Article

Scipion-EM-ProDy: A Graphical Interface for the ProDy Python Package within the Scipion Workflow Engine enabling Integration of Databases, Simulations and Cryo-Electron Microscopy Image Processing

James Michael Krieger [*] , Carlos Oscar S. Sorzano , Jose Maria Carazo [*]

Article

Scipion-EM-ProDy: A Graphical Interface for the ProDy Python Package within the Scipion Workflow Engine Enabling Integration of Databases, Simulations and Cryo-Electron Microscopy Image Processing

James M Krieger [1,*] , **Carlos Oscar S. Sorzano** [1,] , and **Jose Maria Carazo** [1,*]

[1]   Biocomputing Unit, National Centre for Biotechnology (CNB CSIC), Darwin 3, Campus Universidad Autónoma de Madrid, Cantoblanco, Madrid 28049, Spain

[*]   Correspondence: jmkrieger@cnb.csic.es (J.M.K.); carazo@cnb.csic.es (J.M.C.)

Abstract: Macromolecular assemblies, such as protein complexes, undergo continuous structural dynamics, including global reconfigurations critical for their function. Two fast analytical methods are widely used to study these global dynamics, namely elastic network model normal mode analysis and principal component analysis of ensembles of structures. These approaches have found wide use in various computational studies, driving the development of complex pipelines in several software packages. One common theme has been conformational sampling through hybrid simulations incorporating all-atom molecular dynamics and global modes of motion. However, wide functionality is only available for experienced programmers with limited capabilities for other users. We have, therefore, integrated one popular and extensively developed software for such analyses, the ProDy Python application programming interface, into the Scipion workflow engine. This enables a wider range of users to access a complete range of macromolecular dynamics pipelines beyond the core functionalities available in its command-line applications and the normal mode wizard in VMD. The new protocols and pipelines can be further expanded and integrated into larger workflows together with other software packages for cryo-electron microscopy image analysis and molecular simulations. We present the resulting plugin, Scipion-EM-ProDy, in detail, highlighting the rich functionality made available by its development.

Keywords: global protein dynamics; software integration workflows; cryo-electron microscopy; normal mode analysis; elastic network models; ensemble analysis; principal component analysis; hybrid simulations

1. Introduction

Macromolecular complexes exhibit considerable flexibility, which is essential for their diverse functions, including signal transduction, transport of small molecules and ions, enzymatic catalysis and mechanical work, and regulating interactions with other molecules [1]. They undergo a range of motions from small, local rearrangements to large-scale, collective/global conformational changes [1,2]. These motions result in rich ensembles and continuous conformational landscapes, which are starting to be resolved by a range of computational and experimental methods [3–7].

One very exciting area is cryo-electron microscopy (Cryo-EM) single particle analysis (SPA) where thanks to an explosion of continuous heterogeneity analysis methods [6,8–10], it is now possible to start studying this flexibility from single-particle images. These state-of-the-art methods allow each particle image to be assigned some parameters in a multi-dimensional latent space that describe differences between them that are ideally related to their conformation and enable the visualisation of these spaces via dimensionality reduction (dim. red.), giving rise to landscapes where similar particles are usually in close proximity. They also provide tools for navigating the landscapes and clustering particles,

and recovering maps from cluster centres and other points, using either a generative heterogeneous reconstruction autoencoder network [9] or a deformation field [11–13].

1.1. Scipion Flexibility Hub solves challenges with Cryo-EM continuous heterogeneity analysis

Despite the great progress being made, these approaches have several intrinsic limitations (including experimental issues such as potential particle damage/denaturation and enrichment/depletion of particular orientations and conformations as a result of freezing [?] and the air water interface [14]) and it is very challenging to extract biological and physical meaning from the results of these methods as discussed in a recent review [10] (see Table 1, column 1). For example, it is now possible to generate large numbers (tens or hundreds) of maps fairly rapidly. Still, it is far from trivial to follow conformational changes between them and interpret the landscapes in either original or reduced latent coordinates. It is also difficult to compare and combine the results from different methods to come up with consistent conclusions.

Table 1. Challenges in Cryo-EM continuous heterogeneity analysis and potential solutions assisted by Scipion Flexibility Hub[1] and Scipion-EM-ProDy

Problem	Solution	Existing tools	FlexUtils and ProDy
Most analyses of continuous heterogeneity lack biological interpretability and physical meaning (often separate images and maps by non-structural factors)	- Generate representative maps and atomic structures rather than just using landscapes and conformational changes directly from continuous heterogeneity analyses of images - Clustering and dim. red. in structural space	- NMA StructMap for map landscapes[2] - Atomic structure dim. red. and NMA[2] - VMD animation[2] - Theoretical motion statistics from NMA of single structures[2] - Cluster and dim. red. images as structures[2]	- Correlation-based and Zernike3D-based StructMap[1] - PCA, NMA, and deform vectors - NMWiz viewer and comparison tools - GNM and domain decomposition - Zernike3D motion statistics from images[1] - Cluster and dim. red. images as structures[1]
Large numbers of lower quality maps from continuous heterogeneity are challenging to fit with good structures	- Fix rough fits with short simulations, perhaps w/ NMA	- GENESIS NMMD and MD[2]	- ClustENM(D) for OpenMM, with or without NMA
Need to compare standard cryo-EM and continuous heterogeneity outputs to existing structures and those from simulations, and make sense of results in broader context	- Compare cluster representatives from each approach - Compare images directly to projections from maps simulated from structures in a careful way that accounts for prior expectations, such as structural similarity	- BioEM, cryo-BIFE & ensemble reweighting[3]	- Ensemble and trajectory analysis - Atomic structure clustering - Combined PCA - Deformations and landscapes priors[1]

[1] FlexUtils; [2] ContinuousFlex; [3] Outside Scipion.

Some approaches can help by analysing the resulting maps directly, such as structure mapping methods based on correlations or deformations [11,15], quantification of regions experiencing rotations and strains between maps [11,16], or visualisation of morphs between maps in ChimeraX using either standard linear interpolation or optimal transport [17,18]. Nevertheless, even after ruling out compositional heterogeneity with advanced tools, such as the atomic structure-based occupancy mapping approach provided by CryoDRGN [19,20], the differences between maps can also be due to various reasons and be difficult to interpret in an atomistic manner. The latter could be overcome by flexibly fitting atomic models to these maps, and many methods are available to do this [21], but this is also challenging with so many maps.

This can be helped by continuous heterogeneity methods that use atomic models in the interpretation of images and maps via computational biophysics approaches such as normal mode analysis (NMA) and molecular dynamics (MD) simulations (Table 1, column 2). Prominent examples include hybrid electron microscopy normal mode analysis (HEMNMA) [22], HEMNMA-3D [23], NMMD [24] and MDSPACE [25] within the Scipion plugin ContinuousFlex [12]. Another set of approaches use Bayesian inference and computational biophysics for comparing images to atomic structures, including BioEM [26], Cryo-EM Bayesian inference of free energy profiles (cryo-BIFE) [27], and ensemble reweighting [28]. These methods all have limitations to the number of images or maps that can be analysed, but they produce very valuable insights and continue to be improved. For example, a recent deep-learning-accelerated extension called DeepHEMNMA [29] infers HEMNMA parameters for a large image set based on training with the results from a smaller one.

In this context, we recently created the Scipion Flexibility Hub [30], a framework for combining continuous flexibility methods into pipelines with shared interactive analysis tools connected to the rest of the Scipion workflow engine for Cryo-EM image analysis and structural biology [31,32]. This framework provides common tools for several continuous heterogeneity methods, including the CryoDRGN deep reconstruction generative network [19], HEMNMA [22,33], MDSPACE [25] and related methods presented in ContinuousFlex [12], and some approaches based on the Zernike3D deformation framework [11,34]. Altogether, this enables the creation and analysis of both maps and atomic models for representative points (e.g., cluster centres) in the conformational landscapes inferred from the particle images (see Table 1, column 3, and Figure 1).

The atomic model creation is achieved using a fast and approximate flexible fitting approach based on Zernike3D [11,30]. These rough atomic models have many issues as a result of the approximations made in the Zernike3D method, including a lack of any physics in the description and application of the deformations. Nevertheless, these models are a good first approximation of the conformational changes in the images and maps and provide a good starting point for further analyses and improvements, enabled by the new plugin Scipion-EM-ProDy that was briefly introduced with Scipion Flexibility Hub [30] and is presented in more detail here together with the latest developments.

1.2. Scipion-EM-ProDy facilitates better interpretation and simulation of atomic structures through rapid computational biophysics

Having atomic structures enables a connection to the wide range of computational biophysics approaches, which have been developed for extracting and interpreting ensembles and conformational landscapes and inferring functional mechanisms [35]. These cover models and methods over a range of spatial and temporal resolution scales, from quantum mechanics approximations to all-atom molecular dynamics (MD) to coarse-grained approaches.

While all-atom MD simulations are very popular and under considerable development, resulting in a large number of dedicated software packages and force fields [36,37] and a plethora of enhanced sampling methods [38,39], they still have considerable limitations including their computational cost and sensitivity to force fields inaccuracies [37]. Thus, it is challenging to sample events on biologically relevant time scales such as microseconds and milliseconds, especially for large macromolecular assemblies studied by Cryo-EM, and coarse-grained models and methods are often preferable [40,41].

Among these are two fast matrix decomposition methods, elastic network models (ENM) normal mode analysis (NMA) and principal component analysis (PCA) of structural ensembles [42], which are available in a small number of specialised software packages such as ProDy [43,44], Bio3D [45] and ElNemo [46] (used by ContinuousFlex) and web servers such as WEBnm@ [47] and Dynomics [48]. These methods rapidly yield robust modes of motions with clear biological relevance over an extensive collection of macromolecular systems [2,42] and have led to the development of many related methods, including hybrid simulations that combine NMA or PCA with very short (1 to 100 picoseconds [ps]) MD runs [39] including the recently developed NMMD [24] and ClustENMD [49].

