

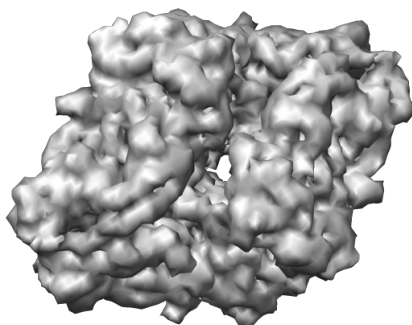


Scipion Tutorial Series

NATIONAL CENTER FOR BIOTECHNOLOGY

BIOCOMPUTING UNIT

Mix-and-Match Tutorial



SCIPION TEAM, January 2018

Intended audience

This tutorial presents a general image processing workflow to obtain 3D models of macromolecular complexes using Electron Microscopy (EM). It is designed to demonstrate how to combine different EM software packages in Scipion. No prior knowledge is required about Scipion, but some basic knowledge about 3DEM image processing is assumed and basic computer skills are required.

We'd like to hear from you

We have tested and verified the different steps described in this tutorial to the best of our knowledge, but since our programs are in continuous development you may find inaccuracies and errors in this text. Please let us know about any errors, as well as your suggestions for future editions, by writing to scipion@cnb.csic.es.

Contents

1	Getting started	3
1.1	Software Installation	3
1.2	Workflow summary	4
2	Creating a Project and Importing Data	5
2.1	Importing Data and Preprocessing	7
2.2	Importing Movies	7
2.3	Movies Alignment	10
2.4	Importing Micrographs	13
2.5	Preprocessing Micrographs	13
3	Estimating CTF	14
3.1	Using CTFFind and Xmipp	14
3.2	Analyzing CTF Results	16
4	Particle Picking	19
4.1	Using Eman2 boxer	20
4.2	Using Xmipp picking	22
4.3	Using Sparx Gaussian	24
4.4	Using Appion DoG picker	24
4.5	Using Relion auto-picking	24
4.6	Extracting Particles	25
5	2D Alignment and Classification	28
5.1	Spider MDA analysis	28
5.2	Xmipp CL2D	36
5.3	Relion 2D classification	38
6	3D Classification and Refinement	45
6.1	Relion 3D classification	45
6.2	Relion 3D auto-refinement	51

1 Getting started

1.1 Software Installation

To follow this tutorial you will need to have SCIPION properly installed on your system. To do so, you can execute the following commands:

```
git clone https://github.com/I2PC/scipion.git
cd scipion
./scipion install -j 5
```

You will also need to install other EM software packages like: motioncor2 (Zheng et al., 2017a) (if you have a GPU), CTFFind4 (Rohou and Grigorieff, 2015a), Spider (Frank et al., 1981), Relion2 (Kimanius et al., 2016) and Eman2 (Tang et al., 2007).

If your server has a GPU additional configuration is needed for packages that can benefit from it (motioncor2 and relion2). Should this be the case edit config/scipion.conf and set the following variables pointing to your CUDA installation:

```
CUDA_BIN = /usr/local/cuda-7.5/bin
CUDA_LIB = /usr/local/cuda-7.5/lib64
```

Also, edit \$HOME/.config/scipion.conf and set the following variables if needed:

```
MOTIONCOR2_CUDA_LIB = None # Fill to override scipion CUDA_LIB if different
RELION_CUDA_LIB = None # Fill to override scipion CUDA_LIB if different
RELION_CUDA_BIN = None # Only for compiling purposes
```

Then install the packages.

```
./scipion install motioncor2 ctffind4 spider relion eman --no-xmipp
```

This command will install the latest version of the packages, if you want to check the available versions simply type

```
./scipion install --help
```

Spider requires installation of csh, in an Ubuntu machine type:

```
sudo apt-get install csh
```

For the full documentation please refer to the Scipion installation page.

1.2 Workflow summary

The main goal of this tutorial is to illustrate the combination of different EM software packages in the same project. Here is a brief summary of the steps that will be done:

1. **Create a new project** with a template workflow loaded. This will simplify the processing of the first 6 steps.
2. **Import movie files** from the Relion tutorial folder. The import parameters are already loaded in the template. The result should be a *SetOfMovies* with 15 items.
3. **Align and average imported movies** using Motioncor2 if a GPU is available or Xmipp3 optical flow protocol if not. Preloaded parameters for the latest are already in the template.
4. **Estimate CTF parameters** using CTFFIND4 and create a subset of 3 micrographs with different defocus.
5. **Pick particles manually** using EMAN2 "swarm" mode from the 3 micrographs to have around 1000 particles.
6. **Extract particles** from previous picking using Xmipp3. Particles will be normalized, contrast-inverted and ranked by a Z-score for visual screening.
7. **Use MDA classification** in SPIDER to obtain 3 or 4 class averages. This step involves several protocols that will be loaded from another workflow template.
8. **Pick particles from all micrographs.** Two programs could be used:
 - (a) Xmipp3 supervised/automatic picking. First step is manual/supervised followed by a completely automatic one.
 - (b) Relion template-based picker. First step calculates the FOM maps and then autopick parameters are adjusted in the next step to pick all micrographs.

9. **Extract all particles** again using Xmipp3 before going into 2D and 3D classification.
10. **Create 2D classes** using different methods and select a subset of particles.
 - (a) Relion 2D classification
 - (b) Xmipp3 CL2D (hierarchical clustering)
11. **Classify in 3D** using Relion and taking as input the subset created previously.
12. **3D Refine** using Relion for one of the 3D classes obtained.

2 Creating a Project and Importing Data

In this demo, we use the *single particle analysis* (SPA) approach to obtain a 3D reconstruction of *beta-galactosidase*. It uses the same test data set as the Relion 1.3 tutorial, which is a subset of the micrographs used in various beta-galactosidase reconstructions (Chen et al., 2013; Vinothkumar et al., 2014). To speed up computations, these data were downsampled by a factor of 2.

The test data may be downloaded and unpacked using the following commands:

```
scipion testdata --download relion13_tutorial
```

After downloading the test data, we will create a project with the pre-loaded workflow by typing the following command:

```
scipion tutorial betagal
```

The project window should appear as shown in Figure 1. The previous command has created a new project and imported an existing workflow template. This way we have a basic template that will make it easier to follow the processing pipeline.

NOTE The ability to export/import workflows in SCIPION is a great way to reproduce previous processing steps. It is particularly useful to repeat steps on similar samples or to share knowledge between users.

In the project window, the left panel displays a tree with processing tasks (protocols) that can be used. The protocols shown can be changed in the *View* list (SPA is the default one) or found by **Ctrl**+**F**. The top right panel displays the sequence of protocols executed (runs) by user and their state: saved, running, finished or aborted. Users can visualize the runs in a list or tree view. Finally, the bottom right panel displays information for the selected run, such as inputs and outputs, execution logs or documentation. The special **Analyze Results** button can be used to visualize outputs and plot results.

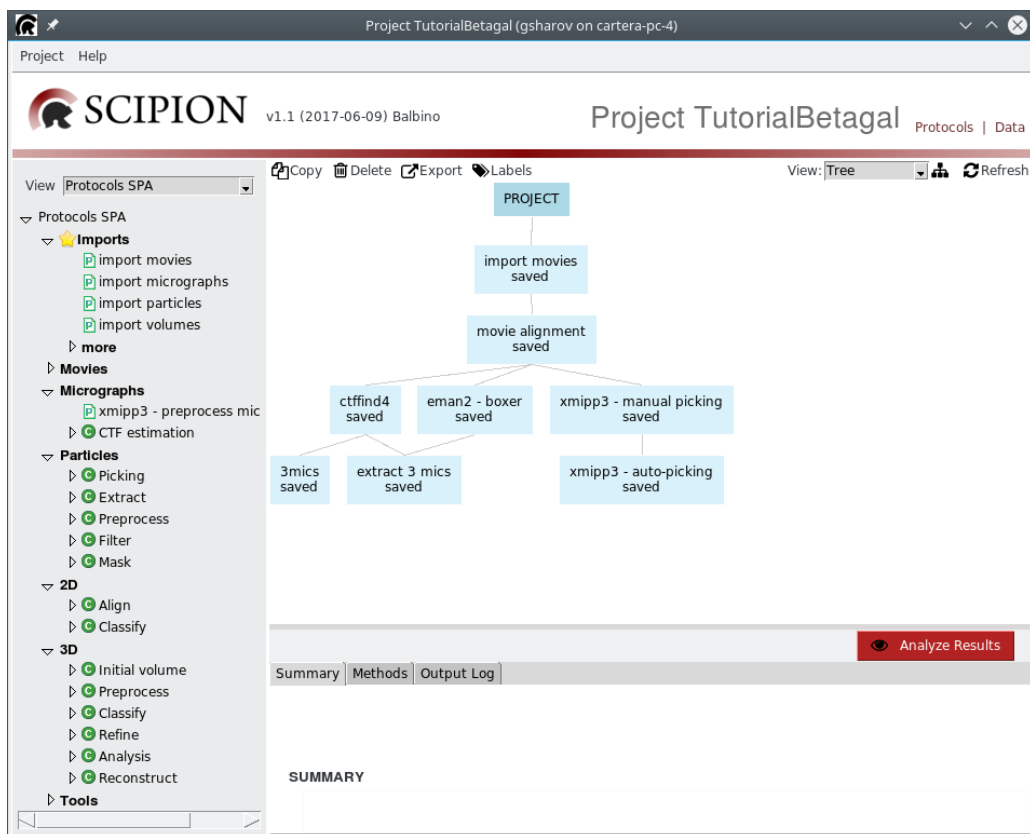


Figure 1: Project window with a pre-loaded workflow.

2.1 Importing Data and Preprocessing

In SCIPION the *import* is almost the only place where the user needs to deal directly with files. In our model each protocol has very well defined inputs and outputs, which are data objects. These objects (SetOfMovies, SetOfParticles, Volume, CTFModel, etc.) encapsulate the underlying files and formats.

When importing data like SetOfMovies, SetOfMicrographs or SetOfParticles, the user provides critical information (such as the Pixel Size). This information will not be requested any more and should be properly propagated.

NOTE Since important information is provided during the *import* step, it is recommended to take your time to check that all input parameters are correct. When importing, the binary files are not copied to the project to avoid data duplication. Instead, soft links are created pointing to the files location. If you move your project to another computer these links may be broken. There is an advanced option where you can set *Copy files?* to Yes. In this case, the project is more self-contained at the price of using more disk space.

2.2 Importing Movies

With the development of direct electron detectors (DD) low-dose images obtained by electron cryo-microscopy (cryo-EM) are recorded as frames of movies. The DDs have confirmed that the beam-induced motion (BIM) of the sample substantially degrades resolution. In this section we will show how to import movies and align them to correct the BIM using a combination of global and local movement corrections.

To import movie files, double-click the box. The protocol form with parameters will open as shown in Figure 2. In this case, the proper values for all the acquisition parameters have been loaded from the template. We only need to provide the path to the movie files of the downloaded Relion 2.1 data set. You can either use the icon to select the path or type it directly in the entry field: `relion13_tutorial/betagal/Micrographs`.

After selecting the path to the movies, we can press the button. This operation should be very fast since it only searches for the files inside the path

that match the selected pattern and register a new SetOfMovies with the provided acquisition information. After the execution, the `import movies` box should become green and the status should be *finished*. Moreover, the summary tab should display some information such as the number of movies imported. Now we can click on the `Analyze Results` button and check the list of movies as shown in Figure 3. We can right-click on any entry and select `Open` to visualize all the frames in that movie.

The screenshot shows the 'Protocol Run: ProtImportMovies' window. The title bar includes a Scipion logo, the text 'Protocol: scipion - import movies', and 'Cite' and 'Help' icons. Below the title bar is a red 'Run' header. The main configuration area includes: 'Run name' (import movies), 'Comment' (empty), 'Run mode' (Continue selected, Restart unselected), 'Host' (localhost), and 'Use queue?' (No selected). An 'Expert Level' section has 'Normal' selected. Below this are three tabs: 'Import' (selected), 'Streaming', and 'Frames'. The 'Import' section contains: 'Files directory' (elion21_tutorial/betagal/Micrographs), 'Pattern' (Falcon*.mrcs), and an 'Acquisition info' section with fields for: 'Have data been phase-flipped?' (No selected), 'Microscope voltage (kV)' (300.0), 'Spherical aberration (mm)' (2.0), 'Amplitude Contrast' (0.1), 'Magnification rate' (39548), 'Sampling rate mode' (From image), 'Pixel size ("sampling rate") (Å/px)' (3.54), 'Dose (e/Å²)' (Initial: 0.0, Per frame: empty), 'Gain image' (empty), and 'Dark image' (empty). Each field has a help icon. At the bottom are 'Close', 'Save', and 'Execute' buttons.

