PLOS Computational Biology

RESEARCH ARTICLE

Automated registration and clustering for enhanced localization atomic force microscopy of flexible membrane proteins

Creighton M. Lisowski[1], **Gavin M. King**[1,2], **Ioan Kosztin**[1]*

1 Department of Physics and Astronomy, University of Missouri, Columbia, Missouri, United States of America, **2** Department of Biochemistry, University of Missouri, Columbia, Missouri, United States of America

* kosztini@missouri.edu

OPEN ACCESS

Citation: Lisowski CM, King GM, Kosztin I (2025) Automated registration and clustering for enhanced localization atomic force microscopy of flexible membrane proteins. PLoS Comput Biol 21(12): e1013277. https://doi.org/10.1371/journal.pcbi.1013277

Editor: Nir Ben-Tal, Tel Aviv University, ISRAEL

Received: June 28, 2025

Accepted: November 17, 2025

Published: December 1, 2025

Data availability statement: Yes - all data are fully available without restriction The data can be freely downloaded at: https://github.com/Kosztin-Research-Group/AFMpy.

Abstract

Atomic Force Microscopy (AFM) can create images of biomolecules under near-native conditions but suffers from limited lateral resolution due to the finite AFM tip size and recording frequency. The recently developed Localization Atomic Force Microscopy or LAFM (Heath et al., Nature 594, 385 (2021)) enhances lateral resolution by reconstructing peak positions in AFM image stacks, but it is less effective for flexible proteins with multiple conformations. Here we introduce an unsupervised deep learning algorithm that simultaneously registers and clusters images by protein conformation, thus making LAFM applicable to more flexible proteins. Using simulated AFM images from molecular dynamics simulations of the SecYEG translocon as a model membrane protein system, we demonstrate improved resolution for individual protein conformations. This work represents a step towards a more general LAFM algorithm that can handle biological macromolecules with multiple distinct conformational states such as SecYEG.

Author summary

Atomic Force Microscopy (AFM) enables high-resolution imaging of biomolecules under near-native conditions but faces lateral resolution limits due to the finite AFM tip size and recording frequency. The recently developed Localization Atomic Force Microscopy (LAFM) method addresses this by reconstructing peak positions from AFM image stacks, achieving almost atomic resolution for membrane proteins like Annexin-V or aquaporin-Z that adopt a single predominant conformation (Heath et al., Nature 594, 385 (2021)). While these proteins exhibit intrinsic flexibility – including conformational adaptation upon membrane binding for Annexin-V and flexible extracellular loops in aquaporin-Z – their protruding regions consistently display a well-defined structural state in AFM images, similar to what would be observed for truly rigid proteins. However, flexible membrane proteins with

Funding: This work was supported by the US National Science Foundation Award #: 2122027 (to GMK and IK). The funder had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

multiple dynamic conformations, such as the SecYEG translocon, which exhibits large and highly mobile cytoplasmic loops that sample distinct conformational states, lead to non-physical smearing in standard LAFM reconstructions. Here, we present a computational framework combining unsupervised deep clustering and LAFM to enhance the lateral resolution of AFM images of flexible membrane proteins. Our neural network algorithm (i) groups AFM images into conformationally homogeneous clusters and (ii) registers images within each cluster. Applying LAFM separately to these clusters minimizes smearing artifacts, yielding high-resolution reconstructions for distinct conformations. We validate this approach using synthetic AFM images generated from all-atom molecular dynamics simulations of SecYEG in a solvated POPE lipid bilayer. This advancement extends LAFM's utility to encompass conformationally diverse proteins.

Introduction

Atomic Force Microscopy (AFM) is a widely used imaging technique for studying membrane protein systems at the single-molecule level in near-native conditions. AFM images are topographic maps generated by scanning surfaces with a sharp probe, achieving near-atomic resolution in the vertical (Z) direction on many biomolecular systems [1,2]. However, the lateral resolution (in the X-Y plane) of these images is constrained by the AFM tip geometry and size as well as the sampling discretization (recording frequency). To address these limitations, Localization Atomic Force Microscopy (LAFM) [3] has emerged as a promising post-acquisition image reconstruction method that enhances in-plane resolution.

LAFM enhances the lateral resolution of conventional AFM images by computationally reconstructing molecular details from a stack of multiple AFM images. In standard AFM imaging, lateral resolution is limited by the finite size of the scanning tip, causing structural details smaller than several nanometers to appear blurred. LAFM overcomes this limitation by exploiting subtle positional fluctuations of surface features across multiple registered AFM images. Specifically, it identifies and precisely localizes peaks corresponding to protein substructures in each individual image within the stack. By statistically combining these localized positions, LAFM reconstructs a higher-resolution image that reveals structural details previously obscured by tip geometry and sampling constraints. This approach significantly improves lateral resolution, enabling visualization of molecular conformations and sub-nanometer features that are inaccessible with traditional AFM analysis.

While effective for protein systems with a single predominant conformation [3], the direct application of LAFM to flexible proteins that undergo significant conformational changes is problematic. Indeed, in spite of its enhanced lateral resolution, the LAFM image obtained from a stack of AFM images corresponding to different protein conformations can be misleading as it blends distinct conformational states together. To mitigate this issue and generalize LAFM to more flexible membrane proteins, here we employ a two-pronged approach consisting of: (i) clustering AFM images based on conformational states, by employing deep learning (DL) algorithms, and (ii) applying

LAFM selectively within clusters. Given a stack of AFM images of membrane protein extracellular domains, our approach effectively discriminates between different proteins, their membrane sidedness, and their distinct conformations. Similar techniques have demonstrated success in the conformational clustering and classification of biomolecular complexes. For instance, CryoDRGN [4] employs an unsupervised DL algorithm to assist in reconstructing 3D density maps from cryo-EM data, effectively capturing and partitioning a continuous conformational landscape. Likewise, HORNET [5] leverages DL to resolve structures of flexible RNA molecules from AFM images. These approaches highlight the versatility and effectiveness of unsupervised DL algorithms, proving them a powerful tool within nanoscale imaging workflows.

Synthetic AFM images generated through molecular dynamics (MD) simulations offer a powerful approach to study flexible proteins under controlled conditions and enable direct registration and benchmarking [6]. Furthermore, the rapid innovation of GPU-accelerated computing allows for the calculation of synthetic AFM images at remarkable speeds [7]. The present study employs all-atom MD simulations of translocon SecYEG embedded in a POPE lipid bilayer. We show how LAFM can be extended to resolve structural details of this flexible membrane protein by combining LAFM methodology with clustering techniques.

Methods

Molecular modeling and MD simulation

To generate a large stack of synthetic AFM images, we performed $\sim 1\mu$s all-atom MD simulation of the protein-conduction channel SecYEG, embedded in a fully solvated 1-palmitoyl-2-oleoyl-phosphatidylethanolamine (POPE) lipid bilayer. The universally conserved SecYEG translocon is a heterotrimeric complex (composed of subunits SecY, SecE, and SecG) that plays a crucial role in mediating the transport of newly synthesized proteins either into or across the bacterial cytoplasmic membrane. The translocon SecYEG was chosen as a model membrane protein for this study because it exhibits large membrane-external loops on the cytoplasmic side of the membrane that are highly dynamic and give rise to distinct conformational states [2,8,9].

The crystal structure of SecYEG (PDB code 3DIN [10]), along with its orientation in the lipid bilayer, was obtained from the OPM database [11]. This structure originally contained both SecYEG and SecA, but we removed all SecA atoms to focus solely on the SecYEG complex. The CHARMM-GUI membrane builder [12–14] was employed to assemble a protein-lipid system. ACE (N-acetyl) and CT2/NME terminal caps were applied to neutralize the peptide termini in the simulation, preventing spurious electrostatic interactions at the membrane–water interface. A homogeneous POPE lipid bilayer, representing about 75% of the native *E. coli* membrane [15], was selected to embed SecYEG while avoiding the complexity of modeling a heterogeneous bilayer. The system was subsequently solvated in a 0.03 M NaCl aqueous solution with 3 nm water padding above and below the membrane.

Next, the SecYEG-POPE system was subjected to all-atom MD simulation using NAMD2 [16], employing the CHARMM36(m) force field [17,18]. Energy minimization and equilibration were carried out in multiple stages, gradually relaxing the restraints on the system in accordance with standard protocols [19]. A production run of unbiased MD simulation was performed at constant pressure (1 atm) and temperature (300 K), regulated by a Langevin piston barostat [20] and a Langevin thermostat [21], respectively. Atomic coordinates were saved every 0.1 ns over 1.2 μs simulation, generating 12,000 frames. To ensure the system was well equilibrated, the first 2,000 frames were discarded, leaving 1 μs of data (10,000 frames) for analysis.

During the simulation, SecYEG displayed translational and rotational drift along the membrane surface. To facilitate analysis, we created two trajectory versions: an unaligned (original) trajectory and an aligned trajectory. For the aligned version, using VMD [22], we minimized the root mean squared displacement (RMSD) of transmembrane helix backbone atoms relative to their initial positions. This alignment was essential since subsequent analyses rely on consistent protein positioning, with the aligned trajectory serving as our reference standard for comparisons.

Simulated AFM images

Simulated AFM (SimAFM) images were generated from all-atom MD trajectories (see Fig 1) using our open-source Python package, AFMpy [23]. The SimAFM implementation in AFMpy employs the MDAnalysis [24,25] library for trajectory processing, NumPy [26] for efficient vectorized calculation of tip contacts, and CuPy [27] for GPU acceleration.