ProDy is a popular Python package with a long history of development and a wide user base as a result of its systematic and rich architecture as an application programming interface (API) and its focus on these fast analytic methods for analysing global protein dynamics [44]. This has enabled several applications to be built on top of it, including internal ProDy and evol applications for core tasks and the normal mode wizard (NMWiz) [50] within VMD [51], the collective MD (coMD) hybrid simulation package [52] that also uses VMD [51] and NAMD [53], and the NMA-assisted docking program LightDock [54].
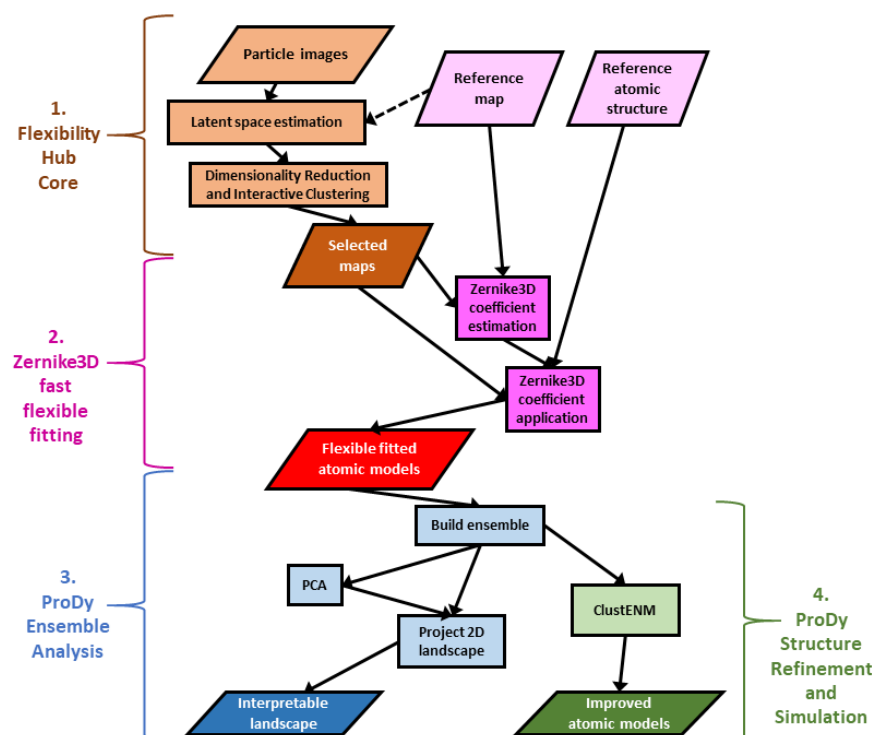

Figure 1. Illustrative workflow using Flexibility Hub and Scipion-EM-ProDy to solve challenges with interpreting landscapes from continuous heterogeneity analysis of single particle images and obtaining reasonable structures. There are 4 main parts. (**1**) A general Flexibility Hub is shown in orange at the top left starts from a set of particles from traditional SPA with contrast transfer function (CTF) and alignment information already provided. The two steps are first to estimate conformational landscapes in high-dimensional latent spaces and then analyse them by dimensionality reduction and interactive clustering, yielding maps corresponding to selected particle subsets. (**2**) A fast, flexible fitting pipeline based on Zernike3D where coefficients are first estimated for the deformation between the reference map and each of the selected maps and then applied to the reference atomic structure. (**3**) ProDy ensemble analysis generates a conformational landscape where the axes are interpretable reaction coordinates from PCA. (**4**) The ProDy ClustENM protocol can be used for improving the rough atomic structures coming from Zernike3D flexible fitting.

ProDy provides several ENMs, including the Gaussian network model (GNM) [55], the anisotropic network model (ANM) [56] and several variants [57,58], and the rotating and translating blocks (RTB) model [59,60], as well as the option to provide others with different distance relationships interactively [61]. It also provides many tools, such as the deformation vector approach to pairwise conformational changes between two structures and a rich ensemble building and analysis toolkit based on sequence and structural alignment and PCA, which allow comparison between experimental and computational results, and the hybrid simulation methods ClustENM and ClustENMD [49,62]. The creation of Scipion-EM-ProDy enables these tools to be more readily available to the general structural biology community, enabling a greater connection between experiments and computational biophysics (see Figure 1), which is critical for obtaining a proper understanding of macromolecular dynamics and function [5]. We illustrate its main functionalities and updates using a system that we have studied recently and extensively, namely the D614G SARS-CoV-2 spike [30,63].

2. Results

We first provide an overview of the Scipion workflow engine and the general ProDy pipeline (Figure 2). Then, we present some example pipelines and workflows, demonstrating the key functionalities that help interpret results from continuous heterogeneity analyses and enable comparison of experimental and simulated structures.

2.1. General overview of Scipion and Scipion-EM-ProDy for building workflows for computational biophysics

The general idea behind Scipion is to integrate many software packages into complex workflows comprised of various interconnected protocols whose outputs are the inputs to subsequent protocols [31]. These protocols are provided by plugins, including the Scipion core plugins [31], as well many others for external software such as Xmipp [64], Relion [65], ContinuousFlex [12] and ChimeraX [17]. A key feature is interoperability by handling data in common objects and converting it between different formats required for different programs. It also streamlines the smooth download and installation of these various software packages in a way that handles and isolates their associated dependencies.

The plugin Scipion-EM-ProDy provides 18 new protocols with associated objects and viewers as well as a menu bar suggesting possible workflows (see Figure 2a). Most of these belong to three main categories corresponding to the three main steps of a protein dynamics analysis workflow: 1. atomic structure operations (atom selection, modelling, and alignment), 2. core dynamics calculations (deformation vector analysis, normal mode analysis, ensemble analysis, Gaussian network model [GNM] analysis, and hybrid simulations), and 3. downstream dynamics analyses (mode comparison, projection onto landscapes and clustering). Each protocol is run through a form, such as that shown in Figure 2b for an ensemble building step.

A first example workflow (Figure 2c) illustrates an ensemble analysis of the D614G spike. As above, there are three main steps coloured in yellow, blue, and green for atomic structure handling (importing and selecting atoms), core dynamics calculations (ensemble construction and PCA), and analysis (landscape projection), respectively. In this case, the selection protocol is used to import three structures from a data set from Benton, Wrobel and co-workers [66] from the protein data bank (PDB) [67]. These represent the three main states of the spike with 3 receptor-binding domains (RBDs) down (PDB: 7BNM), 1 RBD up and 2 RBDs down (PDB: 7BNN), and 2 RBDs up and 1 RBD down (PDB: 7BNO) and the protocols make the trivial selection of the protein atoms in this case as an illustration. The ensemble building protocol takes all these three structures as input and aligns them in a default way using the Cα atoms. A more complicated example following our recent paper [63] is described later. The principal component analysis takes the resulting ensemble of three aligned structures as input. It calculates two components of variation, and the projection protocol projects the atomic coordinates from the ensemble onto the two components.

Many of these steps have associated viewers, such as the ProDy normal mode viewer connecting to NMWiz [43,50] in VMD [51] shown at the bottom left and the projection viewer based on Matplotlib

[68] shown at the top right that yields the plot below it. They can also use viewers from other plugins if they use common objects, such as those from ContinuousFlex.
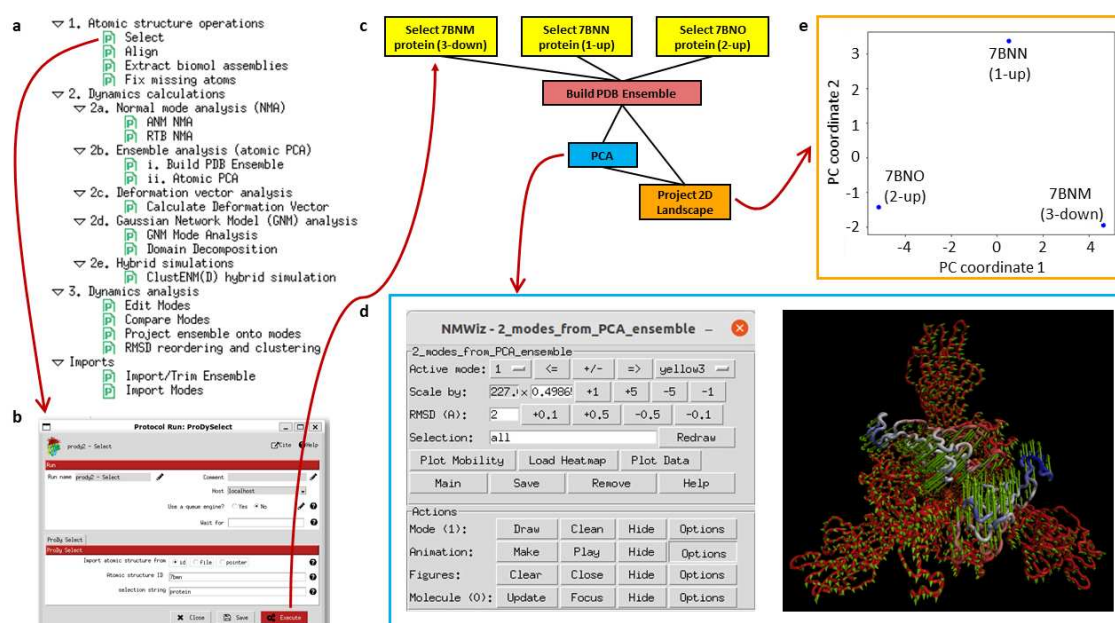

Figure 2. Overview of the Scipion-EM-ProDy plugin with red arrows showing consecutive steps following user interaction. (**a**) List of protocols available within the plugin as shown on the left-hand side of a Scipion window. These are divided into categories with an order indicating how they should be used in pipelines and workflows. Double-clicking on one of these protocols yields a form as shown in panel **b**. (**b**) An example protocol form is shown for atom selection. (**c**) Execution of protocols creates a workflow of inter-connected boxes, as shown. The workflow here illustrates importing of three D614G spike structures with selection of protein atoms (yellow) and ensemble construction and analysis via PCA (blue) and landscape projection (orange). (**d**) PCA results are shown with the ProDy mode viewer that opens up VMD and NMWiz (blue outlined box). The control window for NMWiz (left) has many options for visualising the structure with arrows and creating movies in the VMD structure viewer window (right). The average structure is coloured by the relative size of motion from most rigid in red to most mobile in blue. The green arrows show the first principal component where the two subunits slide past each other to close together (direction shown by arrows; towards 3-down) and open together (opposite direction; towards 2-up). (**e**) A projection plot from Matplotlib (orange outlined box) shows the conformational landscape from the ensemble in the space of the first two PCs, with each point being a structure.

As shown in Figure 2d, the arrows in NMWiz illustrate the active component of motion selected at the top of the window, which can also be animated by clicking the "make" button in the animation row of the bottom part. In this case, principal component (PC) 1 is shown, which shows a concerted closing and opening of two of the RBDs. PC2 shows an anti-correlated opening and closing of these two RBDs, which separates the 1-up structure (PDB: 7BNN) from the others. The landscape from projecting the ensemble onto these two PCs (Figure 2e) shows these separations. It separates the three structures via PC1 along the x-axis, with the 2-up structure (PDB: 7BNO) and the 3-down structures (PDB: 7BNO) being at the extremes, and the 1-up structure (PDB: 7BNN) being near zero as PC1 is not relevant for this structure. Instead, the y-axis corresponding to PC2 separates the 1-up structure (PDB: 7BNN) from the other two structures via the closing of the second RBD as the first RBD opens.

2.2. Ensemble Analysis via PCA downstream of Flexibility Hub enables Interpretation of Cryo-EM Conformational Landscapes

Rather than starting from database structures, ProDy can also analyse structures from Cryo-EM analyses, such as those from the Zernike3D flexible fitting above (see Figure 1). While the resulting atomic models usually have many errors, especially at the level of local details, the global conformational changes should be fairly reasonable.

Following an iterative superposition with the ensemble protocol to converge the mean structure, PCA can then find and clean up the dominant conformational changes, allowing them to be used as new, interpretable coordinates for new conformational landscapes via vector projection (Figure 1, part 3). As each coordinate axis corresponds to a fairly simple mode of motion that can be visualised in NMWiz, this landscape is much more interpretable than the landscapes generated from the continuous heterogeneity methods themselves.

In principle, it should also be possible to project the particles into the PCA landscape via some appropriate interpolation approach. This could be aided by methods such as the conversion between Zernike3D deformations and principal components by thin plate spline interpolation [30].