Figure 2: Import Movies protocol. In this case, only the path needs to be filled.

id	_filename	_samplingRate
1	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-14_33_35_0_movie.mrcs	3.5400
2	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-14_57_34_0_movie.mrcs	3.5400
3	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-15_14_01_0_movie.mrcs	3.5400
4	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-15_41_22_0_movie.mrcs	3.5400
5	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-15_53_09_0_movie.mrcs	3.5400
6	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-15_56_10_0_movie.mrcs	3.5400
7	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-16_26_22_0_movie.mrcs	3.5400
8	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-16_44_07_0_movie.mrcs	3.5400
9	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-16_55_40_0_movie.mrcs	3.5400
10	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-16_59_12_0_movie.mrcs	3.5400
11	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-17_02_43_0_movie.mrcs	3.5400
12	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-17_14_17_0_movie.mrcs	3.5400
13	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-17_17_05_0_movie.mrcs	3.5400
14	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-17_23_32_0_movie.mrcs	3.5400
15	Runs/000002_ProtImportMovies/extra/Falcon_2012_06_12-17_26_54_0_movie.mrcs	3.5400

Figure 3: Visualization of the resulting set of movies. We can open each movie and visualize its frames.

2.3 Movies Alignment

Aligning the individual frames of movies is necessary to correct BIM-induced image blurring and restore important high resolution information. In SCIPION we have integrated several protocols that correct global and local alignment to produce a final averaged micrograph.

The alignment method used by *motioncor2* (Zheng et al., 2017b) combines the correction of both uniform whole-frame motion and anisotropic local motion. This approach is fast when running on GPUs, and at the end, an “average” micrograph is generated for each movie via summation of all corrected frames. The method is appropriate for correction of global sample movements and also if the sample motion is local.

The alignment method proposed in (Abrishami et al., 2015) is based on Optical Flow (OF) and works best at local level. It is therefore particularly suited for those cases in which the BIM pattern presents a high degree of local movements, as in the Falcon II data. If the BIM pattern is characterized primarily by global movements, OF will have only a minor effect on the final average.

In our workflow, we can now open the `movie alignment` box (Figure 4) which corresponds to an Optical Flow run with the parameters already filled, and then execute the protocol. In this tutorial we have chosen *optical flow* because it can be done with and without GPU but if the latter is available, we could also run *motioncor2*. In a general case, we can use the `Search` icon to select the SetOfMovies to be used as input (there is no more file selection at this point and we should select a previously created object, such as SetOfMovies). We can also select to exclude some frames from the alignment process.

When movie alignment is finished, we can again click on the `Analyze Results` button to see the list of the resulting micrographs (Figure 5). The first column contains composite images with half of the PSD of the unaligned micrograph (left side) and half of the PSD of the aligned one (right side). The plots in the second column reflect accumulated shifts of the movie frames.

The name of each resulting micrograph appears in the last column and can also be opened to visual inspection. This protocol should produce as output a new *SetOfMicrographs* that will serve as input for further processing steps.

Protocol Run: XmippProtOFAlignment

Protocol: xmipp3 - optical alignment Cite Help

Run

Run name: movie alignment Comment:

Run mode: Continue Restart Host: localhost

Parallel: Threads MPI 4 Use queue?: Yes No

Expert Level: Normal Advanced

Input: Additional Parameters

Input

Input Movies: import movies,outputMovies

Alignment

Frames to ALIGN from 0 to 0

Use ALIGN frames range to SUM? Yes No

Binning factor: 1.0

Crop offsets (px) X 0 Y 0

Crop dimensions (px) X 0 Y 0

Close Save Execute

Figure 4: Movie Alignment protocol form.

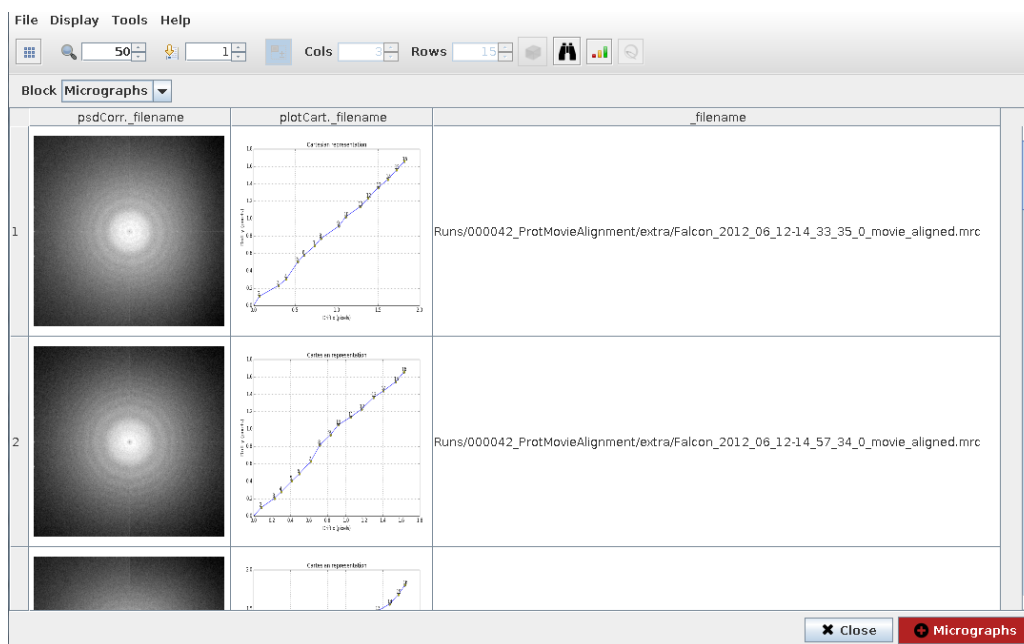


Figure 5: Visualization of the Movie Alignment outputs.

2.4 Importing Micrographs

If you already have your movies aligned, then it is possible to import directly the micrograph data. The import form is very similar to the one for movies, the main difference is that for micrographs we can also import metadata in EMX and Xmipp 3 format. In these cases, CTF information can be also associated with the micrographs. After importing the micrographs we should have an output *SetOfMicrographs* (the same type of output as from the movie alignment) that can be used in further steps.

2.5 Preprocessing Micrographs

Another useful protocol in SCIPION is `Micrographs >> xmipp3 - preprocess micrographs` which combines multiple Xmipp 3 programs in order to perform different operations on the micrographs (Figure 6). Depending on the micrographs, sometimes it is necessary to perform some preprocessing operations such as: reduce the micrograph size (usually referred as *downsampling* or *binning*), crop some pixels from the borders or remove

bad pixels, or apply a filtering operation. In the protocol form we can select input micrographs and the operations to perform and execute, as usual. The result should be a new *SetOfMicrographs* with the transformations applied.

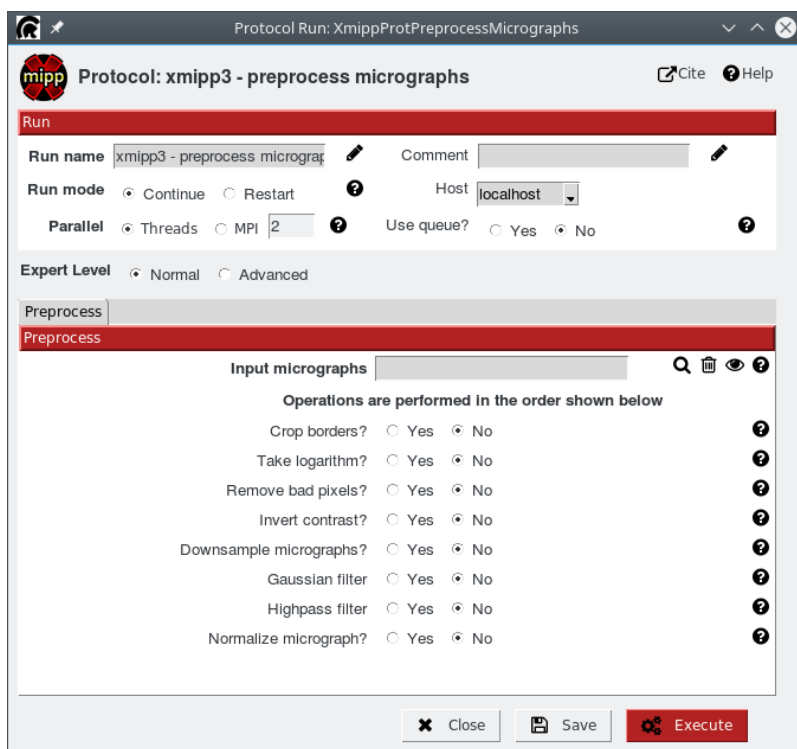


Figure 6: Preprocess Micrographs Protocol.

3 Estimating CTF

3.1 Using CTFFind and Xmipp

The next step is to estimate the CTFs (Contrast Transfer Functions) of the micrographs, either using *CTFFind4* (Rohou and Grigorieff, 2015b) or *Xmipp CTF estimation* (Sorzano et al., 2007). These protocols estimate the PSD (Power Spectral Density) of the micrographs and the parameters of the CTF (defocus U, defocus V, defocus angle, etc.). They cut the micrographs into many smaller images with a

desired window size. After that, they compute the Fourier Transform of each image and make an average.

We have developed the protocols for CTFFind4 and Xmipp in such a way that the input parameters are very similar. To estimate the CTF you will need to select the frequency region to be analyzed (Figure 7). The limiting frequencies must be such that all zeros of the CTF are contained within those frequencies. There is a wizard, shown in Figure 8, that helps in choosing those frequencies. To see all available options, choose the *Advanced* expert level and click on the **Help** button for any specific parameter. The CTFFind protocol allows to use either the **CTFFind3** or **CTFFind4** programs (the latest has been reported to be about 10 times faster than its predecessor). Since we have installed **CTFFind4**, we can choose it.

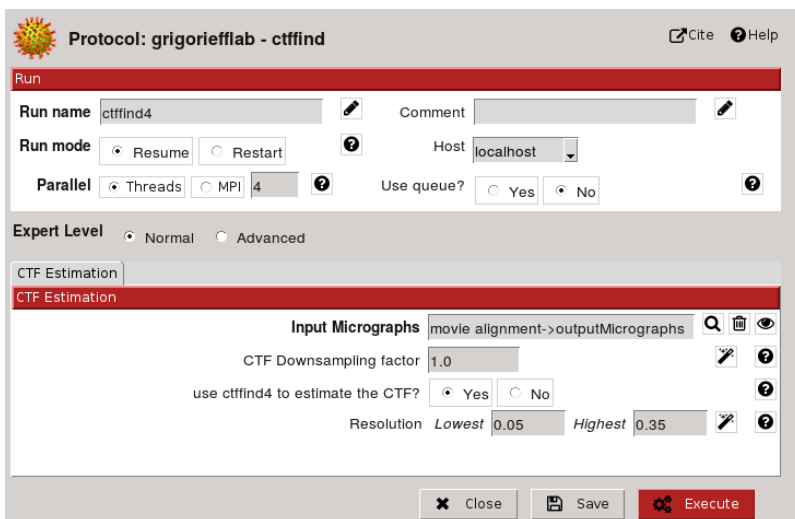


Figure 7: CTFFind4 protocol.

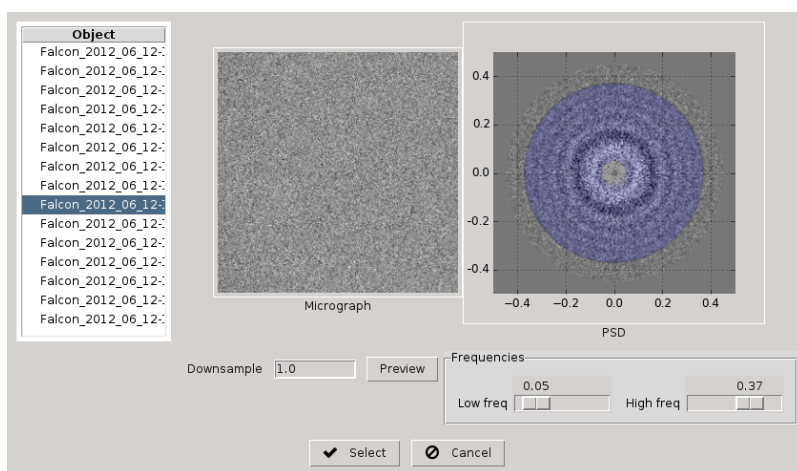


Figure 8: Wizard to help selecting the frequency region.

