode 1 versus mode 3 illustrate structural variation and relationships within the latent space for the RyR1 and thyroglobulin data sets. These plots reveal distinct patterns or clusters that represent conformational states or transitions. High-population PDs show smoother trajectories, reflecting robust and statistically reliable modes, while low-population PDs may appear fragmented due to limited sampling. In the RyR1 data set (Fig. 4*a*) conformational modes derived from the leading eigenfunctions from PDs with lower image counts (117 images) exhibit fragmented, dispersed patterns, indicative of limited sampling that inadequately captures the structural diversity. Conversely, plots from PDs with higher image counts (565 images) display smoother, more continuous trajectories. Similarly, in the thyroglobulin data set (Fig. 4*b*), the PD with 250 images maintains noticeable structural features, although somewhat less defined than the PD with 994 images, which provides well defined and smooth trajectories across modes, highlighting structural clarity and coherence. Comparison between the first and second eigenfunctions for each PD in both data sets further emphasizes the variations in dominant structural modes. Notably, the thyroglobulin data set exhibits stability and coherence across low and high image counts, suggesting a robust manifold construction with consistent structural features, whereas the RyR1 manifold is more sensitive to sampling limitations, with well defined patterns only emerging in high-density PDs.

`manifold-cli nlsa-movie`: Create 2D movies

The `nlsa-movie` command generates 2D movies that visually represent the psi data. It allows users to observe how the manifold embedding evolves over time or as the data set is processed.

```
manifold-cli nlsa-movie --nlsa_fps fps
input_file.toml
```

2.6. Belief propagation by optical flow

Once the individual manifold analyses for each projection direction have been completed, the next task is to align