2.3. A New ClustENM(D) Protocol for Refining Atomic Models and Hybrid Simulations

One recent addition to ProDy is the ClustENM class for the hybrid simulation approach ClustENMD [49], and its predecessor ClustENM [62] using OpenMM [69] for energy minimisation and MD. This method first fixes the starting atomic structure using PDBFixer [69], performs energy minimisation and optionally some picoseconds of MD simulations on it, and then runs several generations of the following steps: 1) generate several new conformations from each starting structure using random linear combinations of ENM normal modes (NMs), 2) cluster the resulting structures, 3) minimise the cluster representative structures and optionally run some picoseconds of MD simulations for each. The combined approach is often referred to as ClustENM(D), with the D in brackets referring to the option of including the MD in ClustENMD or just running energy minimisation in ClustENM. Scipion-EM-ProDy has a ClustENM(D) protocol that can take multiple structures as input and run a ClustENM(D) simulation on each of them.

This protocol provides all the various options available within ProDy, which are spread across three tabs of the form (see Materials and Methods, Section 4.2.2, and Figure 3). Amongst these, there is an option for the number of generations, which can be set to zero to perform energy minimisation and optionally a short MD simulation on the starting structure(s). It also provides the ability to run ClustENM(D) simulations on multiple structures simultaneously and then analyse them together. These two functionalities could be used to refine low-quality structures, such as those from Zernike3D fast flexible fitting, as discussed above (Figure 1, part 4).

2.4. A Combined Workflow for Comparing Structures from Experiments and Simulations

It is often useful to contextualise new experimental data in the context of existing structures, such as those deposited in the PDB or those obtained from simulations. We demonstrate such a workflow here using the set of SARS-CoV-2 spike structures with the D614G mutation available in the PDB (as in our previous paper [63]) and a ClustENM simulation based on one of them (Figure 3a). Structures from continuous heterogeneity analysis could also be included using the workflows presented above (see Figure 1).

After importing the set of 24 D614G spike structures studied previously [63] (grey box), this workflow contains 3 main pipelines. The first two (left half of Figure 3a) are based purely on the analysis of these experimental structures using fast analytical methods, starting with their alignment into an ensemble (red box). The core one (in the green shaded box) is based on PCA as in Figure 2, and the other analyses the structure-encoded dynamics of some particular structures from root-mean-square deviation (RMSD) clustering (left orange box) using ENM NMA to inform the third workflow (blue shaded box) where we use one of these structures in an intermediate state (PDB: 7KEC) for ClustENM

simulations. Ultimately, this allows us to compare the landscapes from the experimental and simulated ensembles by projecting them onto the same set of PCs (bottom right orange box) as shown in Figure 3b. The structure selection is guided by the correlation cosine overlaps between the mode of motion vectors from NMA and PCA (green boxes; Figure 3c).
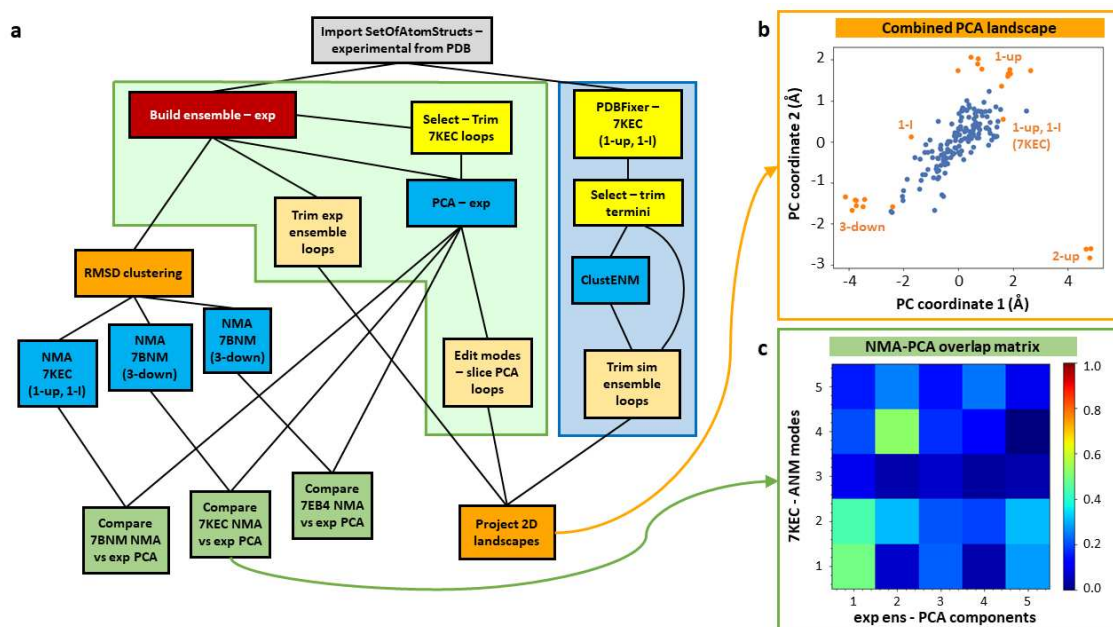

Figure 3. Combined workflow for comparing existing experimental structures and simulations. (**a**) The workflow contains two main parts: experimental structure ensemble analysis (green shading, left) and ClustENM simulations (blue shading, right). The experimental structures are aligned into an ensemble (red box) and analysed by PCA (central blue box) and RMSD clustering (left orange box). Some representative clusters are also used for NMA (3 blue boxes outside green shading) to confirm the best structure for simulations via comparison of NMs and PCs (following green boxes). Some atomic structure tools (yellow) were also used including as a first step to prepare the structure for simulations. PDBFixer adds missing atoms and residues based on information in the PDB or mmCIF file then a selection is applied to trim away the termini. A selection was also applied to the experimental and simulated ensembles and the PCA vectors (yellow-orange) to trim away the flexible loops to enable their comparison. Coloured arrows connect to other panels showing their outputs. (**b**) A combined landscape is shown projecting the experimental ensemble in orange and the simulated one in blue in the space of the first two principal components from the experimental ensemble (axis values are in units of RMSD in Å). The main conformations are labelled including the simulation starting structure (PDB: 7KEC). (**c**) An overlap matrix is shown comparing the first five non-zero ANM normal modes of the 1-up, 1-I structure (PDB: 7KEC) to the first five principal components of the ensemble with low overlaps in blue and high overlaps in green, orange and red. There are reasonably high values for the first two PCs, confirming that its NMs are useful for hybrid simulations to explore a conformational space similar to that observed experimentally.

Ensemble analysis of existing experimental structures can be challenging as different research groups often use different conventions for labelling residues and chains in their structures. However, as ProDy has been extensively used for analysing diverse structures, its advanced ensemble construction and analysis tools provide many options for matching chains [44,70], which are also provided by the Scipion-EM-ProDy plugin (see Materials and Methods, Section 4.1.3 and Figure 4).

One of the chain matching options where the user can provide a custom dictionary option was very useful in our recent D614G spike analysis [63]. In this case, structures largely followed either one of two conventions from the first wildtype (WT) structures, with the first chain with an RBD up conformation as chain A and the other chains labelled B and C in an anticlockwise direction looking

down on the RBDs (PDB: 6VSB) [71] or with the first chain with an RBD up conformation as chain B and the other chains labelled B and C in a clockwise direction (PDB: 6VYB) [72], creating chain orders ABC and BCA (see supplementary Table F in [63] and Figure 4c). There was also one structure from intact virions, which was determined around the same time as these two WT structures and has a different arrangement again, resulting in a chain order BAC [73]. Additionally, there was one paper with two structures with two RBDs up with different chain IDs that needed to be made to match (PDB: 7EB4, 7EB5) [74]. We, therefore, used this approach again here in the ensemble building step (see Figure 4b,c).
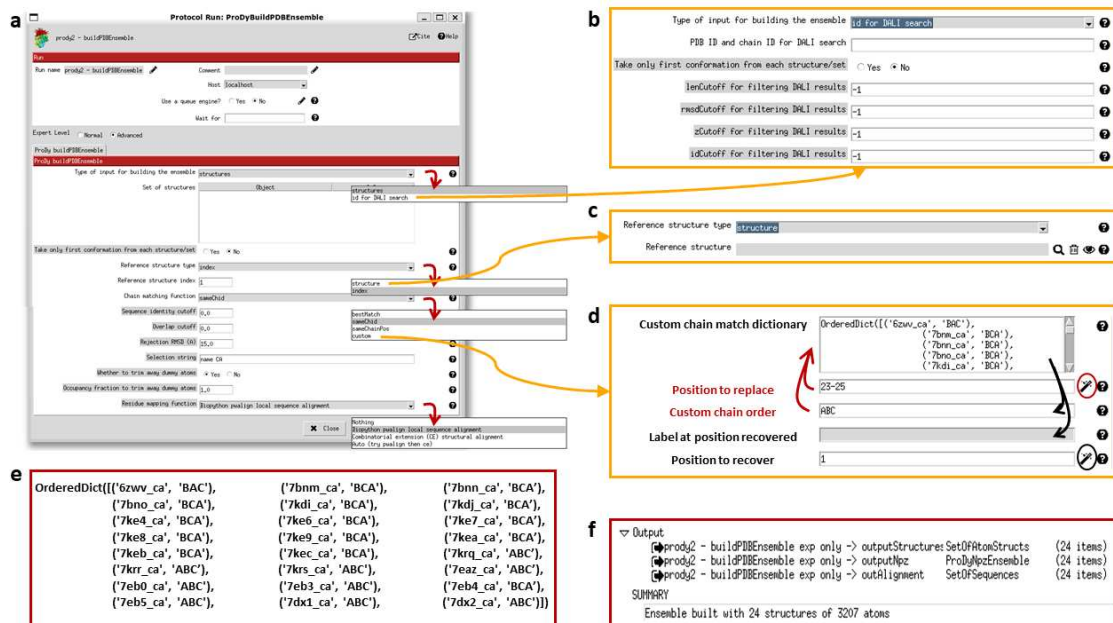

Figure 4. Technical details of the ensemble building form and its usage. (**a**) The ensemble building form is shown with each of the drop-down menus expanded to show the options. The alternative parts of the form that result from selecting particular options are shown in yellow boxes connected to arrows selecting these options. (**b**) Selecting the DALI option for input structures adds new filtering options for the results from the web server. (**c**) Selecting to use another structure as a reference replaces the text box for the reference structure ID with a pointer selector for the new reference structure. (**d**) Selecting the custom option for chain matching brings up the custom chain matching dictionary generation wizards. The first wizard (circled in red) enables the replacement of entries in the dictionary by specifying a position and match order in the two boxes with red labels as shown by the two red arrows. The second wizard (circled in black) recovers items from the dictionary to make it easier to check and update them. It fills the chain order and label boxes as indicated by the black arrows. (**e**) The full custom match dictionary used in the analysis of the 24 D614G structures is shown, matching our previous study [63]. The first D614G structure determined from intact viruses (used as a reference here) is given chain order BAC and the others have orders BCA and ABC to match it. The rest of the structures have similar chain orders, giving two blocks with an exception for one of the structures (PDB: 7EB4). (**f**) The output of the ensemble building protocol for the 24 D614G structures is shown, highlighting the different objects generated and the summary information provided.