3.2 Analyzing CTF Results

The CTFs of good micrographs typically have multiple concentric rings, shown in Figure 9 left, extending from the image center towards its edges. Bad micrographs may lack rings or have very few rings that hardly extend from the image center. A reason to discard micrographs may be the presence of strongly asymmetric rings (astigmatism, Figure 9 center) or rings that fade in a particular direction (drift, Figure 9 right).

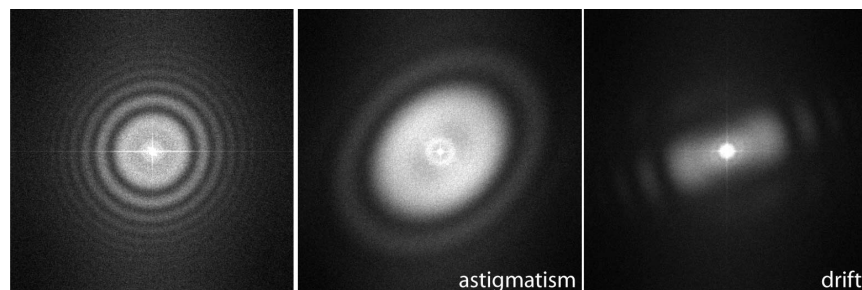


Figure 9: CTFs of good, astigmatic and drift micrographs, respectively.

When the protocol (either CTFFind or Xmipp CTF estimation) is finished you may click on the `Analyze Results` button (Figure 10). To discard micrographs with

bad CTFs you may click with the mouse right button and choose **Disable**. Once you finish the selection, press on the red **Micrographs** button to create a new subset of micrographs with only the enabled ones.

NOTE The creation of user selected subsets is tracked in SCIPION as another operation, so this action is stored in the project workflow. In the EM packages this kind of operations is usually done by editing the metadata files, but the action is no longer tracked.

Sometimes the CTF estimation algorithm may fail to find the rings even if they can be seen by eye. If this is the case, you may help the algorithm to find the rings by clicking on the image with the mouse right-button and choosing **Recalculate CTF** in the menu that appears. A graphical interface will help you to correctly identify the CTF. You must provide the first CTF zero and the search range, and then press OK. When you finish, press red **Recalculate CTFs** button.

It is also possible to analyze the CTF profiles by right-clicking on a micrograph row and selecting the *Show CTF profile* option which should open a window shown in Figure 11. The profile for the Xmipp CTF estimation show some additional options such as showing Envelope function. In the CTFFind case, there is an option to show the CTF fitting as shown in Figure 12.

Metadata: ctf5.sqlite 15 items (256 x 256)

File Display Tools Help

Block CTF

id	enabled	_psdFile	_defocusU	_defocusV	_defocusAngle	_defocusRatio	_ctffind4_ctfResolution	_micObj_filename
1	<input checked="" type="checkbox"/>		34417.6328	34121.7109	123.0873	1.0087	7.5520	Runs/000055_XmippProt0FAlignment/extra/Falcon_2012_C
2	<input checked="" type="checkbox"/>		29905.6426	29659.1152	122.3583	1.0083	6.0416	Runs/000055_XmippProt0FAlignment/extra/Falcon_2012_C
3	<input checked="" type="checkbox"/>		29946.6875	29634.8262	144.9621	1.0105	7.4896	Runs/000055_XmippProt0FAlignment/extra/Falcon_2012_C

Close Recalculate CTFs Micrographs

Figure 10: CTFfind output shows the estimated CTFs for all micrographs.

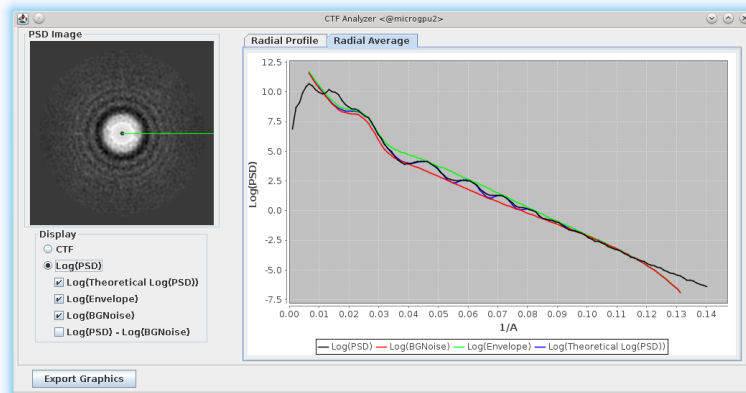


Figure 11: CTF profile of a selected micrograph.

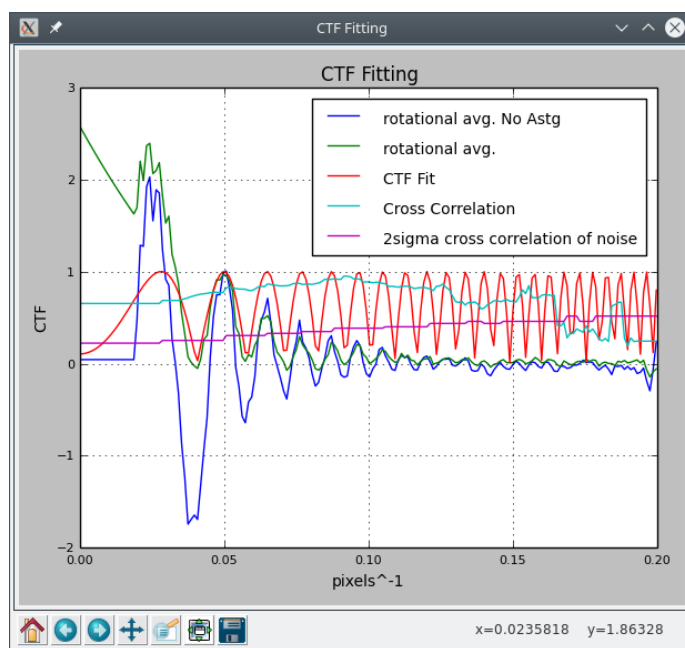


Figure 12: CTF fitting graph displaying the fit between theoretical CTF and PSD of a micrograph, calculated by CTFFind4.

4 Particle Picking

Picking is an important step to select your particles from the micrograph images. Manual picking can be very tedious and many picking tools have been developed that can be more or less convenient depending on the sample and personal preferences. In SCIPION, we have integrated different picking tools, so user can select the one which better fits his needs, or combine some of them to obtain the final coordinates. Currently, the following tools are available: Eman2 boxer, Xmipp supervised/automatic, Appion DoG picker, Sparx Gaussian (packaged with Eman), Relion autopick, gEMpicker autopick, Bsoft manual picking and Gautomatch. In this section we will illustrate how to use some of them.

4.1 Using Eman2 boxer

In this tutorial, before launching the Eman boxer, we will select a subset of 3 micrographs. To do so, we can open the results of the CTF estimation, sort the micrographs by defocus and select one at the top, one at the middle and one at the end of the list. We can register this new subset and assign a meaningful name to it that can be easily remembered. This small set will be used later during Relion picking.

The latest version of Eman2 boxer requires that particles are white over black so for this particular tutorial a previous step of inverting contrast on either micrographs or particles will be needed. Eman2 boxer allows manual picking and several types of automatic picking with fast outcome. Please refer to EMAN webpage for further information.

We need to open the `eman - boxer` box and execute it, after that we should see the e2boxer GUI as in Figure 13. For this tutorial we will use a box size of 64 pixels and the Swarm tool for semi-automated picking.

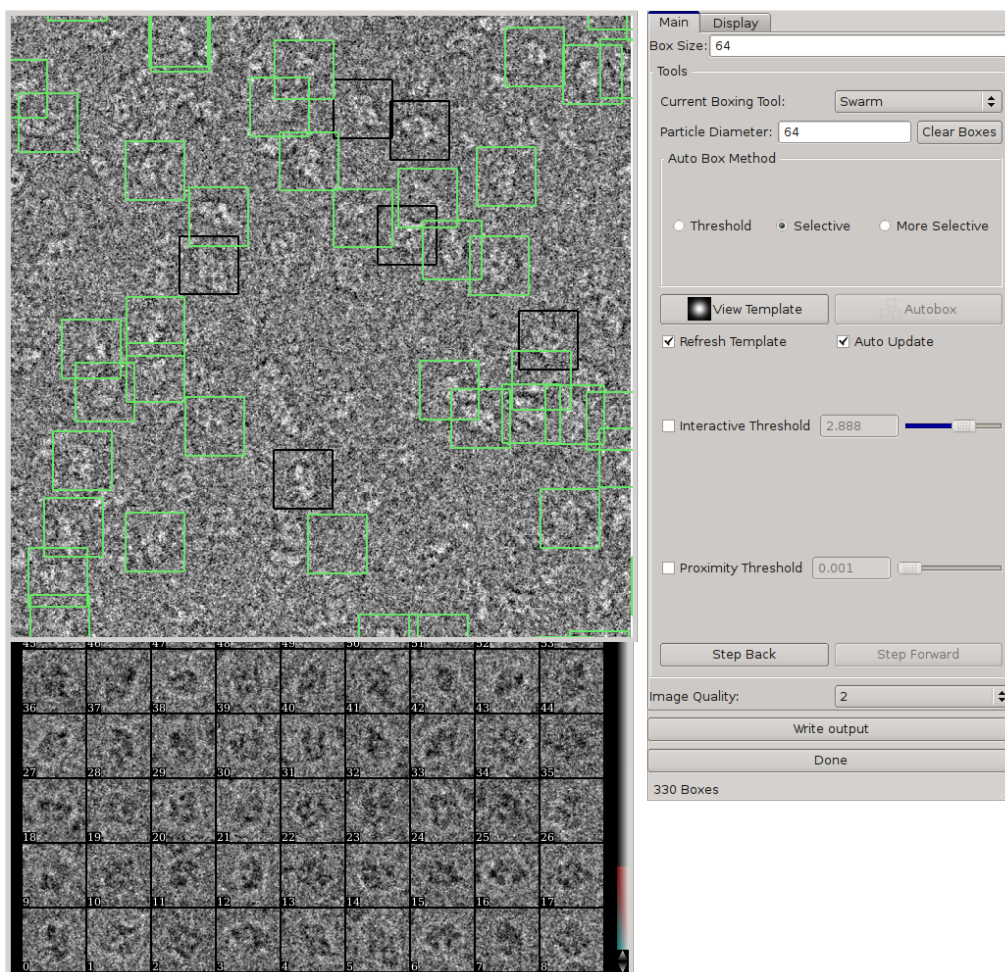


Figure 13: EMAN2 boxer, an interface for particle picking.

After we finished picking particles of one micrograph, we can move to the next one and pick all particles in this small micrograph set. At the end, we can click on the `Done` button in the e2boxer GUI and then a dialog should appear, asking to register the output coordinates in the project. After that, a *SetOfCoordinates* should appear as output of this run in the Summary tab.

4.2 Using Xmipp picking

Xmipp particle picking is divided in two steps: (1) manual/supervised picking and (2) completely automatic picking. For the manual/supervised picking, we open the `xmipp3 - manual picking` box, select the input micrographs and execute it. This box will become light yellow, meaning that this is an interactive job that we can relaunch at any time.

The Xmipp picking GUI (Figure 14) contains a control panel with the list of micrographs and some other parameters. The micrograph that we are picking is displayed in a separate window and we can apply to it a number of filters/enhancements (like Gaussian blurring, Invert contrast, adjust histogram etc.) just to improve the visualization of particles. Following is a summary of the control actions:

- Use `↑` + `mouse wheel` in the overview window to zoom in and out.
- Mark particles with the `mouse left` button. You may move its position by clicking the left mouse button on the selected particle and dragging it to a new position.
- Use `↑` + `left mouse` on a selected particle in order to remove it.
- You can apply filters to the micrographs, so that you may see the particles better. Select in the menu `Filters` as many filters as you like.