Each frame from the MD trajectory is loaded and rotated so that the membrane side of interest faces the $+Z$ direction. The membrane plane is defined by the average Z-coordinate of the phosphorus atoms in the upper leaflet, which serves as the background. Protein atoms located above this background are represented as solid spheres with their van der Waals radii, as specified by the CHARMM force field [28]. The X–Y plane is then divided into a uniform two-dimensional grid according to the chosen scan boundaries and pixel resolution.

The simulated AFM tip is modeled as a cone with a spherical apex, characterized by its radius (R) and a half-angle $\theta = 18°$, matching the typical value for experimental AFM tips. To generate the SimAFM image, the tip is virtually scanned across each point of the grid, and the height at which it contacts the protein surface is recorded. This procedure is repeated for every frame in the trajectory, resulting in a stack of SimAFM images.

Stacks of SimAFM images were generated to visualize the protruding regions of SecYEG on both the cytoplasmic and periplasmic sides of the membrane, using both aligned and unaligned MD trajectories of the SecYEG-POPE system. To mimic a range of experimentally available AFM tips, the simulated tip radius was varied, including an idealized tip with

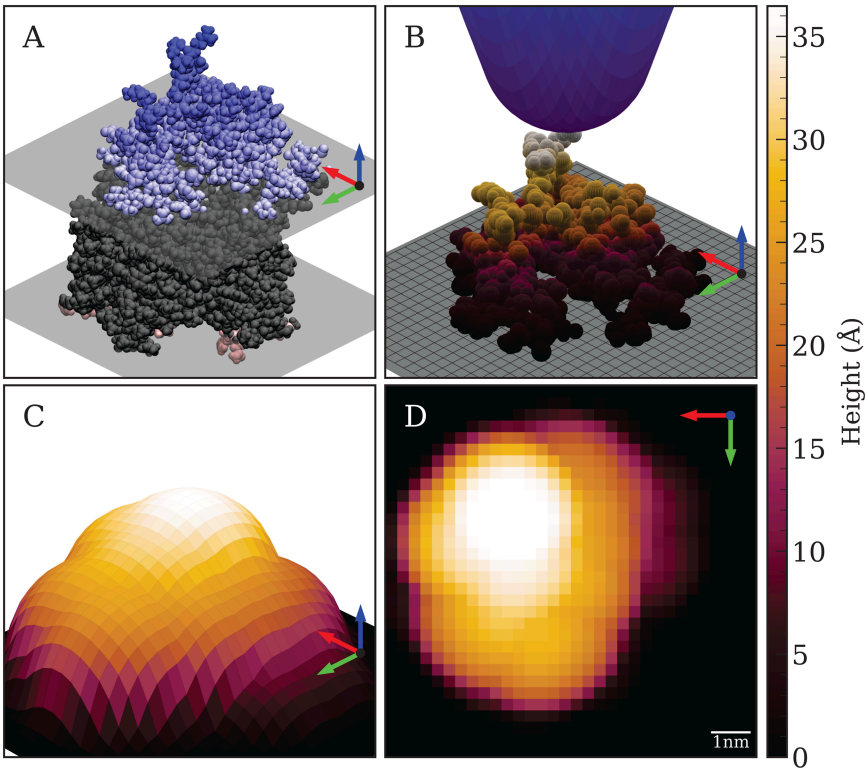

Fig 1. **Simulated Atomic Force Microscopy (SimAFM) of SecYEG. A.** Van der Waals representation of SecYEG rendered in VMD [22]. Average lipid head group positions and transmembrane protein regions are shown in gray, while only the cytoplasmic (blue) and periplasmic (red) protrusions of the protein are accessible to the AFM tip during scanning. **B.** Simulated AFM tip (blue) contacting the cytoplasmic side of SecYEG, with tip contact calculated at each grid point across the X-Y plane. **C.** Surface representation of the SimAFM scan, showing how molecular details are blurred by the finite geometry of the AFM tip. **D.** Top-down view of the SimAFM scan of the cytoplasmic side of SecYEG.

https://doi.org/10.1371/journal.pcbi.1013277.g001

$R = 2$Å. Scan resolution was also systematically adjusted, producing SimAFM stacks across a broad spectrum of resolutions. Specifically, tip radii from 2Å to 80Å and scan resolutions from 2Å/px to 10Å/px were used.

Stacks were assigned a code reflecting their alignment status (Aligned: A; Unaligned: U), membrane side (Cytoplasmic: C; Periplasmic: P), tip radius (in Å), and scan resolution (in Å/px). For instance, a stack of aligned cytoplasmic images simulated with a 20Å tip and a 2Å/px scan resolution is labeled as AC–20–2 for easy identification. Notable stacks include those generated with tip radii of 2Å (approximating the width of a carbon atom), 20Å (representing the PeakForce-HIRS-F-B tip), and 80Å (representing the BioLever Mini AC40 tip).

Localization Atomic Force Microscopy (LAFM)

Each stack of images was processed to generate an enhanced lateral resolution image using our Python implementation of the LAFM algorithm [3]. The workflow consists of several steps: First, using OpenCV [29], the input image stack is enhanced to a higher, user-defined resolution by applying bicubic interpolation to expand each image. Next, peak detection is performed on each interpolated image to identify local maxima, which are then broadened using a Gaussian kernel of user-specified width, producing images where each peak is represented by a 2D Gaussian. The height of each Gaussian is scaled between 0 and 1 according to the detected real-space height relative to the maximum in the stack. These Gaussian images are then averaged pixel-wise across the entire stack to create a cumulative peaking probability image. Finally, this probability image is multiplied by the per-pixel averaged real-space height image to produce the final LAFM image.

Structural Similarity Index Measure (SSIM)

The quality of LAFM images relative to the benchmark (i.e., the LAFM images AP–2–2 and AC–2–2, respectively) was evaluated using the Structural Similarity Index Measure (SSIM) [30], which offers several advantages over simpler metrics such as mean squared error [31]. SSIM is particularly well-suited for this application because its focus on local structural similarity more closely reflects perceived visual quality. However, when applying SSIM to protein AFM images, special care is required: the uniform 0-height background in LAFM images creates large areas of perfect similarity, which can bias the SSIM calculation. To address this, a masked SSIM approach was used. A mask was generated by selecting pixels above a threshold set at 5% of the maximum height in the compared image pair, effectively isolating the protein features from the background. SSIM was then computed over the entire image, and the mask was applied to exclude background pixels. The mean value of the masked SSIM image was reported as the final similarity score.

LAFM images generated from each aligned simulation stack were compared to the benchmark using the masked SSIM score, with the results recorded in 2D arrays called LAFM quality profiles. These profiles are visualized as topographic maps (see Fig 5), illustrating LAFM quality as a function of tip radius (R) and scanning resolution (P).

Deep Spectral Clustering (DSC)

Given an aligned stack of images of a flexible protein exhibiting distinct structural conformations, it is necessary to first identify and separate (i.e., cluster) these conformations before applying the LAFM algorithm. In AFMpy, this conformational clustering is performed using a deep spectral clustering (DSC) algorithm, as illustrated in Fig 2. Our DSC implementation uses the Spectral Clustering module from scikit-learn [32] and employs the Keras API in TensorFlow [33,34] for constructing, training, and validating the deep learning models.

DSC assigns n conformational cluster labels to a stack of images by applying spectral clustering to latent feature vectors (LFVs) extracted by a convolutional autoencoder (CAE) [35]. The CAE is first trained end-to-end to minimize a reconstruction loss, defined as $1 - \text{SSIM}$, where SSIM is used as the similarity metric due to its effectiveness in capturing protein structural features. To properly account for both protein and background pixels, a weighted SSIM loss is employed

$$L(x, x') = \alpha \cdot \text{SSIM}(x, x') + \beta \cdot \text{SSIM}_{\text{masked}}(x, x'),$$

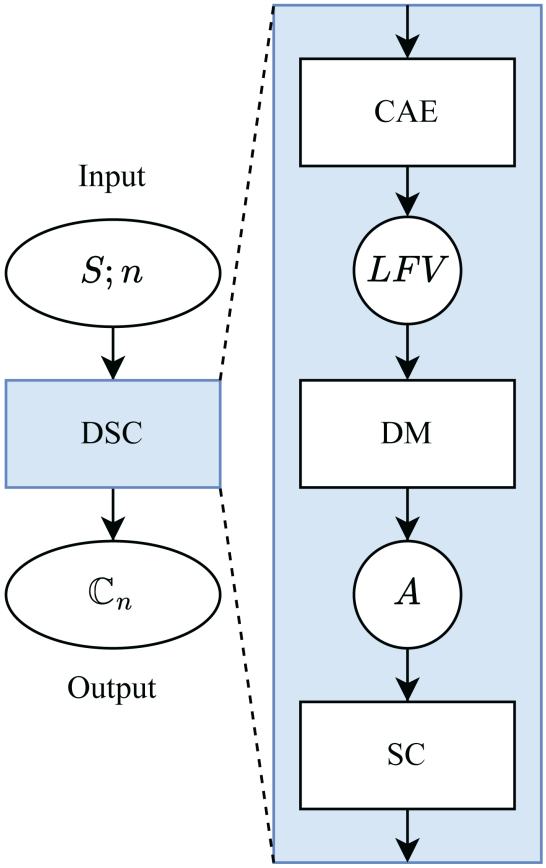

Fig 2. Deep Spectral Clustering (DSC) Algorithm: A convolutional autoencoder (CAE) is trained on input image stack (*S*) to extract latent feature vectors (*LFV*). *LFV* pairs are compared by a given distance metric (DM) to construct an affinity matrix (*A*). Spectral clustering (SC) is applied to *A*, returning *n* cluster labels for the input stack. Images are grouped by label into clusters (*C*) which make up the set of *n* clusters ($\mathbb{C}_n = \{C_1, C_2, \ldots, C_n\}$).

https://doi.org/10.1371/journal.pcbi.1013277.g002

combining standard SSIM (to address background) and masked SSIM (to focus on protein features). The relative weights of each SSIM component can be tuned to balance the contribution of background versus foreground pixels to reconstruction quality. Empirical testing showed that equal weighting performed well for our dataset.

