Scipion-EM-ProDy

Release

James Krieger, She Zhang

June 07, 2024

CHAPTER

ONE

INTRODUCTION

This tutorial shows how to use ProDy within Scipion including experimental ensemble analysis, NMA and ClustENMD simulations (see Figure 3 of *[KJ23]*).

1.1 Required Programs

Scipion[1] 3 is required. How to install it can be found on its website.

1.2 Installation

To install the latest release version of the scipion-em-prody plugin (3.3.0), you can use the following command:

```
scipion3 installp -p scipion-em-prody
```

This will also install the latest release version (2.4.1) of ProDy into environments that Scipion can use along with TEMPy and OpenMM.

The latest development version of the plugin can also be installed as follows:

```
git clone https://github.com/scipion-em/scipion-em-prody.git
scipion3 installp -p ./scipion-em-prody --devel
```

There is also the option to use the release or development version of ProDy with the plugin. This can be achieved by using commands like the following:

```
scipion3 installb prody-2.4.1
scipion3 installb prody-github
```

Alternatively, you can activate the conda environments with one of these names and also the main scipion3 environment and pip install your own ProDy there.

You can generate some example projects and test that everything is working by running the following:

```
scipion3 test prody2.tests
```

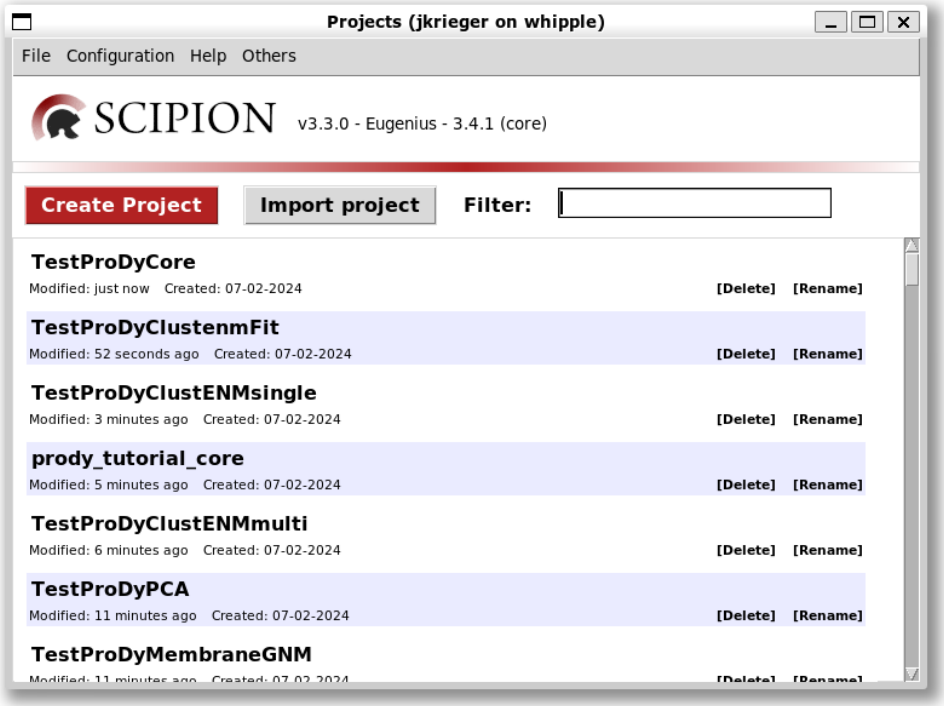
1.3 Getting Started in Scipion

You can start Scipion from the command line as follows:

```
scipion3
```

This will open the Projects window. From here, you can create a new project or open and existing one.

[1] https://scipion-em.github.io/docs/release-3.0.0/



Once you create or select a project, you will get a new window for your project as you can see for TestProDyClustENMsingle:

```
.. figure:: images/intro/2_scipion_TestProDyClustENMsingle.png
```

scale 80%

You can also open up the last project that was open or changed as follows:

```
scipion3 last
```

We will look at the interface more on the next pages but you can also find more information at https://scipion-em.github.io/docs/release-3.0.0/docs/user/scipion-gui.html.

CHAPTER

TWO

CORE CALCULATIONS: SELECT, ANM, DEFORMATION AND COMPARISON

This example shows how to run a simple workflow in Scipion using core ProDy calculations: atom selection, ANM NMA and deformation analysis, and analyse the results. We will use the open and closed states of adenylate kinase.

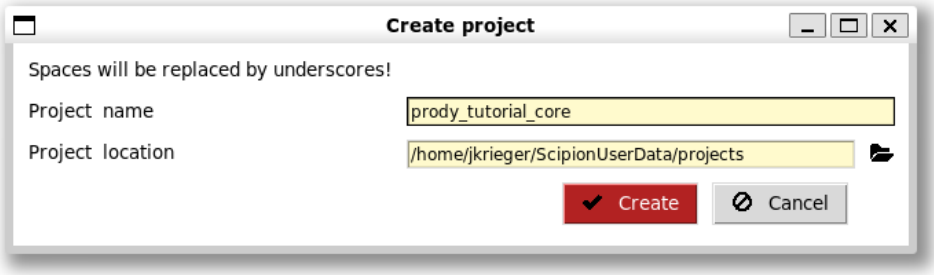
See *[KJ23]* for more information about the Scipion-EM-ProDy plugin and more examples.

2.1 Getting started

You can start Scipion from the command line as follows:

```
scipion3
```

This will open the Projects window. From here, you can create a new project or open and existing one. This time we will create a new project. You can call it something like prody_tutorial_core.



What you will see will be an empty project window with 3 sections:

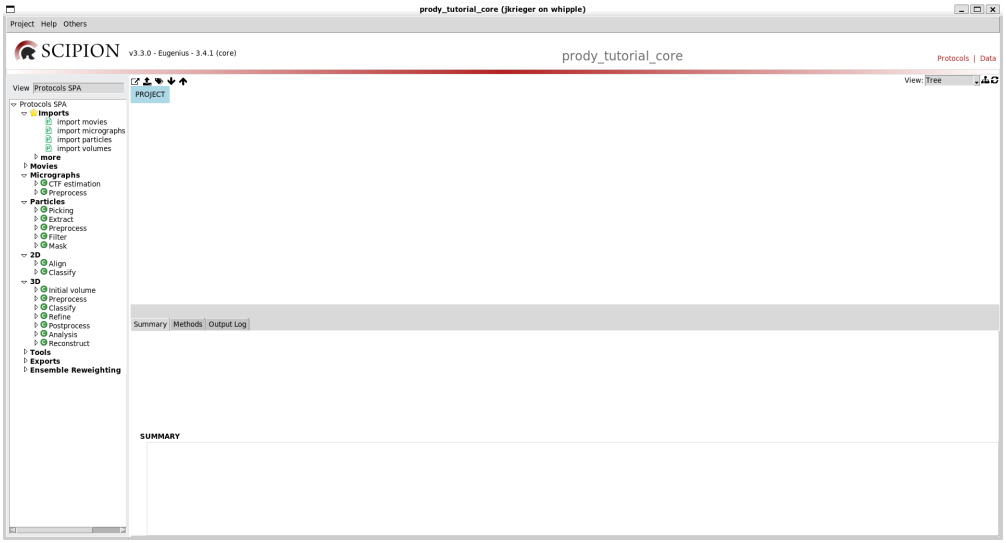
The left-most section is the protocols tree where you can see possible protocols to run for different types of tasks. The default one is Cryo-EM single particle analysis (SPA), which was the original purpose of Scipion.

If you select the dropdown that says Protocols SPA then you can select other types of tasks and there is one that says ProDy. Selecting this then gives you a protocols tree specific to ProDy like the following:

2.2 First workflow from 2 protocols: importing and selecting

As with any ProDy pipeline, we start by parsing an atomic structure and selecting some atoms of interest.

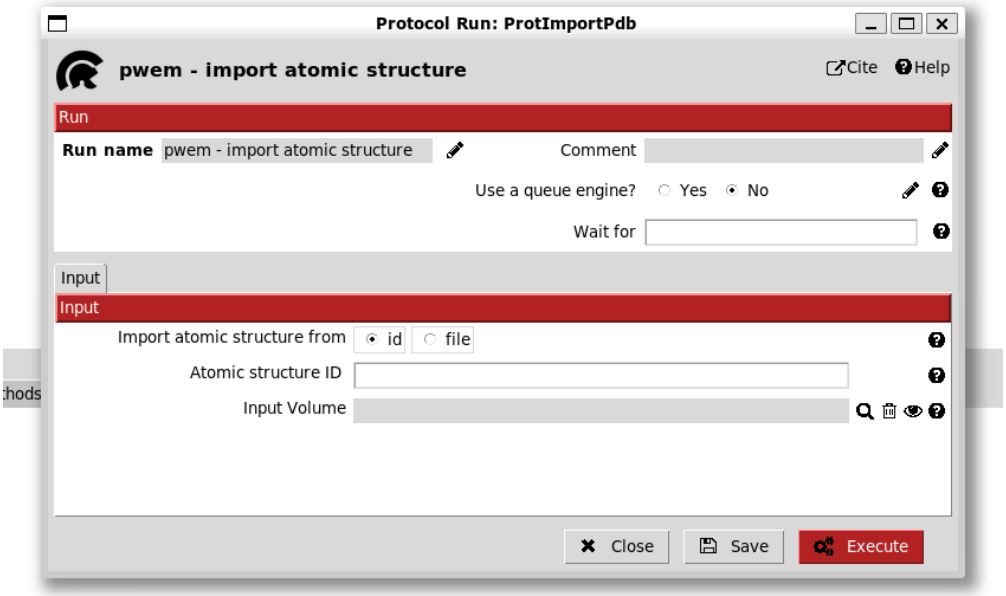
At the top of the tree, we have some optional protocols from the Scipion core for importing atomic structures (step 0). The first of these can import a single atomic structure from the PDB or a file and we will use it for illustration.


prody_tutorial_core (jkrieger on whipple)
Project Help Others
SCIPION v3.3.0 - Eugenius - 3.4.1 (core)
prody_tutorial_core
Protocols | Data
View Protocols SPA
Protocols SPA
Imports
import movies
import micrographs
import particles
import volumes
more
Movies
Micrographs
CTF estimation
Preprocess
Particles
Picking
Extract
Preprocess
Filter
Mask
2D
Align
Classify
3D
Initial volume
Preprocess
Classify
Refine
Postprocess
Analysis
Reconstruct
Tools
Exports
Ensemble Reweighting
PROJECT
View: Tree
Summary Methods Output Log
SUMMARY


prody_tutorial_core (jkrieger on whipple)
Project Help Others
SCIPION v3.3.0 - Eugenius - 3.4.1 (core)
prody_tutorial_core
Protocols | Data
View ProDy
ProDy
0. Import model(s)
Import Structure
Import Set of Atomic Structures
1. Atomic structure operations
Extract biomol assemblies
Fix missing atoms
Add/join atomic structures
Select
Align
2. Dynamics calculations
2a. Normal mode analysis (NMA)
ANM NMA
RTB NMA
2b. Ensemble analysis (PCA or LDA)
i. Build PDB Ensemble
ii. Atomic PCA
iii. Atomic LDA
2c. Deformation vector analysis
Calculate Deformation Vector
2d. Gaussian Network Model (GNM) analysis
GNM Mode Analysis
Domain Decomposition
2e. Hybrid simulations
ClustENM(3D) hybrid simulation
3. Dynamics analysis
Edit Modes
Compare Modes
Project ensemble onto modes
RMSD reordering and clustering
Measure from ensemble
Imports
Import Ensemble
Import Modes
PROJECT
View: Tree
Summary Methods Output Log
SUMMARY

The second can do the same for multiple atomic structures as we will see in the Scipion ensemble analysis tutorial[2].