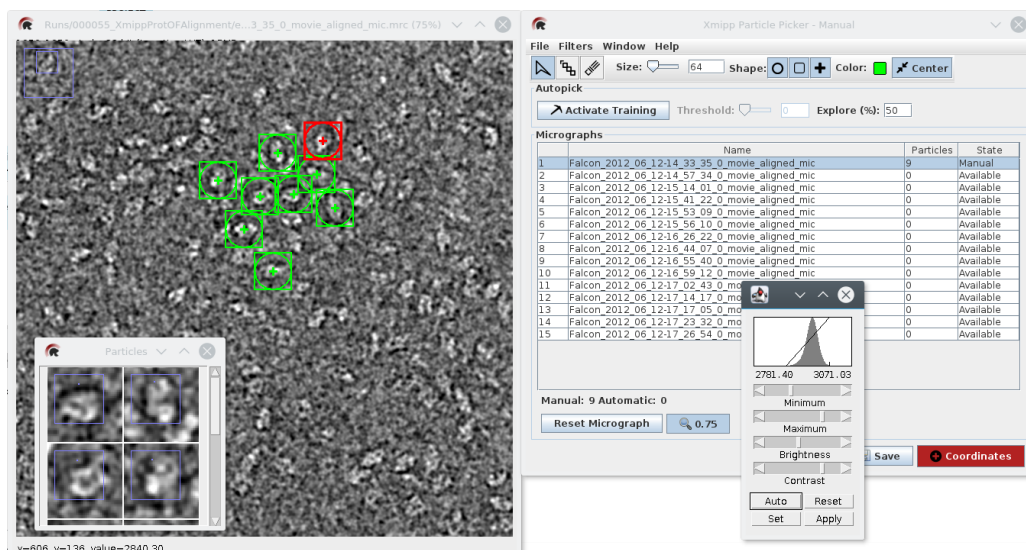


Figure 14: Xmipp picking interface.

In the manual/supervised step, we should start picking manually a few micrographs and then click the *Activate training* button. At this point, the program will train a classifier based on machine learning and will propose some coordinates automatically. You can "correct" the classifier by adding missing particles or removing wrongly picked ones. After training with a few more micrographs, we can register the output coordinates by clicking on the `Coordinates` red button.

Then we can close the GUI and open the `xmipp3 - automatic` box and select both the previous execution of manual/supervised and all micrographs as input (choose *Micrographs to pick: other*). When executing, this will pick the rest of micrographs automatically. At the end, we can review the picking coordinates and we still have a chance to add/remove particles.

NOTE Once we have trained the classifier during the manual/supervised step, we can even pick another set of micrographs of newly collected data in an automated fashion.

4.3 Using Sparx Gaussian

The Sparx Gaussian picking tool is packaged into Eman2. In this protocol, we only need to adjust a few parameters and it will pick particles based on a Gaussian distribution of what is considered as particle. In that sense, it reduces the introduced bias by our eyes or our conception of what we consider as particle.

To use this picking tool, we can open another Eman2 boxer run and then select the Gaussian tool (or launch a new protocol from `Particles` `»` `Picking` `»` `eman2 - sparx gaussian picker`). After adjusting the parameters for a few micrographs, we should click on the `Done` button, answer *Yes* to register the output and it will pick all the input micrographs.

4.4 Using Appion DoG picker

The Appion DoG picker is based on a difference of gaussians and works similarly to the Sparx gaussian picker. To launch the protocol, select in the left panel `Particles` `»` `Picking` `»` `appion - dogpicker`. This picking tool have no own GUI, however the protocol wizards for threshold and particle diameter are interactive: as soon as you click on them, the picking program will start, so that the picked coordinates can be visualized in Xmipp GUI.

4.5 Using Relion auto-picking

The protocol for Relion auto-picking in SCIPION has been divided in two steps: (1) computing the figure-of-merit (FOM) maps and (2) adjusting the parameters and picking the rest of the micrographs. The Relion picker is based on a template-matching approach, so it needs to have some 2D averages to search for particles. So you will need to go a bit ahead to the 2D classification section of this tutorial and produce some classes and then go back to Relion and use them as references.

In the first step, the FOM maps are written to disc for just a few micrographs, which are recommended to be representative of your whole data set (e.g., high and low-defocus one, and/or with thin or thick ice). Then, in the second step, there is a wizard, which will launch again the Relion program now reading the previously created FOM maps in order to speed up the computation. In this way, adjusting the

parameters becomes more interactive. After we are happy with the results for a few micrographs, we can then launch the auto-picking for the full set of micrographs in a completely automatic way. When finished, we can again review the output particles with the Xmipp GUI and add/remove particles.

4.6 Extracting Particles

Once we have a set of coordinates, we can proceed to particle extraction with Xmipp. We need to be careful with the options selected here, since this will affect later steps. The extract protocol (Figure 15) will allow us to extract, normalize and correct the CTF phases of your picked particles, among other things. The options are summarized below:

- The *coordinates* of the particles in the micrographs are taken from the results of the previous step. Also, in the same tab, select the *particle box size* in pixels (in this case 64 px).
- The *invert contrast* flag. If activated, bright regions become dark regions and the other way around. Take into account that you might have inverted already to pick with Eman2.
- The *phase flipping* flag. If activated, the protocol corrects the CTF phases of your particles.
- The *normalize* flag. If activated (recommended), the particles are normalized to have zero mean and a standard deviation of one for the background pixels.

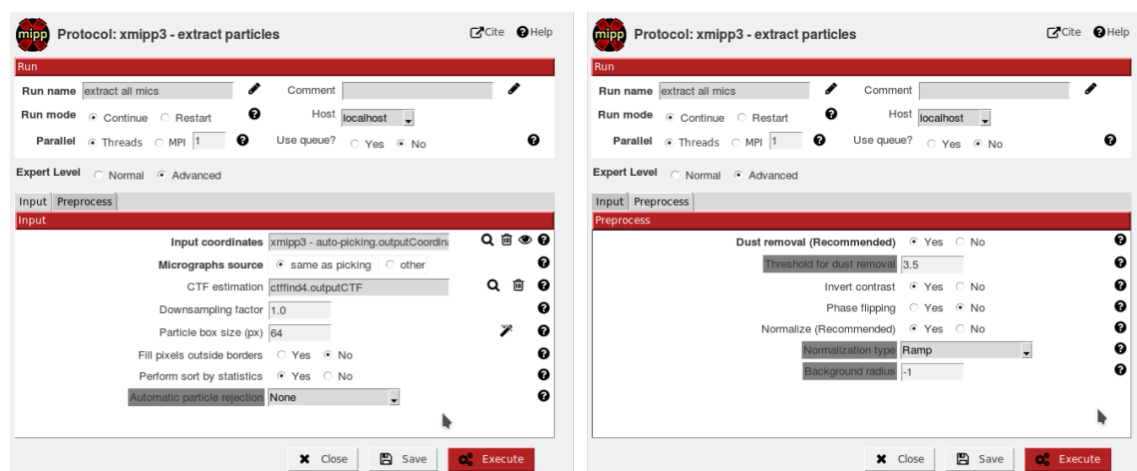


Figure 15: Extract particles protocol. Available options are shown.

In this case, we have chosen to invert the contrast, since we will use Relion later for 2D and 3D classification, and it expects the particles to be white over black. In the case of FREALIGN, it expects the particles to be black on a white background. We have also chosen not to do phase flipping, because Relion also likes to handle that correction internally. If we were going to use some Xmipp programs in the later steps, it is recommended to do phase flipping here.

We have also checked the *Sort by statistics* option, which tells the protocol to sort particles based on general statistics, assigning to each particle a z-score value. Particles with low z-score are reliable and the ones with large z-score are usually outliers. Press the **Analyze Results** button in the main window to check the extracted and normalized images (Figure 16 and 17). If the *Sort by statistics* was checked, the particles will be sorted (ascending) by the z-score value and the z-score distribution over the particle set will also be plotted. If you want to remove some particles, e.g. because they are outliers, you can use **right click** **>> disable**. To create a new set of particles you can click on **Particles** red button.

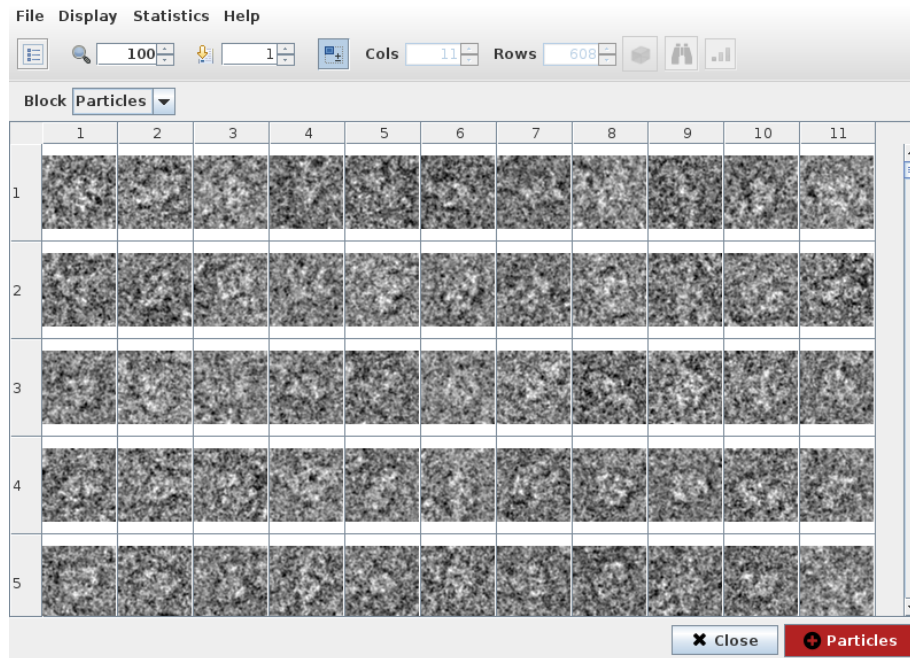


Figure 16: List of particles after extraction, sorted by z-score.

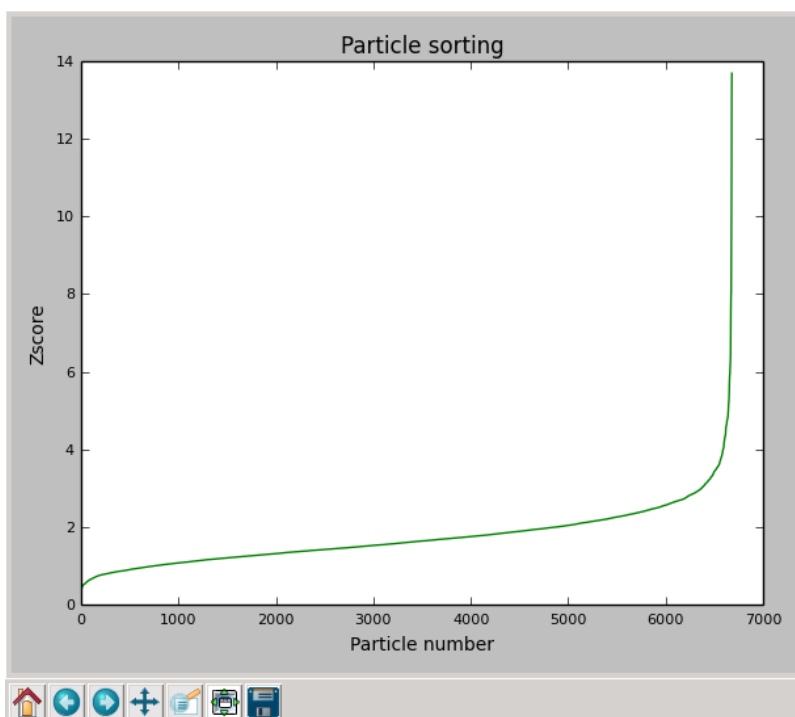


Figure 17: Plot of the z-score values for the particle set.

NOTE The GUI to visualize particles can be used with other protocols to create subsets based on any quality parameter. Particles can be easily ordered by any column and then selected based on that order to discard bad particles.

5 2D Alignment and Classification

2D analysis is a crucial step in the whole processing pipeline. It can serve as an exploratory tool for your data and can be used to throw away bad particles.

5.1 Spider MDA analysis

One of the 2D classification workflows in SPIDER is Multivariate Data Analysis (MDA). The procedures described in this section follow the recommendations from SPIDER web pages.

There are essentially only four steps:

- Low-pass filtration
- Alignment in two dimensions
- Dimension-reduction – expression of an image using only a few terms, i.e., eigenvectors
- Classification

The low-pass filtration is optional, but if you plan to look at individual particles, this step will help. For the classification below to be sensible, the images have to be aligned. The alignment step here is optional if the images have already been aligned.

Even the dimension-reduction step is optional, in theory. In principle, one could classify raw particle images. As an example here, we are using correspondence analysis (CA) for the dimension-reduction. A similar method is principal-component analysis (PCA). For classification, there are three methods illustrated here: Diday's method, Ward's method, and K-means.