After training, LFVs are extracted from the images, and an affinity matrix (A) is constructed by pairwise comparison of these LFVs using locally scaled affinity [36]. Spectral clustering is then performed on this matrix to assign cluster labels to the images.

Because the SimAFM stack captures a continuous range of motion, it contains both images of stable conformations and transient states during transitions. To separate these conformations, DSC was applied in a hierarchical manner. First, $n = 2$ DSC is applied to the stack, and LAFM images are then generated separately for each resulting cluster, henceforth referred to as clustered LAFM images. If the clustered LAFM images have high similarity, i.e., the masked SSIM exceeds a set threshold, the clusters are combined, and the process halts. If the two LAFM images differ, however, the affinity matrix is reclustered for $n = 3$, and new clustered LAFM images are generated. These $n = 3$ clustered LAFM images are cross-compared with the $n = 2$ LAFM images. Images with high similarity represent stable conformations, and the $n = 3$ labeled images are removed from the stack. This process iterates, using the reduced stack and affinity matrix as input, until no stable clusters are found, or the number of remaining unclustered images falls below a set minimum cluster size. We refer to this process as Hierarchical DSC (HDSC).

Registration and Clustering (REC)

Given an unaligned image stack, conformational clustering can be performed using the registration and clustering (REC) algorithm as outlined in Fig 3. The REC algorithm operates in two modes depending on the number of provided registration references. With a single reference and a specified number of clusters *n*, the algorithm registers all images to the reference and applies DSC to identify *n* registered clusters. Image registration was performed using the pystackreg Python package [37], applying rigid body transformations (translation and rotation only). This ensured alignment of images without scaling or shearing. If *n* registration references are provided, the stack is registered separately to each reference, generating *n* registered stacks. Each stack is then clustered independently using DSC, resulting in a superset of registered clusters. Clusters containing their respective registration reference images are considered well-registered and are retained for the final output. Additionally, within each cluster, the image with the highest silhouette score [38] is selected as an improved registration reference, yielding a refined set of registration references.

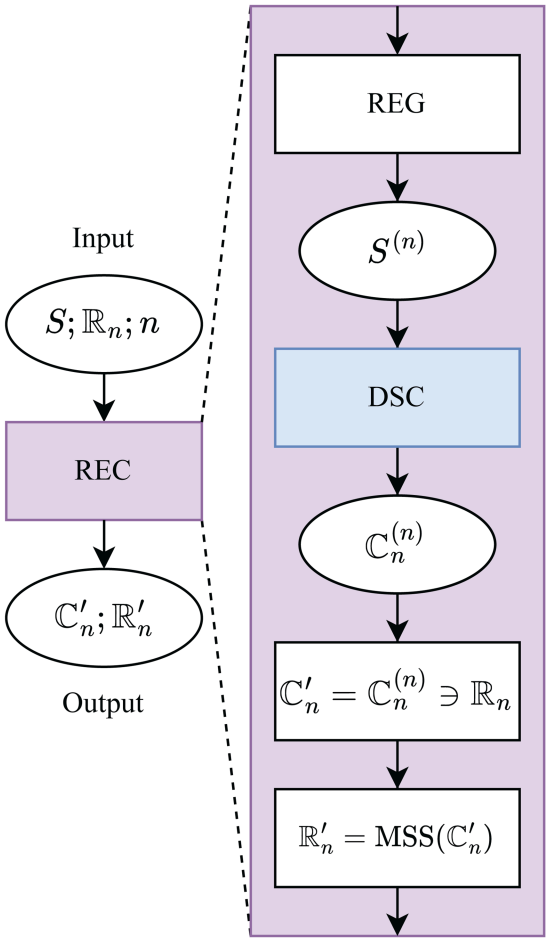

Fig 3. Registration and Clustering (REC) Algorithm: The input image stack (*S*) is registered (REG) using each image in the reference set $\mathbb{R}_n = \{R_1, R_2, \ldots, R_n\}$, resulting in *n* registered stacks ($S^{(n)}$). Each registered stack is then subjected to DSC clustering, producing a set of clusters ($\mathbb{C}_n$) for each reference. These cluster sets are combined into a registered cluster superset ($\mathbb{C}_n^{(n)} = \mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_n$), which contains the clusters generated for each registration reference. From this superset, only the clusters corresponding to their respective registration references ($\mathbb{C}_n^{(n)} \ni \mathbb{R}_n$) are retained to form the refined cluster set ($\mathbb{C}'_n$). Additionally, for each cluster, the image with the maximum *silhouette score* (MSS) is chosen to create a set of refined registration references ($\mathbb{R}'_n$).

https://doi.org/10.1371/journal.pcbi.1013277.g003

Iterative Registration and Clustering (IREC)

Selecting appropriate registration references and determining the optimal number of clusters are challenging tasks. To address this, we developed an iterative registration and clustering (IREC) algorithm. As outlined in Fig 4, IREC iteratively refines both cluster assignments and registration references to simultaneously optimize the registration references and the number of clusters present in a conformationally diverse AFM image stack. The IREC implementation leverages registration methods from the Python module pyStackReg [39] and clustering metrics from scikit-learn [32].

For a given n, the input image stack (S) is first processed by REC using an initial reference (R_0), producing n clusters ($\mathbb{C}_n$) and their corresponding references ($\mathbb{R}_n$). These clusters and references are then refined through iterative cycles: in each iteration, the stack is re-registered and re-clustered according to the refined references from the previous step, yielding new clusters and references. The newly generated references are compared to those from the preceding iteration; if discrepancies are found, the mismatched references are updated and the process repeats. Iteration continues until all references match, cyclic behavior is detected, or a maximum number of iterations is reached, at which point the clusters are considered optimal ($\mathbb{C}_n^\star$). Finally, LAFM is applied to each optimal cluster to generate the set of enhanced images.

Although IREC determines the optimal conformational clusters for a given n, it remains uncertain whether n is the optimal number of clusters required to partition the image stack. To assess the appropriate cluster count, the clustered LAFM images are compared as n increases. First, IREC is applied to the image stack to generate the corresponding LAFM images. Since these LAFM images are generated using different registration references, they may not be mutually

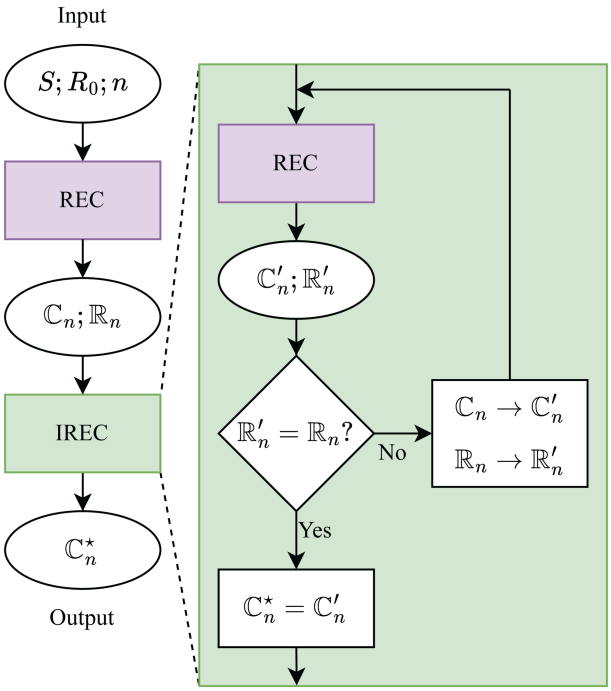

Fig 4. Iterative Registration and Clustering (IREC) Algorithm: The input image stack (S) is first aligned and grouped using registration and clustering (REC) based on an initial reference image (R_0). This yields an initial set of clusters ($\mathbb{C}_n$) and corresponding registration references ($\mathbb{R}_n$), which are then iteratively refined. In each iteration, every reference image in $\mathbb{R}_n$ is used to re-register and re-cluster S via REC, producing updated clusters ($\mathbb{C}'_n$) and new references ($\mathbb{R}'_n$). The new references are compared to those from the previous iteration. If they differ ($\mathbb{R}'_n \neq \mathbb{R}_n$), the references are updated ($\mathbb{R}_n \to \mathbb{R}'_n$) and the process repeats. Iteration continues until the references converge ($\mathbb{R}'_n = \mathbb{R}_n$), a cyclic pattern is detected, or a maximum number of iterations is reached, at which point the algorithm outputs the final, optimized cluster set ($\mathbb{C}_n^\star$).

https://doi.org/10.1371/journal.pcbi.1013277.g004

aligned. To evaluate their similarity, the LAFM images are registered to one another and compared using masked SSIM. If all SSIM scores fall below a predefined similarity threshold (e.g., 0.8), the enhanced images are considered to represent distinct conformations, indicating that at least n clusters are necessary. However, this test does not exclude the possibility that more clusters may be required, so IREC is reapplied with n incremented by one. This process is repeated for increasing values of n until any SSIM score between clustered LAFM images exceeds the similarity threshold, indicating that the optimal number of clusters for the image stack is n–1.