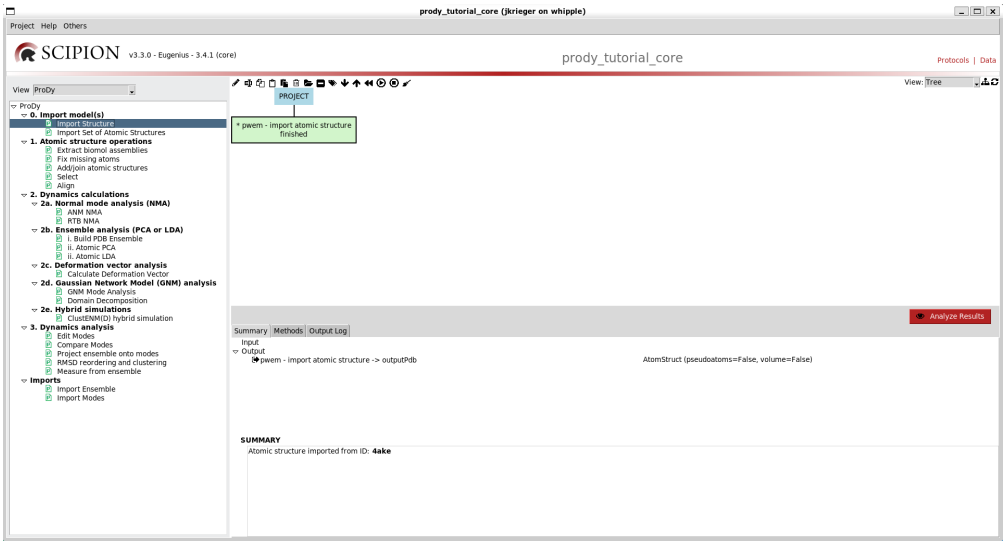
Double clicking this entry in the tree brings up a form for this protocol:



We see that there are options to import from the PDB using an id or from a file. In this case, we enter the id 4ake (the open state adenylate kinase).

Scipion can also associate the structure to a volume but we do not do this now.

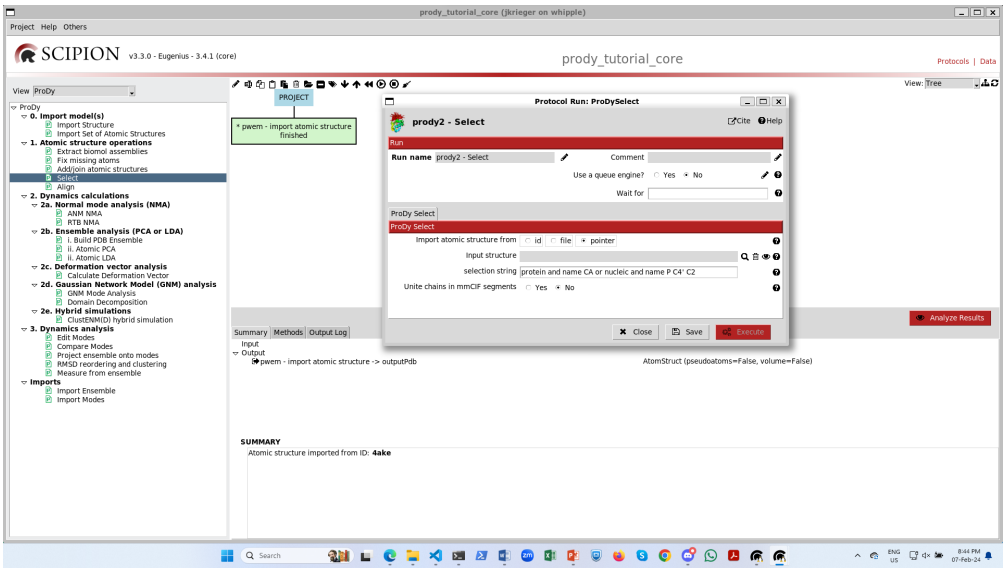
Clicking Execute brings up a box in the main workflow panel, which is initially blue then turns orange while running and green when done. When it is done, we also get an output and summary below:



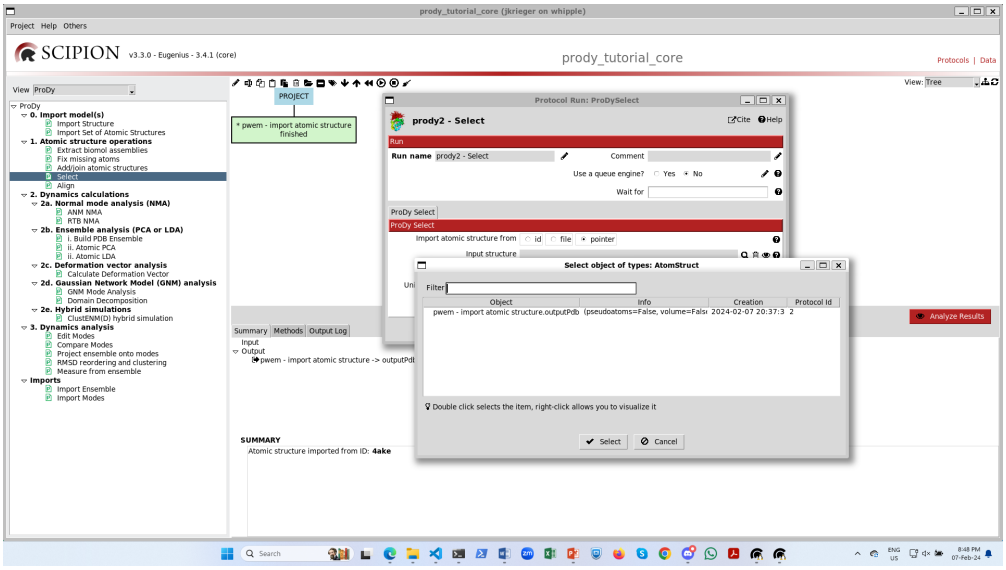
The output is an AtomStruct object, registered in the Scipion sqlite database that stores the path to the associated PDB or mmCIF file.

[2] http://prody.csb.pitt.edu/tutorials/scipion_tutorial/ensembles.html

We can then use this AtomStruct in a following protocol as we show for ProDy atom selection (from step 1). When we open up this protocol, we see that the default option is to import a structure into ProDy using a pointer and, below that, the form has a grey box with a magnifying glass:



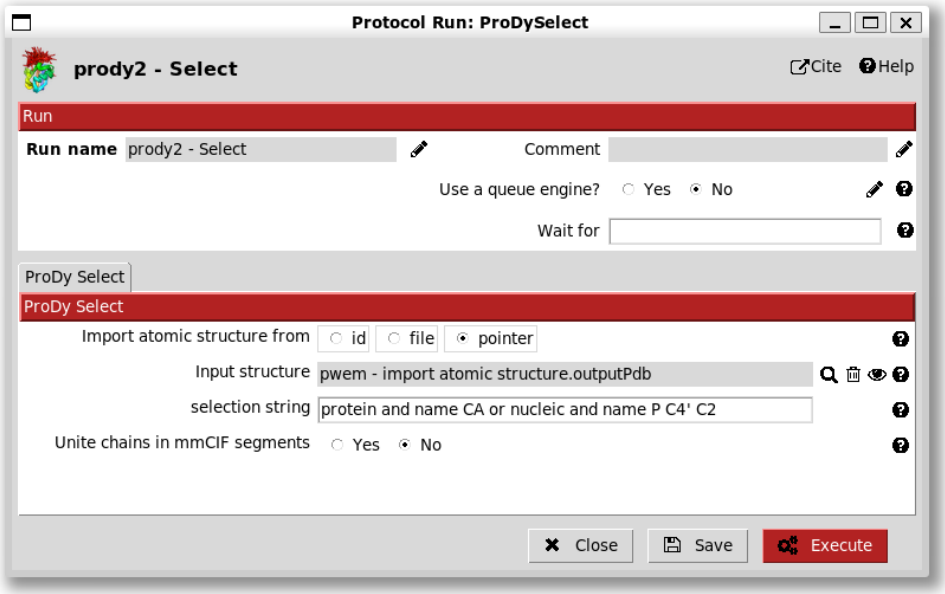
If you click on the magnifying glass, you get a new window with the list of AtomStruct objects to point to:

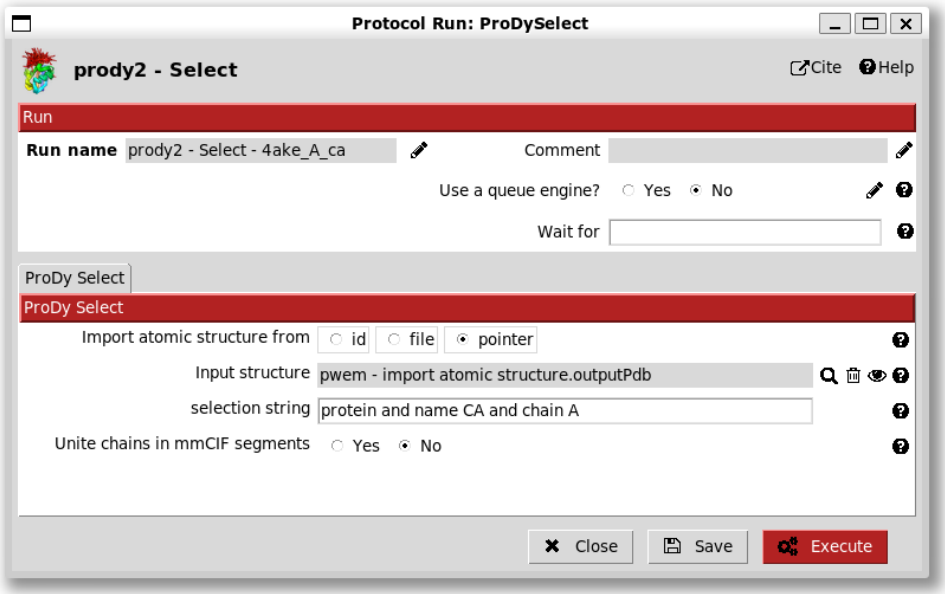


Upon selecting it, we then have it on our form and the red Execute button becomes available:

We can then change the selection string to include only chain A (and remove the nucleic part) and change the run name to change the box label to be more informative.

We now get another box in the workflow connected to the first with another output AtomStruct for a new file from running the prody select app, and information about the atoms selected.