In the traditional SPIDER processing pipeline all four steps are done using SPIDER scripts, while in Scipion the tools from other packages can be combined. Another advantage is that we can load a predefined workflow with all the steps as a template, that will help us to execute the steps in the correct order. To do so, go to the main project window and choose in the menu `Project >> Import workflow`. Next, select the file:

```
$SCIPION/config/workflows/workflow_betagal2.json
```

However, if you choose not to use the template workflow, all the protocols described further can be found in the left panel of the project window (select **MDA workflow** in View mode) or by using `Ctrl + F`.

Now we can use any set of particles to perform the MDA workflow. In this case we are going to use the small set of particles obtained by picking from a few micrographs. Double click on `spider - filter particles` box. Figure 18 shows the form for this first step: filtering the particles. You can select any type of filter and preview the operation in

the wizard window as shown in figure 19. After filtering the particles, they should look like in figure 20.

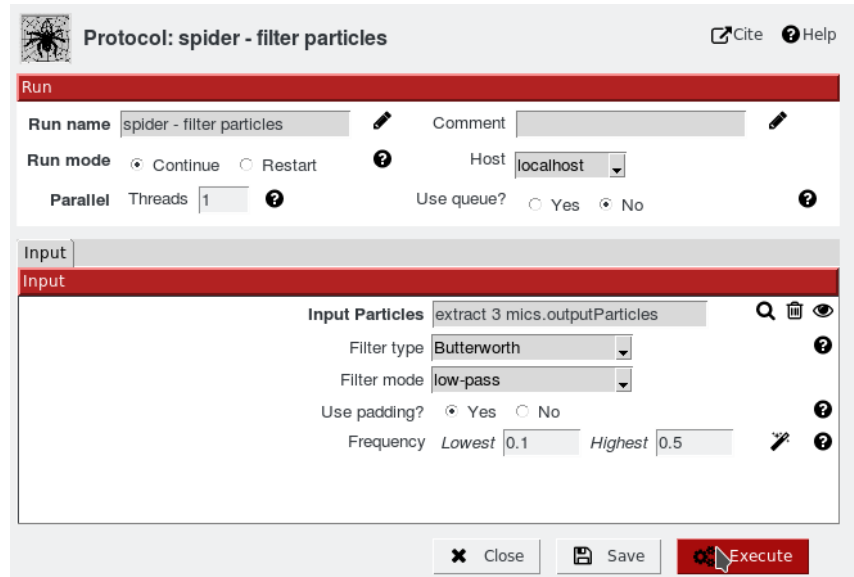


Figure 18: SPIDER filter protocol form. It is possible to select different types of filters and their parameters.

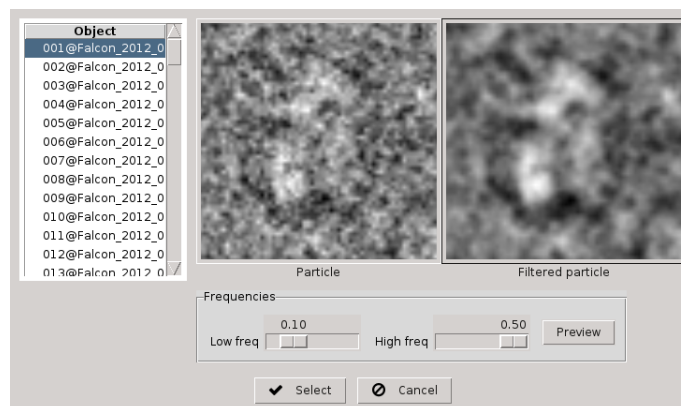


Figure 19: Wizard showing in real time how the filter operation will affect input particles.

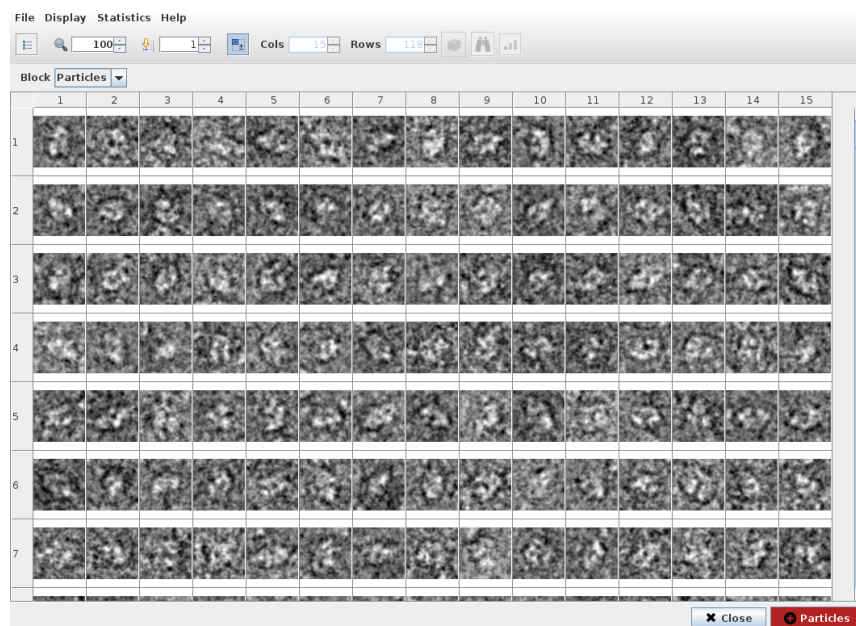


Figure 20: Visualization of the particles after the filter operation

The filtered particles will be used for the reference-free 2D alignment step. At this point, we will try to find the alignment parameters (x and y shifts and rotation angle) of each image against a common reference. Two Spider protocols are available for this task: `spider - align apsr` and `spider - align pairwise`, however any other protocol that align 2D images can be used (like `xmipp - align with cl2d`). We are going to use here the `xmipp - align with cl2d` protocol, but you can try a different one. Figure 21 shows the parameters form where we mainly need to select the input particles, that in our case are the result of the filter operation. In this protocol it is also possible to select a reference or the maximum number of pixels to consider for alignment. After this operation, the `Analyze results` button should display a list of particles, now aligned, and also the average image, similar to the one shown in figure 22.

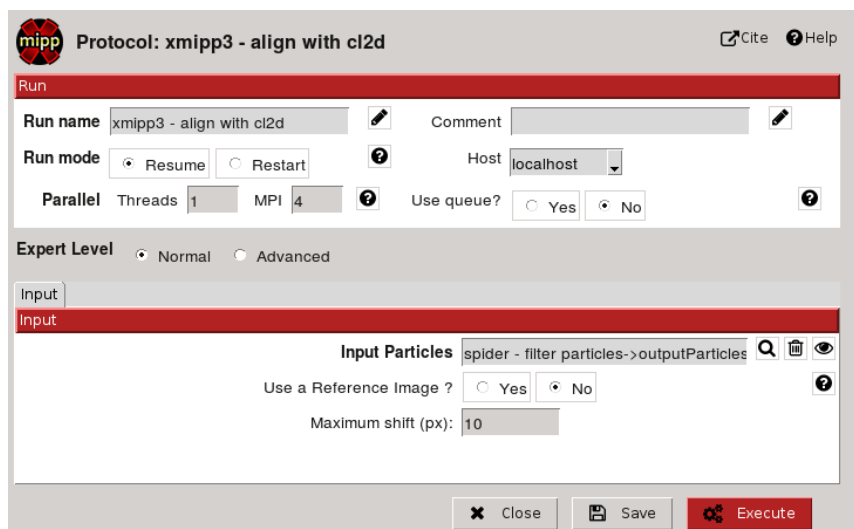


Figure 21: Xmipp - Align with CL2D protocol form.

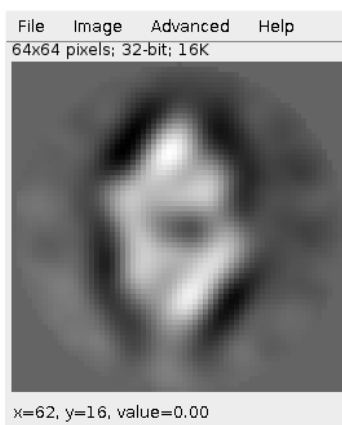


Figure 22: Particles average after alignment

Before going to dimension-reduction, we can apply a mask to aligned particles. Masking out areas outside the particle ensures that only those areas in which structural information resides will be analyzed. Exclusion of peripheral noise increases the SNR and makes the eigenvalue spectrum more compact. In other words, masking assures that existing variations within the molecule set are represented by the smallest number of factors. Another purpose of the mask is to focus on one region of the

molecule while deliberately ignoring the variations in the remaining regions. It is possible to create a mask tailored to the average image using either `spider - custom mask` protocol or an interactive mask design tool from the menu (when visualizing the average image). Another option is just to use a circular mask by specifying a radius of your particle.

The next protocol performs correspondence analysis (CA) or principal component analysis (PCA) or iterative PCA (IPCA). What this means, briefly, is that systematic pixel density variations are reduced to an arbitrary number of factors (e.g., 25 used here); in this case, the factors can be expressed as images, or "eigenimages." Each original image can be reconstituted as a linear combination of these eigenimages, using the proper weights for each factor. CA and PCA differs in the way these weights are calculated. The "importance" of each factor is the percent of density variation throughout the data set. The eigenimages of lower importance typically correspond to noise. Thus, an image reconstituted from the strongest eigenimages can be thought of as a kind of filtered image, where some noise contribution has been excluded. For a more complete description of multivariate data analysis see Chapter 4 in (Frank, 2002)

After executing the `spider - capca` protocol, the `Analyze results` will show up a form for visualization of output, as illustrated in Figure 23, where the first option will display the computed eigenimages. It is also possible to display the eigenfactors histogram or the factor maps selected by pairs (Figure 24).

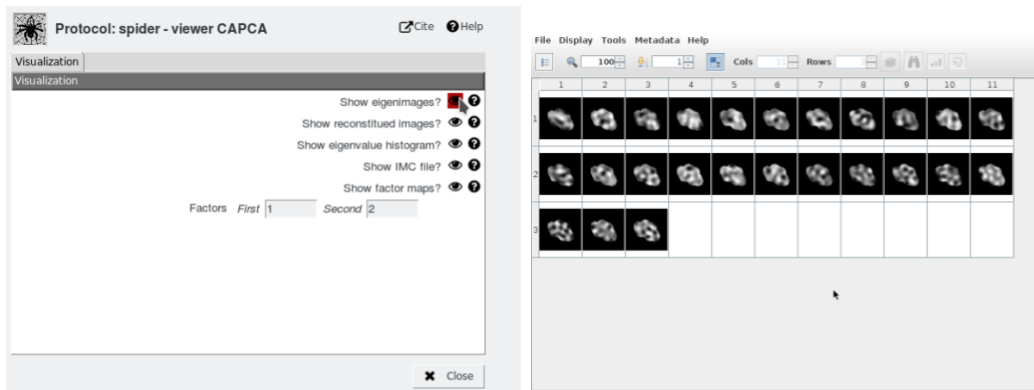


Figure 23: Output of Spider CA/PCA protocol showing the eigenimages.

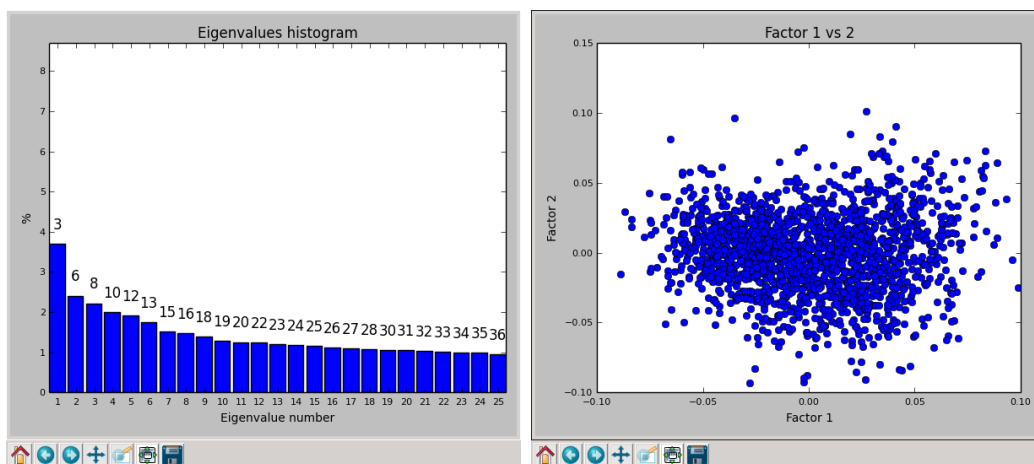


Figure 24: Histogram of eigenvalues (left) and 2D factor map (right).

