National Center for Biotechnology
Biocomputing Unit

Mix-and-Match Tutorial

Scipion Team
**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.

**We’d like to hear from you**

We have tested and verified the different steps described in this demo 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 3

2 Creating a Project and Importing Data 4
  2.1 Importing Data and Preprocessing 6
  2.2 Importing Movies 7
  2.3 Movies Alignment 9
  2.4 Importing Micrographs 12
  2.5 Preprocessing Micrographs 12

3 Estimating CTF 13
  3.1 Using CTFFind and Xmipp 13
  3.2 Analyzing CTF Results 15

4 Particle Picking 18
  4.1 Using Eman boxer 18
  4.2 Using Xmipp picking 20
  4.3 Using Sparx Gaussian 22
  4.4 Using Appion DoG picker 22
  4.5 Using Relion auto-picking 22
  4.6 Extracting Particles 23

5 2D Alignment and Classification 25
  5.1 Spider MDA analysis 25
  5.2 Xmipp CL2D 33
  5.3 Relion 2D classification 35

6 3D Classification and Refinement 41
  6.1 Relion 3D classification 41
  6.2 Relion 3D auto-refinement 47
1 Getting started

1.1 Software Installation

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

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

You will also need to install other EM software packages like: CTFFind4, Spider, Relion (Scheres, 2012) and Eman2. 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. Following is a brief summary of the steps that will be done:

1. **Create a new project** with a template workflow loaded. This will ease to follow the processing 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. **Average the imported movies** using Xmipp3 optical flow protocol.

4. **Estimate CTF parameters** using CTFFIND4 and create a subset with 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** using Xmipp3 from previous picking. Particles will be normalized, contrast-inverted and ranked to 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 will be used:
   
   (a) Xmipp3 supervised/automatic picking. First step is manual/supervised and then completely automatic.

   (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. **Compare picking results** through a consensus protocol that keeps particles selected by both algorithms.

10. **Extract all particles** again using Xmipp3 before going into 2D and 3D classification.

11. **Create 2D classes** using different methods and select a subset of particles.

   (a) Relion 2D classification

   (b) Xmipp3 CL2D (hierarchical clustering)

12. **Classify in 3D** using Relion and taking as input the subset created previously.

13. **Refine angular assignment** 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 than 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 reduce the computational load, these data were downsampled by a factor of 2.

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

\begin{verbatim}
wget ftp://ftp.mrc-lmb.cam.ac.uk/pub/scheres/relion13_tutorial.tar.gz
tar -xzf relion13_tutorial.tar.gz
\end{verbatim}

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

\begin{verbatim}
scipion tutorial betagal
\end{verbatim}

The project windows should appear as shown in Figure 1. The previous command has done two steps: (1) create a new project, (2) import an existing workflow template. In 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 experimental users.

In the project window, the left panel displays a tree with the processing tasks (protocols) that can be used. The protocols shown can be filtered by perspective (SPA is the default one) or found by `Ctrl`+`F`. The top right panel displays the sequence of protocols executed (runs) by the user and its state: 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.
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, CTF-Model, 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 will be properly propagated.

**NOTE** Since important information is provided during the *import* step, it is recommended to take your time to check that all provided 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 file locations. 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. 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 the movie files, double-click the `import movies` box. The protocol form will be open to fill the parameters 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 from the downloaded Relion 1.3 data set. You can either use the `Browse` icon to select the path or type in the entry field.

After selecting the path to the movies, we can press the `Execute` button. This operation should be very fast since it only searches for the files inside the path that matches 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 in any entry and select Open to visualize all the frames in that movie.
Figure 2: Import Movies protocol. In this case, only the path needs to be filled.
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 the BIM image blurring and restore important high resolution information. In SCIPION we have developed a protocol that combines both the global and local alignment to produce the final averaged micrograph.

The alignment method used in [Li et al., 2013] consists of a pure in-plane drift correction in which a step of the sub-frame translational alignment is introduced by dividing each frame into a number of sub-frames. This approach is fast if running on GPUs, and at the end, an “average” micrograph is generated for each movie via the summation of all corrected frames. The method is certainly appropriate for global sample movements but is not the best option if the sample motion is local.
The alignment method proposed in (Abrishami et al., 2015) is based on Optical Flow (OF), works best at a local level and 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. Still, even for those latter cases, we have found it advantageous to use the (Li et al., 2013) method combined with OF, to obtain an additional level of refinement and an intuitive graphical representation of the total BIM pattern.