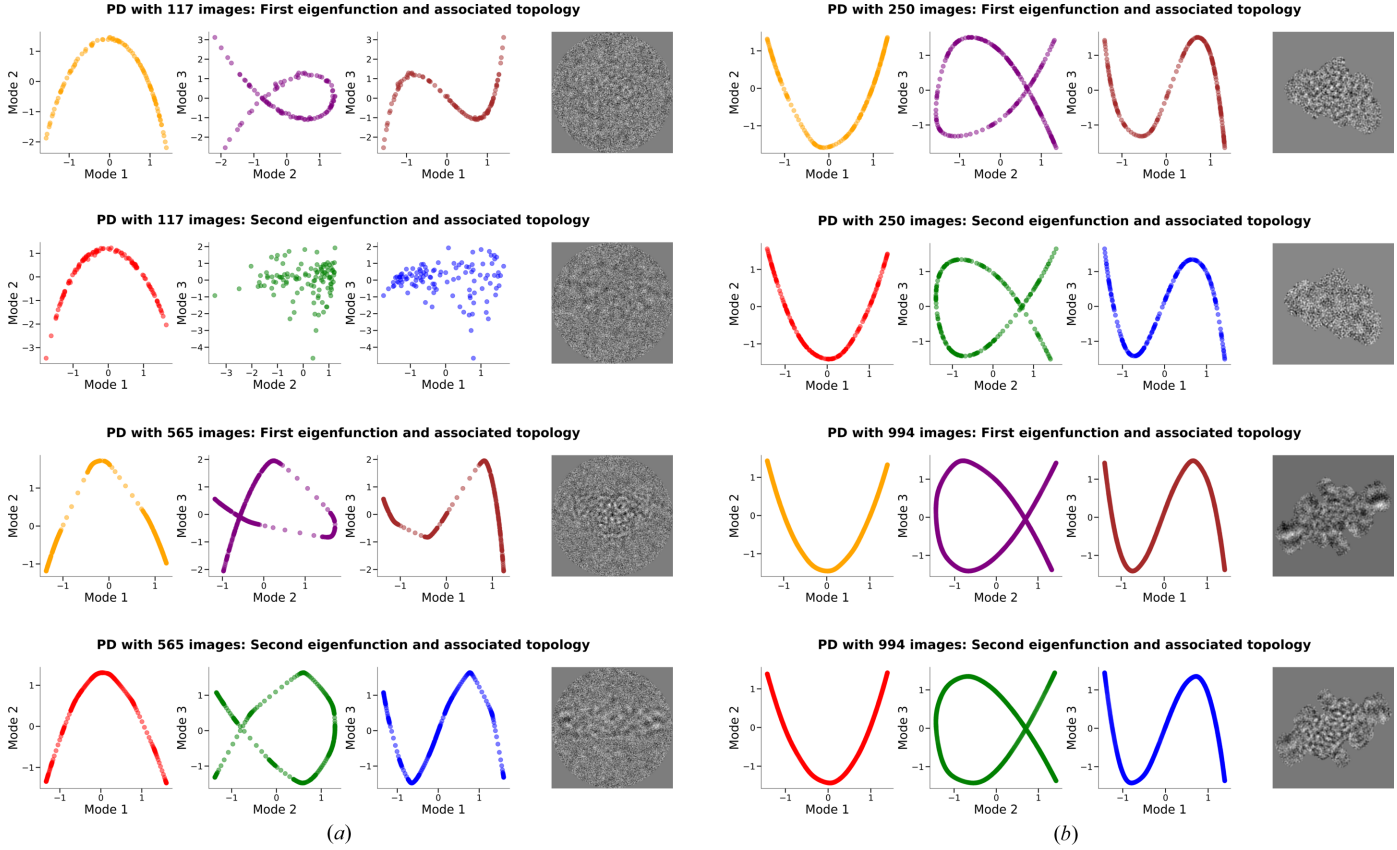


Figure 4
Scatter plots of the first three conformational modes derived from the first two eigenfunctions for selected projection directions within (*a*) the RyR1 and (*b*) the thyroglobulin data sets. For each data set, rows represent projection directions with different image counts (117 and 565 images for RyR1 and 250 and 994 images for thyroglobulin), displaying pairwise comparisons of the conformational modes, mode 1 versus mode 2 (left), mode 2 versus mode 3 (center), and mode 1 versus mode 3 (right), highlighting structural variation and latent space relationships in high-dimensional data.

conformational changes across PDs to construct a unified representation of the conformational state space, enabling 3D volume reconstruction. This alignment process, termed belief propagation (Yedidia *et al.*, 2000, 2003), ensures consistent motion direction and transitions across PDs, facilitating smooth analysis of conformational heterogeneity. The term 'belief' here refers to an inferred, consistent interpretation of motion directions and magnitudes across PDs. Employing optical flow (Otte & Nagel, 1994; Beauchemin & Barron, 1995) for tracking motion, the belief-propagation algorithm presented in the prior work (Maji *et al.*, 2020) iteratively aligns the directional changes observed in each PD, ensuring that motion patterns are consistently represented across the data set. This approach significantly reduces manual alignment requirements, as only a handful of nodes needs to be assigned manually rather than each of the potentially hundreds or thousands of individual PDs, transforming PD alignment into an accessible process achievable within a few hours.

`manifold-cli find-ccs`: Find conformational coordinates

The `find-ccs` command identifies the conformational coordinates within the data set, highlighting the distinct states present and helping to differentiate between various structural conformations.

```
manifold-cli find-ccs input_file.toml
```

2.6.1. Optical flow for motion estimation

Each PD is a member of a larger cluster, for which the user assigns an 'anchor node' (a reasonable default is the PD of highest occupancy within a cluster) to define subsequent motions within that cluster, and by extension across the data set. To estimate molecular motion across PDs, optical flow is computed to determine the velocity field, $u(x, y)$, between consecutive images. Given two images, $I(x, y, t)$ and $I(x, y, t + \Delta t)$, the optical flow is derived from the intensity-change equation (equation 21), providing the direction and magnitude of molecular movements within each PD.

$$\frac{\partial I}{\partial x}u_x + \frac{\partial I}{\partial y}u_y + \frac{\partial I}{\partial t} = 0. \tag{21}$$

2.6.2. Histogram of oriented gradients (HOG) for feature representation

To quantify conformational shifts, histograms of oriented gradients (HOGs; Dalal & Triggs, 2005; Tomasi, 2012) are computed from optical flow vectors within each PD. Each PD image is divided into cells, and the HOG features are calculated according to equation (22).

$$\mathrm{HOG}(\mathrm{PD}_i) = \{h_j\}_{j=1}^{M}, \tag{22}$$

where h_j represents the histogram for the jth cell and M is the number of cells in PD_i.

2.6.3. Clustering of PDs based on motion similarity

PDs with similar motion patterns are grouped based on HOG features, where the pairwise distance, $d(\mathrm{PD}_i, \mathrm{PD}_j)$, between PDs is given by equation (23). A similarity graph is constructed, where nodes represent PDs and edges are weighted by $d(\mathrm{PD}_i, \mathrm{PD}_j)$. PDs with comparable motions are clustered, leading to coordinated motion analysis within each group.

$$d(\mathrm{PD}_i, \mathrm{PD}_j) = \|\mathrm{HOG}(\mathrm{PD}_i) - \mathrm{HOG}(\mathrm{PD}_j)\|. \tag{23}$$

2.6.4. Belief propagation across PD clusters

Belief propagation iteratively updates the conformational state of each PD based on neighboring PDs within the cluster. Let ψ_i represent the conformational state of PD_i. Neighboring PDs, PD_j, influence ψ_i through the belief-propagation update rule (equation 24), ensuring the alignment of motion direction across all PDs achieves consistency within each cluster.

$$\psi_i^{(t+1)} = \frac{1}{Z_i}\sum_{j\in\mathrm{neighbors}}(i)\exp\left(-\frac{d(\mathrm{PD}_i, \mathrm{PD}_j)}{\sigma}\right)\psi_j^{(t)}, \tag{24}$$

where σ is a smoothing parameter and

$$Z_i = \sum_{j\in\mathrm{neighbors}}(i)\exp\left(-\frac{d(\mathrm{PD}_i, \mathrm{PD}_j)}{\sigma}\right)$$

is a normalization constant.