In the next step we will try to separate images into homogeneous classes. There is a trade-off between homogeneity of the classes and size of the classes (which is related to the improvement of signal-to-noise ratio in the class averages). In other words, if too few classes are used, dissimilar classes will be grouped together; if too many classes are used, class averages will be likely redundant and noisy. Thus, there is some degree of subjectivity involved in the classification.

For any of the classification protocols presented here, choose the number of factors to use based on the appearance of the eigenimages and the strength of the factors (based on the histogram), including those factors that you think represent true structural differences. Using weaker factors will probably not make a appreciable difference, however.

There are three classification protocols presented here: Ward (recommended), K-means and Diday. For hints about the usage of the different methods, see the Spider MDA and Scipion wiki web pages. Both Ward and Diday are hierarchical classification methods and their results are analyzed with a similar form, such as in Figure 25, where we can display a clustering dendrogram cut at any desired height. In the case of Ward protocol, we can also display the class averages tree from which is possible to select classes or particles (see Figure 26).

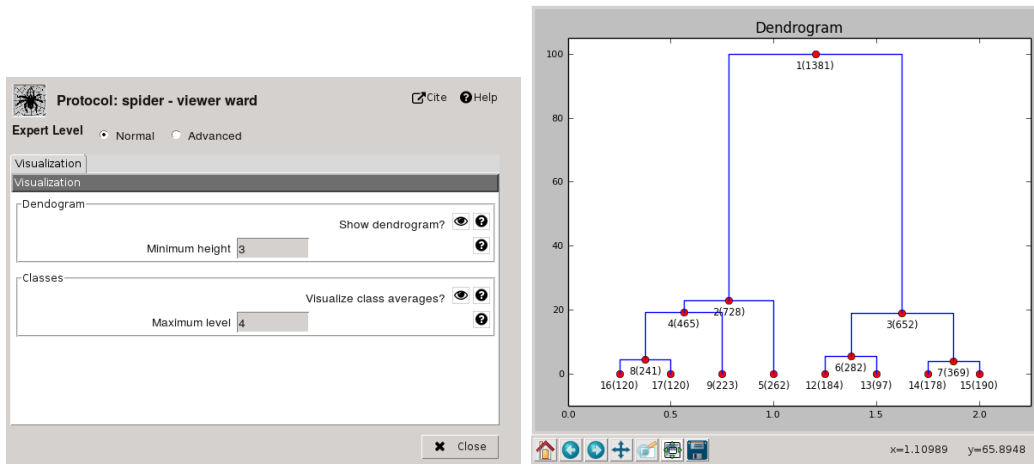


Figure 25: Results of Spider - classify Ward protocol, showing the dendrogram of clusters.

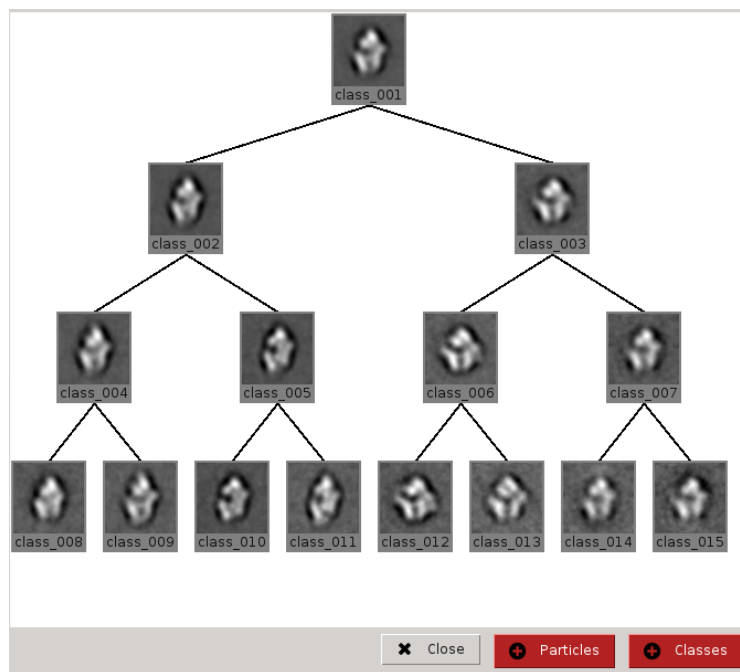


Figure 26: Another form of the results representation: class averages tree.

5.2 Xmipp CL2D

Another classification algorithm is `xmipp - cl2d` (Clustering 2D) that computes multiple classes in a very flexible way (Sorzano et al., 2010). As a similarity metric you can choose either correlation or correntropy. As a rule of thumb, correlation is well suited for images in which the noise is Gaussian (there are no neighbouring particles, black spots etc.), while correntropy is more robust to non- Gaussianity violations. You can also choose either classical (an image is assigned to a cluster with maximum similarity) or robust clustering criterion (an image is assigned to a cluster where it fits better than its neighbours. By default, correlation method and classical criterion are chosen (see “Advanced options” to change these values). The algorithm progresses in a hierarchical way: a few clusters are computed (by default, 4), then once the clustering has converged, it splits the largest classes so that the number of classes is doubled. This process is iterated until the desired number of classes is reached. Depending on the image quality, we recommend to calculate a number of classes so that on average each class has between 50 and 200 particles (classes with fewer particles are too noisy and may be unstable; classes with more particles may not be homogeneous enough, simply because they are composed of too many particles that might be different).

In general, 2D classification works better on “as good as possible” images. That means we might apply a band pass filter (see in SPA view, `Particles` \gg `Filter` \gg `xmipp3 - filter particles`) to allow the algorithm to concentrate on the important image features and not to get distracted by high-frequency noise or background gradient.

In this tutorial we are going to use the set of particles extracted from a few micrographs to speed up calculations. We should select the “Advanced options” and classify into 2 to 8 classes, the rest of the parameters are the default ones, as shown in Figure 27.

The screenshot shows the 'mipp' software interface for the 'Protocol: xmipp3 - cl2d'. The 'Run' section includes fields for 'Run name' (xmipp3 - cl2d 2), 'Comment', 'Run mode' (Resume selected, Restart unselected), 'Host' (localhost), 'Parallel' (MPI 4), and 'Use queue?' (No selected). The 'Expert Level' is set to 'Advanced'. The 'Input' section is active, showing 'Input images' (3mics particles), 'Number of classes' (8), 'Random initialization of classes' (Yes selected, No unselected), 'Number of initial classes' (2), 'Number of iterations' (10), 'Comparison method' (correlation), 'Clustering method' (classical), and 'Additional parameters'. At the bottom, there are 'Close', 'Save', and 'Execute' buttons.

Figure 27: CL2D parameters form.

The results are organized by levels (see Figure 28, in this case, first level (0) has 2 classes, second level - 4 classes and third level - 8 classes). For each level it is possible to check image information such as the similarity and the changing class number. A good indication of having a good clustering structure is that the number of images changing class assignment decreases over iterations and reach a level of about 5-10%. The levels visualized as standard classification from which we can group particles belonging to several classes to create subsets, this topic is explained in the web page *Creating subsets*.

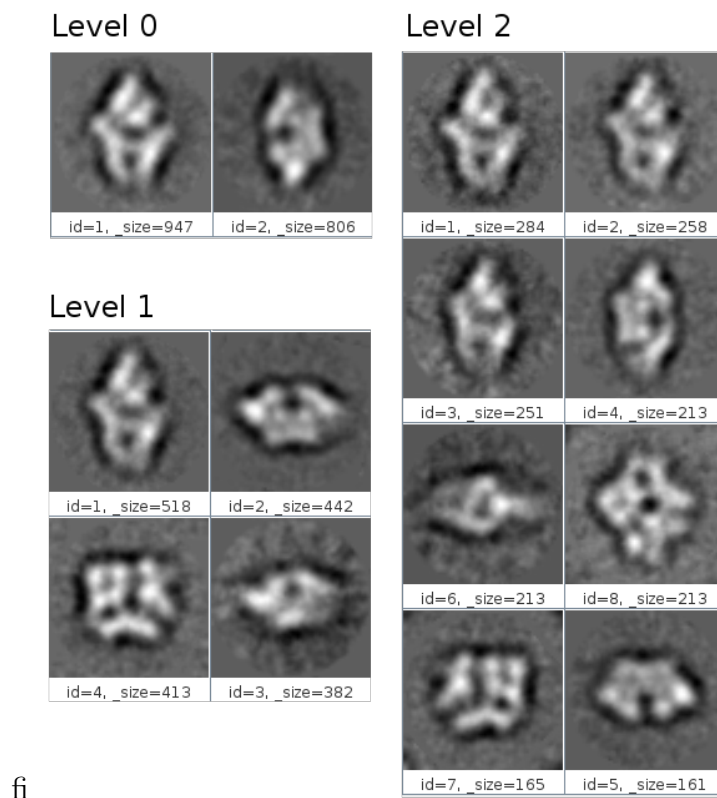


Figure 28: Classes averages of different levels after CL2D classification.

5.3 Relion 2D classification

The reference-free 2D class averaging in Relion is a great tool to throw away bad particles. Although it is recommended to try (very hard!) to only include good particles from the extraction step by inspecting and sorting the particles. Nevertheless, most of the time there are still particles in the data set that do not belong there. Because they do not average well together, they often go to relatively small classes that yield ugly 2D class averages. Throwing those away then becomes a good way of cleaning up your data.

In this tutorial we can run this job to either generate the 2D template averages for picking (using a smaller set of particles) or just to classify the whole data set and clean it up before going to 3D refinement. The latter case is more CPU intensive

and it is recommended to use a cluster or a multi-core computer, or even better a GPU if possible.

To open the protocol GUI, find **2D** » **Classify** » **relion - 2D classification**. The Figure 29 shows the form to launch this protocol. In this case we are using the whole dataset to split it in 16 classes. The diameter of the mask used for particles can be selected via the wizard shown in Figure 30.

The screenshot shows the RELION GUI for the 'relion - 2D classification' protocol. The window title is 'RELION Protocol: relion - 2D classification'. The 'Run' section includes fields for 'Run name' (relion - 2d classify 3), 'Comment', 'Run mode' (Resume selected), 'Host' (localhost), 'Parallel' settings (Threads: 4, MPI: 8), and 'Use queue?' (Yes selected). The 'Expert Level' is set to 'Normal'. The 'Input' section has tabs for 'Input', 'CTF', 'Optimisation', 'Sampling', and 'Additional'. Under 'Input', there are options for 'Continue from a previous run?' (No selected), 'Input particles' (screened particles), 'Particle mask diameter (A)' (192), and 'Number of classes' (16). At the bottom are 'Close', 'Save', and 'Execute' buttons.

Figure 29: Relion 2D classification protocol form.

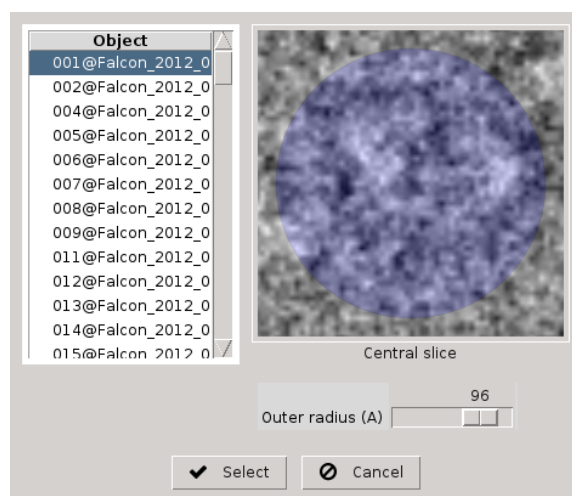


Figure 30: Wizard for particle mask radius selection.

Relion 2D classification protocol form has several tabs that you need to go through before launching the job. In the CTF tab, set:

- **Phase-flip:** In Scipion the phase-flip is a property of the particles and it is automatically detected here.
- **Do CTF-correction?:** Yes, this will perform full phase+amplitude CTF correction inside Relion.
- **Ignore CTFs until first peak?:** No. This option is only occasionally useful, when amplitude correction gives spuriously strong low-resolution components.

In the Optimisation tab, set:

- **Number of iterations:** 25. The default value is rarely changed.
- **Regularisation parameter T:** 2. For the exact definition of T, please refer to (Scheres, 2012). For cryo-EM 2D classification we typically use values of $T=1-2$, and for 3D classification values of $3-4$. For negative stain sometimes slightly lower values are better. In general, if your class averages appear noisy, then lower T; if your class averages display very low resolution, then increase T. The main thing is to be aware of overfitting high-resolution noise.