In our workflow, we can now open the movie alignment box (Figure 4) and, since the parameters are filled, we can execute the protocol. In a general case, we can use the Search icon to select the SetOfMovies to be used as input (the options to select are the objects of this type registered, no more file selection at this point). In this tutorial we have chosen to use optical flow only to be able to use with/without GPU. If the latter is available, then we can try combining both dosefgpu + optical flow. We can also select to exclude some frames from the alignment process.

When finished, we can again click on the Analyze Results button to see the list of the resulting micrographs (Figure 5). The first column should be a composite image with half of the PSD of the unaligned micrograph (left side) and half of the PSD of the aligned one (right side). The plot reflect the accumulated shifts of 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 to further processing steps.
Figure 4: Movie Alignment Protocol.

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 in EMX format and Xmipp 3 metadata. In these cases, CTF information can also be 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 preprocess - micrographs which combines multiple Xmipp 3 programs in order to perform different operations over 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 applying a filtering operation. In the protocol form we can select the input micrographs and the operations to perform and execute, as usual. The result should be a new SetOfMicrographs with the transformations applied.
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 CTFFind (Mindell and Grigorieff 2003) or Xmipp CTF estimation (cite). 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 plenty of images with the desired window size. After that, they compute the Fourier Transform of each image and make an average.

We have developed the protocols for CTFFind and Xmipp in such a way that the 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 PSD are contained within those frequencies. There is a wizard, shown in Figure 8, that helps in choosing those frequencies. To see the full 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 ten times faster than its predecessor).

Figure 7: CTFFind 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.](image)

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 press **Disable**. Once you finish the selection, press on the **Micrographs** button to create a 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 not 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** on 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 the Recalculate CTFs button.

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

Figure 10: Protocol CTFFind output that shows the estimated CTFs for all micrographs.
Figure 11: Protocol CTFFind output that shows the estimated CTFs for all micrographs.

Figure 12: Protocol CTFFind output that shows the estimated CTFs for all micrographs.
4 Particle Picking

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

4.1 Using Eman 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 from the CTF estimation, order the micrographs by defocus and select one at the top, one at the middle and one at the end. We can register this new subset and put a meaningful name that can be easily remembered. This small set will be used later during Relion picking.

Eman boxer has manual picking and several modes of automatic picking, with fast outcome. Please refer to its webpage for further information.

We need to open the eman - boxer box and executed it, after that we should see the boxer GUI as in Figure. For this tutorial we will use a box size of 64 pixels and the Swarm tool.
After we finished with picking particles with one micrograph, we can move to the next one and pick all of this small set. At the end, we can click in the [Done] button in the Boxer GUI and then a dialog should appear asking to register the output 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) completed automatic picking. For the manual supervised picking, we open the Xmipp picking GUI (Figure 14) contains a control panel with the list of micrographs and some other parameters. The micrograph where we are picking is displayed in a separated window and we can apply a number of filters/enhancements (like Gaussian blurring, Invert contrast, adjust histogram) just to improve the visualization of particles and its selection. Following is a summary of the control actions:

- Use $[\text{Ctrl}]+\text{mouse wheel}$ in the overview window to zoom in and out.
- Mark particles with the $\text{mouse left}$ button. You may move its position by clicking the left mouse button on the selected particle and dragging it to the new position.
- Use $[\text{Ctrl}]+\text{left mouse}$ over 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 $\text{Filters}$ as many filters as you like.
In the manual/supervised step, we should start picking manually a few micrographs and then mark the Autopick checkbox. At this point, it will train a classifier based in machine learning and will propose some coordinates automatically. You can ”correct” the classifier by adding missing particles or removing wrongly picked. 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 the previous execution of manual/supervised as input. When executing, this will pick the rest of micrographs completely automatic. At the end, we can review the picking coordinates and we still have the chance to add/remove particles.

**NOTE** Once we have trained the classifier in the manual/supervised step, we can even pick another set of micrographs of new collected data in an automatic fashion.
4.3 Using Sparx Gaussian

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

To use this picking, we can open another Eman boxer run and then select the Gaussian tool. 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 in a similar way of the Sparx gaussian. This picking tool have no GUI, so we should try with some parameters and later visualize the picked coordinates using the Xmipp GUI.

4.5 Using Relion auto-picking

The protocols for Relion auto-picking in SCIPION have been divided in two steps: (1) computing the figure-of-merit (FOM) maps and (2) Adjusting the parameter 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 section and produce some classes and 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(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 but now reading the previous FOM maps in order to speed up the computation. On this way, adjusting the parameters becomes more interactive. After we are happy with the results for this few micrographs, we can then launch the auto-picking for the full set of micrographs in a completed 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 any set of coordinates, we can proceed to extract these particles 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 phase of your picked particles, among other things. The options are summarized below:

- The *coordinates* of the particles in the micrographs, which are taken from the results of the previous step. Also in the same tab, 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.