Computational performance

Benchmark computations were performed to assess the acceleration and scalability of AFMpy functions across CPU and GPU processing. All benchmarks ran on a Linux workstation with an Intel Xeon CPU (w9–3495X, with 56 cores) and two NVIDIA RTX A5500 GPUs (connected via NVLink). Table 1 summarizes execution times for SimAFM, DSC, Hierarchical DSC, and IREC in both CPU-only and GPU-enabled modes.

The 10,000-frame stacks of 32×32 simulated AFM images for periplasmic (110 protruding atoms) and cytoplasmic SecYEG (∼2000 protruding atoms) were generated using SimAFM from the 1 μs MD trajectory. Simulation time increased with system size; GPU acceleration provided greater speedup for the larger cytoplasmic SecYEG.

Benchmarking of DSC, Hierarchical DSC, and IREC functions on the cytoplasmic SecYEG stack showed that GPU acceleration substantially reduced computation times for all deep learning-based analyses, with speedups of 7.9–11.5. These functions employ TensorFlow (GPU-accelerated) and scikit-learn (CPU-based) components. Based on the obtained results, we recommend prioritizing GPU resources for AFMpy workflows, especially for deep learning components and SimAFM simulations of large systems or long trajectories.

Results and discussion

LAFM quality assessment

To quantify the influence of the AFM tip radius (R) and pixel resolution (P) on the quality of LAFM images, we compared images derived from the aligned cytoplasmic and periplasmic SecYEG SimAFM stacks to their respective benchmark LAFMs (from the AC–2–2 and AP–2–2 stacks) using masked SSIM. The results are shown as density plots in the left panel of Fig 5. These LAFM quality profiles, for both periplasmic and cytoplasmic SecYEG, show that the LAFM image quality increases nonlinearly with decreasing R and P. In both cases an $R \sim 2$ nm tip scanning at $P = 4$

A/px provide a good quality LAFM image (with SSIM≈0.7). While $R \sim 2$ nm represents a practical lower bound for current AFM tips, reducing P below 4 Å/px is achievable with state-of-the-art HS-AFM. As shown in Fig 5, for $P < 4$ Å/px, good quality periplasmic LAFM image can be obtained with $R > 2$ nm tips, whereas a good cytoplasmic LAFM image requires $R \sim 2$ nm tips regardless of further reductions in P. This indicates that for the cytoplasmic side, minimizing tip radius is essential and cannot be compensated by increased pixel resolution alone. Conversely, for the periplasmic side, comparable image quality can be achieved with larger tip radii by improving pixel resolution (reducing P), providing greater experimental flexibility. Thus, optimizing R is crucial for cytoplasmic imaging, while periplasmic imaging benefits from balancing both R and P.

Table 1. CPU vs. GPU runtime and speedup.

Process	CPU Time (s)	GPU Time (s)	GPU Speedup
Periplasmic SimAFM	43	31	1.4
Cytoplasmic SimAFM	344	68	5.1
DSC	376	33	11.5
Hierarchical DSC	392	50	7.9
IREC	4598	493	9.3

https://doi.org/10.1371/journal.pcbi.1013277.t001

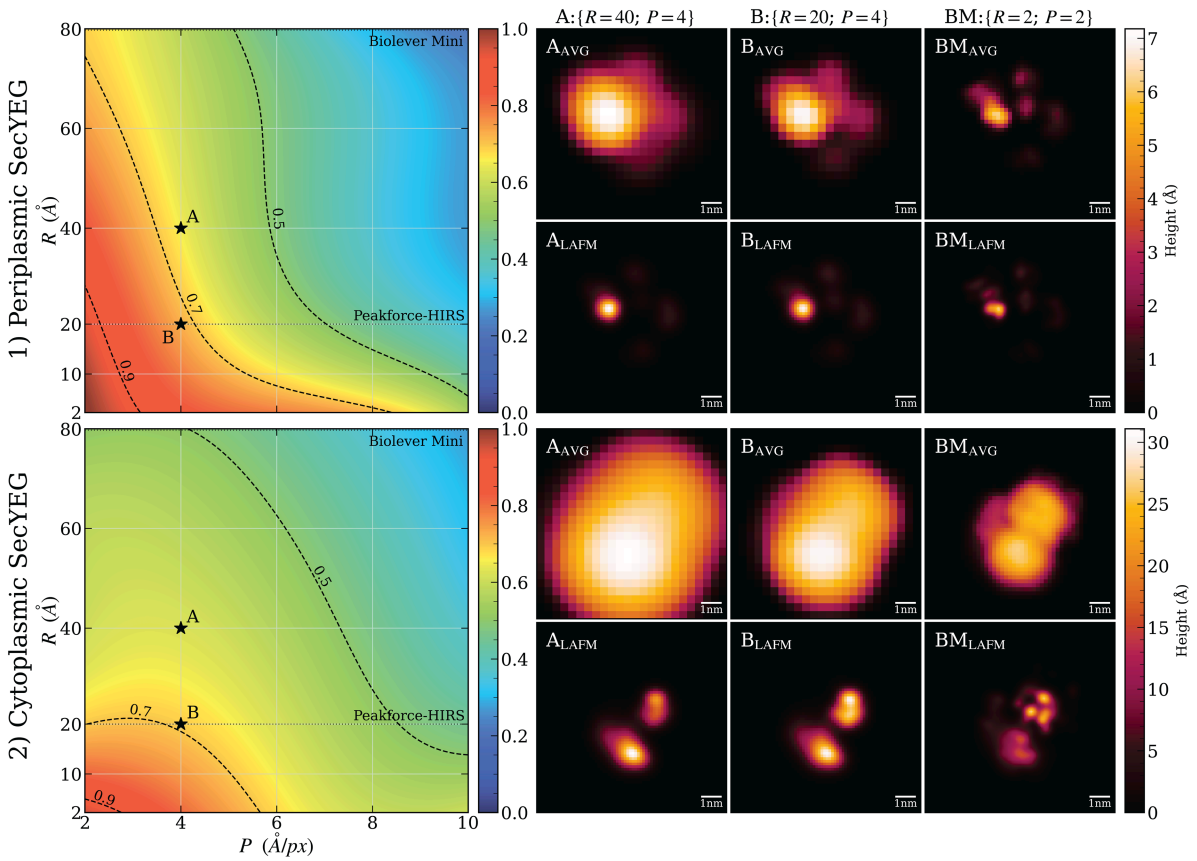

Fig 5. **Left: Quality profile of LAFM images for 1) periplasmic (top) and 2) cytoplasmic (bottom) SecYEG.** Color-coded density plots display masked SSIM scores of simulated LAFM images as functions of AFM tip radius (R) and pixel resolution (P), benchmarked against reference LAFM images acquired with $R = 2$ and $P = 2$. The dotted horizontal lines indicate the nominal radii of a commercial AFM tip (PeakForce-HIRS tip, $R = 20$). The radius of another common tip (Biolever Mini, $R = 80$), coincides with the top axes of the plots. Dashed contour lines at SSIM values of 0.9, 0.7, and 0.5 delineate the boundaries between "very good", "good", and "acceptable" LAFM quality regions. **Right:** Per-pixel averaged (AVG) AFM and LAFM images of 1) periplasmic (top) and 2) cytoplasmic (bottom) SecYEG for A:{$R = 40$; $P = 4$}, B:{$R = 20$; $P = 4$} and benchmark BM:{$R = 2$; $P = 2$} image stacks.

https://doi.org/10.1371/journal.pcbi.1013277.g005

As shown in the right panel of Fig 5, LAFM effectively resolves five distinct domains in periplasmic SecYEG using the AP–40–4 dataset (SSIM = 0.670) and further enhances feature accuracy in the AP–20–4 stack (SSIM = 0.748). For cytoplasmic SecYEG, the average AFM image appears as a featureless blob, whereas the corresponding LAFM image reveals distinct domains. However, this improvement is modest: the AC–40–4 image yields a relatively low SSIM of 0.632, and reducing the tip radius to 2 nm (AC–20–4) results in only a marginal increase to 0.662.

A notable challenge in the cytoplasmic case arises from conformational changes in the approximately 35-amino acid loops connecting transmembrane helices 6-7 and 8-9 within the SecYEG structure. These dynamic regions cause the bottom domain to appear smeared upward and to the left in the LAFM image, which inherently blends multiple structural states into a single composite representation. To address this limitation, deep spectral clustering approaches will be essential for distinguishing and separating the distinct conformational states present within the image stack, thereby enabling more precise structural characterization of dynamic membrane protein complexes.