Protocol Run: ProDySelect
prody2 - Select
Cite
Help
Run
Run name
prody2 - Select
Comment
Use a queue engine?
Yes
No
Wait for
ProDy Select
ProDy Select
Import atomic structure from
id
file
pointer
Input structure
pwem - import atomic structure.outputPdb
selection string
protein and name CA or nucleic and name P C4' C2
Unite chains in mmCIF segments
Yes
No
Close
Save
Execute

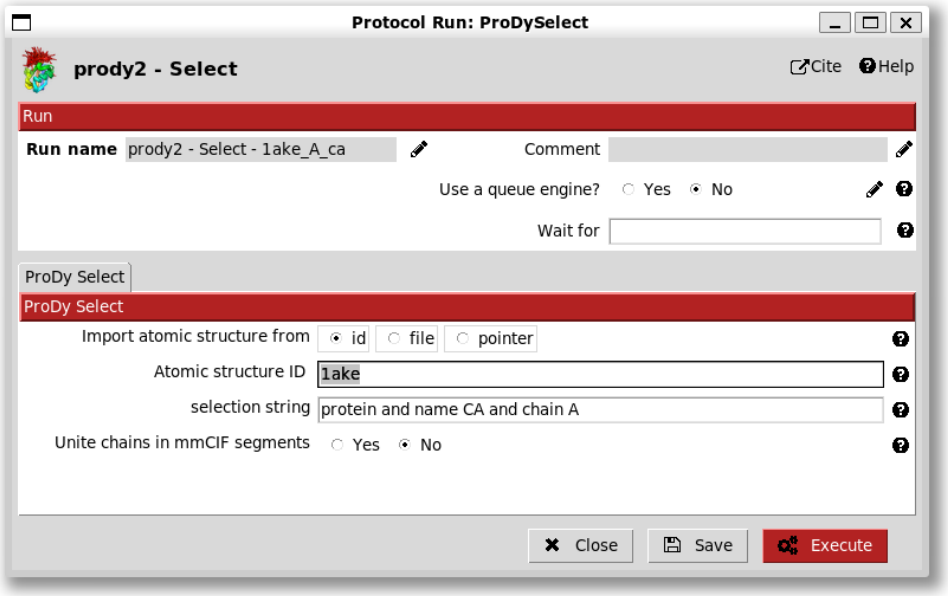

Protocol Run: ProDySelect
prody2 - Select
Cite
Help
Run
Run name
prody2 - Select - 4ake_A_ca
Comment
Use a queue engine?
Yes
No
Wait for
ProDy Select
ProDy Select
Import atomic structure from
id
file
pointer
Input structure
pwem - import atomic structure.outputPdb
selection string
protein and name CA and chain A
Unite chains in mmCIF segments
Yes
No
Close
Save
Execute

2.3 Another way of parsing and selecting

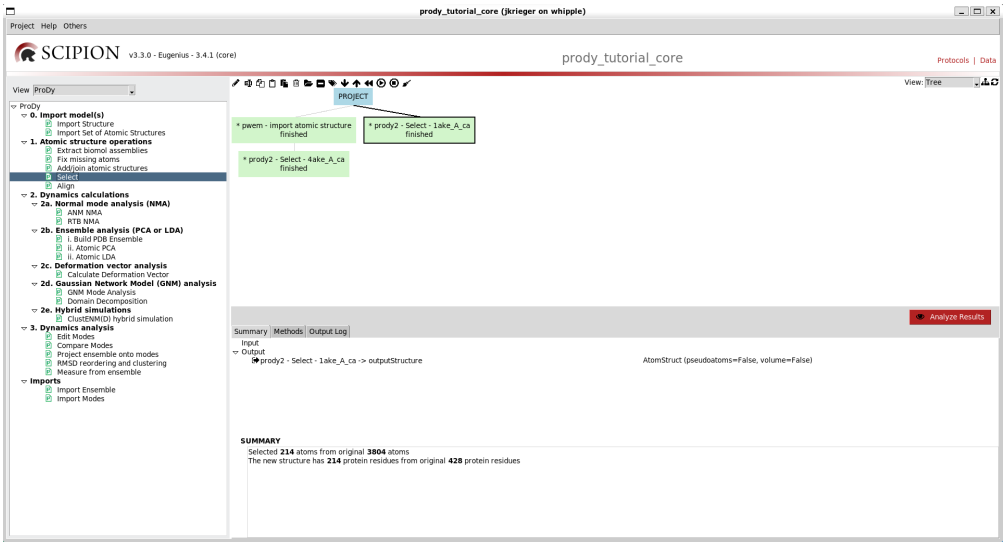
Rather than using the Scipion core protocol for importing atomic structures, we can also provide an id or file straight to any of the ProDy protocols related to atomic operations (excluding the alignment one, which takes two AtomStruct pointers, see below).

We will do this now for PDB id 1ake (the closed adenylate kinase). To do this, we can either select the protocol from the list again, or use ctrl + f to find it, or right-click it and click Duplicate, which has a shortcut ctrl + d.

We again select CA atoms from chain A and change the run name accordingly:



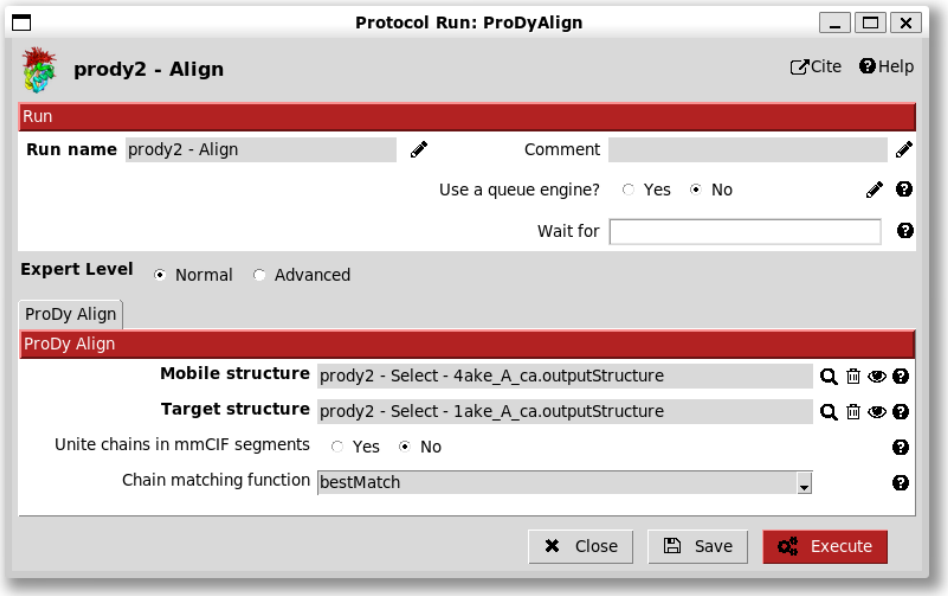
Executing this now connects it to the top of the tree again:



2.4 Step 2: dynamics calculations - NMA and deformation vector analysis

We are now ready to perform standard ANM and deformation vector analyses to see how the intrinsic dynamics of the open system drives transitions to the closed state as in the deformation analysis tutorial[3].

The first step here is to align the two structures. We do this by providing the two pointers to the align protocol:



We see here that there are options for chain matching, which we will look at later in the ensemble analysis tutorial[4] as this is a trivial case.

There's also an option of an advanced expert level. Selecting this brings up more options, which are again not useful here:

Now, we can perform ANM NMA and deformation analysis as usual. The ANM NMA protocol (from protocol tree option 2a) also has various advanced options (below) but, as in the ProDy API and apps, we are usually fine to leave these alone.

It can be noted that explicit membrane ANM is also an option here if we provide a structure aligned to membrane like that from opm/ppm.

There is also another tab that controls options for animations using `traverseModel()`:

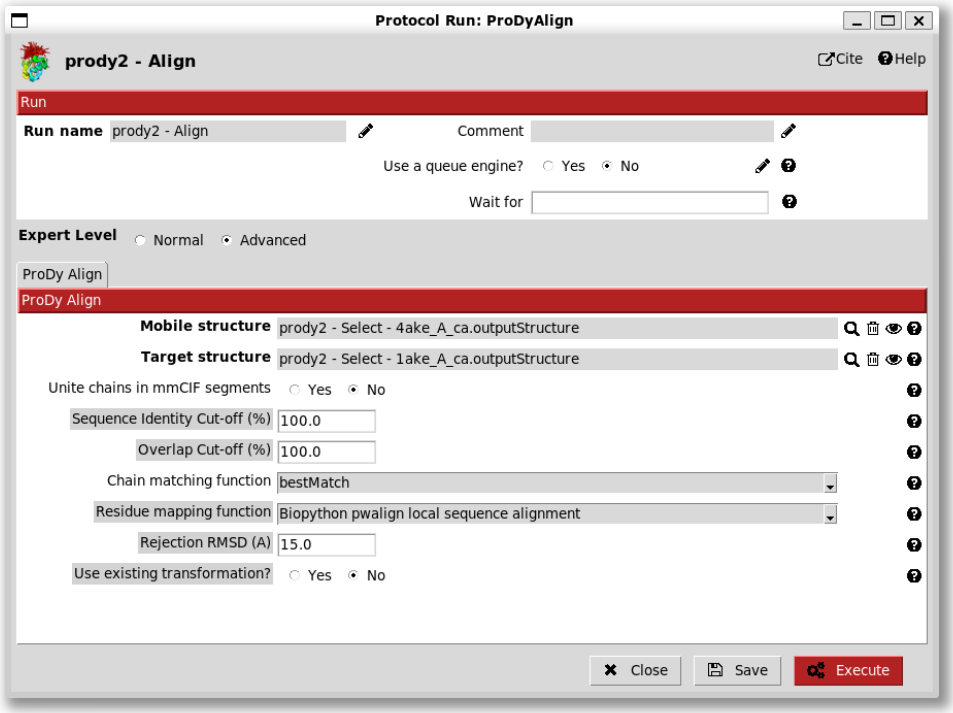
Similar options to the previous two protocol forms are available on that for deformation vector analysis (option 2c), so go ahead and run that too. We now have a tree like the following:

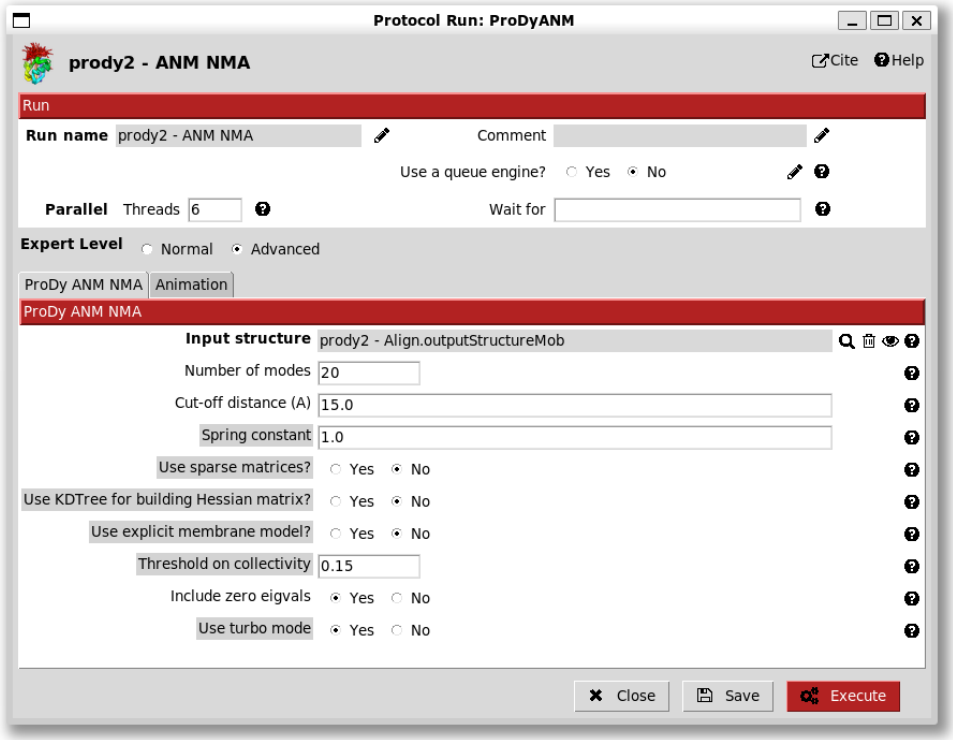
We can make it nicer by dragging the boxes around or clicking the tree icon at the top right:

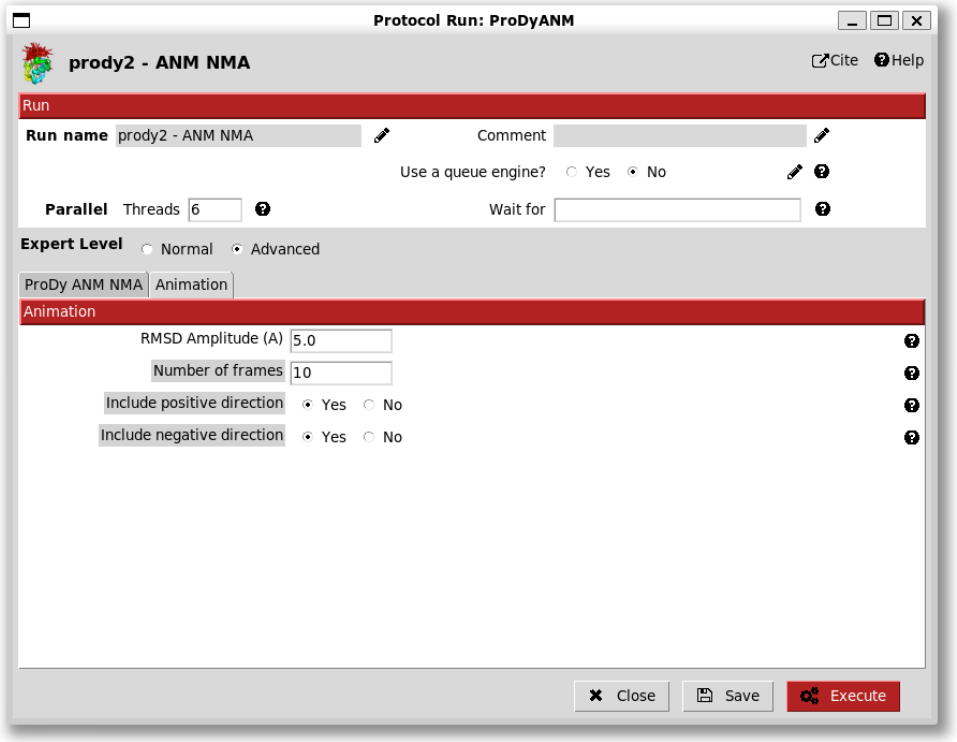
We can also colour our boxes to make the workflow easier to follow using labels, which are in the middle of the set of icons at the top left of the workflow panel (the other icons are for run control, e.g. stopping and starting and copying protocols):

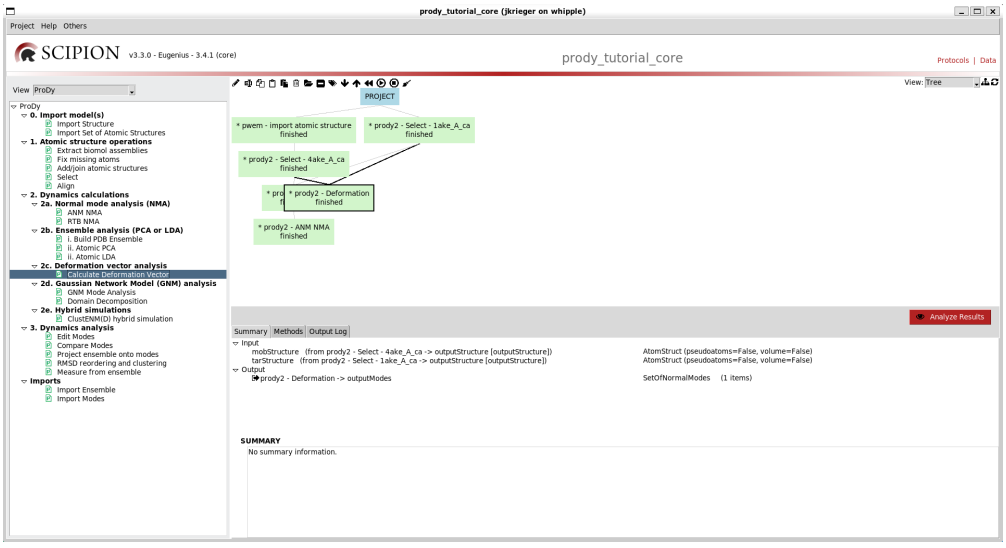
We can add now labels with different colours for different steps. There is a default set of colours for each label in order that usually works quite well. We can give the first (red) one a label "import", for example:

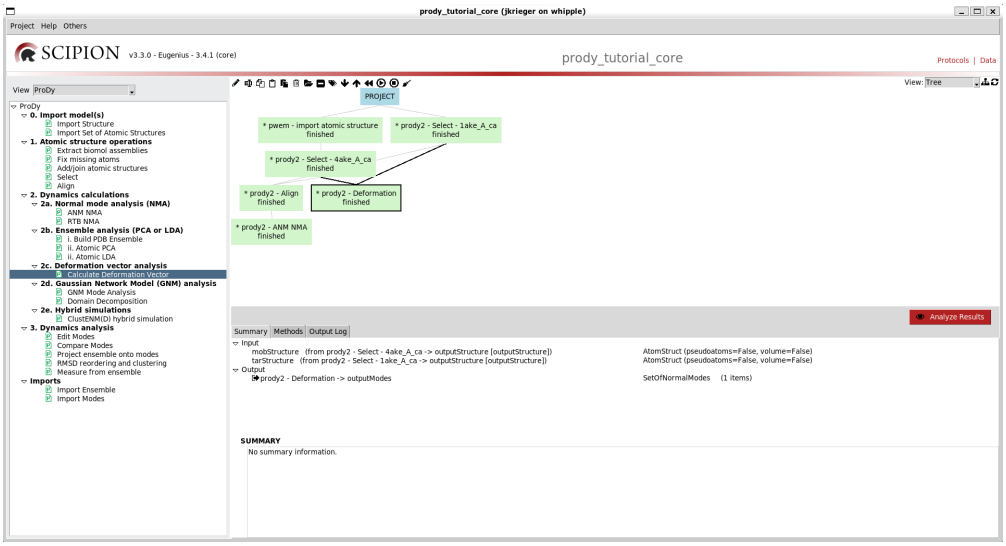
[3] https://prody.csb.pitt.edu/tutorials/enm_analysis/deformation.html
[4] http://prody.csb.pitt.edu/tutorials/scipion_tutorial/ensembles.html

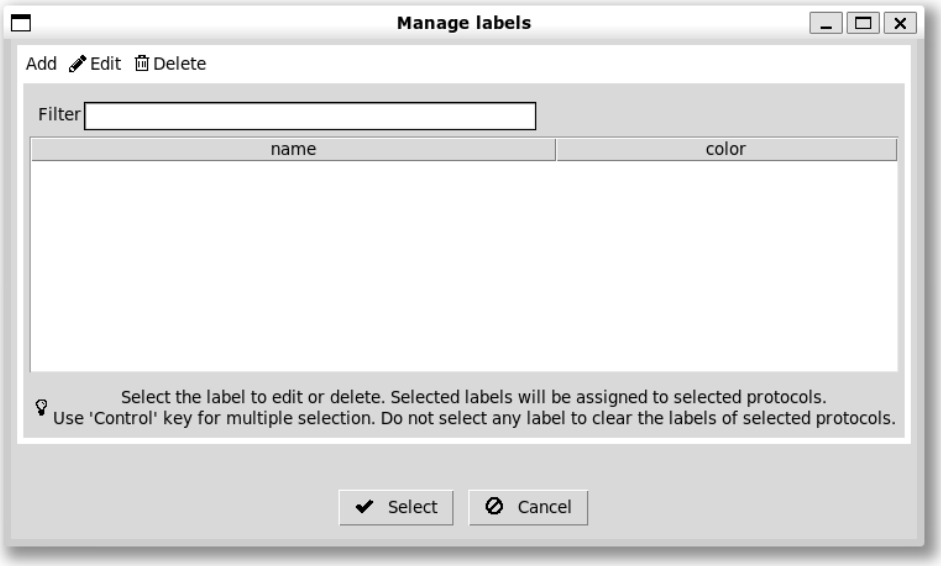

Protocol Run: ProDyAlign
prody2 - Align
Cite
Help
Run
Run name prody2 - Align
Comment
Use a queue engine? Yes No
Wait for
Expert Level Normal Advanced
ProDy Align
ProDy Align
Mobile structure prody2 - Select - 4ake_A_ca.outputStructure
Target structure prody2 - Select - 1ake_A_ca.outputStructure
Unite chains in mmCIF segments Yes No
Sequence Identity Cut-off (%) 100.0
Overlap Cut-off (%) 100.0
Chain matching function bestMatch
Residue mapping function Biopython pwalign local sequence alignment
Rejection RMSD (A) 15.0
Use existing transformation? Yes No
Close
Save
Execute

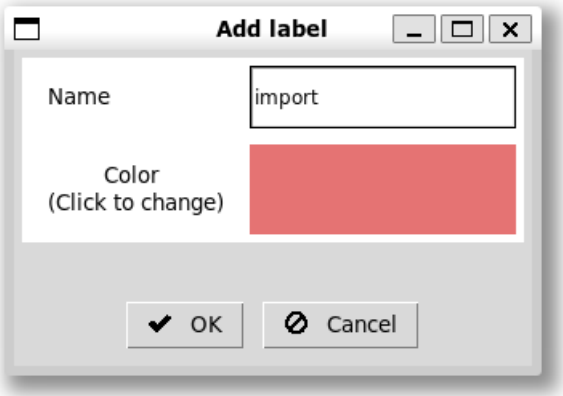

Protocol Run: ProDyANM
prody2 - ANM NMA
Cite
Help
Run
Run name prody2 - ANM NMA
Comment
Use a queue engine? Yes No
Parallel Threads 6
Wait for
Expert Level Normal Advanced
ProDy ANM NMA
Animation
ProDy ANM NMA
Input structure prody2 - Align.outputStructureMob
Number of modes 20
Cut-off distance (A) 15.0
Spring constant 1.0
Use sparse matrices? Yes No
Use KDTree for building Hessian matrix? Yes No
Use explicit membrane model? Yes No
Threshold on collectivity 0.15
Include zero eigvals Yes No
Use turbo mode Yes No
Close
Save
Execute


Protocol Run: ProDyANM
prody2 - ANM NMA
Cite
Help
Run
Run name
prody2 - ANM NMA
Comment
Use a queue engine?
Yes
No
Parallel
Threads
6
Wait for
Expert Level
Normal
Advanced
ProDy ANM NMA
Animation
Animation
RMSD Amplitude (A)
5.0
Number of frames
10
Include positive direction
Yes
No
Include negative direction
Yes
No
Close
Save
Execute

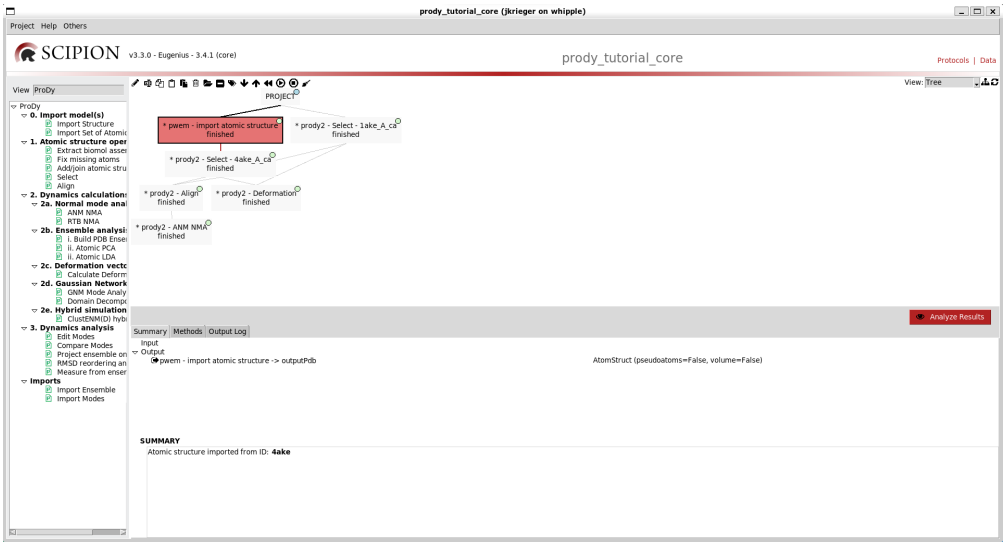

prody_tutorial_core (jkrieger on whipple)
Project
Help
Others
SCIPION
v3.3.0 - Eugenius - 3.4.1 (core)
prody_tutorial_core
Protocols | Data
View
ProDy
View: Tree
PROJECT
* pwem - import atomic structure finished
* prody2 - Select - 1ake_A_ca finished
* prody2 - Select - 4ake_A_ca finished
* prody2 - Deformation finished
* prody2 - ANM NMA finished
Analyze Results
Summary
Methods
Output Log
Input
mobStructure (from prody2 - Select - 4ake_A_ca -> outputStructure [outputStructure])
tarStructure (from prody2 - Select - 1ake_A_ca -> outputStructure [outputStructure])
Output
prody2 - Deformation -> outputModes
AtomStruct (pseudoatoms=False, volume=False)
AtomStruct (pseudoatoms=False, volume=False)
SetOfNormalModes (1 items)
SUMMARY
No summary information.