- The *phase flipping* flag. If activated, the protocol corrects the CTF phase 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.](image-url)
In this case, we have choose 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 expect the particles on the other way, black over white. We have also choose 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 tell the protocol to sort the 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 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 will also be plot. If you want to remove some particles, eg: because they are outliers, you can use [right click] [disable]. To create a new set of particles you can click on [Particles] red button.

![Image](image.png)

Figure 16: List of particles after extraction, sorted by z-score.
NOTE The GUI to visualize particles can be used with other protocols to create subsets based on some quality parameter. Particles can be easily ordered by any column and then make selection base 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 of your data and 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 recomendations in the SPIDER web pages [MDA] and [RCT].
There are essentially only four steps:

- Low-pass filtration
- Alignment in two dimensions
- Dimension-reduction – expression of a mxn 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 will need to have been aligned. The alignment step here is optional if the images have been aligned already.

The dimension-reduction step is even optional, in theory. In principle, one could classify the raw images. As an example here, I’m 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, all the four steps are done using SPIDER scripts while in Scipion, 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 guide us to execute the steps in the correct order.

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 from picking just a few micrographs. Figure 18 shows the form for the first step: filter the particles. You can select any type of filter and preview the operation in the wizard window as shown in 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 the input particles.
The filtered particles will be used to the reference-free 2D alignment step. At this point, we will try to find the alignment parameters (x and y shifts and rotational angle) of each image against a common reference. Two Spider protocols are available for this task: `spider - align apsr`, `spider - align pairwise`, or can be used any other protocol that align 2D images (like `xmipp - align with cl2d`). We are going to use here the `xmipp - align with cl2d` protocol, but you can try with any of the others. Figure 21 shows the parameters form where we mainly need to select the input particles that in our case are the result from the filter operation. In this protocol 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 but now aligned and also the average image, similar to the one shown in figure 22.
Before going to dimension-reduction, we can apply a mask to the aligned particles. Masking off areas outside the particle ensures that only those areas in which structural information resides will participate in the analysis. Exclusion of peripheral noise increases the SNR and makes the eigenvalue spectrum more compact. In other words, the masking assures that existing variations within the molecule set are represented by the smallest number of factors. Another use 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 tailed 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. For more details about creating a mask see http://scipion.cnb.csic.es/bin/view/TWiki/Create2DMask

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

After executing the protocol, the Analyze results will show up a form for visualization as illustrated in Figure 23, where the first option will display the computed eigenfactors. It is also possible to display the eigenfactors histogram or the factor maps selected by pairs (24).
At the classification step we will attempt to separate the images into homogeneous subclasses. There is a tradeoff between homogeneity of the subclasses and size of the subclasses (which is related to the improvement of signal-to-noise in the class averages). In other words, if too few classes are used, dissimilar classes will be grouped together, and 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 options 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 believe to represent true structural differences. Use of weaker factors will probably not make a appreciable difference, however.

There are three classification options given here: Ward (recommended), K-means and Diday. For hints about the usage of the different methods, see the MDA web page. 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 the clustering dendrogram cut a any desired height. In the case of Ward, we can also display the class averages tree from which is possible to select classes or particles. (See Figure 26

Figure 25
5.2 Xmipp CL2D

Another classification algorithm is \texttt{xmipp - cl2d} (Clustering 2D) that computes many classes in a very flexible way (Sorzano et al., 2010). As similarity metric you can choose whether to use correlation or correntropy (Sorzano et al., 2010). As a rule of thumb, correlation is well suited to images in which the noise is Gaussian (there are no neighbouring particles, black spots, ...), while correntropy is more robust to non-Gaussianity violations. You can also choose classical (an image is assigned to the cluster with maximum similarity) or robust clustering criterion (an image is assigned to the cluster where it is better than its neighbours. By default, correlation and classical are chosen (show “Advanced options” to change these values). The algorithm progresses in a hierarchical way, that is, 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 till the desired number of classes is reached. Depending on the image quality, we recommend to
calculate a number of classes so that in average each class has between 50 and 200 particles (classes defined with fewer particles are too noisy and may be unstable; classes defined with more particles may not be homogeneous enough simply because it is composed of too many particles that might be different).