Hierarchical deep spectral clustering results

Hierarchical DSC was applied to the cytoplasmic side images (AC–20–4), identifying four conformational clusters. The corresponding LAFM images are shown in Fig 6. Compared to the unclustered LAFM image, where different conformations are blended together, clustering clearly separates these states, with each cluster producing a unique LAFM image

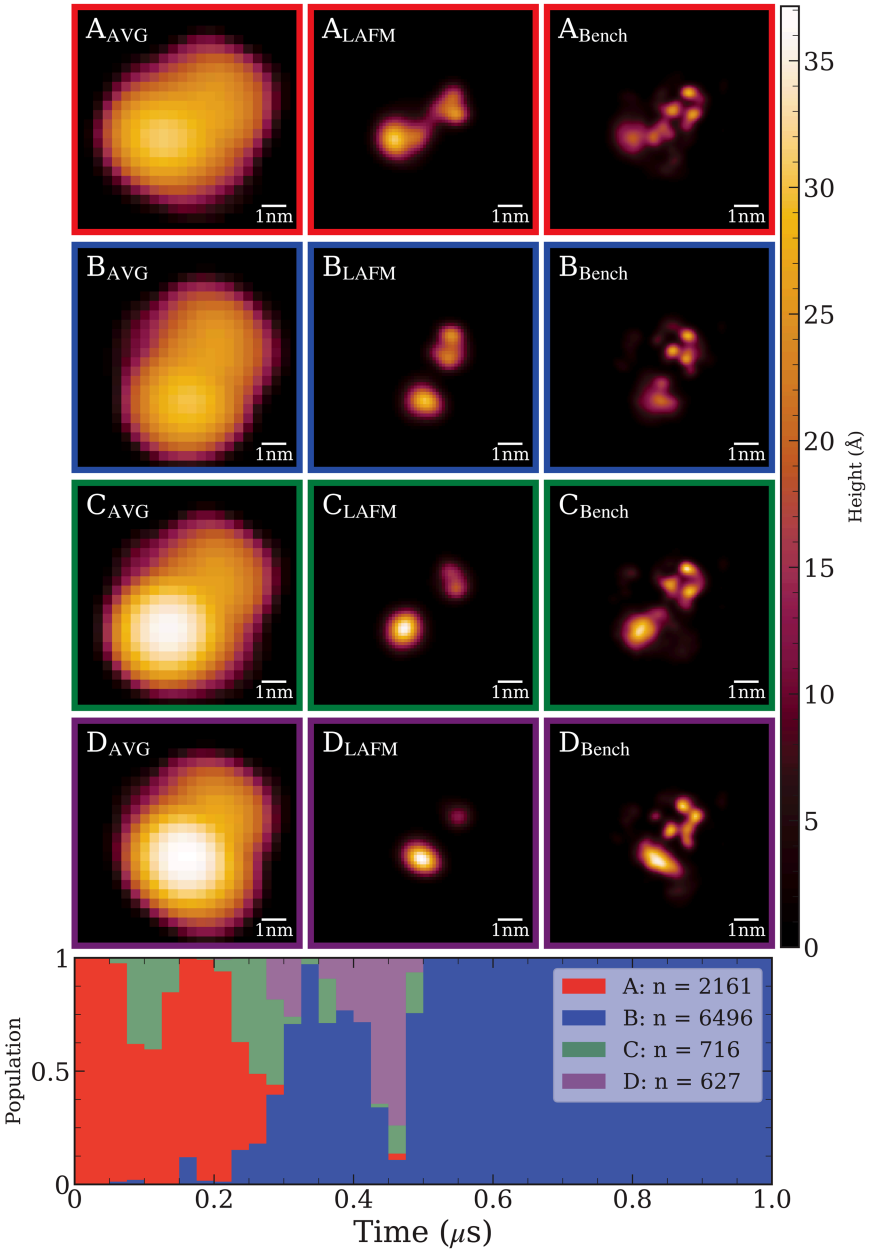

Fig 6. **Top: Hierarchical deep spectral clustering results for AC–20–4.** Each row shows the average, LAFM, and benchmark LAFM images for the detected conformational clusters. **Bottom:** Cluster occupancy over the trajectory, shown in discrete $t = 0.025\mu s$ bins. Despite the input frames being shuffled during clustering, the reconstructed time series displays a clear progression between states. The cytoplasmic side of SecYEG starts in conformation A. Clusters C and D, whose populations fall below the minimum stable cluster size, are considered transient states that the protein passes through before settling into the stable conformation B.

https://doi.org/10.1371/journal.pcbi.1013277.g006

representing a specific conformation. The clusters are shown in order of decreasing population. The stack is dominated by clusters A and B, which together account for 86.5% of the total, indicating that these conformations represent metastable states. In contrast, clusters C and D are less populated (collectively 13.5%), and most likely represent short-lived transition states. Furthermore, since we know the true sequence of frames, the temporal progression of cluster occupancy indicates that AC–20–4 starts in conformation A and transitions to conformation B, briefly passing through conformations C and D. However, it is important to emphasize that our DSC algorithm produces consistent results regardless of image sequence, yielding the same outcome when applied to the AFM image stack even after randomizing the order of frames.

For completeness, we also applied hierarchical DSC to periplasmic side images (AP–40–4) and identified two distinct conformational clusters, as shown in Fig 7. The resulting LAFM images display similar features in the top and leftmost regions, but the right domain is noticeably taller in conformation B, suggesting that the protruding loop extends outward, perpendicular to the membrane surface. Overall, the periplasmic side of SecYEG exhibits minimal large-scale conformational changes. Cluster A is predominant, representing 97.9% of the stack, while cluster B appears only briefly, accounting for 2.1%. Although the transition between these states is subtle, DSC reliably distinguishes them, demonstrating the algorithm's sensitivity to minor conformational differences.

The pixel size of ∼ 1.3 Å in the LAFM images (Fig 7) suggests that individual amino acid residues should be resolvable. However, only residues at prominent peak positions can be reliably identified due to the large AFM tip radius of $R = 2$ nm, which is comparable in size to the region of interest (ROI) shown as a green box in Fig 8. This observation captures both

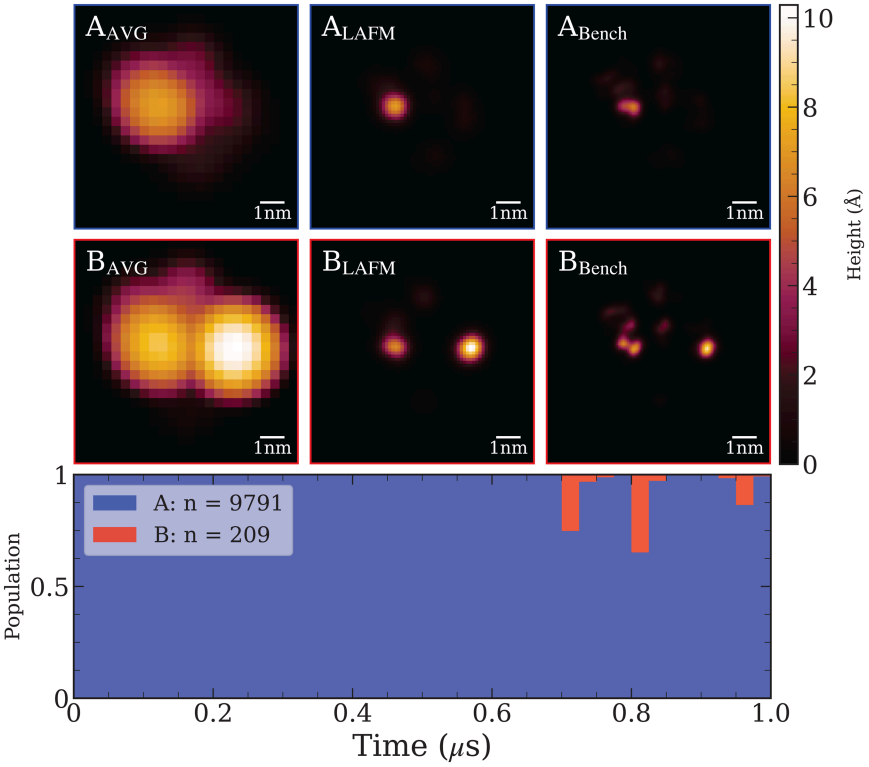

Fig 7. **Top: Hierarchical deep spectral clustering results for AP–20–4.** Each row shows the average, LAFM, and benchmark LAFM images for the detected conformational clusters. **Bottom:** Cluster distribution over the trajectory (binned in 0.025μs intervals). The periplasmic side of SecYEG predominantly adopts the stable conformation represented by cluster A throughout most of the simulation. Around 0.7μs, SecYEG briefly transitions to the conformation of cluster B, with subsequent intermittent switching between clusters A and B. Since cluster B contains fewer frames than the minimum threshold for a stable cluster, it is classified as a transient conformation.

https://doi.org/10.1371/journal.pcbi.1013277.g007

the strength and the fundamental limitation of LAFM: while the method can localize structural features with sub-nanometer lateral precision, tip convolution effects restrict this capability to topographically prominent residues that protrude sufficiently above their neighbors. The Hierarchical DSC results for AC-20-4 is shown in Fig 8 with each row displaying (left to right) the average AFM image, the corresponding LAFM reconstruction, the VMD surface representation of the atomic structure from the MD trajectory frame with maximum silhouette score, and zoomed views highlighting prominent residues within the ROI. The upper right domain exhibits small lateral fluctuations and contains five prominent residues: I307, K309, L317, R318, and P319. LAFM accurately tracks the peak positions of these residues in clusters A and B, where more than 2000 images provide sufficient statistics, but becomes less reliable in clusters C and D due to fewer available images and obstruction by nearby tall features that prevent the AFM tip from sampling these residues effectively. The bottom left domain contains the long flexible arm of SecYEG that undergoes large-scale conformational changes, which are accurately detected by hierarchical DSC. The prominent residues consistently contributing to peak detections in this region are Q215, W216, R218 (R_1), R219 (R_2), R223 (R_3), and R224 (R_4). Absolute heights of the highlighted residues in structures A–D, measured in Å from the membrane surface, are provided in S1 Table.