prody_tutorial_core (jkrieger on whipple)
Project Help Others
SCIPION v3.3.0 - Eugenius - 3.4.1 (core)
prody_tutorial_core
Protocols | Data
View ProDy
ProDy
0. Import model(s)
Import Structure
Import Set of Atomic Structures
1. Atomic structure operations
Extract biomol assemblies
Fix missing atoms
Add/join atomic structures
Select
Align
2. Dynamics calculations
2a. Normal mode analysis (NMA)
ANM NMA
RTB NMA
2b. Ensemble analysis (PCA or LDA)
i. Build PDB Ensemble
ii. Atomic PCA
ii. Atomic LDA
2c. Deformation vector analysis
i. Calculate Deformation Vector
2d. Gaussian Network Model (GNM) analysis
GNM Mode Analysis
Domain Decomposition
2e. Hybrid simulations
ClustENM(D) hybrid simulation
3. Dynamics analysis
Edit Modes
Compare Modes
Project ensemble onto modes
RMSD reordering and clustering
Measure from ensemble
Imports
Import Ensemble
Import Modes
View: Tree
PROJECT
* pwem - import atomic structure finished
* prody2 - Select - 1ake_A_ca finished
* prody2 - Select - 4ake_A_ca finished
* prody2 - Align finished
* prody2 - Deformation finished
* prody2 - ANM NMA finished
Analyze Results
Summary Methods Output Log
Input
mobStructure (from prody2 - Select - 4ake_A_ca -> outputStructure [outputStructure])
tarStructure (from prody2 - Select - 1ake_A_ca -> outputStructure [outputStructure])
AtomStruct (pseudoatoms=False, volume=False)
AtomStruct (pseudoatoms=False, volume=False)
Output
prody2 - Deformation -> outputModes
SetOfNormalModes (1 items)
SUMMARY
No summary information.


Manage labels
Add
Edit
Delete
Filter
name
color
Select the label to edit or delete. Selected labels will be assigned to selected protocols.
Use 'Control' key for multiple selection. Do not select any label to clear the labels of selected protocols.
Select
Cancel

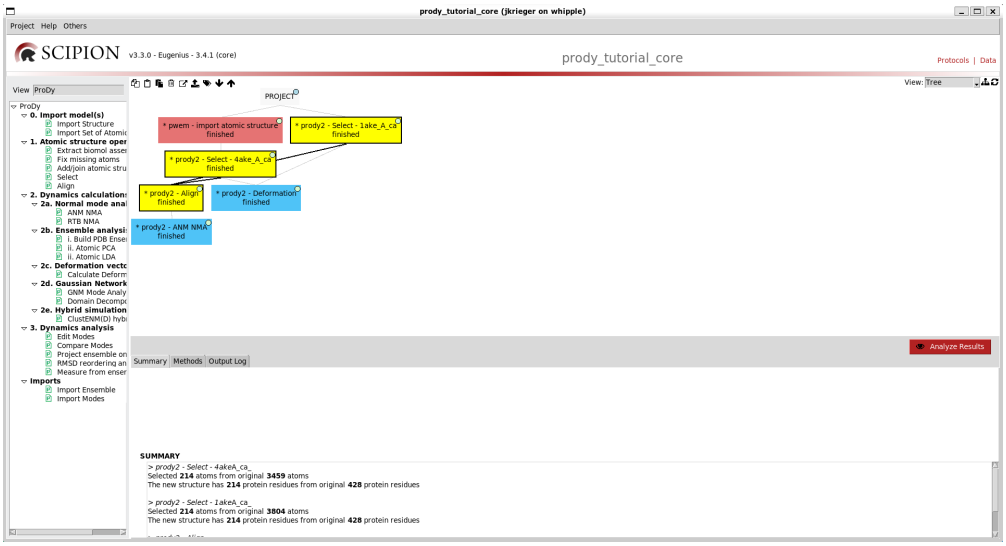

Add label
Name
import
Color
(Click to change)
OK
Cancel

We can then click our first box and select this label for it. At this stage, nothing happens except that the box gets a shape on it. However, if we go to the menus at the top and toggle colour mode, then we get this box coloured red and the rest white.



As in the Scipion-EM-ProDy paper, we can then make a blue label for dynamics and a yellow one (manually selecting to change the colour) for atomic structure operations.

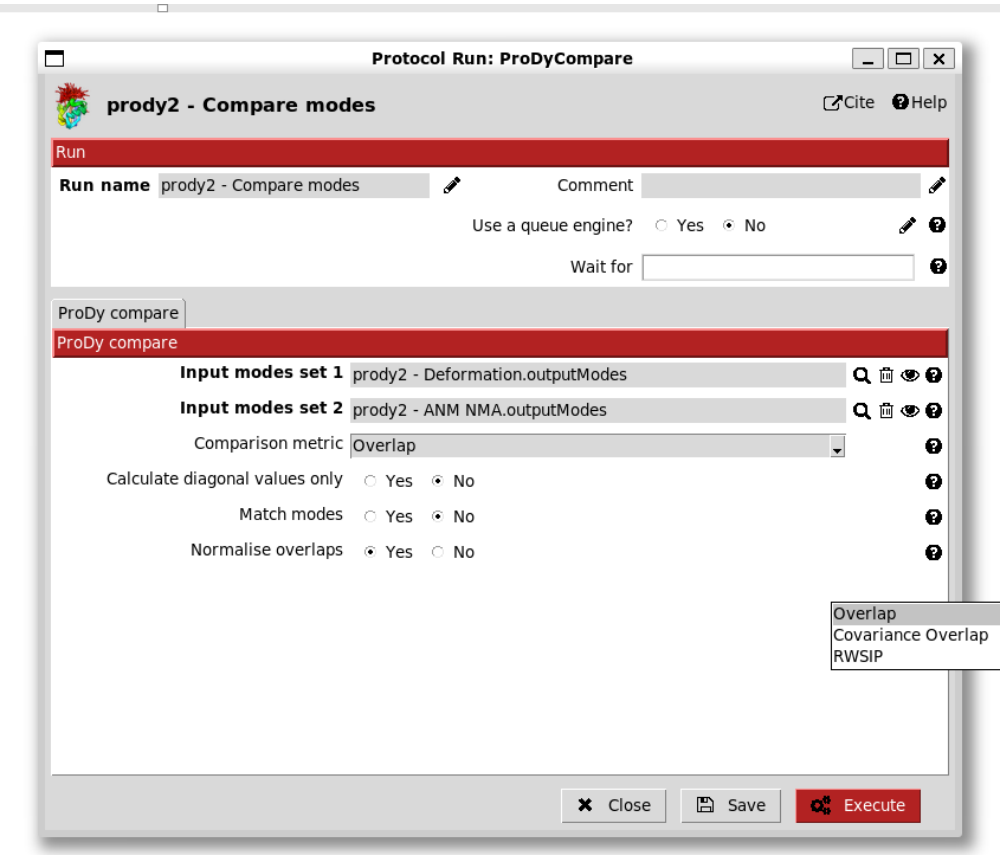
We can then select multiple boxes with ctrl + click and select labels for them to paint our workflow:

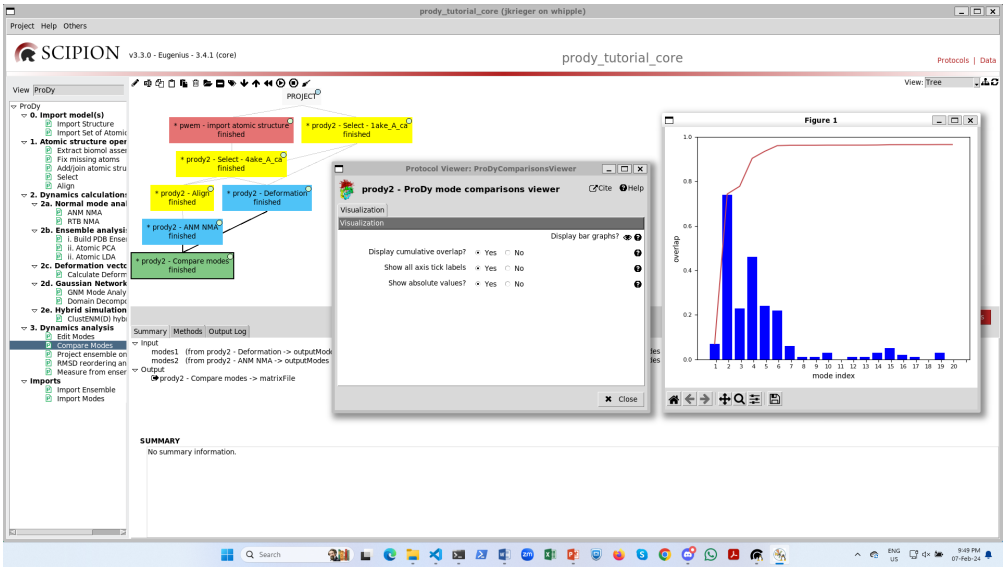


2.5 Step 3: analysis

Lastly, we can create a comparison protocol and paint it with a green label. This form has several options for comparisons as in the ProDy API:

Finally, this completes the workflow and we can analyse our results with the big red button, which opens up a viewer. If we select "yes" to cumulative overlaps and click the eye, then we get a graph as in the ProDy API:


Protocol Run: ProDyCompare
prody2 - Compare modes
Cite
Help
Run
Run name
prody2 - Compare modes
Comment
Use a queue engine?
Yes
No
Wait for
ProDy compare
ProDy compare
Input modes set 1
prody2 - Deformation.outputModes
Input modes set 2
prody2 - ANM NMA.outputModes
Comparison metric
Overlap
Calculate diagonal values only
Yes
No
Match modes
Yes
No
Normalise overlaps
Yes
No
Overlap
Covariance Overlap
RWSIP
Close
Save
Execute


prody_tutorial_core (jkrieger on whipple)
SCIPION
prody_tutorial_core
Protocol Viewer: ProDyComparisonsViewer
prody2 - ProDy mode comparisons viewer
Figure 1
SUMMARY
No summary information.

We can also click either of the (blue) two dynamics protocols and click the red button there to open up the NMWiz viewer in VMD (see the NMWiz tutorial[5]). Note that this will lock up your Scipion window.

We can also right click any input or output object and get options for other viewers.

[5] https://prody.csb.pitt.edu/tutorials/nmwiz_tutorial/

CHAPTER

THREE

ENSEMBLE ANALYSIS: DYNAMICS FROM PCA OF MULTIPLE STRUCTURES.

This example shows how to run an ensemble analysis workflow in Scipion by importing several structures and building an ensemble, and analysing it via PCA. We will use the SARS-CoV-2 spike as in *[KJ23]*.

See *[KJ23]* for more information about the Scipion-EM-ProDy plugin and more examples.