In order to compare against simulations, we selected one of these structures for several generations of ClustENM and projected both the experimental and simulated ensembles onto the first two PCs from the experimental ensemble (see Figure 3c). To have a good starting point with relevant flexibility, we picked a structure with one RBD up and one RBD in an intermediate (I) state (1-up/1-I) (PDB: 7KEC) [75]. To inform this decision, we performed an RMSD clustering on the ensemble using an UPGMA tree and a cutoff of 2.5 Å, yielding 7 cluster representatives belonging to a range of states including 3-down, 1-I, 1-up, 1-up/1-I, and 2-up. We then ran NMA for the most distinct cluster representatives

and compared the first five NMs to the PCs from the ensemble. In contrast to the 3-down state, where there were only considerable overlaps to PCs 4 and 5, the first NMs of the 1-up/1-I state showed overlaps > 0.40 for the first two PCs (see Figure 3d). A similar behaviour with slightly higher overlaps was seen for the 2-up structure (PDB: 7EB4), but we decided not to use that state as it is more different to the others.

Prior to ClustENM, we used the separate PDBFixer protocol to fix missing residues and atoms and then selected the core structure without the termini. We then ran the ClustENM simulation with 6 modes and 5 generations of 50 conformers, each having an average RMSD of 2 Åfrom the previous conformer, and used the maximum number of clusters option with values of 10, 20, 30, 40 and 50 for the five generations. All other parameters were left at their default values.

In order to compare the two ensembles, we also trimmed away flexible loops that were missing in the experimental ensemble (see Materials and Methods and Figure 3e) and sliced the PCA vectors accordingly. This allowed the experimental and simulated structures to be projected onto the same landscape (Figure 3d), showing that the simulation (blue) explores the region between 3-down and 1-up, but does not explore as wide a region as the experimental structures (orange) and does not approach the 2-up state.

3. Discussion

Cryo-electron microscopy has advanced enormously over the last decade and now has the capability to fulfill its promise as a single particle structural biology technique with a host of new computational methods capitalising on the potential to learn continuous structural heterogeneity information from each of the particle images in a data set [6,9,10]. All these methods are generating landscapes that distinguish these images by various criteria, including compositional and conformational differences, and can generate estimates of maps in different conformational states for different regions or particles in these landscapes, but it is often difficult to interpret these results. We introduced a new plugin for ProDy within the Scipion workflow engine to help with this as illustrated in our various example workflows. In particular, we show how principal component analysis of atomic structure ensembles including existing structures can help contextualise conformational landscapes and illustrate that the hybrid simulation method ClustENM(D) can be used to rapidly improve structures and sample conformational space.

4. Materials and Methods

We first review the underlying implementation of Scipion-EM-ProDy and then highlight some key protocols and their associated pipelines, focusing on those with the richest functionality, namely those for 1) pairwise alignment of atomic structures and ensemble construction from multiple atomic structures; 2) ANM NMA and PCA; 3) GNM analysis and domain decomposition; and 4) ClustENM(D) hybrid simulations. We also include examples of key parameters, including those used in the example cases presented in Results Sections 2.1 (Figure 2) and 2.4 (Figure 3).

All analyses presented here are based on publicly available structures from the protein databank (PDB) [67] and use standard computational biophysics methods available in ProDy as previously described in detail [44,76]. Scipion-EM-ProDy was installed in development mode, providing the latest development versions of ProDy and the plugin, similar to release versions 2.4.1 and 3.3.0, respectively.

4.1. Integration of ProDy pipelines into Scipion workflows

All software was written in Python. Scipion-EM-ProDy is a new Scipion plugin following standard Scipion plugin conventions and is available at https://github.com/scipion-em/scipion-em-prody as briefly described in [30]. Some changes have also been made to the Scipion-EM core package found at https://github.com/scipion-em/scipion-em, including the addition of the new object SetOfPrincipalComponents. Collaborative software development for these was performed

through branches on the same fork with pull requests into the main devel branch and later into the master master branches as is standard for Scipion.

ProDy has already existed for over a decade [44] and has been developed further using its standard conventions and existing Github repositories including https://github.com/prody/ProDy and https://github.com/jamesmkrieger/ProDy. Collaborative development follows a personal forks and branches approach with pull requests to the main prody/ProDy master branch as is generally the standard for ProDy.

Scipion-EM-ProDy downloads ProDy from GitHub in two different ways. Either it downloads the tar.gz archive for the latest compatible release (currently v2.3.1) or it clones the latest compatible development code from github.com/jamesmkrieger/ProDy/tree/scipion. These are then both installed in the same way via two commands for building the C/C++ extensions in place and installing ProDy with pip in editable development mode.

4.1.1. Building upon ProDy classes, functions and apps to create Scipion protocols and workflows

The Scipion-EM-ProDy plugin ensures a smooth hand-over of data from Scipion to ProDy and back through the use of corresponding objects (see Table 2). For Scipion, there are various classes of objects that point to files or items within them (among others) through the use of SQLite tables [31]. These pointers have file name attributes that can be passed on to programs such as ProDy to read the data from those files directly into memory as their own objects.

For example, the common step of a parsing an atomic structure uses the Scipion AtomStruct object that points to a file containing an atomic structure (in either PDB or PDBx/mmCIF format) that ProDy parses with the function parsePDB into Atomic objects (such as AtomGroup or Selection) containing information about all the atoms in the structure such as residue names and numbers and Cartesian coordinates. After some calculations based on this structure, the results from ProDy are written to new files and registered in Scipion as metadata and file pointers in the SQLite tables. In the case of atom selection, a new PDB file containing the subset of atoms, which is registered as another AtomStruct.

Table 2. Table relating Scipion and ProDy objects

Scipion object	ProDy objects
AtomStruct, SetOfAtomStructs	Atomic[1], AtomGroup, Selection
NormalMode	VectorBase[1], Vector, Mode
SetOfNormalModes	NMA[1], ANM, RTB, GNM
SetOfPrincipalComponents[2]	PCA
TrajFrame[3]	Conformation, Frame
SetOfTrajFrames[3]	Ensemble, Trajectory
ProDyNpzEnsemble[3]	PDBEnsemble

[1] Base/parent class. [2] New class based on SetOfNormalModes. [3] Part of Scipion-EM-ProDy at the time of writing.

The main way to use ProDy is via the Python application programming interface (API), a Python package with some C and C++ extensions, providing a rich set of classes and functions for programmers to incorporate into pipelines either interactively or in scripts. There are also command line applications, which handle many of the key steps such as fetching structures from the PDB, selecting atoms, running ANM NMA, and performing PCA on simple ensembles, and are also accessible through an even more limited graphical interface inside NMWiz. Scipion-EM-ProDy gains the benefits of both of these two components: The Python API is useful for steps requiring more flexibility and not included in the apps such as alignment and ensemble construction using structures with multiple chains, and using the apps provides better control for the simpler and more computationally expensive activities and makes it easier to terminate the processes.

The Scipion-EM-ProDy plugin has been developed hand-in-hand with further developments to ProDy to make this as smooth a process as possible. This includes the addition of new ProDy functions

writeScipionModes and parseScipionModes for writing and parsing normal modes and related objects in Scipion format, where each eigenvector is written to its own file within a particle folder with one line of three values per atom and a single SQLite file with entries pointing to each one provides additional metadata such as eigenvalue, collectivity and a score based on the two from HEMNMA [22].

The apps were also updated to include additional features that are useful for including them in workflows, such as a limit to number of processors used and the option to include Hessian matrices in the NPZ (NumPy [77] pickle with gzip compression) files that ProDy uses for use in later steps such as vibrational subsystem analysis (see Section 4.2.1.1).

We next present the protocols in the four main categories in which they are used in a pipeline as shown in Figure 2a.

4.1.2. Protocols for atomic structure operations

The first step in a ProDy pipeline is to parse one atomic structure or multiple atomic structures and perform some operations on them, such as reconstructing biological molecular assemblies, fixing the structure to account for missing atoms, selecting atoms, and aligning structures to each other in pairs or ensembles. These steps can also be used later in a process to help with the analysis as in Figure 3. The corresponding protocols can fetch files from the PDB, read files from their paths or take pointers to Scipion objects as inputs. The outputs are one or more PDB files registered as AtomStruct or SetOfAtomStructs pointers as discussed below and there is also a summary of the number of structures and atoms.

The Scipion protocol for importing sets of atomic structures is especially useful here, allowing the download of several structures into files specified in a SetOfAtomStructs object via its associated SQLite table. This was particularly useful for the 24 D614G spike structures using the following PDB ids: 6ZWV, 7BNM, 7BNN, 7BNO, 7KDI, 7KDJ, 7KE4, 7KE6, 7KE7, 7KE8, 7KE9, 7KEA, 7KEB, 7KEC, 7KRQ, 7KRR, 7KRS, 7EAZ, 7EB0, 7EB3, 7EB4, 7EB5, 7DX1, 7DX2.

Reconstructing biological molecular assemblies

In several cases, especially when structures come from X-ray crystallography, the structure stored in the files downloaded from the PDB does not correspond to the biologically relevant assemblies (of which there can sometimes be several) and some symmetry operations are needed to reconstruct them. ProDy handles this through some arguments in parsePDB and the plugin provides a protocol to do this. It writes a PDB file for each assembly found and the Scipion output is a SetOfAtomStructs object pointing to them.

Atom Selection

It is often helpful to focus on particular chains, residues or atom types to obtain more meaningful results or efficient calculations. For example, NMA and PCA are often performed with Cα atoms as the other atoms are not needed to describe global motions [76] and the global rearrangements of particular domains can be important for signalling as is the case with N-terminal venus fly trap domains from various receptors [78].

The plugin uses the select app, creating a new file, and the output is a new AtomStruct. The form has a box for entering a ProDy selection string, which is similar to that used by VMD. The syntax is described extensively with examples in tutorials on the ProDy website and a link to the relevant tutorial page is provided under the help option on the form. The default selection is "protein and name CA or nucleic and name P C4' C2" in line with ProDy itself, based on various previous studies.

For the combined workflow in Figure 3, the selection string for trimming the termini prior to ClustENM simulations was "protein and resnum 26 to 1147". We used the following selection string to trim loops to match the ClustENM ensemble to the experimental one, based on an analysis of missing atoms in the experimental ensemble in PyMOL: "name CA and resnum 27 to 66 82 to 113 116 to 140 166 to 172 186 to 196 200 to 211 215 to 242 264 to 442 491 to 501 503 to 515 522 to 620 641 to 676 690 to

811 813 to 827 856 to 939 944 to 956 958 to 1006 1008 to 1146". The new structures have 2,745 Cα atoms as compared to the original 51,737 atoms from 3,366 residues in the full ClustENM structures.

4.1.3. Pairwise alignment and ensemble construction

Many comparisons should be made following alignment of atomic structures, either in pairs or larger numbers. This requires appropriate matching of residues and chains prior to structural superposition. ProDy provides several options for each type of matching [44,70] and performs superpositions using the Kabsch algorithm with the matching points [79].

The Scipion-EM-ProDy plugin provides two protocols for this: one for pairwise alignment and one for ensemble building. As in the ProDy API (where the function buildPDBEnsemble provides a wrapper for alignChains), these tasks and their protocol forms are very similar and we mostly present the ensemble building one here (see Figure 4).

The pairwise alignment protocol has two fields for inputs, which select Scipion pointers for the mobile and target structures. The ensemble building one has the option to either take inputs from existing structures inside Scipion (which can be any combination of individual structures and sets of them) or to search for new ones using the DALI web server [80]. Currently, the DALI option cannot handle multiple chains as in ProDy itself, but single chain analyses can also be very informative and DALI alignments are pretty good as shown in our signature dynamics (SignDy) paper [70].