Finally, we point out that the hierarchical DSC method demonstrates robust noise tolerance against imperfections in experimentally acquired AFM images that persist despite careful filtering. To quantify this robustness, Gaussian noise

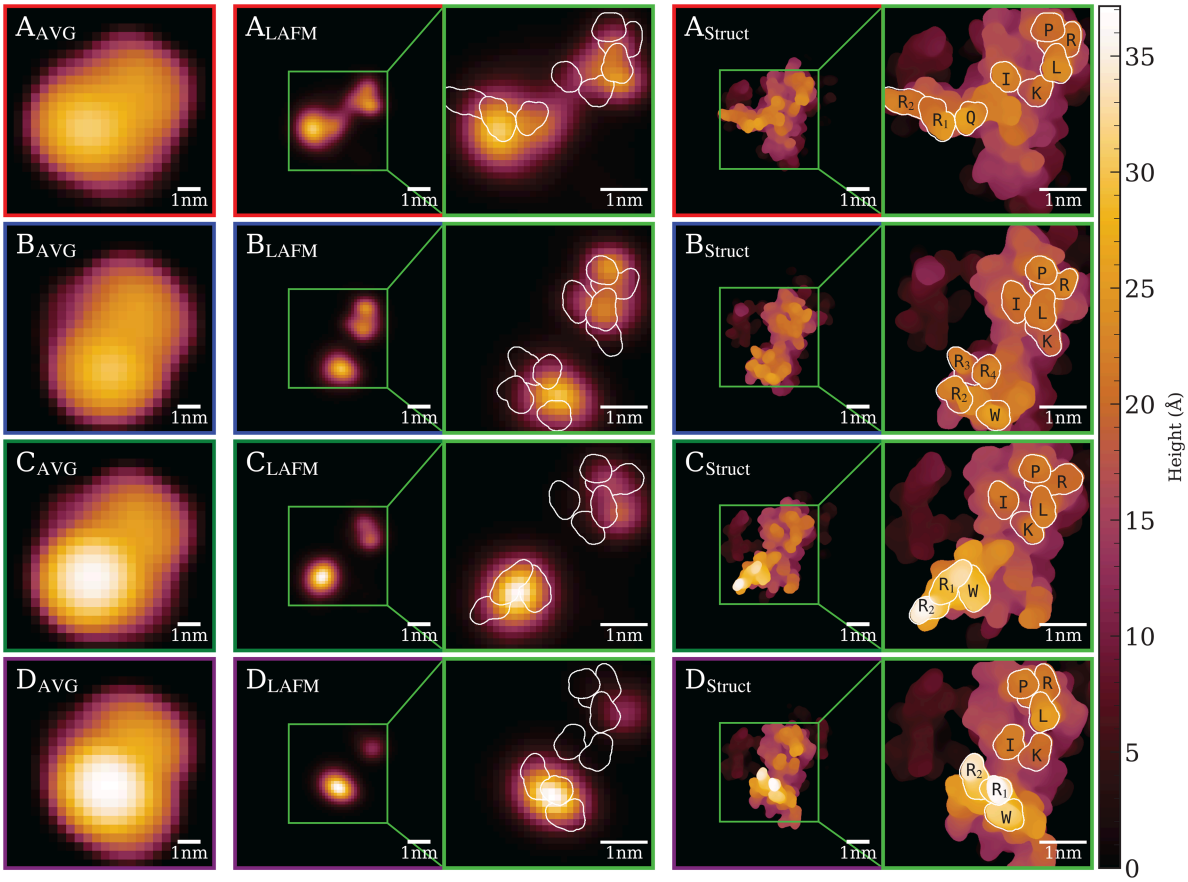

Fig 8. Hierarchical Deep Spectral Clustering results for AC-20-4. Left to right: average AFM image, LAFM image, VMD surface representation of atomic structure with maximum silhouette score in each cluster, including zoomed views of ROI (green boxes) outlining prominent residues. LAFM accurately tracks the peak position of these residues in clusters A and B but is less reliable in clusters C and D.

https://doi.org/10.1371/journal.pcbi.1013277.g008

(1–5%) was added to the AC-20-4 AFM image stack, and the Jaccard index (JI) was computed and plotted in S1 Fig for the four clusters (A-D) identified by hierarchical DSC (Fig 6). The two principal clusters (A and B) maintained high consistency with JI values exceeding 0.9 even at 5% noise. Furthermore, S2 Fig shows that clear cluster separation persists up to at least 10% noise, confirming the strong noise tolerance and stability of the hierarchical DSC clustering procedure.

Iterative Registration and Clustering (IREC) results

We applied IREC to the UC–20–4 stack, generating cytoplasmic LAFM images for increasing numbers of clusters. The results are presented in Fig 9. The unclustered LAFM image (first row) alone cannot establish whether a single stable conformation exists; therefore, we applied IREC with $n = 2$ clusters. The resulting two-cluster LAFM images (second row) were compared using SSIM, yielding scores below the 0.9 threshold, thereby confirming that these images represent distinct conformations and that at least two clusters are required. Subsequently, we applied IREC with $n = 3$ clusters. The three resulting LAFM images (third row) again exhibited SSIM scores below 0.9, validating the presence of distinct conformations and indicating that three clusters are necessary. Finally, we tested $n = 4$ clusters. Cross-comparison of the four-cluster LAFM images (fourth row) revealed that clusters D_2 and D_3 shared an SSIM score of 1.0, indicating they represent identical conformations and that one clusters is redundant; hence, $n = 3$ is the optimal clustering solution for this system.

Visual comparison of the three IREC-derived clusters with those obtained from hierarchical DSC analysis of the aligned cytoplasmic stack (Fig 6) reveals strong correspondence: IREC cluster C_1 corresponds to DSC cluster B, C_2 to cluster C, and C_3 to cluster A. Notably, all three major conformational states are successfully resolved despite the use of unaligned

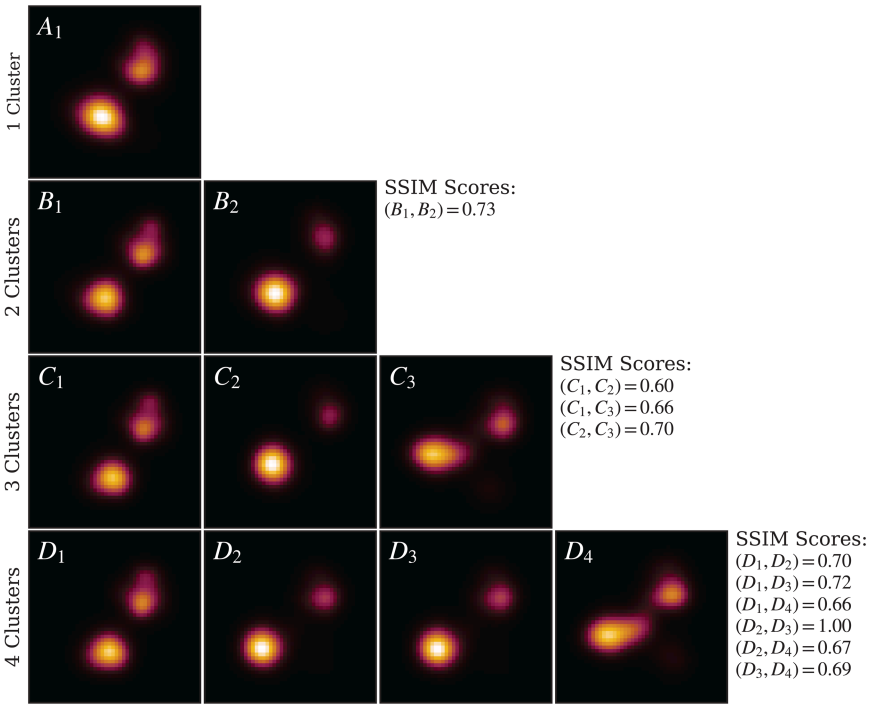

Fig 9. IREC applied to UC–20–4. LAFM images for each detected conformational cluster are shown, with rows corresponding to increasing numbers of clusters. For $n = 3$, the clustered LAFM images have SSIM scores below the 0.9 threshold and closely resemble those obtained from hierarchical DSC on the aligned stack. With $n = 4$, two of the clustered LAFMs (D_2 and D_3) exhibit a similarity score of 1.0, indicating redundancy and confirming that $n = 3$ is the optimal clustering solution.

https://doi.org/10.1371/journal.pcbi.1013277.g009

input data. The absence of DSC cluster *D* in the IREC analysis suggests that frames originally assigned to this cluster are redistributed among neighboring clusters during processing of the unaligned stack.

We also applied IREC to the UP–40–4 stack (Fig 10). The first row displays the unclustered LAFM image, which lacks definitive evidence for a single, stable conformation. Consequently, we applied IREC with $n = 2$, yielding two LAFM images with an SSIM score of 0.50, indicating distinct conformational states and validating the requirement for at least two clusters. Upon increasing to $n = 3$, cross-comparison analysis revealed that clusters C_2 and C_3 exhibited an SSIM score of 1.0, demonstrating they represent identical conformations. Therefore, $n = 2$ constitutes the optimal clustering solution.

Visual comparison between these two IREC-derived clusters and the two-cluster solution obtained from hierarchical DSC (aligned stack) demonstrates that IREC cluster B_1 corresponds to DSC cluster *B*, while B_2 corresponds to cluster *A*. This correspondence confirms comprehensive capture of all conformational states without missing clusters, thereby validating IREC's accuracy for the periplasmic dataset.

To further evaluate IREC, we combined the UP–20–4 and UC–20–4 stacks and applied IREC to this merged dataset (Fig 11). With $n = 2$, IREC successfully separated cytoplasmic and periplasmic images into distinct clusters. Since the SSIM between these clusters was substantially below the 0.9 similarity threshold, we systematically increased the number of clusters to $n = 3$, which preserved the compartmental separation while subdividing the cytoplasmic images into two distinct conformational clusters. All pairwise SSIM scores remained below the established cutoff, confirming the identification of genuinely distinct conformations.