2.6.5. Anchor-node selection and manual determination of its 'sense'

Within each cluster, a single anchor node, generally the PD with the highest occupancy, is selected to define the relevant molecular motions. This anchor node is a reference for ensuring consistent motion sense across the entire cluster. This selection is critical to connecting the state space among all PDs by aligning the conformational changes represented in the individual PD manifold analysis. As an example of performing this process manually, the initial step involves identifying the PD with the highest image count, observing it and defining the motion captured in the primary mode, ψ_1. For instance, if the motion depicts the 'wings' of the system moving from an upward to a downward position, this movement should be assigned a positive 'sense'. Following this initial step, it is essential to ensure that motions across all other PDs are defined consistently with this reference. For instance, if the ψ_1 motion represents a down-to-up movement in a particular PD, it should be assigned a negative 'sense' to align it with the reference and allow coherent analysis across PDs. Similarly, if this motion appears in ψ_2 instead of ψ_1 in another PD, it is necessary to designate ψ_2 as the anchor node to maintain consistency. This approach ensures that the defined motion propagates accurately into downstream analysis.

Manual sense determination is illustrated in the optical flow visualizations for the RyR1 and thyroglobulin data sets (Fig. 5). For the RyR1 data set, the anchor node ψ_2 is chosen instead of ψ_1, optimizing alignment with the conformational

changes within this orientation. Sparse transitions appear in the lower-occupancy PD (Fig. 5*a*), indicating limited sampling, while the higher-occupancy PD (Fig. 5*b*) shows smoother, continuous flows indicative of robust sampling. In the thyroglobulin data set, ψ_1 is selected as the anchor node to maintain coherent motion analysis across the conformational landscape. The lower-occupancy PD (Fig. 5*c*) displays sparse transitions, whereas the higher-occupancy PD (Fig. 5*d*) reveals a continuous, well defined flow, enhancing the structural clarity.

2.7. Probability distribution estimation and volume reconstruction

Once belief propagation has been performed, the final step is generating the probability distribution and performing the volume reconstruction along a 1D trajectory within this distribution. Below, the steps for this are described in detail.

2.7.1. Parameterization and state occupancy

For each selected PD, the conformational trajectory data, represented by the eigenvector τ, are divided into N uniformly spaced bins. Each bin corresponds to a unique NLSA-derived conformational state, and the occupancy of each state, n_s, is determined by counting the number of snapshots that fall within each bin. The cumulative occupancy, $h_{\rm un}$, across all PDs is computed according to