For residue alignment, there are options to use Biopython pairwise sequence alignment [81] or combinatorial extension (CE) structural alignment [82] or to try both. There is also an option to use the mappings from DALI [80]. For chain matching, the options are 1) select same chain ID; 2) select same chain position; 3) use an automated matching scheme that optimises various features including sequence ID and alignment coverage for each chain and overall RMSD; and 4) manually input a custom dictionary of corresponding chain orders.

There are several cutoff options for accepting or rejecting alignments, which could rule out alignments from particular methods and favour the outputs of others or exclude particular structures from the ensemble altogether. In most cases, these are sequence identity and coverage for each chain and an RMSD for the whole structure. When using DALI, additional options are available such as the DALI Zscore (see Figure 4b). These comparisons are all relative to the reference structure, which can be selected using an index from the provided structures or as another structure (see Figure 4c). There is also an option for whether to include dummy atoms when the reference structure has atoms that are not found in other structures or to trim them away depending on an occupancy cutoff and a selection string, which allows the selection of particular chains, residues or atoms.

Figure 4d shows the custom chain matching implementation, which is based on two wizards (triggered by magic wand icons). The use of either wizard first results in the custom chain matching dictionary is populated automatically. Clicking the first wizard icon (circled in red) allows the dictionary to be updated by filling in the first two boxes with red labels and arrows. Values can also be recovered for checking or updating the matching order using the second wizard (circled in black) as indicated by the black arrows. The final dictionary used for the the D614G analysis is shown in Figure 4e.

The ensemble building protocol uses buildPDBEnsemble, which aligns each structure to the reference and then iteratively superposes the matching atoms to the average structure until the average converges as this is necessary for PCA. The outputs (shown in Figure 4f) are 1) a SetOfAtomStructs object pointing to a number of PDB files with aligned structures based on the starting structures; 2) a ProDyNpzEnsemble object, which points to entries in ProDy's specialised NPZ file for the ensemble; and 3) a sequence alignment where each row is a structure from the ensemble. The summary gives the number of structures and atoms in the ensemble, which in this case includes dummy atoms.

4.1.4. Protocols for calculating global modes of motion

ProDy uses matrix decomposition methods to calculate global modes of motion and these all work in similar ways, but with different matrices. Each one has a class based on the NMABase parent class with methods for setting coordinates, building the relevant matrices and calculating the modes from eigenvalue decomposition. This is handled by an internal utilities function solveEig, which calls relevant methods from Scipy [83] if available or otherwise from NumPy [77]. These classes are initialised and used in the corresponding apps, which are used the Scipion-EM-ProDy plugin. We briefly describe these here, but refer the reader to previous reviews (such as [39,76,84]) for more details. All these protocols produce NPZ files, which include the matrices and can be read back into ProDy, as well as NMD files for visualisation in NMWiz and sets of files in Scipion format with corresponding SetOfNormalModes and SetOfPrincipalComponents pointers. These pointers also have associated AtomStruct pointers to keep track of the starting structures.

The protocol output also includes animations generated using ProDy's traverseMode function and corresponding VMD scripts and motion statistics for visualisation with the modes viewer from ContinuousFlex. All these forms look very similar too, with one tab for the main parameters and another for controlling the animations, e.g. setting the motion size in RMSD or the direction. We also provide a new ProDy mode viewer, which opens VMD and NMWiz to analyse the results. It can also read data from other SetOfNormalModes objects and write out new NMD files if they are not present.

Deformation vector analysis

As well as calculating modes of motion from matrix decomposition, there is also an option of calculating a simple $3N$-dimensional deformation vector $\mathbf{q}$ between two structures, describing the conformational change between them by subtracting the positions of the N corresponding atoms after alignment and superposition. This is equivalent to a morph and is useful for both visualisation and comparison to the results from the other methods.

Principal component analysis

PCA is based on decomposition of the $3N \times 3N$ covariance matrix of atomic positions from an ensemble of structures. This matrix is the average of the dot products $\mathbf{q} \cdot \mathbf{q}^\mathsf{T}$ of the deformation vectors between each of the structures and the converged average structure. PCA decompositions finds the components of this covariance matrix that best explain the variance, which are usually interpretable conformational changes similar to those observed in morphs and more complicated methods. The input to this protocol is a pointer to an aligned ensemble as either a SetOfAtomStructs (which can also come from ContinuousFlex or FlexUtils) or ProDyNpzEnsemble object. It also creates a fractional variance plot, which is saved as an image.

Normal mode analysis protocols

NMA is based on decomposition of the $3N \times 3N$ Hessian matrix of second derivatives of the interatomic interaction potential energy. This matrix can be built using various potential functions, but the most common and efficient ones are elastic network models (ENMs). Protocols for two common ENMs are provided here, namely the anisotropic network model (ANM) [56,85,86] and the rotating translating blocks (RTB) model [59,60]. Both of these are also provided by ContinuousFlex and we have confirmed that it is possible to obtain very similar results, but ProDy is more flexible about options and can be more efficient in some cases. For the spike structures, we used the ANM with default parameters, including a cutoff of 15 Åand skipping zero eigenvalue modes. It is also possible to provide spring constant (gamma) and cutoff parameters as text to call other ENMs, such as GammaED and "2.9 * math.log(214) - 2.9" (replacing 214 for the number of residues in the protein used) for the essential dynamics (ED)-refined ENM [58] or GammaStructureBased for a version that takes into account secondary structure [57].

ContinuousFlex automatically chooses the model based on the input structure and uses the RTB for atomic structures and the regular ANM for pseudoatoms [33], whereas ProDy can use either model for either situation. ProDy also does not provide an automatic relative cutoff option like ContinuousFlex either, but it does provide suggestions on potential cutoff distances based on previous studies, such as 15 or 18 Å for Cα atoms [86], 5 to 7 Å for all atoms [85], or longer distances for fewer atoms depending on the level of coarse-graining [87]. ProDy also has additional options, such as optimising blocks in RTB by splitting merging based on distance and minimum and maximum size or selecting them based on secondary structures. This overcomes some problems we have come across from having blocks with small numbers of residues or residues at opposite ends of loops with missing residues.

4.2. Gaussian network model analysis and domain decomposition

The GNM is a related elastic network model from statistical mechanics of polymer networks [55], which has an $N \times N$ Kirchoff matrix and thus does not provide three-dimensional information on movements. It is based on the assumption of isotropic Gaussian fluctuations of atom positions and distance vectors and incorporates more realistic constraints that result in better agreement with a wide range of experimental data [42]. This approach can be very useful for predicting mean square fluctuations and cross-correlations and identifying critical regions such as hinge sites [42]. Two types of cross-correlation matrices are returned by the protocol: a raw covariance matrix and a normalised orientation cross-correlation matrix.

Scipion-EM-ProDy has a dedicated GNM viewer (see Figure 5a), which provides options for viewing mean square fluctuations (MSFs), root mean square fluctuations (RMSFs), and covariance and orientational cross-correlation matrices, estimated from either the full set of GNM modes or any individual mode or range of modes, as well as a per-atom maximum distance profile like that from ContinuousFlex and the GNM mode shape, which shows regions moving in opposite directions (Figure 5b), and to open NMWiz with structures coloured by GNM modes (Figure 5c).

GNM analysis can also be used in various pipelines similarly to NMA and PCA as shown in Figure 5d for an example using adenylate kinase (PDB: 4AKE, chain A). One useful application of GNM is dynamical domain decomposition, based on a spectral clustering using GNM modes [61,88], which is also provided in a protocol in Scipion-EM-ProDy. The output of this protocol is a PDB file with the b-factor column replaced by dynamical domain label and a VMD script to show a the structure as beads coloured by b-factors. There is then a viewer that displays this coloured structure (Figure 5e).
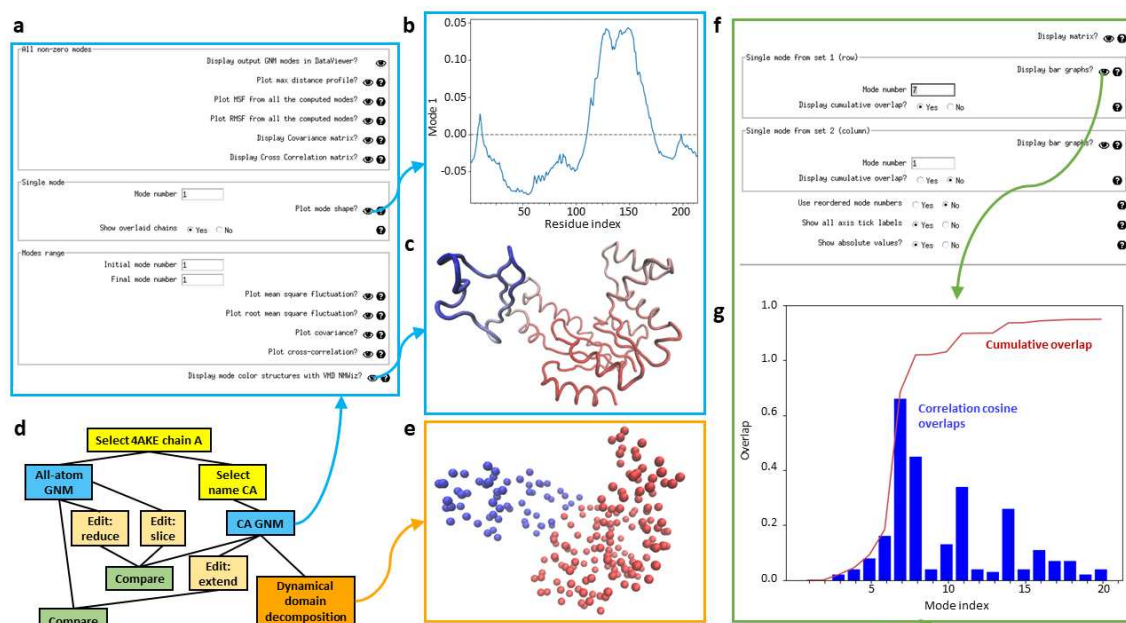
Figure 5. Illustration of GNM and mode comparison viewers. (**a**) The GNM viewer form is shown for the Cα atom GNM from the workflow in panel **d** based on an open adenylate kinase structure (PDB: 4AKE, chain A; blue arrow going up). The form has different sections for properties based on all modes or individual modes or mode ranges. Certain results are highlighted by arrows and shown in panels **b** and **c**. (**b**) The mode shape is plotted as a line with atoms moving in opposite directions have positive and negative signs. (**c**) The NMWiz viewer colours the structure by the mode shape. (**d**) An example workflow using GNM is shown using open adenylate kinase (PDB: 4AKE). Two different atoms selections in yellow precede two GNM calculations in blue using all atoms of chain A and only the Cα atoms. The latter is used for dynamical domain decomposition (orange) and the two of them are used to illustrate mode editing (light orange) and comparison (green). (**e**) The output from the dynamical domain decomposition wizard using two modes (including the zero mode) is a structure coloured by dynamical domains in line with the behaviour in the first non-zero mode shown above. (**f**) The mode comparison viewer is shown for the comparison between the all-atom GNM and the extended CA GNM. (**g**) The result from selecting the option to display the row from comparing mode 7 (non-zero mode 1) from set 1 to all the others with cumulative overlaps enabled is shown. Correlation cosine overlaps are blue bars and the cumulative overlap is shown with a dark red line.