We extended this analysis to $n = 4$ and $n = 5$ clusters. At $n = 5$, two clusters exhibited an SSIM of 0.96, exceeding our similarity threshold and indicating they likely represent the same conformational state. Therefore, $n = 4$ was

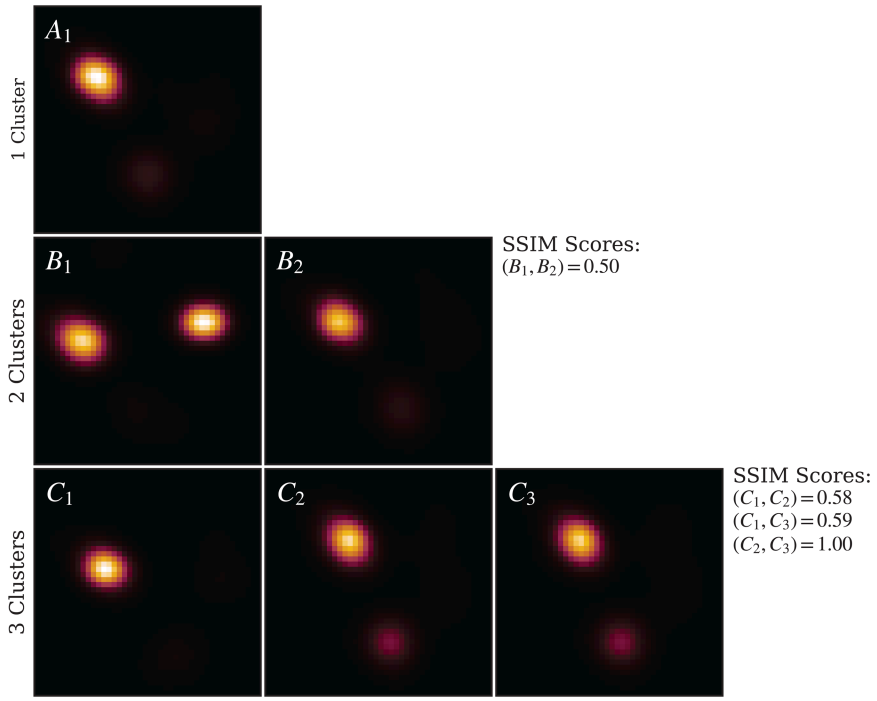

Fig 10. **IREC applied to UP–40–4.** LAFM images for each detected conformational cluster are shown, with rows corresponding to increasing numbers of clusters. For $n = 2$, the clustered LAFM images exhibit SSIM scores below the 0.9 threshold and closely resemble those obtained from hierarchical DSC on the aligned stack. With $n = 3$, two clustered LAFMs show a 0.99 SSIM score, indicating redundancy and confirming that $n = 2$ is the optimal clustering solution.

https://doi.org/10.1371/journal.pcbi.1013277.g010

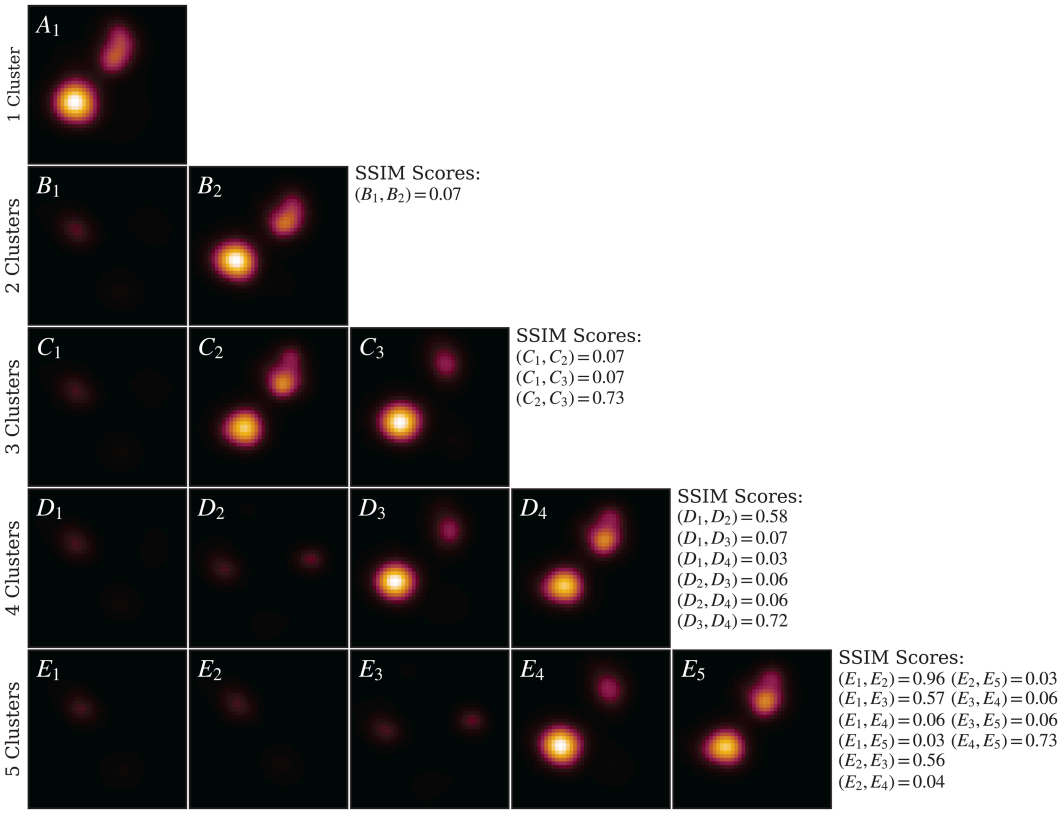

Fig 11. IREC applied to combined UP-20-4 and UC-20-4. LAFM images for detected conformational clusters are shown, with rows corresponding to increasing cluster numbers. At $n = 2$, clusters perfectly separate cytoplasmic and periplasmic sides, generating distinct LAFM images for each orientation. $n = 3$ retains side separation while resolving two cytoplasmic conformational clusters. $n = 4$ further splits both cytoplasmic and periplasmic orientations into two clusters each. $n = 5$ reproduces existing clusters but duplicates a periplasmic conformation, confirming $n = 4$ as the optimal clustering solution.

https://doi.org/10.1371/journal.pcbi.1013277.g011

determined to be the optimal clustering solution, providing the maximum resolution of distinct conformational states without over-partitioning the data.

This approach successfully identified two periplasmic and two cytoplasmic clusters. However, the third cytoplasmic conformation observed in the UC–20–4 dataset was not detected. Analysis of the initial references employed by REC revealed an inherent bias: the algorithm utilized two cytoplasmic and three periplasmic references, which during iterative refinement failed to converge on the missing cytoplasmic conformation. These findings indicate that when analyzing datasets containing structurally distinct morphologies (e.g., periplasmic versus cytoplasmic conformations), IREC should be applied hierarchically — first partitioning the data by major structural class, then performing separate clustering within each subset to resolve finer conformational distinctions.

Conclusion

The full power of the atomic force microscope – directly visualizing individual atoms – is typically realized only when imaging extremely flat and inert solid-state surfaces such as silicon or mica. The development of LAFM, which enables the distinction of individual amino acid residues in hydrated protein specimens, marks a significant advance for force microscopy in biophysical research. However, the application of LAFM has thus far been largely restricted to conformationally stable proteins with relatively rigid structures.

This work represents a significant step toward extending LAFM to a broader class of biological macromolecules that exhibit greater conformational flexibility and populate multiple structural states. Leveraging the AFMpy Python package [23] developed here, we combined LAFM with deep learning algorithms to both identify and enhance the lateral resolution of individual protein conformations. Given a stack of AFM images of membrane protein protrusions, our software is highly effective in discriminating autonomously between different proteins, their conformational states, and the respective membrane sides of their protrusions. Using the core component of the general secretory system of *E. coli*, the translocon SecYEG, as a model system, we demonstrated the feasibility of this approach on simulated AFM image stacks.

While our results are based on simulated AFM images over computationally accessible timescales, this work lays the foundation for future studies employing experimental data, such as high-speed AFM (HS-AFM) images, to probe protein conformational landscapes over biologically relevant timescales. Ultimately, these advances may enable the routine application of LAFM to a wider array of dynamic biomolecular systems.

Supporting information

S1 Table. Heights (measured in Å from the membrane surface) of prominent residues for representative all-atom protein structures from each of the four clusters, as shown in S1 Fig.
(XLSX)

S1 Fig. Jaccard index (JI) values for four clusters identified by HDSC analysis of the AC-20-4 image stack under varying levels of Gaussian noise (1–5%). The consistently high JI values, particularly for clusters A and B, demonstrate the robustness of the DSC clustering method against noise perturbations.
(PDF)

S2 Fig. Stability of conformational clusters from Hierarchical Deep Spectral Clustering (HDSC) of AC-20-4 image stacks under added Gaussian noise. Structural similarity (SSIM) values between LAFM image pairs within each cluster (A–D) are shown versus the noise level. Clusters above the 0.65 SSIM threshold (horizontal dashed line) were indistinguishable to HDSC (× marker). All clusters remained distinct below 3 % noise; at 3 %, clusters A–D merged, and at 4 %, clusters C–D merged, leaving two clusters (A, B) stable up to ∼10 % noise.
(PDF)

S3 Fig. Tilt angle (θ) fluctuations of the transmembrane axis of SecYEG during a 1 μs MD trajectory. Top panels: aligned [A] trajectory; bottom panels: unaligned [U] trajectory. Left: time series $\theta(t)$; right: distributions $P(\theta)$. The mean and standard deviation of θ are slightly larger for [U] than for [A], with both distributions being nearly normal.
(PDF)

Author contributions

Conceptualization: Creighton M. Lisowski, Gavin M. King, Ioan Kosztin.