$$h_{\rm un} = \sum_{x=1}^{x_{\rm active}} h_x, \tag{25}$$

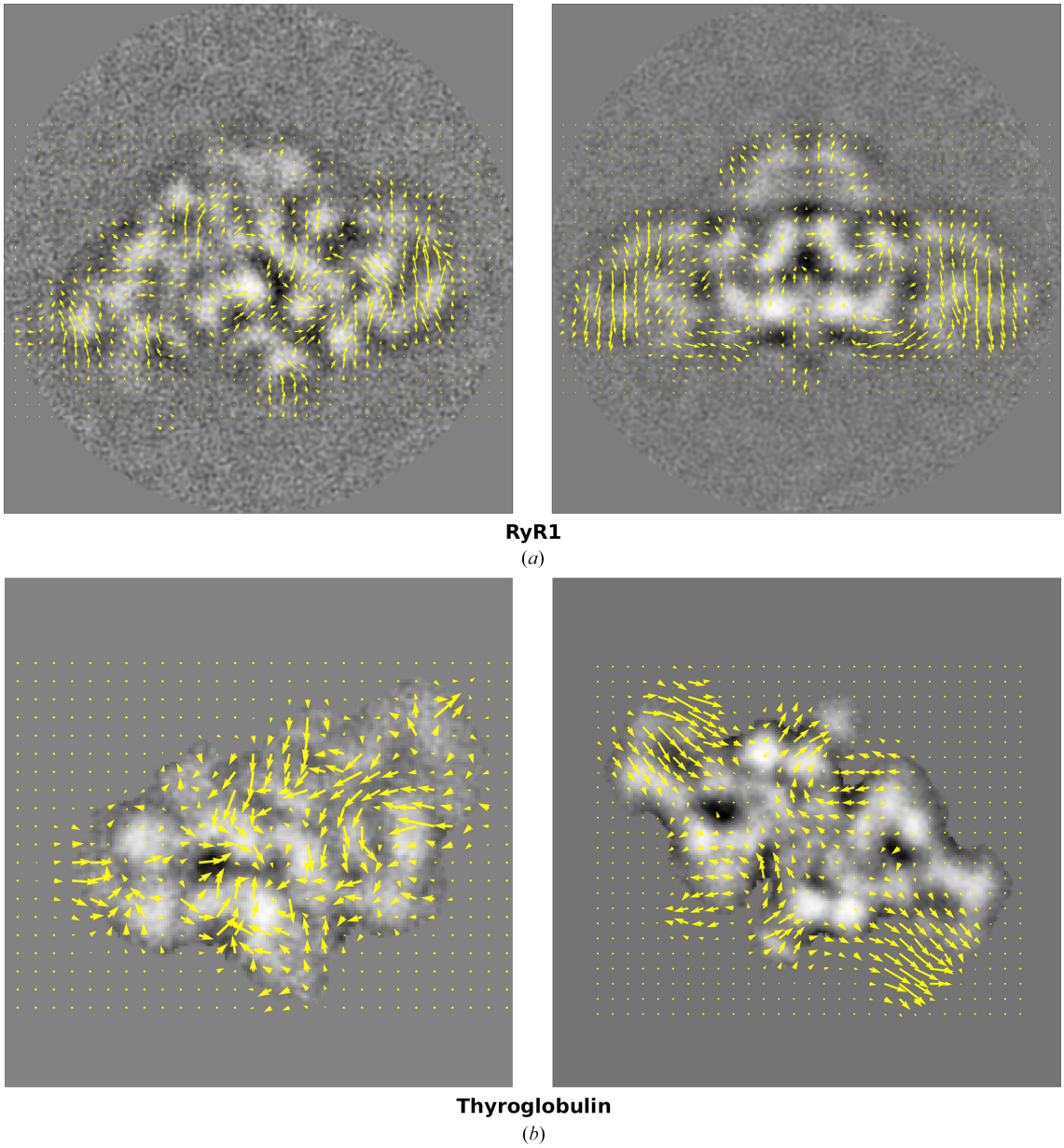


Figure 5
Optical flow visualization for the RyR1 and thyroglobulin projection directions. (*a*) RyR1 PD with 117 images. (*b*) RyR1 PD with 565 images. (*c*) Thyroglobulin PD with 250 images. (*d*) Thyroglobulin PD with 994 images.

where h_x is the histogram of τ values for the xth PD and x_{active} denotes the number of active PDs.

2.7.2. Normalization of trajectory data

To maintain consistency across PDs and ensure that conformational states are uniformly represented across the energy landscape, each trajectory τ is normalized to the range [0, 1] (equation 26).

$$\tau' = \frac{\tau - \min(\tau)}{\max(\tau) - \min(\tau)}. \tag{26}$$

2.7.3. Probability distribution of conformational states

With the occupancy h_{un} calculated, the probability P_s of each state s is given by

$$P_s = \frac{n_s}{n_0}, \tag{27}$$

where n_s is the occupancy of state s and n_0 is the maximum occupancy across all states. To avoid singularities, the normalized state occupancy, $\rho = \max(h_{\text{un}}, 1)$, is used, where h_{un} represents the unnormalized occupancy for a given state. This ensures that ρ is always at least 1, preventing issues with logarithmic calculations and enabling robust estimation of state probabilities across the conformational landscape.

`manifold-cli calc-probabilities`: Probability distributions of conformational states

The `calc-probabilities` command computes the probability landscape from the manifold embedding to represent relative state occupancies across the conformational space.

```
manifold-cli calc-probabilities input_file.toml
```

where
`--con_order_range`: Coarse-graining factor for binning the probability landscape (default: 50)
`--states_per_coord`: Number of states partitioned within each 1D reaction coordinate (default: 50)

`manifold-cli trajectory`: Calculate conformational trajectories

The `trajectory` command calculates the conformational pathways and reconstructs the transitions between different states, offering a dynamic view of the structural changes of the system over time.

```
manifold-cli trajectory input_file.toml
```

where

`--nlsa_fps`: Frames per second for the generated movies

Figs. 6(*a*) and 6(*b*) depict the probability distributions of conformational states for RyR1 and thyroglobulin along their respective reaction coordinates. Key conformational states are annotated with markers at regions of high and low occupancy, highlighting transitions between distinct states.

2.7.4. Optional: free-energy calculation

With the occupancy, h_{un}, in hand, one can calculate the free energy, ΔG, of each state by employing the Boltzmann factor. The relative free energy, ΔG, of each state, s, is given by

$$\frac{\Delta G}{k_{\text{B}}T} = -\ln\left(\frac{n_s}{n_0}\right), \tag{28}$$

where n_s is the occupancy of the current state s, n_0 is the maximum occupancy across states, k_{B} is the Boltzmann constant (1.987×10^{-3} kcal mol^{-1} K^{-1}) and T is the temperature in kelvin. The Boltzmann factor yields a free energy E for each state given by