- **Mask particles with zeros?:** Yes.
- **Limit resolution E-step to (A):** -1. If a positive value is given, then no frequencies beyond this value will be included in the alignment. This can also be useful to prevent overfitting. Here we don't really need it, but it can be set to 10-15Å.

The default values in the Sampling tab are usually OK. Five degrees angular sampling is enough for most cases, although some large icosahedral viruses may benefit from finer angular sampling. In that case, one could first run 25 iterations with a sampling of 5 degrees, and then Continue from previous run for an additional five iterations (by setting Number of iterations: 30 in the Optimisation tab) with a sampling of say 2 degrees. The total number of requested CPUs is the number of threads multiplied by the number of MPI processors. Threads offer the advantage of more efficient RAM usage, whereas MPI parallelization scales better than threads. Read the original Relion tutorial for more details about the threads and MPI usage. Relion2 offers the possibility to process using GPUs, which greatly speed up the most computationally intensive steps of cryo-EM structure determination workflow, such as classification and refinement. GPU tuning parameters are found in the Additional tab, for more information refer to Relion2 documentation (<ftp://ftp.mrc-lmb.cam.ac.uk/pub/scheres/betaGuide.pdf>).

It is possible to analyze the results while the job is running or when it finishes. The form shown in Figure 31 allows to access the different visualization options. By default, the last iteration is displayed, but you can select any previous one(s).

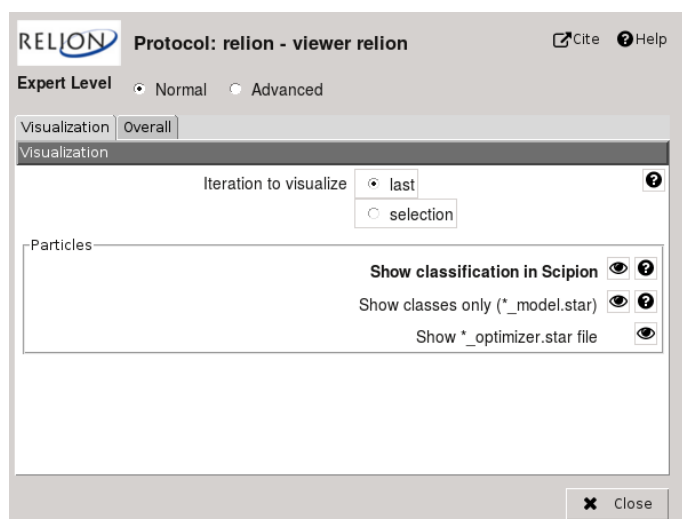


Figure 31: Relion viewer form for results visualization.

The following options are available:

- Show classification in Scipion:** This option will show the classification (classes and images assigned) of last iteration. It will generate a `SetOfClasses` (in Scipion) by converting the last `_data.star` file from Relion. This option may take few minutes depending on the dataset size. By default the classification is displayed in gallery mode with class averages (as shown in Figure 32) and sorted in reverse order of the number of particles assigned to each class. If you compare the number of assigned particles with the class average, you will see that classes with few particles have lower resolution, while classes with many particles are of higher resolution. This is an important feature of the Bayesian approach, as averaging over fewer particles will naturally lead to lower signal-to-noise ratios in the average. From this view you can easily group particles from different classes and create a new set of particles to be used in further steps (see *Creating subsets* web page for more details).
- Show classes only (*_model.star):** Display the `*_model.star` file that contains the model parameters that are refined besides the actual class averages (i.e. the distribution of the images over the classes, the spherical average of

the signal-to-noise ratios in the reconstructed structures, the noise spectra of all groups, etc. By default the class averages are rendered in reverse order of the `rlnClassDistribution` value, which is equivalent to the number of particles assigned to each class. The view can be changed to table mode as shown in Figure 33 and sorted by any column. You can also access other data tables like `data_model_class_N` where the estimated noise spectra for each group are stored. The table `data_model_groups` stores a refined intensity scale-factor for each group: groups with values higher than one have a stronger signal than the average, relatively low-signal groups have values lower than one. These values are often correlated with the defocus, but also depend on accumulated contamination and ice thickness.

- **Show `*_optimiser.star` file:** Display the `*_optimiser.star` file (see Figure 34) that contains some general information about the refinement process. From this view, you can easily open Relion star files in a new window. For example, the `*_data.star` file contains all metadata related to the individual particles. Besides the information of the input `SetOfParticles`, this file has additional information about the optimal orientations, the optimal class assignment, the contribution to the log-likelihood, etc. The `*_sampling.star` contains information about the employed sampling rates.

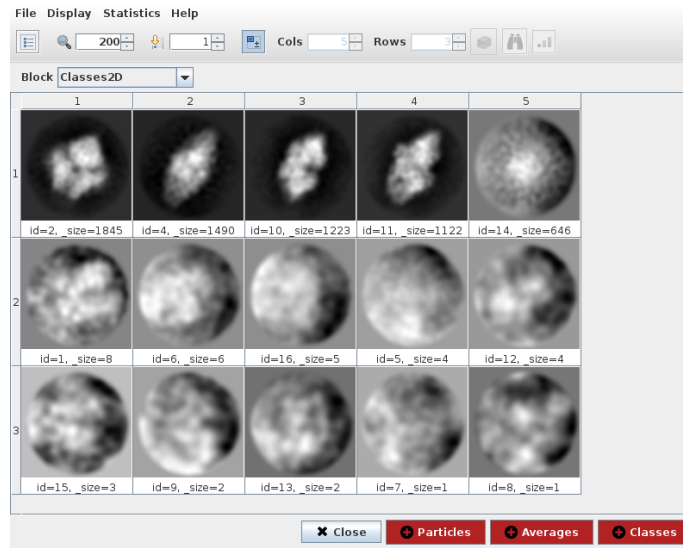


Figure 32: Relion classes displayed as Scipion classification

The screenshot shows the Scipion interface displaying a table view of a Relion model.star file. The table has the following columns: rlnReferenceImage, rlnClassDistribution, rlnAccuracyRotations, rlnAccuracyTranslations, rlnClassPriorOffsetX, and rlnClassPriorOffsetY. The data is as follows:

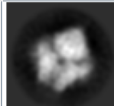
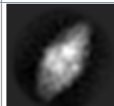

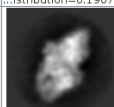
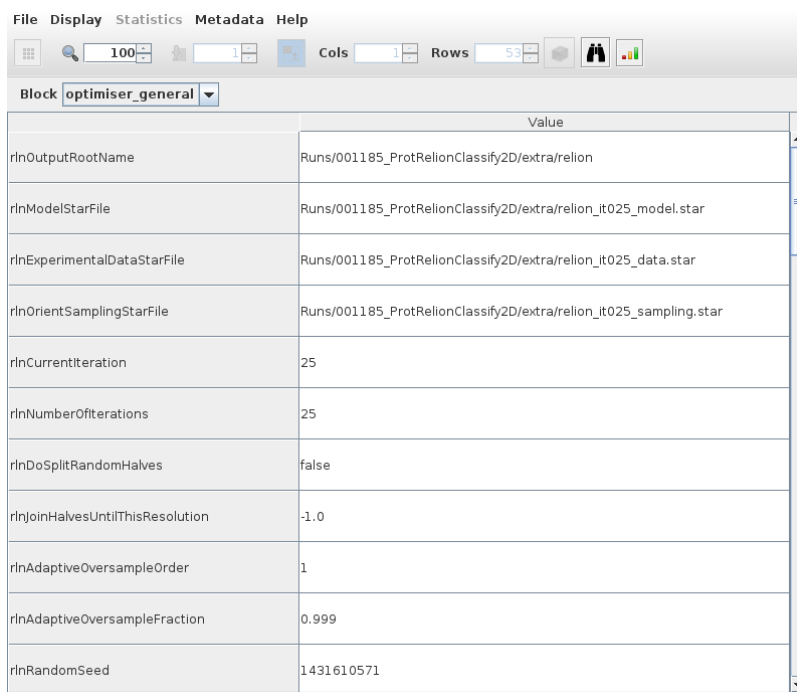
	rlnReferenceImage	rlnClassDistribution	rlnAccuracyRotations	rlnAccuracyTranslations	rlnClassPriorOffsetX	rlnClassPriorOffsetY
1		0.2899	4.6000	0.9700	-0.0251	
	...tribution=0.289933					
2		0.2406	5.0000	1.2200	0.0483	
	...tribution=0.240627					
3		0.1907	4.1000	0.8500	-0.1589	
	...tribution=0.19074					
4		0.1763	3.2000	0.7700	0.0188	
	...tribution=0.176306					

Figure 33: Relion model.star file shown in table mode.



	Value
rlnOutputRootName	Runs/001185_ProtRelionClassify2D/extra/relion
rlnModelStarFile	Runs/001185_ProtRelionClassify2D/extra/relion_it025_model.star
rlnExperimentalDataStarFile	Runs/001185_ProtRelionClassify2D/extra/relion_it025_data.star
rlnOrientSamplingStarFile	Runs/001185_ProtRelionClassify2D/extra/relion_it025_sampling.star
rlnCurrentIteration	25
rlnNumberOfIterations	25
rlnDoSplitRandomHalves	false
rlnJoinHalvesUntilThisResolution	-1.0
rlnAdaptiveOversampleOrder	1
rlnAdaptiveOversampleFraction	0.999
rlnRandomSeed	1431610571

Figure 34: Relion optimiser.star file display, from which files from other iterations can be accessed.

6 3D Classification and Refinement

3D Classification and Refinement are the last steps, and more time and resources consuming, to obtain a 3D map at the highest resolution possible. To achieve this, your data should have sufficient homogeneity, i.e. your data should represent only a single conformation of your sample.

6.1 Relion 3D classification

To separate the heterogeneity of the sample, RELION's 3D multi-reference refinement procedure provides a powerful unsupervised 3D classification approach. You will need a single, low-resolution 3D initial model to start with.

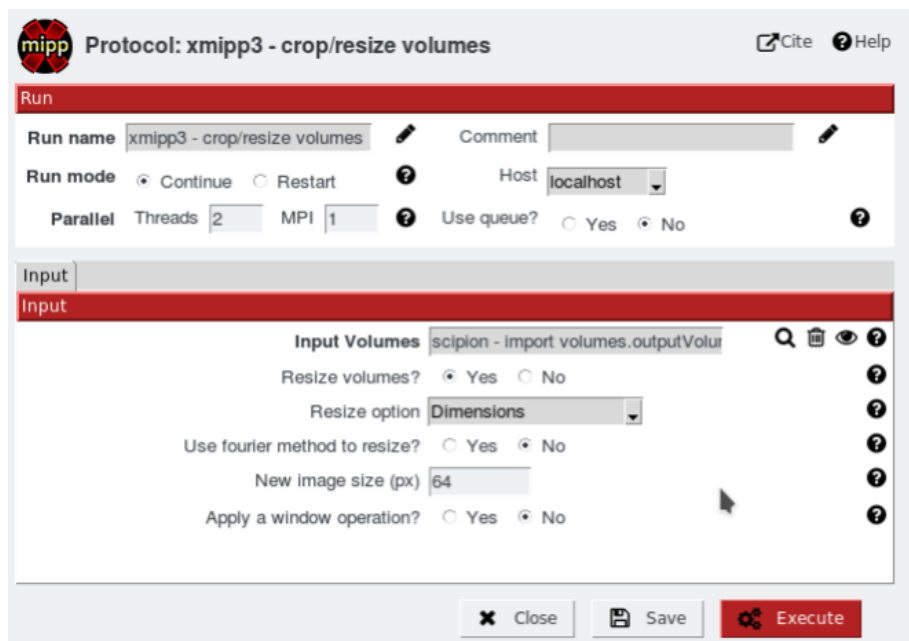
To import an initial volume click on **Imports >> Import Volume**. You will need to

provide the path to the volume file downloaded from the Relion 1.3 data set:

```
relion13_tutorial/betagal/3i3e_lp50A.mrc
```

Also, you must provide the sampling rate (3.54 Å/px) of the volume, that has the same value as the imported movies. In a typical project you may not have a good initial model. In that case you would need to generate one, using one of the several protocols that are implemented in Scipion: initial model (EMAN), ransac (Xmipp) or reconstruct significant (Xmipp).

An extra step is required since the volume just imported has 100 pixels whereas the extracted particles are 64 pixels. In order to resize volume the Xmipp Preprocess volume can be used (found under 3D/Preprocess protocols menu) and it can be seen on Figure 35.