4.2.1. Protocols for downstream analysis

Scipion-EM-ProDy also provides a number of protocols to assist with downstream analysis including editing the modes to have similar numbers of atoms to others, comparing modes from different calculations, and projecting ensembles onto modes to visualise conformational landscapes.

Mode editing

It is often useful to apply slice operations to the eigenvectors from a set of modes to ensure they have the same number of atoms as other modes or structures or ensembles as we did for the comparison between experimental and simulated structures. ProDy also has an extend operation that allows eigenvectors calculated for Cα atoms to be copied to all atoms in the same residues, and the slice operation can be applied to do the opposite.

Another useful mode editing method is a Hessian reduction method, also known as subsystem-environment analysis or vibrational subsystem analysis, which calculates an effective Hessian for part of the system taking into account the effects of the rest of the structure as its environment [89–91]. This method is useful for assessing the impact on a particular domain from its interactions with the rest of the structure [91] as well as for accounting for the effect of a membrane [92,93].

These three mode editing methods have been incorporated as options in a single protocol along with additional method for thin-plate spline interpolation, which has been shown to be useful for extending coarse-grained modes based on pseudoatoms to atoms [94], that we added into ProDy as well. This protocol requires 3D modes and cannot be used with GNM modes.

These protocols take a SetOfNormalModes or SetOfPrincipalComponents object as input along with a new AtomStruct containing the new nodes and compare them to find corresponding atoms using the same chain IDs and the default residue alignment, which first tries matching residue number then Biopython sequence alignment followed by CE structural alignment. The output is a new SetOfNormalModes or SetOfPrincipalComponents with an associated AtomStruct corresponding to the new nodes. Thus, these methods are currently limited to comparing modes based on atomic structures, but could be extended to pseudoatoms with methods such as nearest neighbours [61].

Mode comparison

Modes of motion can be compared in various ways [70,95], many of which are provided by ProDy. Scipion-EM-ProDy currently provides the correlation cosine overlap (also called inner products or projection products) [96], the root-weighted square inner product [97] and the covariance overlap [98]. The resulting matrix is written to a text file using ProDy's writeArray function and is registered in Scipion using the generic EMFile pointer. There is also an option to match and reorder modes based on linear assignment [70], which returns an additional file with mode matches and another SetOfNormalModes with the old IDs stored in the metadata. It is also possible to only calculate the diagonal values of the comparison matrix, equivalent to comparing the matching modes only. Lastly, there is an option to normalise overlaps or retain the raw projection products.

This protocol has an associated viewer (see Figure 5f) that can show the whole matrix as a heatmap (see Figure 3c) or particular rows (or the diagonal) using bar graphs along with the cumulative overlaps as the square root of the sum of the squared overlaps if desired (see Figure 5g).

Landscape projection

Projection of an ensemble of structures onto a set of modes is a simple operation based on the dot product of the mode vectors and the deformation vectors from the average structures, which can be normalised or scaled to represent RMSDs. This can be shown as a set of points in a 2D or 3D scatter plot for two or three modes (see Figure 3b) or as a line for a single mode with the y-axis representing the projection and the x-axis being the conformation index. Alternatively, these can be approximated with a kernel density estimate for two modes or a histogram for one mode.

4.2.2. ClustENM(D) hybrid simulations

The ClustENM class in ProDy enables ClustENM and ClustENMD hybrid simulations using OpenMM. Scipion-EM-ProDy has a protocol for running these simulations using this class and the many parameters are spread across different tabs of the protocol form: one for general parameters, one for normal mode parameters and one for all-atom minimisation and MD simulation parameters. It has the option to take multiple structures as inputs and run an independent simulation for each one, making it useful for refining a set of structures from continuous heterogeneity analysis (see Figure 1).

4.2.3. Protocols for imports

Additionally, Scipion-EM-ProDy has two protocols for importing modes or ensembles calculated outside that Scipion project. The ensemble import protocol can also be used for trimming ensembles as done in the pipeline in Figure 3a.

Data Availability Statement: Scipion and ProDy are open source software and all code is available on Github at https://github.com/prody/ProDy, https://github.com/scipion-em/scipion-em and https://github.com/scipion-em/scipion-em-prody. Stable master versions are also available on the Python package index (PyPI) and can be installed with pip and the Scipion plugin manager.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANM	Anisotropic network analysis
API	Application programming interface
PDB	Protein data bank
CE	combinatorial extension
Cryo-EM	Cryo-electron microscopy
dim. red.	dimensionality reduction
GNM	Gaussian network analysis
I	intermediate state for RBD
NMA	Normal mode analysis
NM	Normal mode
NMWiz	Normal mode wizard
PCA	Principal component analysis
PC	Principal component
ps	picoseconds
PDB	Protein data bank
RBD	Receptor-binding domain
RMSD	Root-mean-square deviation
RTB	Rotating and translating blocks
SPA	Single particle analysis
VMD	Visual molecular dynamics

References

1. Henzler-Wildman, K.; Kern, D. Dynamic personalities of proteins. *Nature* **2007**, *450*, 964–972. doi:10.1038/nature06522.
2. Bahar, I.; Cheng, M.H.; Lee, J.Y.; Kaya, C.; Zhang, S. Structure-Encoded Global Motions and Their Role in Mediating Protein-Substrate Interactions. *Biophysical Journal* **2015**, *109*, 1101–1109. doi:10.1016/j.bpj.2015.06.004.
3. Harpole, T.; Delemotte, L. Conformational landscapes of membrane proteins delineated by enhanced sampling molecular dynamics simulations. *Biochim Biophys Acta Biomembr* **2018**, *1860*, 909–926.
4. van den Bedem, H.; Fraser, J. Integrative, dynamic structural biology at atomic resolution–it's about time. *Nat Methods.* **2015**, *12*, 307–18. doi:10.1038/nmeth.3324.
5. Srivastava, A.; Tiwari, S.; Miyashita, O.; Tama, F. Integrative/Hybrid Modeling Approaches for Studying Biomolecules. *J Mol Biol.* **2020**, *432*, 2846–2860. doi:10.1016/j.jmb.2020.01.039.
6. Toader, B.; Sigworth, F.J.; Lederman, R.R. Methods for Cryo-EM Single Particle Reconstruction of Macromolecules Having Continuous Heterogeneity. *Journal of Molecular Biology* **2023**, p. 168020. doi:https://doi.org/10.1016/j.jmb.2023.168020.

7. Bonomi, M.; Vendruscolo, M. Determination of protein structural ensembles using cryo-electron microscopy. *Curr Opin Struct Biol.* **2019**, *56*, 37–45. doi:10.1016/j.sbi.2018.10.006.

8. Sorzano, C.; Jiménez, A.; Mota, J.; Vilas, J.; Maluenda, D.; Martínez, M.; Ramírez-Aportela, E.; Majtner, T.; Segura, J.; Sánchez-García, R.; Rancel, Y.; Del Caño, L.; Conesa, P.; Melero, R.; Jonic, S.; Vargas, J.; Cazals, F.; Freyberg, Z.; Krieger, J.; Bahar, I.; Marabini, R.; Carazo, J. Survey of the analysis of continuous conformational variability of biological macromolecules by electron microscopy. *Acta Crystallogr F* **2019**, *75*, 19–32. doi:10.1107/S2053230X18015108.

9. Donnat, C.; Levy, A.; Poitevin, F.; Zhong, E.; Miolane, N. Deep generative modeling for volume reconstruction in cryo-electron microscopy. *J Struct Biol.* **2022**, *214*, 107920. doi:10.1016/j.jsb.2022.107920.

10. Tang, W.S.; Zhong, E.D.; Hanson, S.M.; Thiede, E.H.; Cossio, P. Conformational heterogeneity and probability distributions from single-particle cryo-electron microscopy. *Current Opinion in Structural Biology* **2023**, *81*, 102626. doi:https://doi.org/10.1016/j.sbi.2023.102626.

11. Herreros, D.; Lederman, R.; Krieger, J.; Jiménez-Moreno, A.; Martínez, M.; Myška, D.; Strelak, D.; Filipovic, J.; Bahar, I.; Carazo, J.; Sanchez, C. Approximating deformation fields for the analysis of continuous heterogeneity of biological macromolecules by 3D Zernike polynomials. *IUCrJ* **2021**, *8*, 992–1005. doi:10.1107/S2052252521008903.

12. Harastani, M.; Vuillemot, R.; Hamitouche, I.; Moghadam, N.B.; Jonic, S. ContinuousFlex: Software package for analyzing continuous conformational variability of macromolecules in cryo electron microscopy and tomography data. *Journal of Structural Biology* **2022**, *214*, 107906. doi:https://doi.org/10.1016/j.jsb.2022.107906.

13. Punjani, A.; Fleet, D. 3DFlex: determining structure and motion of flexible proteins from cryo-EM. *Nat Methods* **2023**, *20*, 860–870. doi:10.1038/s41592-023-01853-8.

14. Klebl, D.; Gravett, M.; Kontziampasis, D.; Wright, D.; Bon, R.; Monteiro, D.; Trebbin, M.; Sobott, F.; White, H.; Darrow, M.; Thompson, R.; Muench, S. Need for Speed: Examining Protein Behavior during CryoEM Grid Preparation at Different Timescales. *Structure* **2020**, *28*, 1238–1248.e4. doi:10.1016/j.str.2020.07.018.

15. Sanchez Sorzano, C.O.; Alvarez-Cabrera, A.L.; Kazemi, M.; Carazo, J.M.; Jonić, S. StructMap: Elastic Distance Analysis of Electron Microscopy Maps for Studying Conformational Changes. *Biophysical Journal* **2016**, *110*, 1753–1765. doi:https://doi.org/10.1016/j.bpj.2016.03.019.

16. Sorzano, C.; Martin-Ramos, A.; Prieto, F.; Melero, R.; Martin-Benito, J.; Jonic, S.; Navas-Calvente, J.; Vargas, J.; Oton, J.; Abrishami, V.; de la Rosa-Trevin, J.; Gomez-Blanco, J.; Vilas, J.; Marabini, R.; Carazo, J. Local analysis of strains and rotations for macromolecular electron microscopy maps. *Journal of Structural Biology* **2016**, *195*, 123–128.

17. Pettersen, E.; Goddard, T.; Huang, C.; Meng, E.; Couch, G.; Croll, T.; Morris, J.; Ferrin, T. UCSF ChimeraX: Structure visualization for researchers, educators, and developers. *Protein Sci* **2021**, *30*, 70–82.

18. Ecoffet, A.; Poitevin, F.; Dao Duc, K. MorphOT: transport-based interpolation between EM maps with UCSF ChimeraX. *Bioinformatics* **2021**, *36*, 5528–5529. doi:10.1093/bioinformatics/btaa1019.

19. Zhong, E.; Bepler, T.; Berger, B.; Davis, J. CryoDRGN: reconstruction of heterogeneous cryo-EM structures using neural networks. *Nature Methods* **2021**, *18*, 176–185. doi:https://doi.org/10.1038/s41592-020-01049-4.

20. Kinman, L.; Powell, B.; Zhong, E.; .; Berger, B.; Davis, J. Uncovering structural ensembles from single-particle cryo-EM data using cryoDRGN. *Nat Protoc* **2023**, *18*, 319–339. doi:10.1038/s41596-022-00763-x.