$$E = -k_{\text{B}}T\ln(\rho), \tag{29}$$

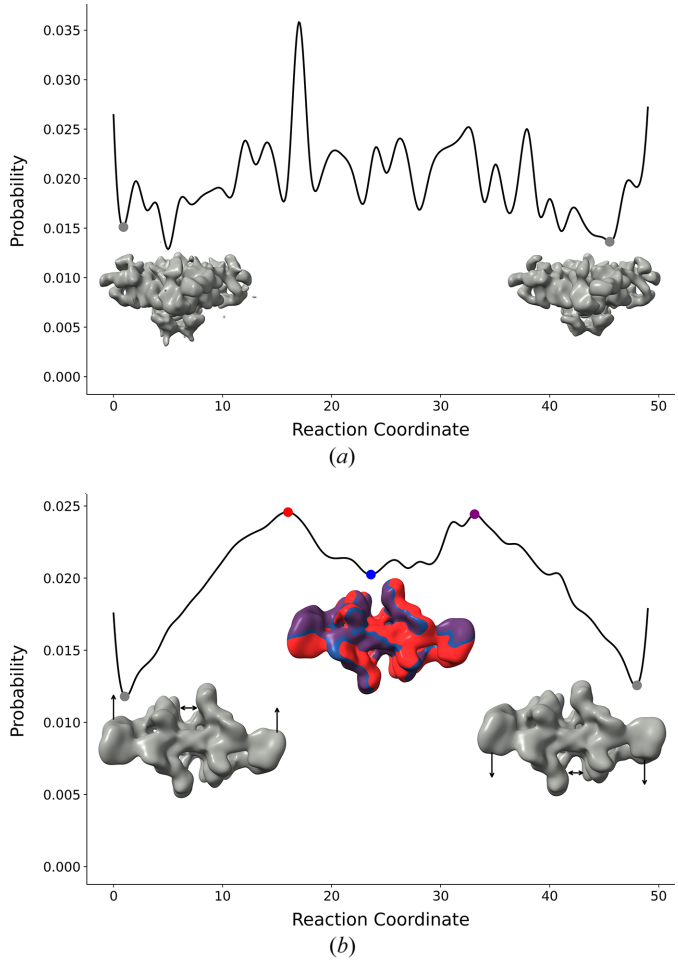


Figure 6
(*a*) Probability distribution of RyR1 conformations along the reaction coordinate, representing the relative occupancy of states derived from *ManifoldEM* analysis. Key states are shown in gray, with volume renderings depicting the structural features of these states. (*b*) Probability distribution of thyroglobulin conformations along the reaction coordinate. Key states are marked with colored indicators: gray for low-occupancy states, red and purple for frequently sampled states (local peaks in probability) and blue for a state with intermediate occupancy. Inset volume renderings, collapsed into a single panel for frequently sampled states, illustrate conformational transitions inferred from the probability landscape. Single and double arrows represent unidirectional and bidirectional transitions, respectively.

where $\rho = \max(h_{un}, 1)$ to avoid singularities and k_BT is the thermal energy. Note that free-energy profiles are not calculated by default, as this depends on the assumption that a sample is representative of the equilibrium distribution at a known temperature, the validity of which has been brought into question (Grübmuller & Bock, 2023). However, the user still has access to this calculation, as it can be derived from the probability distribution with the above equation.

2.7.5. Volume reconstruction

Volume reconstruction involves generating 3D volumes from the images corresponding to the selected points along the 1D coordinate of the probability distribution of conformational states (50 points in the case of RyR1 and thyroglobulin). This process employs the pseudo images and pseudo STAR files produced by *ManifoldEM*. Tools for volume reconstruction are provided using the `relion_reconstruct` command. However, alternative reconstruction methods can be used, as this calculation is external to *ManifoldEM*.

`manifold-cli utility mrcs2mrc`: Volume reconstruction

The `mrcs2mrc` utility converts a stack of .mrcs files into a reconstructed volume along the 1D coordinate, facilitating visualization and post-analysis of 3D conformations derived from projection images.

```
manifold-cli utility mrcs2mrc input_file.toml
```

3. Overview of changes

3.1. Usability improvements

ManifoldEM is now pip-installable, ensuring a smoother setup across diverse computing environments. The addition of `manifold-cli` enables direct command-line execution, enhancing flexibility for automation and scripting beyond GUI reliance. Configuration files are standardized in Tom's obvious minimal language (`.toml`) format, making parameters easily editable and accessible for both end-users and developers. Stability improvements address frequent crashes, particularly in handling large PD clusters, resulting in a more stable software experience. *ManifoldEM* accommodates multiple projects within a single directory, with outputs organized systematically for straightforward data handling and streamlined workflows. A remote visualization mode (`manifold-gui -V`) allows 3D visualizations to be disabled, conserving bandwidth, reducing system demands, and allowing remote execution of the GUI via X-forwarding in headless environments. Further optimizations include enhanced plotting routines with customizable division planes that exploit image symmetry for faster processing and tailored visualization control.

3.2. Pipeline optimization

The adoption of Numba, a just-in-time compiler, combined with optimized mathematical computations using NumPy, has substantially accelerated the processing speed of the pipeline. Optimizations to the `rotate_fill` routines deliver up to a tenfold speedup in image handling, providing efficient processing for large data sets. Replacing the histogram of oriented gradients (HOG) function from SciPy with a custom-built FastHOG library achieves a 100-fold performance boost, enabling rapid image analysis across high-volume data sets. This library is hosted on GitHub by the Flatiron Institute at https://github.com/flatironinstitute/fasthog. The pipeline now includes parallel processing capabilities in every computationally intensive step, allowing major steps to execute simultaneously while reducing the overall runtime. Storage requirements have been optimized to reduce file size and count while minimizing I/O operations, further decreasing the data footprint and improving speed. The image-output process is streamlined by utilizing imageio in place of Matplotlib, eliminating unnecessary overhead and enhancing the performance of image generation and saving. The pipeline has been optimized to process large-scale data sets efficiently. A single-particle data set containing 1.3 million particles with 280 × 280 pixel images was processed on AMD EPYC 9004 Series (Genoa) processors with 96 cores, completing in approximately 4–6 h and demonstrating the capability of the pipeline to handle modern cryo-EM data sets.