Data curation: Creighton M. Lisowski.

Formal analysis: Creighton M. Lisowski, Gavin M. King, Ioan Kosztin.

Methodology: Creighton M. Lisowski, Ioan Kosztin.

Project administration: Ioan Kosztin.

Software: Creighton M. Lisowski.

Validation: Creighton M. Lisowski.

Visualization: Creighton M. Lisowski, Ioan Kosztin.

Writing – original draft: Creighton M. Lisowski.

Writing – review & editing: Creighton M. Lisowski, Gavin M. King, Ioan Kosztin.

References

1. Bippes CA, Muller DJ. High-resolution atomic force microscopy and spectroscopy of native membrane proteins. Rep Prog Phys. 2011;74(8):086601. https://doi.org/10.1088/0034-4885/74/8/086601
2. Sanganna Gari RR, Chattrakun K, Marsh BP, Mao C, Chada N, Randall LL, et al. Direct visualization of the E. coli Sec translocase engaging precursor proteins in lipid bilayers. Sci Adv. 2019;5(6):eaav9404. https://doi.org/10.1126/sciadv.aav9404 PMID: 31206019
3. Heath GR, Kots E, Robertson JL, Lansky S, Khelashvili G, Weinstein H, et al. Localization atomic force microscopy. Nature. 2021;594(7863):385–90. https://doi.org/10.1038/s41586-021-03551-x PMID: 34135520
4. Zhong ED, Bepler T, Berger B, Davis JH. CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. Nat Methods. 2021;18(2):176–85. https://doi.org/10.1038/s41592-020-01049-4 PMID: 33542510
5. Degenhardt MFS, Degenhardt HF, Bhandari YR, Lee Y-T, Ding J, Yu P, et al. Determining structures of RNA conformers using AFM and deep neural networks. Nature. 2025;637(8048):1234–43. https://doi.org/10.1038/s41586-024-07559-x PMID: 39695231
6. Amyot R, Flechsig H. BioAFMviewer: an interactive interface for simulated AFM scanning of biomolecular structures and dynamics. PLoS Comput Biol. 2020;16(11):e1008444. https://doi.org/10.1371/journal.pcbi.1008444 PMID: 33206646
7. Amyot R, Kodera N, Flechsig H. Atom filtering algorithm and GPU-accelerated calculation of simulation atomic force microscopy images. Algorithms. 2024;17(1):38. https://doi.org/10.3390/a17010038
8. Sanganna Gari RR, Frey NC, Mao C, Randall LL, King GM. Dynamic structure of the translocon SecYEG in membrane: direct single molecule observations. J Biol Chem. 2013;288(23):16848–54. https://doi.org/10.1074/jbc.M113.471870 PMID: 23609442
9. Weaver DR, King GM. Atomic force microscopy reveals complexity underlying general secretory system activity. Int J Mol Sci. 2022;24(1):55. https://doi.org/10.3390/ijms24010055 PMID: 36613499
10. Zimmer J, Nam Y, Rapoport TA. Structure of a complex of the ATPase SecA and the protein-translocation channel. Nature. 2008;455(7215):936–43. https://doi.org/10.1038/nature07335 PMID: 18923516
11. Lomize MA, Pogozheva ID, Joo H, Mosberg HI, Lomize AL. OPM database and PPM web server: resources for positioning of proteins in membranes. Nucleic Acids Res. 2012;40(Database issue):D370-6. https://doi.org/10.1093/nar/gkr703 PMID: 21890895
12. Jo S, Kim T, Iyer VG, Im W. CHARMM-GUI: a web-based graphical user interface for CHARMM. J Comput Chem. 2008;29(11):1859–65. https://doi.org/10.1002/jcc.20945 PMID: 18351591
13. Wu EL, Cheng X, Jo S, Rui H, Song KC, Dávila-Contreras EM, et al. CHARMM-GUI membrane builder toward realistic biological membrane simulations. J Comput Chem. 2014;35(27):1997–2004. https://doi.org/10.1002/jcc.23702 PMID: 25130509
14. Lee J, Cheng X, Swails JM, Yeom MS, Eastman PK, Lemkul JA, et al. CHARMM-GUI input generator for NAMD, GROMACS, AMBER, OpenMM, and CHARMM/OpenMM simulations using the CHARMM36 additive force field. J Chem Theory Comput. 2016;12(1):405–13. https://doi.org/10.1021/acs.jctc.5b00935 PMID: 26631602
15. Rowlett VW, Mallampalli VKPS, Karlstaedt A, Dowhan W, Taegtmeyer H, Margolin W, et al. Impact of membrane phospholipid alterations in escherichia coli on cellular function and bacterial stress adaptation. J Bacteriol. 2017;199(13):e00849-16. https://doi.org/10.1128/JB.00849-16 PMID: 28439040
16. Phillips JC, Hardy DJ, Maia JDC, Stone JE, Ribeiro JV, Bernardi RC, et al. Scalable molecular dynamics on CPU and GPU architectures with NAMD. J Chem Phys. 2020;153(4):044130. https://doi.org/10.1063/5.0014475 PMID: 32752662
17. Huang J, MacKerell AD Jr. CHARMM36 all-atom additive protein force field: validation based on comparison to NMR data. J Comput Chem. 2013;34(25):2135–45. https://doi.org/10.1002/jcc.23354 PMID: 23832629
18. Huang J, Rauscher S, Nawrocki G, Ran T, Feig M, de Groot BL, et al. CHARMM36m: an improved force field for folded and intrinsically disordered proteins. Nat Methods. 2017;14(1):71–3. https://doi.org/10.1038/nmeth.4067 PMID: 27819658
19. Schmidt-Krey I, Gumbart JC. Structure and function of membrane proteins. New York, NY: Springer US; 2021.
20. Feller SE, Zhang Y, Pastor RW, Brooks BR. Constant pressure molecular dynamics simulation: the Langevin piston method. The Journal of Chemical Physics. 1995;103(11):4613–21. https://doi.org/10.1063/1.470648
21. Grønbech-Jensen N, Farago O. A simple and effective Verlet-type algorithm for simulating Langevin dynamics. Molecular Physics. 2013;111(8):983–91. https://doi.org/10.1080/00268976.2012.760055
22. Humphrey W, Dalke A, Schulten K. VMD: visual molecular dynamics. J Mol Graph. 1996;14(1):33–8, 27–8. https://doi.org/10.1016/0263-7855(96)00018-5 PMID: 8744570
23. Kosztin Research Group. Kosztin-Research-Group/AFMpy. 2025.
24. Gowers R, Linke M, Barnoud J, Reddy T, Melo M, Seyler S, et al. MDAnalysis: a python package for the rapid analysis of molecular dynamics simulations. In: Proceedings of the Python in Science Conference. 2016. p. 98–105. https://doi.org/10.25080/majora-629e541a-00e
25. Michaud-Agrawal N, Denning EJ, Woolf TB, Beckstein O. MDAnalysis: a toolkit for the analysis of molecular dynamics simulations. J Comput Chem. 2011;32(10):2319–27. https://doi.org/10.1002/jcc.21787 PMID: 21500218

26. Harris CR, Millman KJ, van der Walt SJ, Gommers R, Virtanen P, Cournapeau D, et al. Array programming with NumPy. Nature. 2020;585(7825):357–62. https://doi.org/10.1038/s41586-020-2649-2 PMID: 32939066
27. Okuta R, Unno Y, Nishino D, Hido S, Loomis C. CuPy: a NumPy-compatible library for NVIDIA GPU calculations. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Thirty-First Annual Conference on Neural Information Processing Systems (NIPS). 2017.
28. Brooks BR, Brooks CL 3rd, Mackerell AD Jr, Nilsson L, Petrella RJ, Roux B, et al. CHARMM: the biomolecular simulation program. J Comput Chem. 2009;30(10):1545–614. https://doi.org/10.1002/jcc.21287 PMID: 19444816
29. Bradski G. The OpenCV Library. Dr Dobb's Journal of Software Tools. 2000;25(11):120–5.
30. Wang Z, Simoncelli EP, Bovik AC. Multiscale structural similarity for image quality assessment. In: The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003. https://doi.org/10.1109/acssc.2003.1292216
31. Zhou Wang, Bovik AC. Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures. IEEE Signal Process Mag. 2009;26(1):98–117. https://doi.org/10.1109/msp.2008.930649
32. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: machine learning in python. Journal of Machine Learning Research. 2011;12(85):2825–30.
33. Abadi M, Barham P, Chen J, Chen Z, Davis A, Dean J. TensorFlow: a system for large-scale machine learning. 2016.
34. Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al. TensorFlow: large-scale machine learning on heterogeneous distributed systems. 2016.
35. Masci J, Meier U, Cireşan D, Schmidhuber J. Stacked convolutional auto-encoders for hierarchical feature extraction. In: Honkela T, Duch W, Girolami M, Kaski S, editors. Artificial Neural Networks and Machine Learning – ICANN 2011. Berlin, Heidelberg: Springer; 2011. p. 52–9.
36. Zelnik-manor L, Perona P. Self-tuning spectral clustering. Advances in Neural Information Processing Systems. MIT Press; 2004.
37. Lichtner G. Pystackreg: Image registration tool (Python implementation of the ImageJ/FIJI plugin TurboReg/StackReg). 2024.
38. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational and Applied Mathematics. 1987;20:53–65. https://doi.org/10.1016/0377-0427(87)90125-7
39. Thévenaz P, Ruttimann UE, Unser M. A pyramid approach to subpixel registration based on intensity. IEEE Trans Image Process. 1998;7(1):27–41. https://doi.org/10.1109/83.650848 PMID: 18267377