21. Malhotra, S.; Träger, S.; Dal Peraro, M.; Topf, M. Modelling structures in cryo-EM maps. *Curr Opin Struct Biol* **2019**, *58*, 105–114. doi:10.1016/j.sbi.2019.05.024.

22. Jin, Q.; Sorzano, C.; de la Rosa-Trevín, J.; Bilbao-Castro, J.; Núñez Ramírez, R.; Llorca, O.; Tama, F.; Jonić, S. Iterative Elastic 3D-to-2D Alignment Method Using Normal Modes for Studying Structural Dynamics of Large Macromolecular Complexes. *Structure* **2014**, *22*, 496–506.

23. Harastani, M.; Eltsov, M.; Leforestier, A.; Jonić, S. HEMNMA-3D: Cryo Electron Tomography Method Based on Normal Mode Analysis to Study Continuous Conformational Variability of Macromolecular Complexes. *Frontiers in Molecular Biosciences* **2021**, *8*, 663121.

24. Vuillemot, R.; Miyashita, O.; Tama, F.; Rouiller, I.; Jonic, S. NMMD: Efficient Cryo-EM Flexible Fitting Based on Simultaneous Normal Mode and Molecular Dynamics atomic displacements. *Journal of Molecular Biology* **2022**, *434*, 167483.

25. Vuillemot, R.; Mirzaei, A.; Harastani, M.; Hamitouche, I.; Fréchin, L.; Klaholz, B.; Miyashita, O.; Tama, F.; Rouiller, I.; Jonic, S. MDSPACE: Extracting Continuous Conformational Landscapes from Cryo-EM Single Particle Datasets Using 3D-to-2D Flexible Fitting based on Molecular Dynamics Simulation. *J Mol Biol* **2023**, *435*, 167951. doi:10.1016/j.jmb.2023.167951.

26. Cossio, P.; Hummer, G. Bayesian analysis of individual electron microscopy images: towards structures of dynamic and heterogeneous biomolecular assemblies. *J Struct Biol* **2013**, *184*, 427–437. doi:10.1016/j.jsb.2013.10.006.

27. Giraldo-Barreto, J.; Ortiz, S.; Thiede, E.; Palacio-Rodriguez, K.; Carpenter, B.; Barnett, A.; Cossio, P. A Bayesian approach to extracting free-energy profiles from cryo-electron microscopy experiments. *Sci Rep* **2021**, *11*, 13657. doi:10.1038/s41598-021-92621-1.

28. Tang, W.; Silva-Sánchez, D.; Giraldo-Barreto, J.; Carpenter, B.; Hanson, S.; Barnett, A.; Thiede, E.; Cossio, P. Ensemble Reweighting Using Cryo-EM Particle Images. *J Phys Chem B* **2023**, *127*, 5410–5421. doi:10.1021/acs.jpcb.3c01087.

29. Hamitouche, I.; Jonić, S. DeepHEMNMA: ResNet-based hybrid analysis of continuous conformational heterogeneity in cryo-EM single particle images. *Frontiers in Molecular Biosciences* **2022**, *9*, 965645.

30. Herreros, D.; Krieger, J.M.; Fonseca, Y.; Conesa, P.; Harastani, M.; Vuillemot, R.; Hamitouche, I.; Serrano Gutiérrez, R.; Gragera, M.; Melero, R.; Jonic, S.; Carazo, J.M.; Sorzano, C.O.S. *Scipion* Flexibility Hub: an integrative framework for advanced analysis of conformational heterogeneity in cryoEM. *Acta Crystallographica Section D* **2023**, *79*, 569–584. doi:10.1107/S2059798323004497.

31. Conesa, P.; Fonseca, Y.; Jiménez de la Morena, J.; Sharov, G.; de la Rosa-Trevín, J.; Cuervo, A.; García Mena, A.; Rodríguez de Francisco, B.; del Hoyo, D.; Herreros, D.; Marchan, D.; Strelak, D.; Fernández-Giménez, E.; Ramírez-Aportela, E.; de Isidro-Gómez, F.; Sánchez, I.; Krieger, J.; Vilas, J.; del Cano, L.; Gragera, M.; Iceta, M.; Martínez, M.; Losana, P.; Melero, R.; Marabini, R.; Carazo, J.; Sorzano, C. Scipion3: a workflow engine for cryo-electron microscopy image processing and structural biology. *Biological Imaging* **2023**, pp. 1–22. doi:10.1017/S2633903X23000132.

32. Jimenez-Moreno, A.; Del Cano, L.; Martinez, M.; Ramirez-Aportela, E.; Cuervo, A.; Melero, R.; Sanchez-Garcia, R.; Strelak, D.; Fernandez-Gimenez, E.; de Isidro-Gomez, F.P.; Herreros, D.; Conesa, P.; Fonseca, Y.; Maluenda, D.; Jimenez de la Morena, J.; Macias, J.R.; Losana, P.; Marabini, R.; Carazo, J.M.; Sorzano, C.O.S. Cryo-EM and Single-Particle Analysis with Scipion. *J Vis Exp* **2021**, p. e62261.

33. Harastani, M.; Sorzano, C.O.S.; Jonic, S. Hybrid Electron Microscopy Normal Mode Analysis with Scipion. *Protein Science* **2020**, *29*, 223–236. doi:10.1002/pro.3772.

34. Herreros, D.; Lederman, R.; Krieger, J.; Jiménez-Moreno, A.; Martínez, M.; Myška, D.; Strelak, D.; Filipovic, J.; Sorzano, C.; Carazo, J. Estimating conformational landscapes from Cryo-EM particles by 3D Zernike polynomials. *Nat Commun.* **2023**, *14*, 154. doi:10.1038/s41467-023-35791-y.

35. Orozco, M. A theoretical view of protein dynamics. *Chem. Soc. Rev.* **2014**, *43*, 5051–5066. doi:10.1039/C3CS60474H.

36. Hollingsworth, S.; Dror, R. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99*, 1129–1143.

37. Collier, T.; Piggot, T.; Allison, J. Molecular Dynamics Simulation of Proteins. *Methods Mol Biol* **2020**, *2073*, 311–327. doi:10.1007/978-1-4939-9869-2_17.

38. Hénin, J.; Lelièvre, T.; Shirts, M.R.; Valsson, O.; Delemotte, L. Enhanced Sampling Methods for Molecular Dynamics Simulations [Article v1.0]. *Living Journal of Computational Molecular Science* **2022**, *4*, 1583. doi:10.33011/livecoms.4.1.1583.

39. Krieger, J.; Doruker, P.; Scott, A.; Perahia, D.; Bahar, I. Towards gaining sight of multiscale events: utilizing network models and normal modes in hybrid methods. *Curr Opin Struct Biol* **2020**, *64*, 34–41. doi:10.1016/j.sbi.2020.05.013.

40. Atilgan, C. Chapter Two - Computational Methods for Efficient Sampling of Protein Landscapes and Disclosing Allosteric Regions. In *Computational Molecular Modelling in Structural Biology*; Karabencheva-Christova, T.G.; Christov, C.Z., Eds.; Academic Press, 2018; Volume 113, Advances in Protein Chemistry and Structural Biology; pp. 33–63. doi:https://doi.org/10.1016/bs.apcsb.2018.06.001.

41. Dill, K.; Jernigan, R.L.; Bahar, I. *Protein Actions: Principles and Modeling*; Garland Science: New York, 2017. doi:https://doi.org/10.1201/9781315212210.

42. Bahar, I.; Lezon, T.R.; Bakan, A.; Shrivastava, I.H. Normal Mode Analysis of Biomolecular Structures: Functional Mechanisms of Membrane Proteins. *Chem Rev* **2010**, *110*, 1463–1497.

43. Bakan, A.; Meireles, L.M.; Bahar, I. ProDy: protein dynamics inferred from theory and experiments. *Bioinformatics* **2011**, *27*, 1575–1577.

44. Zhang, S.; Krieger, J.M.; Zhang, Y.; Kaya, C.; Kaynak, B.; Mikulska-Ruminska, K.; Doruker, P.; Li, H.; Bahar, I. ProDy 2.0: Increased Scale and Scope after 10 Years of Protein Dynamics Modelling with Python. *Bioinformatics* **2021**, *37*, 3657–3659.

45. Grant, B.J.; Skjaerven, L.; Yao, X.Q. The Bio3D packages for structural bioinformatics. *Protein Sci* **2021**, *30*, 20–30.

46. Suhre, K.; Sanejouand, Y.H. ElNemo: a normal mode web server for protein movement analysis and the generation of templates for molecular replacement. *Nucleic Acids Research* **2004**, *32*, W610–W614.

47. Tiwari, S.P.; Fuglebakk, E.; Hollup, S.M.; Skjaerven, L.; Cragnolini, T.; Grindhaug, S.H.; Tekle, K.M.; Reuter, N. WEBnm@ v2.0: Web server and services for comparing protein flexibility. *BMC Bioinformatics* **2014**, *15*, 427.

48. Li, H.; Chang, Y.Y.; Lee, J.Y.; Bahar, I.; Yang, L.W. DynOmics: dynamics of structural proteome and beyond. *Nucleic Acids Res* **2017**, *45*, W374–W380.

49. Kaynak, B.T. Zhang, S.; Bahar, I.; Doruker, P. ClustENMD: efficient sampling of biomolecular conformational space at atomic resolution. *Bioinformatics* **2021**, *37*, 3956–3958. doi:10.1093/bioinformatics/btab496.

50. Bakan, A.; Dutta, A.; Mao, W.; Liu, Y.; Chennubhotla, C.; Lezon, T.R.; Bahar, I. Evol and ProDy for bridging protein sequence evolution and structural dynamics. *Bioinformatics* **2014**, *30*, 2681–2683.

51. Humphrey, W.; Dalke, A.; Schulten, K. VMD: visual molecular dynamics. *J Mol Graph* **1996**, *14*, 33–38. doi:10.1016/0263-7855(96)00018-5.

52. Gur, M. Madura, J.; Bahar, I. Global transitions of proteins explored by a multiscale hybrid methodology: application to adenylate kinase. *Biophys J.* **2013**, *105*, 1643–1652. doi:10.1016/j.bpj.2013.07.058.

53. Phillips, J.; Hardy, D.; Maia, J.; Stone, J.; Ribeiro, J.; Bernardi, R.; Buch, R.; Fiorin, G.; Hénin, J.; Jiang, W.; McGreevy, R.; Melo, M.; Radak, B.; Skeel, R.; Singharoy, A.; Wang, Y.; Roux, B.; Aksimentiev, A.; Luthey-Schulten, Z.; Kalé, L.; Schulten, K.; Chipot, C.; Tajkhorshid, E. Scalable molecular dynamics on CPU and GPU architectures with NAMD. *J Chem Phys* **2020**, *153*, 044130. doi:10.1063/5.0014475.

54. Jiménez-García, B.; Roel-Touris, J.; Romero-Durana, M.; Vidal, M.; Jiménez-González, D.; Fernández-Recio, J. LightDock: a new multi-scale approach to protein-protein docking. *Bioinformatics* **2018**, *34*, 49–55. doi:10.1093/bioinformatics/btx555.

55. Bahar, I.; Atilgan, A.R.; Erman, B. Direct evaluation of thermal fluctuations in proteins using a single-parameter harmonic potential. *Fold Des* **1997**, *2*, 173–181.