3.3. Developer-focused enhancements

Extensive code cleanup eliminated thousands of lines of redundant code, enhancing maintainability and efficiency for streamlined modifications. The GUI is refactored with a modular, developer-friendly structure, facilitating easier customization. Parameter management is now unified, with centralized storage that removes code duplication, simplifies data organization and improves usability. Automated help generation for CLI commands provides current guidance and boosts accessibility for developers. Project metadata, such as rotation, projection directions and image indices, are now centrally stored, enhancing organization and accessibility. Conjugate image handling is optimized using transformation flags and half-S^2 representations, aligning directly with the input stack and simplifying workflows. Comprehensive function documentation and added type hints increase code clarity, supporting effective collaboration on future development.

4. Discussion and conclusions

ManifoldEM adopts a distinctive approach to cryo-EM heterogeneity by using per-PD analysis, enabling detailed conformational reconstruction within orientation-specific subspaces. That is, the conformational states are recovered for each PD, setting *ManifoldEM* apart from other methods. *ManifoldEM* requires moderate CPU resources, achieving high-resolution conformational insights without the extensive GPU demands of deep learning-based tools. The employment of NLSA with Gaussian kernel-based Laplacian matrices in *ManifoldEM* enhances its ability to map conformational heterogeneity and produce interpretable free-energy landscapes, which are essential for connecting structural and energetic features. Further additions within this Python

framework, such as the implementation of 2D landscapes, multi-data-set analysis and support for tomography data as inputs, are clear next steps. To facilitate all of these, the groundwork presented here, including CLI and GUI modes, TOML configuration and improved modularity, make it adaptable across computing environments and support future integration with well supported Python tools.

The versatility of *ManifoldEM*, especially in this developer-friendly implementation, and its focus on per-PD analysis make it well suited for extensions. One obvious example is explicit integration with MD simulation. The benefits of combining MD and *ManifoldEM* have been exploited in past work on RyR1 (Dashti *et al.*, 2020) and the SARS-CoV-2 spike protein (Sztain *et al.*, 2021). However, a practical integration of the two methods for wider use to tackle biological problems remains to be performed. A related recent advancement in this area is the use of Bayesian ensemble reweighting to obtain free-energy landscape estimation by reweighting MD-generated distributions to match that present in a single-particle cryo-EM data set of the same system (Tang, Silva-Sánchez *et al.*, 2023). This is a compelling demonstration of the promise of combining MD and cryo-EM to understand the conformational heterogeneity present in these experimental data sets. However, per-particle noise and other aspects of the process of conducting ensemble reweighting using cryo-EM particle images suggest that a per-PD approach may be advantageous. Furthermore, biasing MD simulations directly by the outputs of *ManifoldEM* offers a powerful framework for capturing biomolecular conformational landscapes: merging cryo-EM data with simulation-driven kinetics and thermodynamics. Enhanced simulation methods, such as milestoning (Ojha, Votapka *et al.*, 2023; Votapka *et al.*, 2022; Ojha, Srivastava *et al.*, 2023) and weighted ensemble (Zuckerman & Chong, 2017; Zwier *et al.*, 2015; Ojha, Thakur *et al.*, 2023) simulations for kinetic and thermodynamic analysis, would reveal dynamic behaviors across *ManifoldEM*-identified conformational states.

Since the first application of *ManifoldEM* to a single-particle cryo-EM data set (Dashti *et al.*, 2014), many other approaches to continuous conformational heterogeneity in cryo-EM have been proposed (Tang, Zhong *et al.*, 2023). However, in recent studies using an artificial data set as a controlled ground truth (Dsouza *et al.*, 2023) and a well studied experimental single-particle cryo-EM data set of TRPV1 (Astore *et al.*, 2024) *ManifoldEM* still performs on a par with or, at times, even better than the state-of-the-art methods such as *cryoDRGN* (Zhong *et al.*, 2021). An indicator of the growth of this field is the recent push for further benchmarking (Joosten *et al.*, 2024; Jeon *et al.*, 2024) and challenges (Astore *et al.*, 2023) for heterogeneity in cryo-EM. Indeed, as the frontiers of cryo-EM push into *in situ* and cryo-electron tomography, as well as time-resolved cryo-EM (Mäeots & Enchev, 2022), better methods for resolving heterogeneity and conformational landscapes with fewer and fewer particles will be critical to developing a mechanistic understanding of the molecular mechanisms of the biological world.

The *ManifoldEM* Python suite is an open-source repository hosted on GitHub by the Flatiron Institute at https://github.com/flatironinstitute/ManifoldEM.