The screenshot shows the 'Protocol: xmipp3 - crop/resize volumes' form. The 'Run' section includes fields for 'Run name' (xmipp3 - crop/resize volumes), 'Comment', 'Run mode' (Continue selected), 'Host' (localhost), 'Parallel' (Threads: 2, MPI: 1), and 'Use queue?' (No selected). The 'Input' section includes 'Input Volumes' (scipion - import volumes.outputVolu), 'Resize volumes?' (Yes selected), 'Resize option' (Dimensions), 'Use fourier method to resize?' (No selected), 'New image size (px)' (64), and 'Apply a window operation?' (No selected). The form has 'Close', 'Save', and 'Execute' buttons at the bottom.

Figure 35: Xmipp preprocess volume protocol form.

Figure 36 shows the form to launch the Relion 3D classification job. In this case, we are using a subset of particles previously selected from best 2D classes after Relion 2D classification. The 3D classification is very CPU intensive (even more than 2D) and it is recommended to use a cluster or a multi-core computer with sufficient RAM

memory or a GPU.

Figure 36: Relion 3D classification protocol form.

The options in this protocol are mostly the same as for Relion 2D classification, except some additional parameters that make sense in 3D. Just to remember them, on the CTF tab, set:

- **Do CTF-correction?:** Yes, this will perform full phase+amplitude CTF correction inside Relion.
- **Has reference been CTF-corrected?:** No. This means that in the first iteration, the Fourier transforms of the reference projections are not multiplied by the CTFs. Set Yes if the reference map represents CTF-unaffected density, e.g. it was created inside RELION or comes from a PDB.

- **Phase-flip:** In Scipion the phase-flip is a property of the particles and it is automatically detected here.
- **Ignore CTFs until first peak?:** No. This option is only occasionally useful, when amplitude correction gives spuriously strong low-resolution components.
- **Do manual grouping ctf?:** No. Use this option only when you need to regroup your particles because are a few (e.g., less than 10) of them in one or more micrographs.

On the Optimisation tab, set:

- **Number of iterations:** 25. The default value is rarely changed.
- **Regularisation parameter T:** 2. For the exact definition of T, please refer to (Scheres, 2012). For cryo-EM 2D classification we typically use values of T=1-2, and for 3D classification values of 3-4. For negative stain sometimes slightly lower values are better. In general, if your class averages appear noisy, then lower T; if your class averages display very low resolution, then increase T. The main thing is to be aware of overfitting high-resolution noise.
- **Mask particles with zeros?:** Yes.
- **Reference Mask (Optional):** No mask is given. In this case, a soft spherical mask based on the diameter provided in the Input tab will be applied. If you want to use a mask, it must have the same dimensions as the reference, and pixel density values between 0 and 1, with 1 being 100% protein and 0 being 100% solvent. The reconstructed reference map will be multiplied by this mask.
In some cases, e.g. non-empty icosahedral viruses, it is also useful to use a second mask.
- **Second reference mask (Optional):** Leave this empty. For all white (value 1) pixels in this second mask the corresponding pixels in the reconstructed map are set to the average value of these pixels. Thereby, for example, the higher density inside the virion may be set to a constant. Note that this second mask

should have one-values inside the virion and zero-values in the capsid and the solvent areas.

- **Limit resolution E-step to (A):** -1. If a positive value is given, then no frequencies beyond this value will be included in the alignment. This can also be useful to prevent overfitting. Here we don't really need it, but it can be set to 10-15Å.

The default values in the Sampling tab are usually used. The default value of the angular sampling is enough for most projects. Note that you can only select a discrete defined range of angular samplings values. For some large icosahedral viruses, you may benefit from using finer angular samplings. In that case, one could first run 25 iterations with a sampling of 3.7 degrees, and then Continue from previous run for an additional five iteration (by setting Number of iterations: 30 in the Optimisation tab) with a sampling of say 1.8 degrees.

The total number of requested CPUs or GPU parameters are the same as explained for the Relion 2D classification. Read the original Relion tutorial for more details about the threads and MPI usage.

The form shown in Figure 37 allows to access different visualization options. By default, the last iteration is displayed, but you can select any previous one(s). Also, as in Relion 2D classification, you can view intermediate results while the job is running.

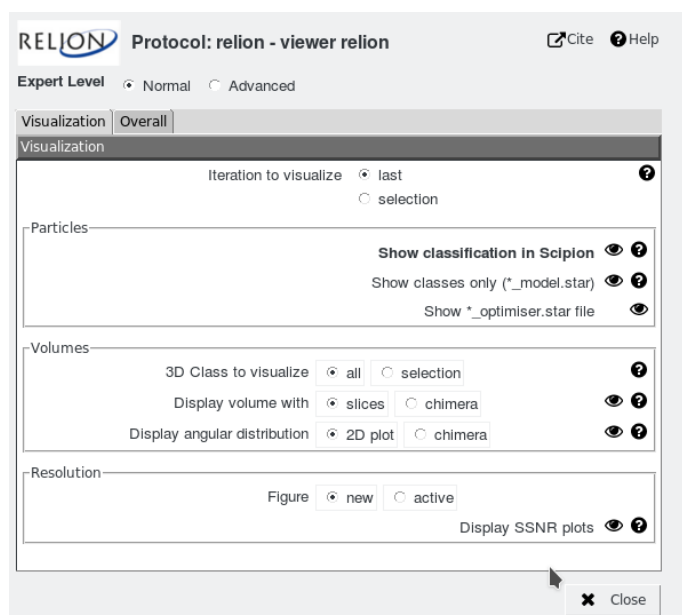
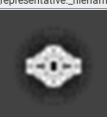




Figure 37: Relion viewer form for results visualization of 3D classification.

enabled	id	_size	_representative_filename	_inClassDistribution	_inAccuracyRotations	_inAccuracyTranslations
<input checked="" type="checkbox"/>	3	4894	 _size=4894, id=3	0.5630	3.4250	0.5880
<input checked="" type="checkbox"/>	1	2844	 _size=2844, id=1	0.3306	3.5700	0.6290
<input checked="" type="checkbox"/>	2	913	 _size=913, id=2	0.1063	4.4500	1.1030

At the bottom of the table, there are buttons for 'Close', 'Particles', 'Volumes', and 'Classes'.

Figure 38: Relion 3D classes displayed as Scipion classification

Compared to the 2D classification results, the following additional options are available:

- **3D Class to visualize:** You may select 3D classes to display.
- **Display volume with:** This option show the 3D classes, selected above, in slices along z axis or as surfaces in Chimera.
- **Display angular distribution:** Display the angular distribution of the classes in a 2D plot or using Chimera with red spheres.
- **Display SSNR plots:** Display signal-to-noise ratio plots.

6.2 Relion 3D auto-refinement

Once a subset of particles with sufficient homogeneity has been selected, one may use `3D >> Refine >> Relion - 3D auto-refine` protocol to refine this subset to high resolution in a fully automated manner. This procedure employs so-called gold-standard Fourier Shell Correlation (FSC) calculations to estimate resolution, so that over-fitting may be almost completely avoided. Combined with a novel procedure to estimate the accuracy of the angular assignments, the protocol automatically determines when a refinement has converged. Thereby, this procedure requires very little user input, i.e. it remains objective, and has been observed to yield excellent maps for “normal” data sets. Another advantage is that one typically needs to run it once, as there are hardly any parameters to optimize.

The input parameters shown in Figure 39 largely remain the same as for the 3D classification protocol, although some of them are no longer available.

Figure 39: Relion 3D auto-refine form.

The *Angular sampling interval* option in the Sampling tab will only be used in the first few iterations, from there on the algorithm will automatically increase the angular sampling rate until convergence. Therefore, for all refinements with less than octahedral or icosahedral symmetry, we typically use the default values of *Angular sampling interval*: 7.5 degrees for global searches, and *Local searches from auto-sampling* of 1.8 degrees. Only for higher symmetry refinements, we use 3.7 degrees and perform local searches from 0.9 degrees.

There are another Tab (Movies) with respect to Relion 3D classification form. It is used when you want align the movie-particles of each frame to run the particle polishing protocol later.

The Analyse result form is basically the same as for Relion 3D classification, as shown in Figure 40, although it has in addition a FSC plot.

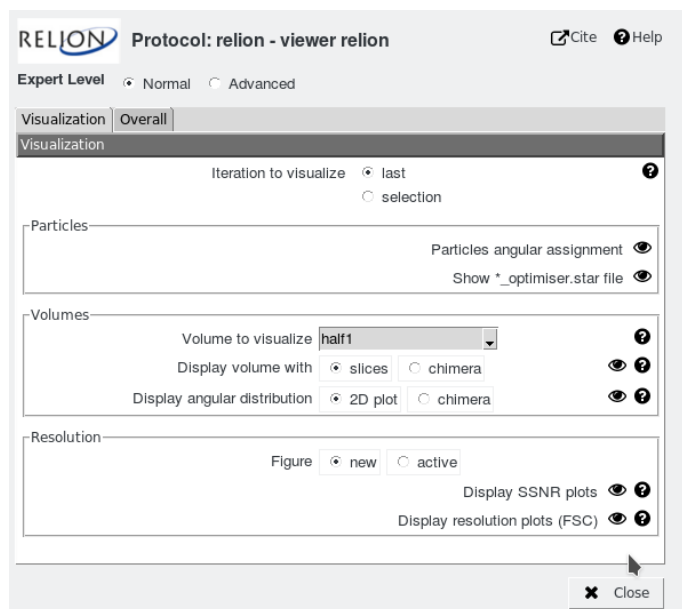


Figure 40: Analyse Result of Relion 3D auto-refine.

References

- Abrishami, V., Vargas, J., Li, X., Cheng, Y., Marabini, R., Óscar Sánchez Sorzano, C., and Carazo, J. M. (2015). Alignment of direct detection device micrographs using a robust optical flow approach. *Journal of Structural Biology*, 189(3):163 – 176.
- Chen, S., McMullan, G., Faruqi, A. R., Murshudov, G. N., Short, J. M., Scheres, S. H., and Henderson, R. (2013). High-resolution noise substitution to measure overfitting and validate resolution in 3d structure determination by single particle electron cryomicroscopy. *Ultramicroscopy*, 135(0):24 – 35.
- Frank, J. (2002). Single-particle imaging of macromolecules by cryo-electron microscopy. *Annu. Rev. Biophys. Biomol. Struct.*, pages 303–319.
- Frank, J., Shimkin, B., and Dowse, H. (1981). Spider a modular software system for electron image processing. *Ultramicroscopy*, 6:343–358.

- Kimanius, D., Forsberg, B. O., Scheres, S., and Lindahl, E. (2016). Accelerated cryo-em structure determination with parallelisation using gpus in relion-2. *Elife*, 5.
- Rohou, A. and Grigorieff, N. (2015a). Ctffind4: Fast and accurate defocus estimation from electron micrographs. *J Struct Biol*, 192(2):216–221.
- Rohou, A. and Grigorieff, N. (2015b). CTFFIND4: Fast and accurate defocus estimation from electron micrographs. *J. Struct. Biol.*, 192(2):216–221.
- Scheres, S. H. (2012). Relion: Implementation of a bayesian approach to cryo-em structure determination. *Journal of Structural Biology*, 180(3):519 – 530.
- Sorzano, C. O. S., Bilbao-Castro, J. R., Shkolnisky, Y., Alcorlo, M., Melero, R., Caffarena-Fernández, G., Li, M., Xu, G., Marabini, R., and Carazo, J. M. (2010). A clustering approach to multireference alignment of single-particle projections in electron microscopy. *J Struct Biol*, 171(2):197–206.
- Sorzano, C. O. S., Jonic, S., Boisset, N., and Carazo, J. M. (2007). Fast, robust, and accurate determination of transmission electron microscopy contrast transfer function. *J Struct Biol*, 160(2):249–262.
- Tang, G., Peng, L., Baldwin, P., Mann, D., Jiang, W., Rees, I., and Ludtke, S. (2007). EMAN2: an extensible image processing suite for electron microscopy. *J. Struc. Bio.*, 157:38–46.
- Vinothkumar, K. R., McMullan, G., and Henderson, R. (2014). Molecular mechanism of antibody-mediated activation of i2-galactosidase. *Structure*, 22(4):621 – 627.
- Zheng, S., Palovcak, E., Armache, J.-P., Verba, K., Cheng, Y., and Agard, A. (2017a). MotionCor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nature Methods*, 14:331–332.
- Zheng, S., Palovcak, E., Armache, J.-P., Verba, K., Cheng, Y., and Agard, D. (2017b). Motioncor2: anisotropic correction of beam-induced motion for improved cryo-electron microscopy. *Nature methods*, 14(4):331–332.