56. Atilgan, A.R.; Durell, S.R.; Jernigan, R.L.; Demirel, M.C.; Keskin, O.; Bahar, I. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophys J* **2001**, *80*, 505–515.

57. Lezon, T.R.; Bahar, I. Using entropy maximization to understand the determinants of structural dynamics beyond native contact topology. *PLoS Comput Biol* **2010**, *6*, e1000816. doi:10.1371/journal.pcbi.1000816.

58. Orellana, L.; Rueda, M.; Ferrer-Costa, C.; Lopez-Blanco, J.; Chacón, P.; Orozco, M. Approaching Elastic Network Models to Molecular Dynamics Flexibility. *J Chem Theory Comput* **2010**, *6*, 2910–2923. doi:10.1371/journal.pcbi.1000816.

59. Durand, P.; Trinquier, G.; Sanejouand, Y.H. A new approach for determining low-frequency normal modes in macromolecules. *Biopolymers* **1994**, *34*, 759–771.

60. Tama, F.; Gadea, F.X.; Marques, O.; Sanejouand, Y.H. Building-block approach for determining low-frequency normal modes of macromolecules. *Proteins* **2000**, *41*, 1–7.

61. Zhang, Y.; Krieger, J.; Mikulska-Ruminska, K.; Kaynak, B.; Sorzano, C.O.S.; Carazo, J.; Xing, J.; Bahar, I. State-dependent sequential allostery exhibited by chaperonin TRiC/CCT revealed by network analysis of Cryo-EM maps. *Progress in Biophysics and Molecular Biology* **2021**, *160*, 104–120.

62. Kurkcuoglu, Z.; Bahar, I.; Doruker, P. ClustENM: ENM-Based Sampling of Essential Conformational Space at Full Atomic Resolution. *J Chem Theory Comput.* **2016**, *12*, 4549–4562. doi:10.1021/acs.jctc.6b00319.

63. Ginex, T.; Marco-Marín, C.; Wieczór, M.; Mata, C.; Krieger, J.; Ruiz-Rodriguez, P.; López-Redondo, M.; Francés-Gómez, C.; Melero, R.; Sánchez-Sorzano, C.; Martínez, M.; Gougeard, N.; Forcada-Nadal, A.; Zamora-Caballero, S.; Gozalbo-Rovira, R.; Sanz-Frasquet, C.; Arranz, R.; Bravo, J.; Rubio, V.; Marina, A.; IBV-Covid19-Pipeline.; Geller, R.; Comas, I.; Gil, C.; Coscolla, M.; Orozco, M.; Llácer, J.; Carazo, J. The structural role of SARS-CoV-2 genetic background in the emergence and success of spike mutations: The case of the spike A222V mutation. *PLoS Pathogens* **2022**, *18*, e1010631. doi:10.1371/journal.ppat.1010631.

64. de la Rosa-Trevín, J.; Otón, J.; Marabini, R.; Zaldívar, A.; Vargas, J.; Carazo, J.; Sorzano, C. Xmipp 3.0: An improved software suite for image processing in electron microscopy. *Journal of Structural Biology* **2013**, *184*, 321–328.

65. Scheres, S. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. *Journal of Structural Biology* **2012**, *180*, 519–530.

66. Benton, D.; Wrobel, A.; Roustan, C.; Borg, A.; Xu, P.; Martin, S.; Rosenthal, P.; Skehel, J.; Gamblin, S. The effect of the D614G substitution on the structure of the spike glycoprotein of SARS-CoV-2. *Proc Natl Acad Sci U S A* **2021**, *118*, e2022586118. doi:10.1073/pnas.2022586118.

67. Berman, H.M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E. The Protein Data Bank. *Nucleic Acids Res* **2000**, *28*, 235–242.

68. Hunter, J.D. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering* **2007**, *9*, 90–95. doi:10.1109/MCSE.2007.55.

69. Eastman, P.; Swails, J.; Chodera, J.; McGibbon, R.; Zhao, Y.; Beauchamp, K.; Wang, L.; Simmonett, A.; Harrigan, M.; Stern, C.; Wiewiora, R.; Brooks, B.; Pande, V. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Comput Biol* **2017**, *13*, e1005659. doi:10.1371/journal.pcbi.1005659.

70. Zhang, S.; Li, H.; Krieger, J.M.; Bahar, I. Shared Signature Dynamics Tempered by Local Fluctuations Enables Fold Adaptability and Specificity. *Mol Biol Evol* **2019**, *36*, 2053–2068.

71. Wrapp, D.; Wang, N.; Corbett, K.; Goldsmith, J.; Hsieh, C.; Abiona, O.; Graham, B.; McLellan, J. Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* **2020**, *367*, 1260–1263. doi:10.1126/science.abb2507.

72. Walls, A.; Park, Y.; Tortorici, M.; Wall, A.; McGuire, A.; Veesler, D. Structure, Function, and Antigenicity of the SARS-CoV-2 Spike Glycoprotein. *Cell* **2020**, *181*, 281–292.e6. doi:10.1016/j.cell.2020.02.058.

73. Ke, Z.; Oton, J.; Qu, K.; Cortese, M.; Zila, V.; McKeane, L.; Nakane, T.; Zivanov, J.; Neufeldt, C.; Cerikan, B.; Lu, J.; Peukes, J.; Xiong, X.; Krausslich, H.; Scheres, S.; Bartenschlager, R.; Briggs, J. Structures and distributions of SARS-CoV-2 spike proteins on intact virions. *Nature* **2020**, *588*, 498–502. doi:10.1038/s41586-020-2665-2.

74. Yang, T.; Yu, P.; Chang, Y.; Hsu, S. D614G mutation in the SARS-CoV-2 spike protein enhances viral fitness by desensitizing it to temperature-dependent denaturation. *J Biol Chem* **2021**, *297*, 101238–101238. doi:10.1016/j.jbc.2021.101238.

75. Gobeil, S.; Janowska, K.; McDowell, S.; Mansouri, K.; Parks, R.; Manne, K.; Stalls, V.; Kopp, M.; Henderson, R.; Edwards, R.; Haynes, B.; Acharya, P. D614G Mutation Alters SARS-CoV-2 Spike Conformation and Enhances Protease Cleavage at the S1/S2 Junction. *Cell Rep* **2021**, *34*, 108630. doi:10.1016/j.celrep.2020.108630.

76. Krieger, J.; Sorzano, C.; Carazo, J.; Bahar, I. Protein dynamics developments for the large scale and cryoEM: case study of ProDy 2.0. *Acta Crystallogr D Struct Biol.* **2022**, *78*, 399–409. doi:10.1107/S2059798322001966.

77. Harris, C.; Millman, K.; van der Walt, S.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M.; Brett, M.; Haldane, A.; Fernández del Río, J.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. Array programming with NumPy. *Nature* **2020**, *585*, 357–362.

78. Krieger, J.; Greger, I.; Bahar, I. Structure, Dynamics, and Allosteric Potential of Ionotropic Glutamate Receptor N-Terminal Domains. *Biophys J* **2015**, *109*, 1136–1148. doi:10.1016/j.bpj.2015.06.061.

79. Kabsch, W. A solution for the best rotation to relate two sets of vectors. *Acta Cryst* **1976**, *A32*, 922–923. doi:10.1107/S0567739476001873.

80. Holm, L.; Laakso, L. Dali server update. *Nucleic Acids Res* **2016**, *44*, W351–W355. doi:10.1093/nar/gkw357.

81. Cock, P.; Antao, T.; Chang, J.T.; Chapman, B.; Cox, C.J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M.J. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422–1423.

82. Shindyalov, I.; Bourne, P. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein engineering* **1998**, *11*, 739–747. doi:10.1107/S0567739476001873.

83. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; van der Walt, S.J.; Brett, M.; Wilson, J.; Millman, K.J.; Mayorov, N.; Nelson, A.R.J.; Jones, E.; Kern, R.; Larson, E.; Carey, C.J.; Polat, İ.; Feng, Y.; Moore, E.W.; VanderPlas, J.; Laxalde, D.; Perktold, J.; Cimrman, R.; Henriksen, I.; Quintero, E.A.; Harris, C.R.; Archibald, A.M.; Ribeiro, A.H.; Pedregosa, F.; van Mulbregt, P.; SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272. doi:10.1038/s41592-019-0686-2.

84. Bahar, I.; Lezon, T.R.; Eyal, E. Global dynamics of proteins: bridging between structure and function. *Annu Rev Biophys* **2010**, *39*, 23–42.

85. Tirion, M.M. Low-amplitude elastic motions in proteins from a single-parameter atomic analysis. *Phys Rev Lett* **1996**, *77*, 1905–1908.

86. Eyal, E.; Yang, L.W.; Bahar, I. Anisotropic network model: systematic evaluation and a new web interface. *Bioinformatics* **2006**, *22*, 2619–2627.

87. Doruker, P.; Jernigan, R.L.; Bahar, I. Dynamics of large proteins through hierarchical levels of coarse-grained structures. *J Comput Chem* **2002**, *23*, 119–127.

88. Sauerwald, N.; Zhang, S.; Kingsford, C.; Bahar, I. Chromosomal dynamics predicted by an elastic network model explains genome-wide accessibility and long-range couplings. *Nucleic Acids Research* **2017**, *45*, 3663–3673.

89. Hinsen, K. Analysis of domain motions by approximate normal mode calculations. *Proteins* **1998**, *33*, 417–429.

90. Ming, D.; Wall, M.E. Allostery in a coarse-grained model of protein dynamics. *Phys Rev Lett* **2005**, *95*, 198103.

91. Woodcock, H.L.; Zheng, W.; Ghysels, A.; Shao, Y.; Kong, J.; Brooks, B.R. Vibrational subsystem analysis: A method for probing free energies and correlations in the harmonic limit. *J Chem Phys* **2008**, *129*, 214109.

92. Zhang, Y.; Zhang, S.; Xing, J.; Bahar, I. Normal mode analysis of membrane protein dynamics using the vibrational subsystem analysis. *J Chem Phys* **2021**, *154*, 195102. doi:10.1063/5.0046710.

93. Lezon, T.; Bahar, I. Constraints imposed by the membrane selectively guide the alternating access dynamics of the glutamate transporter GltPh. *Biophys J* **2021**, *102*, 1331–1340. doi:10.1016/j.bpj.2012.02.028.

94. Stember, J.; Wriggers, W. Bend-twist-stretch model for coarse elastic network simulation of biomolecular motion. *J Chem Phys* **2009**, *131*, 074112. doi:10.1063/1.3167410.

95. Fuglebakk, E.; Tiwari, S.; Reuter, N. Comparing the intrinsic dynamics of multiple protein structures using elastic network models. *Biochim Biophys Acta* **2015**, *1850*, 911–922. doi:10.1016/j.bbagen.2014.09.021.

96. Marques, O.; Sanejouand, Y. Hinge-bending motion in citrate synthase arising from normal mode calculations. *Proteins* **1995**, *23*, 557–560. doi:10.1002/prot.340230410.

97. Carnevale, V.; Pontiggia, F.; Micheletti, C. Structural and Dynamical Alignment of Enzymes with Partial Structural Similarity. *J. Phys. Condens. Matter* **2007**, *19*, 285206. doi:10.1088/0953-8984/19/28/285206.

98. Hess, B. Convergence of sampling in protein simulations. *Phys. Rev. E.* **2002**, *65*, 031910. doi:10.1103/PhysRevE.65.031910.