Acknowledgements

We would like to acknowledge the support and encouragement of Joachim Frank and the previous work of those in his group who have worked on the *ManifoldEM* code. Similarly, this work is indebted to the previous work of Abbas Ourmazd and his group in building the original manifold embedding algorithm for cryo-EM implemented in *MATLAB*.

Funding information

The work of PS was supported in part by the US National Science Foundation under award STC1231306. The Flatiron Institute is a division of the Simons Foundation.

References

Astore, M. A., Blackwell, R., Silva-Sánchez, D., Cossio, P. & Hanson, S. M. (2024). *bioRxiv*, 2024.10.07.617120.
Astore, M. A., Woollard, G., Silva-Sánchez, D., Zhao, W., Duc, K. D., Grigorieff, N., Cossio, P. & Hanson, S. M. (2023). *The Inaugural Flatiron Institute Cryo-EM Heterogeneity Community Challenge.* https://doi.org/10.17605/OSF.IO/8H6FZ.
Beauchemin, S. S. & Barron, J. L. (1995). *ACM Comput. Surv.* **27**, 433–466.
Bhamre, T., Zhang, T. & Singer, A. (2016). *J. Struct. Biol.* **195**, 72–81.
Chu, X., Su, X., Liu, M., Li, L., Li, T., Qin, Y., Lu, G., Qi, L., Liu, Y., Lin, J. & Shen, Q.-T. (2022). *Proc. Natl Acad. Sci. USA*, **119**, e2111231119.
Citterio, C. E., Targovnik, H. M. & Arvan, P. (2019). *Nat. Rev. Endocrinol.* **15**, 323–338.
Cody, V. (1984). *Endocr. Res.* **10**, 73–88.
Coifman, R. R. & Lafon, S. (2006). *Appl. Comput. Harmon. Anal.* **21**, 5–30.
Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F. & Zucker, S. W. (2005). *Proc. Natl Acad. Sci. USA*, **102**, 7426–7431.
Crowther, R. A., DeRosier, D. J. & Klug, A. (1970). *Proc. R. Soc. London A*, **317**, 319–340.
Dalal, N. & Triggs, B. (2005). *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1, pp. 886–893. Piscataway: IEEE.
Dashti, A., Mashayekhi, G., Shekhar, M., Ben Hail, D., Salah, S., Schwander, P., des Georges, A., Singharoy, A., Frank, J. & Ourmazd, A. (2020). *Nat. Commun.* **11**, 4734.
Dashti, A., Schwander, P., Langlois, R., Fung, R., Li, W., Hosseinizadeh, A., Liao, H. Y., Pallesen, J., Sharma, G., Stupina, V. A., Simon, A. E., Dinman, J. D., Frank, J. & Ourmazd, A. (2014). *Proc. Natl Acad. Sci. USA*, **111**, 17492–17497.
Dsouza, R., Mashayekhi, G., Etemadpour, R., Schwander, P. & Ourmazd, A. (2023). *Sci. Rep.* **13**, 1372.
Giannakis, D. & Majda, A. J. (2012). *Proc. Natl Acad. Sci. USA*, **109**, 2222–2227.
Gilles, M. & Singer, A. (2024). *bioRxiv*, 2023.10.28.564422.
Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2020). *Commun. ACM*, **63**, 139–144.
Grant, T., Rohou, A. & Grigorieff, N. (2018). *eLife*, **7**, e35383.
Grübmuller, H. & Bock, L. V. (2023). *Biophys. J.* **122**, 488a.