3.1 Getting started

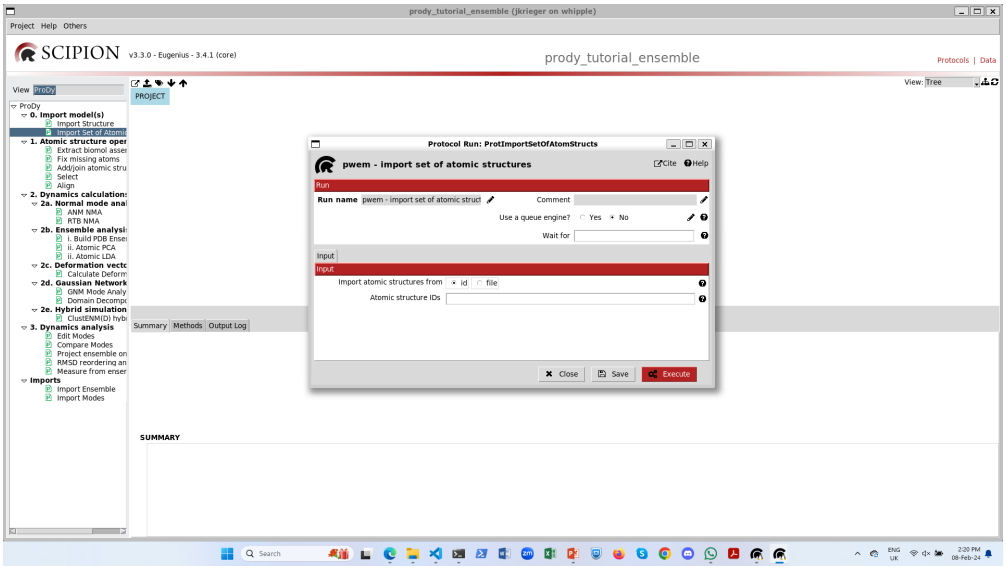
You can open the Scipion Projects window from the command line as follows:

```
scipion3
```

From here, create a new project and call it prody_tutorial_ensemble. Then, select the ProDy protocols tree from the dropdown on the left.

3.2 Simple case, steps 0 and 2-i: importing related structures and building an ensemble

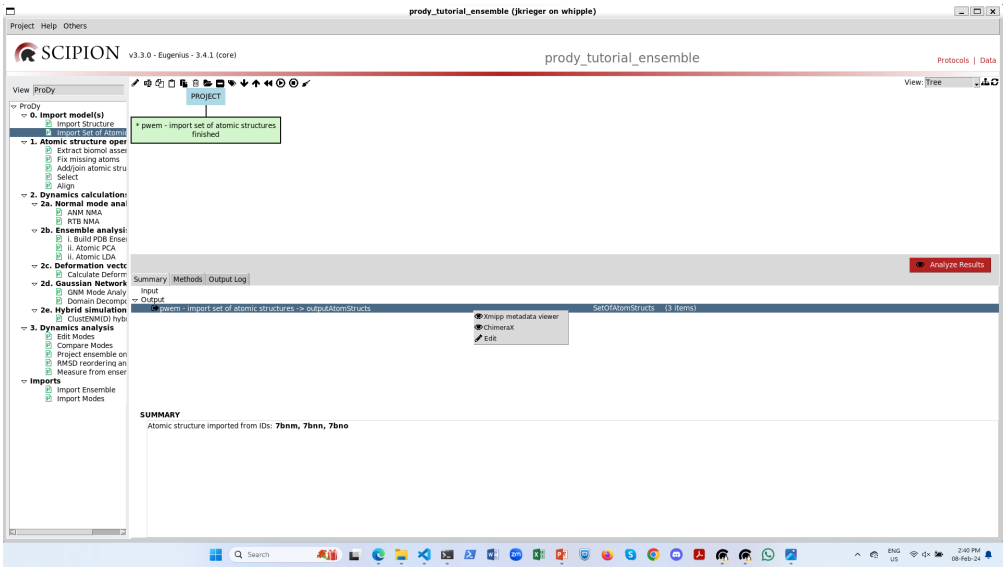
As with any ProDy pipeline, we start by parsing atomic structures. In this case, we use the second protocol from the Scipion core to importing multiple atomic structures at once.



For the simple case, we will just parse the structures of the D614G spike without ACE2 from Benton and Wrobel et al. 2021 (*[BW21]*). Coming from the same paper, these have the same chain IDs and can be easily aligned into an ensemble as in Figure 2 of *[KJ23]*.

We just provide the 3 PDB IDs separated by commas: *7bnm, 7bnn, 7bno*.

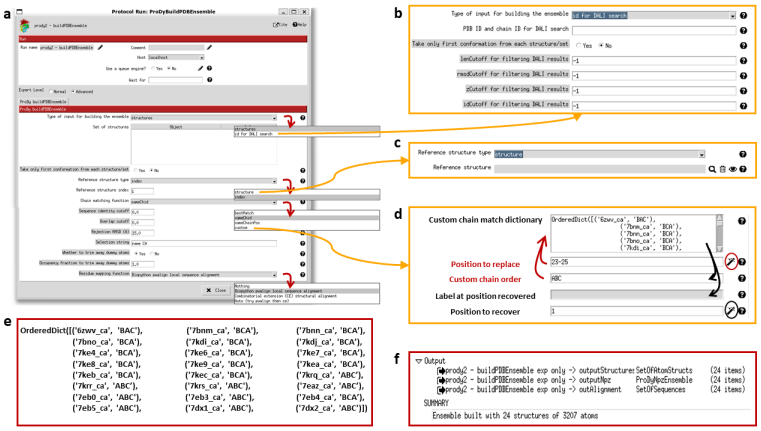
The output is a SetOfAtomStructs with 3 items and if we right-click it, then we get options of viewers:



The Xmipp metadata viewer is very useful for seeing the contents of sets and any associated metadata. At the moment, we see that each item points to an mmCIF file containing the corresponding structures that we can pass on to ProDy:

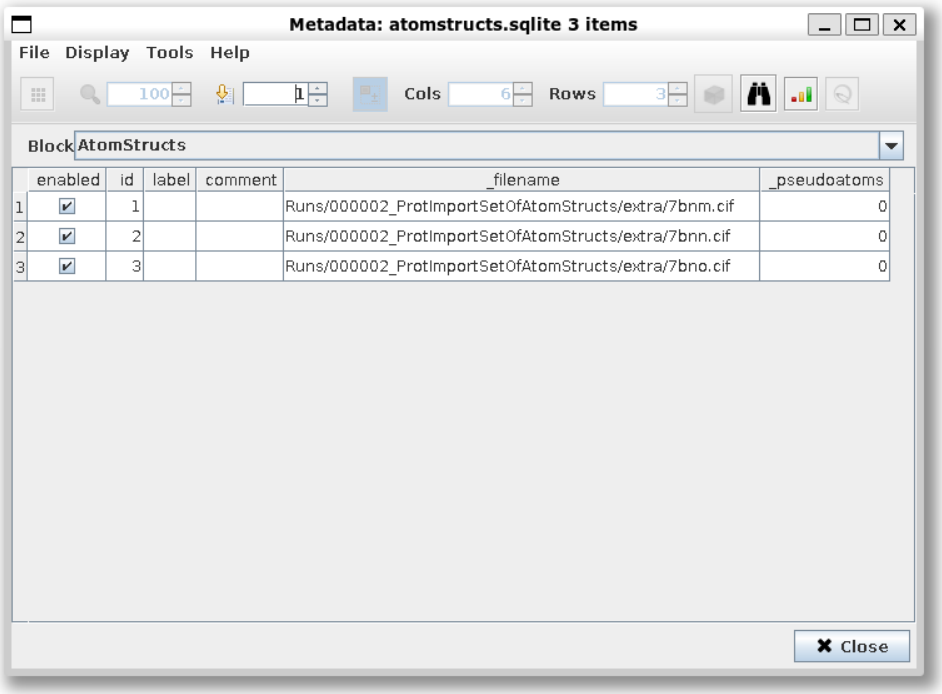
In this case, we skip step 1 as atom selection and alignment happens when we build the ensemble in the first protocol of step 2b. We provide the SetOfAtomStructs as a pointer in the list of input structures and keep the default options. Alternatively, we could also provide multiple AtomStruct or SetOfAtomStructs pointers, e.g. structures after selecting particular regions or sets from experiments and simulations.

In this case, the default chain matching function `sameChid()` works well as they all have the same chain nomenclature, but there are many other options as shown in Figure 4 of *[KJ23]* below.



We will explore the custom chain matching in the next section.

By default, the protocol will trim away dummy atoms where there is no occupancy for some structures. We change this in the advanced parameters, which then allows us to compare the PCs later. Likewise, we change unite_chains to yes.



We will also change the reference to index 2, which corresponds to the 1-up structure to aid with alignment, especially later.

The output is a ProDyNpzEnsemble object, which is also a type of set that can be opened with the metadata viewer. Alternatively, this object and the ensemble building protocol also have a viewer in VMD for visualising the ensemble via a DCD file. We also get a SetOfSequences, which we can use to check the alignment.

There is also an option on the form to write a set of atomic structures, which can be registered as an output. If you also install Scipion-Chem then you also have an option to write out a trajectory and register it as an MDSystem for further analyses.

3.3 Advanced case: many heterogeneous structures

If we import structures from another paper that uses a different nomenclature for chains, then it becomes more difficult to align them. One good solution for this is to provide a custom chain matching dictionary as follows.

Let's rename and duplicate the import protocol and import the structures from Zhang and Cai et al. 2021 (*[ZC21]*): 7krq, 7krr, 7krs.

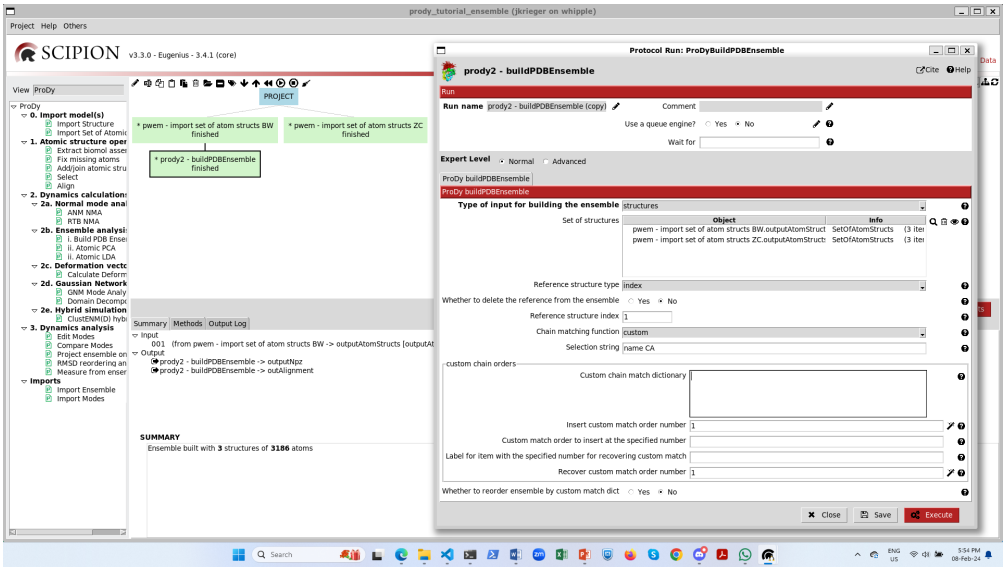
When we select the custom match option on the form, then we get an extra section with a wizard for creating a matching dictionary:

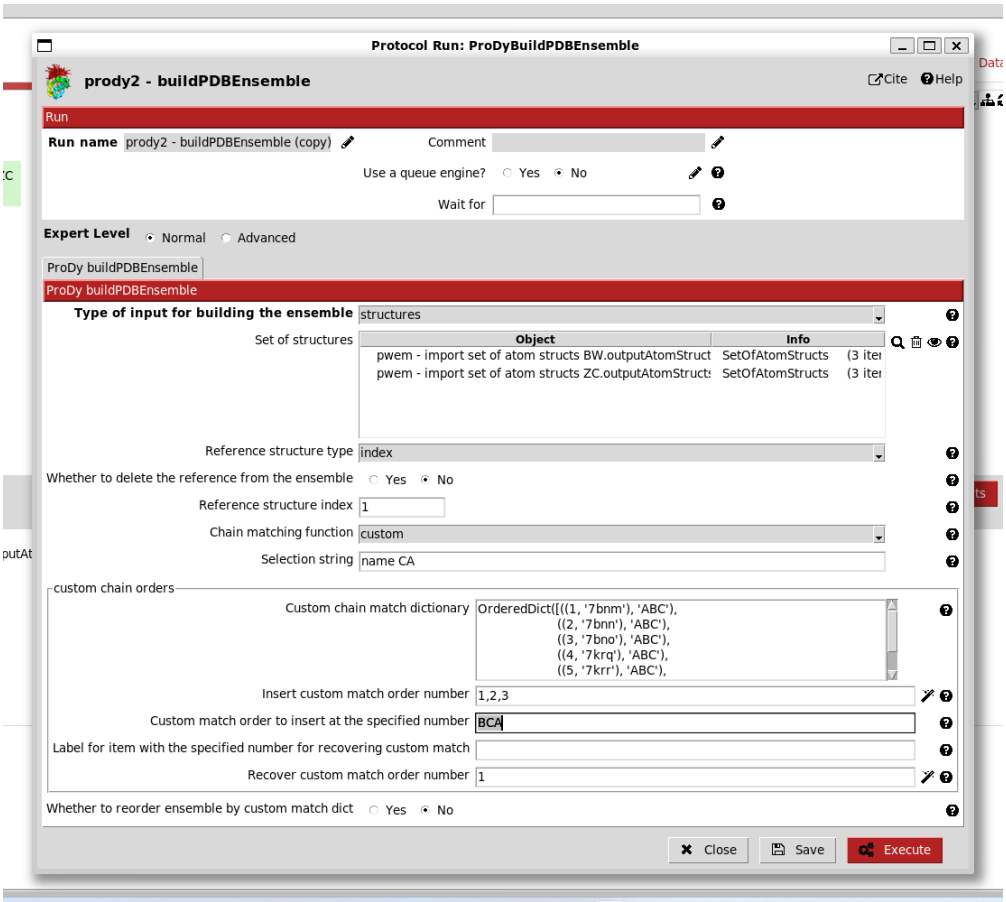
Initially, there is a big empty box, but when we click the magic wand underneath it, then it populates with the chain orders in each of the structures.

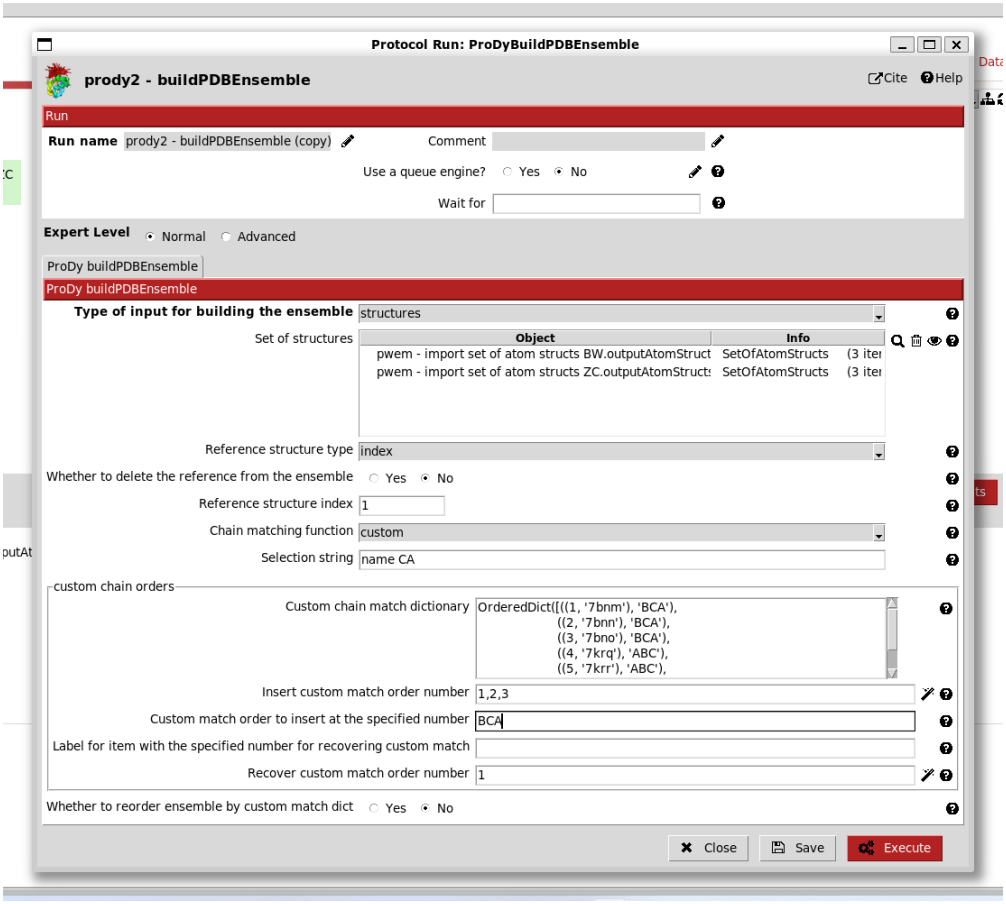
Following our previous studies (*[KJ23]* and *[GT22]*), we know that chain order BCA in the first structures matches chain order ABC in the second set. Therefore, we can change the orders at position numbers 1,2,3 to BCA as follows:

When we then click the magic wand again, the match dictionary changes accordingly:

This applies when we use the unite_chains advanced option, so we set this to yes.


prody_tutorial_ensemble (jkrieger on whipple)
Project Help Others
SCIPION v3.3.0 - Eugenius - 3.4.1 (core)
View ProDy
PROJECT
* pwem - import set of atom structs BW finished
* pwem - import set of atom structs ZC finished
* prody2 - buildPDBEnsemble finished
Summary Methods Output Log
SUMMARY
Ensemble built with 3 structures of 3186 atoms
Protocol Run: ProDyBuildPDBEnsemble
prody2 - buildPDBEnsemble
Cite Help
Run
Run name prody2 - buildPDBEnsemble (copy)
Comment
Use a queue engine? Yes No
Wait for
Expert Level Normal Advanced
ProDy buildPDBEnsemble
Type of input for building the ensemble structures
Set of structures
Object Info
pwem - import set of atom structs BW.outputAtomStruct SetOfAtomStructs
pwem - import set of atom structs ZC.outputAtomStruct SetOfAtomStructs
Reference structure type index
Whether to delete the reference from the ensemble Yes No
Reference structure index 1
Chain matching function custom
Selection string name CA
custom chain orders
Custom chain match dictionary
Insert custom match order number 1
Custom match order to insert at the specified number
Label for item with the specified number for recovering custom match
Recover custom match order number 1
Whether to reorder ensemble by custom match dict Yes No
Close Save Execute


Protocol Run: ProDyBuildPDBEnsemble
prody2 - buildPDBEnsemble
Cite Help
Run
Run name prody2 - buildPDBEnsemble (copy)
Comment
Use a queue engine? Yes No
Wait for
Expert Level Normal Advanced
ProDy buildPDBEnsemble
ProDy buildPDBEnsemble
Type of input for building the ensemble structures
Set of structures
Object Info
pwem - import set of atom structs BW.outputAtomStruct SetOfAtomStructs (3 ite
pwem - import set of atom structs ZC.outputAtomStruct SetOfAtomStructs (3 ite
Reference structure type index
Whether to delete the reference from the ensemble Yes No
Reference structure index 1
Chain matching function custom
Selection string name CA
custom chain orders
Custom chain match dictionary
OrderedDict([((1, '7bnm'), 'ABC'),
((2, '7bnn'), 'ABC'),
((3, '7bno'), 'ABC'),
((4, '7krq'), 'ABC'),
((5, '7krr'), 'ABC'),
Insert custom match order number 1,2,3
Custom match order to insert at the specified number BCA
Label for item with the specified number for recovering custom match
Recover custom match order number 1
Whether to reorder ensemble by custom match dict Yes No
Close Save Execute


Protocol Run: ProDyBuildPDBEnsemble
prody2 - buildPDBEnsemble
Cite
Help
Run
Run name
prody2 - buildPDBEnsemble (copy)
Comment
Use a queue engine?
Yes
No
Wait for
Expert Level
Normal
Advanced
ProDy buildPDBEnsemble
ProDy buildPDBEnsemble
Type of input for building the ensemble
structures
Set of structures
Object
Info
pwem - import set of atom structs BW.outputAtomStruct
SetOfAtomStructs
pwem - import set of atom structs ZC.outputAtomStruct
SetOfAtomStructs
Reference structure type
index
Whether to delete the reference from the ensemble
Yes
No
Reference structure index
1
Chain matching function
custom
Selection string
name CA
custom chain orders
Custom chain match dictionary
OrderedDict([((1, '7bnm'), 'BCA'),
((2, '7bnn'), 'BCA'),
((3, '7bno'), 'BCA'),
((4, '7krq'), 'ABC'),
((5, '7krr'), 'ABC'),
Insert custom match order number
1,2,3
Custom match order to insert at the specified number
BCA
Label for item with the specified number for recovering custom match
Recover custom match order number
1
Whether to reorder ensemble by custom match dict
Yes
No
Close
Save
Execute

We can now execute the protocol and get the correct ensemble. The box gets the same name as the other one with "(copy)" added on the end, which is good enough for now.

When we click the red analyse button, we now see that all 1-up structures have the same RBD up and that RBD is also up in the 2-up structure.

3.4 Step 2-ii: PCA

Now we are ready to analyse the ensembles by PCA, which is step 2-ii. We can do this with the default options for each ensemble, copying the protocol again for the second one.

By default, it uses half the threads, so we can run both at the same time.

Scipion also has random failures sometimes, so if this happens, just right-click the protocol box and select restart.

The protocol and output SetOfPrincipalComponents, like the NMA protocol and SetOfNormalModes, can open the NMWiz viewer or with the metadata viewer. In this case, the metadata viewer can be helpful for looking at the fractional variance.

In the NMWiz viewer, we can see that the first mode features concerted opening and closing of two RBDs, while the second features anti-correlated opening of these two RBDs.

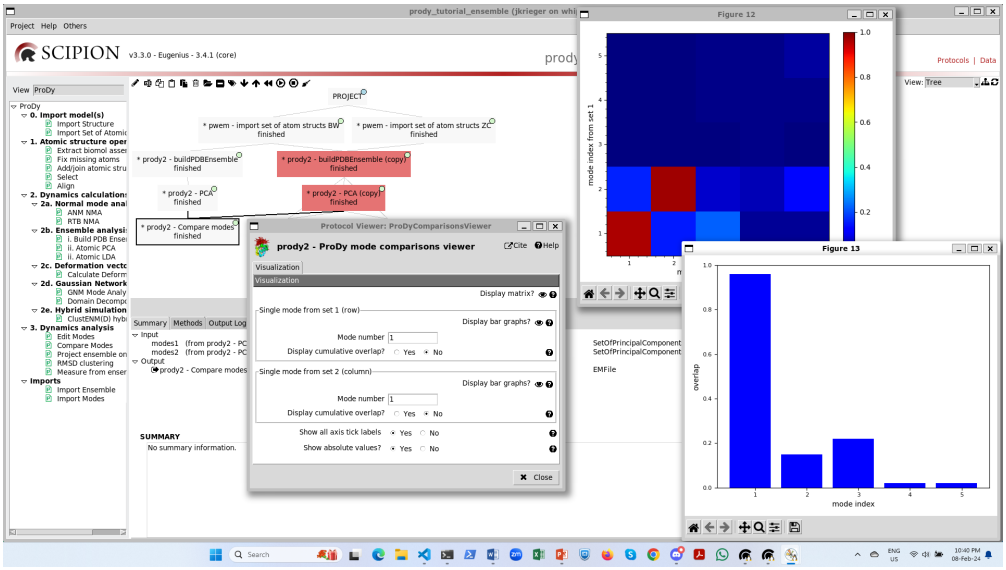
For the PCA from the larger ensemble, we also have additional meaningful modes of variation, particular PC3 which shows a locking and unlocking of the RBDs with respect to each other and the NTDs.