Hart, J. C., Francis, G. K. & Kauffman, L. H. (1994). *ACM Trans. Graph.* **13**, 256–276.
Hu, M., Zhang, Q., Yang, J. & Li, X. (2020). *J. Struct. Biol.* **212**, 107601.
Jeon, M., Raghu, R., Astore, M., Woollard, G., Feathers, R., Kaz, A., Hanson, S. M., Cossio, P. & Zhong, E. D. (2024). *arXiv*:2408.05526.
Joosten, M., Greer, J., Parkhurst, J., Burnley, T. & Jakobi, A. J. (2024). *IUCrJ*, **11**, 951–965.
Jordanger, L. A. & Tjøstheim, D. (2022). *J. Am. Stat. Assoc.* **117**, 1010–1027.
Kermode, H., Williams, A. J. & Sitsapesan, R. (1998). *Biophys. J.* **74**, 1296–1304.
Kingma, D. P. & Welling, M. (2019). *Mach. Learn.* **12**, 307–392.
Lafon, S. S. (2004). PhD thesis. Yale University.
Liu, Y. (2018). PhD thesis. University of Illinois at Urbana-Champaign.
Mäeots, M.-E. & Enchev, R. I. (2022). *Acta Cryst.* D**78**, 927–935.
Maji, S., Liao, H., Dashti, A., Mashayekhi, G., Ourmazd, A. & Frank, J. (2020). *J. Chem. Inf. Model.* **60**, 2484–2491.
McPherson, P. S., Kim, Y.-K., Valdivia, H., Knudson, C. M., Takekura, H., Franzini-Armstrong, C., Coronadot, R. & Campbell, K. P. (1991). *Neuron*, **7**, 17–25.
Ng, A., Jordan, M. & Weiss, Y. (2001). *NIPS'01: Proceedings of the 15th International Conference on Neural Information Processing Systems: Natural and Synthetic*, edited by T. Dietterich, S. Becker & Z. Ghahramani, pp. 849–856. Cambridge: MIT Press.
Ogawa, Y. (1994). *Crit. Rev. Biochem. Mol. Biol.* **29**, 229–274.
Ojha, A. A., Srivastava, A., Votapka, L. W. & Amaro, R. E. (2023). *J. Chem. Inf. Model.* **63**, 2469–2482.
Ojha, A. A., Thakur, S., Ahn, S.-H. & Amaro, R. E. (2023). *J. Chem. Theory Comput.* **19**, 1342–1359.
Ojha, A. A., Votapka, L. W., Huber, G. A., Gao, S. & Amaro, R. E. (2023). *Living J. Comput. Mol. Sci.* **5**, 2359.
Otte, M. & Nagel, H. H. (1994). *Computer Vision - ECCV '94*, edited by J.-O. Eklundh, pp. 49–60. Berlin, Heidelberg: Springer-Verlag.
Punjani, A. & Fleet, D. J. (2021). *J. Struct. Biol.* **213**, 107702.
Punjani, A., Rubinstein, J. L., Fleet, D. J. & Brubaker, M. A. (2017). *Nat. Methods*, **14**, 290–296.
Scheres, S. H. W. (2012). *J. Struct. Biol.* **180**, 519–530.
Scheres, S. H. W. & Chen, S. (2012). *Nat. Methods*, **9**, 853–854.
Schwander, P., Fung, R. & Ourmazd, A. (2014). *Phil. Trans. R. Soc. B*, **369**, 20130567.
Seitz, E., Acosta-Reyes, F., Maji, S., Schwander, P. & Frank, J. (2022). *IEEE Trans. Comput. Imaging*, **8**, 462–478.
Seitz, E., Frank, J. & Schwander, P. (2023). *Digit. Discov.* **2**, 702–717.
Sigworth, F. J. (1998). *J. Struct. Biol.* **122**, 328–339.
Stewart, M. (1990). *Modern Microscopies: Techniques and Applications*, edited by P. J. Duke & A. G. Michette, pp. 9–39. New York: Plenum Press.
Sztain, T., Ahn, S.-H., Bogetti, A. T., Casalino, L., Goldsmith, J. A., Seitz, E., McCool, R. S., Kearns, F. L., Acosta-Reyes, F., Maji, S., Mashayekhi, G., McCammon, J. A., Ourmazd, A., Frank, J., McLellan, J. S., Chong, L. T. & Amaro, R. E. (2021). *Nat. Chem.* **13**, 963–968.
Tang, W. S., Silva-Sánchez, D., Giraldo-Barreto, J., Carpenter, B., Hanson, S. M., Barnett, A. H., Thiede, E. H. & Cossio, P. (2023). *J. Phys. Chem. B*, **127**, 5410–5421.
Tang, W. S., Zhong, E. D., Hanson, S. M., Thiede, E. H. & Cossio, P. (2023). *Curr. Opin. Struct. Biol.* **81**, 102626.
Tomasi, C. (2012). *Computer Vision Sampler*, pp. 1–6.
Van Heel, M. & Frank, J. (1981). *Ultramicroscopy*, **6**, 187–194.
Van Petegem, F. (2012). *J. Biol. Chem.* **287**, 31624–31632.
Votapka, L. W., Stokely, A. M., Ojha, A. A. & Amaro, R. E. (2022). *J. Chem. Inf. Model.* **62**, 3253–3262.
Yedidia, J. S., Freeman, W. T. & Weiss, Y. (2000). *NIPS'00: Proceedings of the 14th International Conference on Neural Information Processing Systems*, pp. 668–674. Cambridge: MIT Press.
Yedidia, J. S., Freeman, W. T. & Weiss, Y. (2003). *Exploring Artificial Intelligence in the New Millennium*, edited by G. Lakemeyer & B. Nebel, pp. 239–269. San Francisco: Morgan Kaufmann.
Zhong, E. D., Bepler, T., Berger, B. & Davis, J. H. (2021). *Nat. Methods*, **18**, 176–185.
Zhu, J., Penczek, P. A., Schröder, R. & Frank, J. (1997). *J. Struct. Biol.* **118**, 197–219.
Zuckerman, D. M. & Chong, L. T. (2017). *Annu. Rev. Biophys.* **46**, 43–57.
Zwier, M. C., Adelman, J. L., Kaus, J. W., Pratt, A. J., Wong, K. F., Rego, N. B., Suárez, E., Lettieri, S., Wang, D. W., Grabe, M., Zuckerman, D. M. & Chong, L. T. (2015). *J. Chem. Theory Comput.* **11**, 800–809.