3.5 Step 3: analysis 1 - compare modes

The comparison between two sets of modes is slightly different to the comparison of a single vector against a set of modes. In this case, we have an overlap matrix.

This gives us additional options of matching modes and slicing out the diagonal, but we will not pick these now.

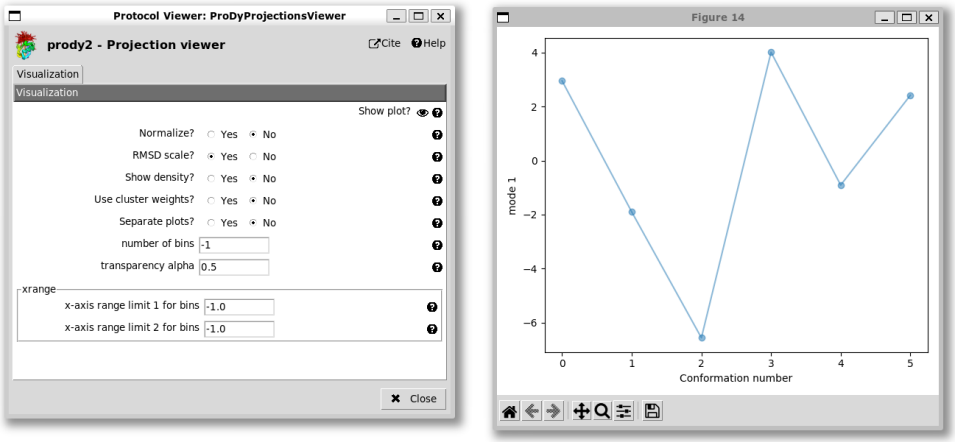
In the end, the viewer gives us ways of interacting with this matrix. Either, we can plot the colour map like `showOverlapTable()` or we can select rows to plot like `showOverlap()` as in the single deformation vector case. We then get the following outputs:



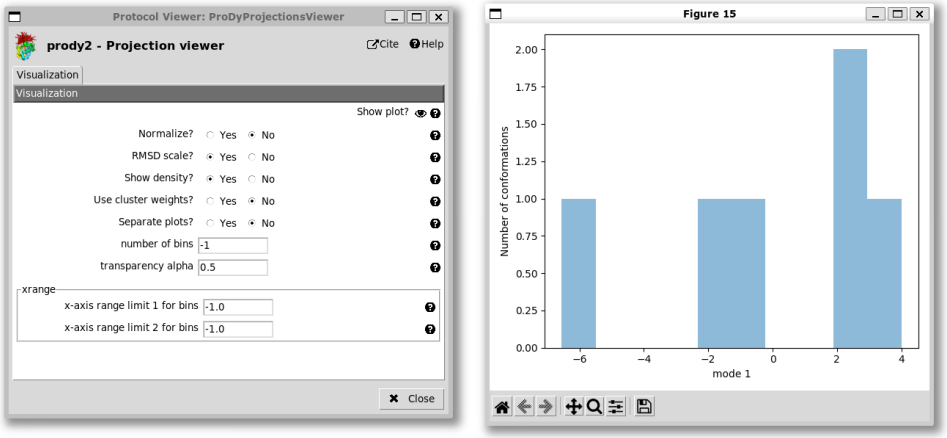
3.6 Step 3: analysis 2 - project ensembles on PCA landscapes

Finally, we can run the projection protocol to project the members of the ensemble onto the PCA components, allowing us to visualise the conformational landscapes. We have options of projecting onto 1, 2 or 3 modes.

If we project onto 1 mode, then the default option is to get a line graph with points for each structure with the conformation number on the x-axis and the PC coordinate along the y-axis:



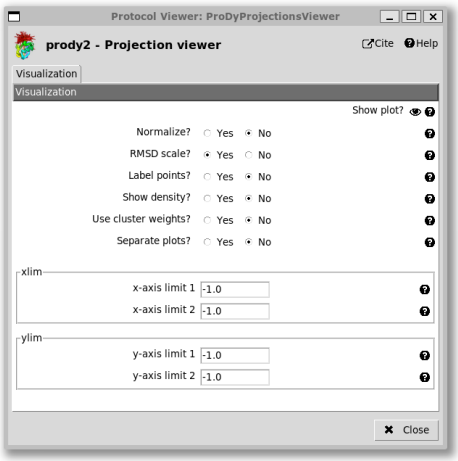
We can change this to a histogram by setting "Show density" to yes:



For the 2D case, the default plot is a scatter graph with each point being a structure and the axes being the two selected PCs:

Unlike the 1D case, it's not easy to know which point is which, so we have an option to label the points. As in the ProDy API, we can adjust the labels if they are overlapping, but that's not necessary here. We also have the option of setting the axis ranges, which is useful for getting a more equal view, e.g. setting the x-axis from -7 to +7 and the y-axis from -3.5 to +3.5 as we do here:

We see that the 2-up structure (PDB: 7bno) locates to the bottom left corner and the 3-down structures (PDB: 7bnm and 7krq) are near the the bottom right corner, separated mostly by PC1.


Protocol Viewer: ProDyProjectionsViewer
prody2 - Projection viewer
Cite
Help
Visualization
Visualization
Show plot?
Normalize? Yes No
RMSD scale? Yes No
Label points? Yes No
Show density? Yes No
Use cluster weights? Yes No
Separate plots? Yes No
xlim
x-axis limit 1 -1.0
x-axis limit 2 -1.0
ylim
y-axis limit 1 -1.0
y-axis limit 2 -1.0
Close

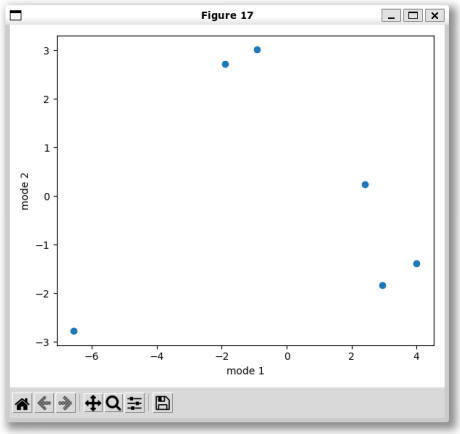
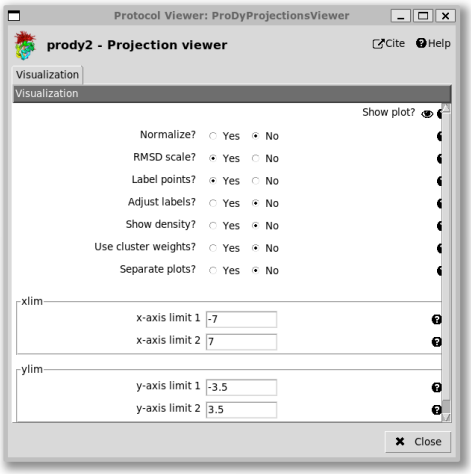


Figure 17
mode 2
mode 1


Protocol Viewer: ProDyProjectionsViewer
prody2 - Projection viewer
Cite
Help
Visualization
Visualization
Show plot?
Normalize? Yes No
RMSD scale? Yes No
Label points? Yes No
Adjust labels? Yes No
Show density? Yes No
Use cluster weights? Yes No
Separate plots? Yes No
xlim
x-axis limit 1 -7
x-axis limit 2 7
ylim
y-axis limit 1 -3.5
y-axis limit 2 3.5
Close

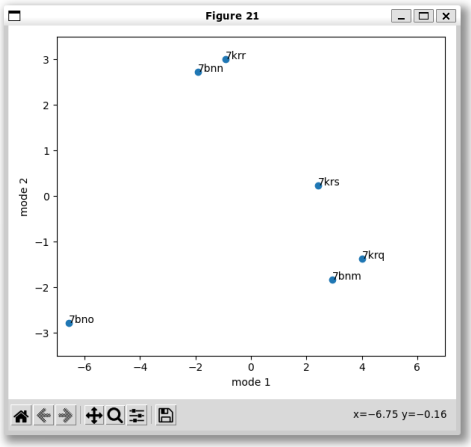
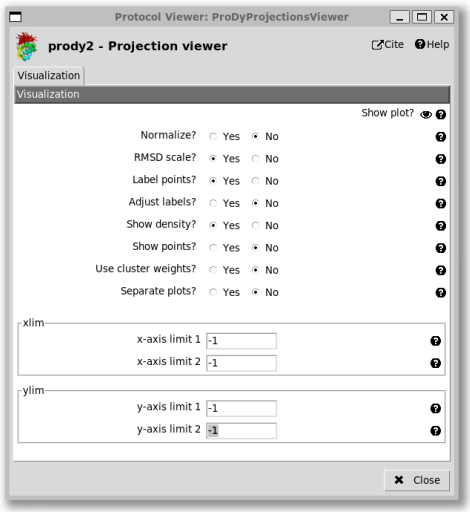
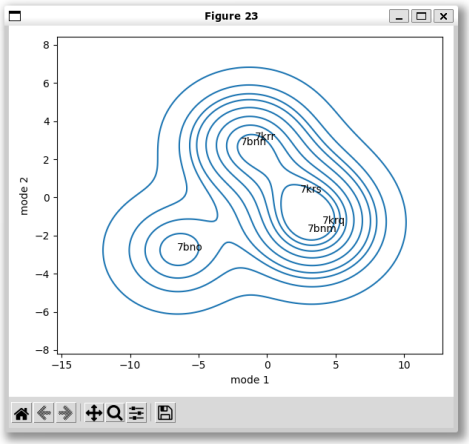


Figure 21
7krr
7bnn
7krs
7krq
7bnm
7bno
mode 2
mode 1
x=−6.75 y=−0.16

The 1-up structures (PDB: 7bnn and 7krr) lie near the middle of the top, being separated from the 3-down structures by a combination of PCs 1 and 2. The 1-up, 1-intermediate structure (PDB: 7krs) lies along the line between the two end states as expected.

We also have the option of showing a kernel density estimate of the landscape, which requires a bigger plot space, so we set the x and y limits back to -1 to allow them to take default values:





The 3D projection works in a similar way to the 2D with the absence of the density representation. It gives a 3D scatter plot that can be rotated as in the ProDy API (not shown here), which shows that PC3 mostly separates the structures from the two papers (excluding the 2-up), which may be due to differences in the constructs such as stabilising proline mutations in *[BW21]* and absent in *[ZC21]*.

There are also other options for calculating distances, angles and dihedrals from these ensembles and clustering them by RMSD.

../template/acknowledgments.rst

[KJ23] Krieger JM, Sorzano COS, Carazo JM. 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. *Int. J. Mol. Sci.* **2023** 24(18):14245.

[BW21] Benton DJ, Wrobel AG, Roustan C, Borg A, Xu P, Martin SR, Rosenthal PB, Skehel JJ, Gamblin SJ. The effect of the D614G substitution on the structure of the spike glycoprotein of SARS-CoV-2. *Proc. Natl. Acad. Sci. USA* **2021** 118(9): e2022586118.

[ZC21] Zhang J, Cai Y, Xiao T, Lu J, Peng H, Sterling SM, Walsh RM Jr, Rits-Volloch S, Zhu H, Woosley AN, Yang W, Sliz P, Chen B. Structural impact on SARS-CoV-2 spike protein by D614G substitution. *Science* **2021** 372(6541): 525-530.

[GT22] Ginex T, Marco-Marín C, Wieczór M, Mata CP, Krieger J, Ruiz-Rodriguez P, López-Redondo ML, Francés-Gómez C, Melero R, Sánchez-Sorzano CÓ, 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 JL, Carazo JM. 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 Pathog.* **2022** 18(7): e1010631.