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

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 some 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 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.
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 times 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 set and clean up before going to 3D refinement. The last case is even more CPU intensive and is
recommended to be launched in a cluster or in a multi-core computer.

The Figure 29 shows the form to launch the Relion 2D classification job. In this case we are using the whole dataset and using 16 classes. The radius of the particles can be selected using the wizard shown in Figure 30.

Figure 29: Relion parameters form.

On the CTF tab, set:
• **Phase-flip**: In Scipion the phase-flip is a property of the particles and then is not needed to be selected here.

• **Do CTF-correction?**: Yes, which will perform full phase+amplitude correction inside the Bayesian framework.

• **Ignore CTFs until first peak?**: No. This option is only occasionally useful, when amplitude correction gives spuriously strong low-resolution components.

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)](Scheres2012). 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 remain too-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 could have been set to 10-15A anyway.

The defaults on the Sampling tab are usually used. Five degrees angular sampling is enough for most projects, although some large icosahedral viruses may benefit from finer angular samplings. 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 iteration (by setting Number of iterations: 30 on 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.
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.

![Relion parameters form](image)

Figure 31: Relion parameters form.

The following options are available:

- **Show classification in Scipion**: This option will show the classification (classes and images assigned) of last iteration. This option will generate a SetOfClasses (in Scipion) by converting the last `_data.star` file from Relion. This option may take some minutes depending of the dataset size. By default the classification is displayed in gallery model with the 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 are low-resolution, while classes with many particles are high-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 right-click over other iteration Relion files and open 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

Figure 33: Relion model.star file shown in table mode.
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 as possible. To achieve this, your data should have sufficient homogeneity, i.e. your data may 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 from the downloaded Relion 1.3 data set. Also,
you must provide the sampling rate 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, as initial model (EMAN), ransac (Xmipp) or reconstruct significant (Xmipp).

Figure 35 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 output of Relion 2D classification. The 3D classification is very CPU intensive (even more than in 2D) and it is recommended to be launched in a cluster or a multi-core computer with sufficient RAM memory.

![Figure 35: Relion parameters form.](image)

The options in this protocol are mostly the same as 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, which will perform full phase+amplitude correction inside the Bayesian framework.

- **Has reference been CTF-corrected?**: No, which mean 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 using Wiener filtering inside RELION or from a PDB.

- **Phase-flip**: In Scipion the phase-flip is a property of the particles and then is not needed to be selected 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 ctfs?**: 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 remain too-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 radius of the mask for the experimental images will be applied. If give the mask, must have the same dimensions as the reference, and 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 (Å):** -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 could have been set to 10-15Å anyway.

The defaults on the Sampling tab are usually used. The 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 may benefit from 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 on the Optimisation tab) with a sampling of say 1.8 degrees.

The total number of requested CPUs is 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 36 allows to access the different visualization options. By default, the last iteration is displayed, but you can select any previous one. Also, as Relion 2D classification, you can view intermediates results while the job is running or when it finishes.
The following options are available:

- **Show classification in Scipion**: This option will show the classification (classes and images assigned) of last, or selected, iteration(s). This option will generate a SetOfClasses (in Scipion) by converting the _data.star file(s) from Relion. This option may take some minutes depending on the dataset size. By default the classification is displayed in table model with the 3D class models (as shown in Figure 37) and sorted in reverse order of the number of particles assigned to each class, as for the Relion 2D classification. As before, smaller classes will be low-pass filtered more strongly than large classes. 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)**: The the *_model.star file are basically the same as for the 2D classification run. By default the 3D class are rendered in reverse order of the rlnClassDistribution value, which is equivalent to the
number of particles assigned to each class. As before, smaller classes will be
low-pass filtered more strongly than large classes, and the spectral signal-to-
oise ratios are stored in the data_model_class_N table.

- **Show *optimiser.star file**: Display the *optimiser.star file that contains
some general information about the process. From this view, you can easily
right-click over other iteration Relion files and open 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 ad-
ditional information about the optimal orientations, the optimal class assign-
ment, the contribution to the log-likelihood, etc. The *sampling.star contains
information about the employed sampling rates.

![Figure 37: Relion classes displayed as Scipion classification](image)

- **3D Class to visualize**: You may select show all 3D classes or a selection.
- **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 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 sufficient homogeneity has been selected, one may use 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 completely avoided. Combined with a novel procedure to estimate the accuracy of the angular assignments, it 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 only needs to run it once, as there are hardly any parameters to optimize.

The input parameters shown in [38] largely remain the same as for the 3D classification protocol, although some of them are no longer available.

![Relion 3D auto-refine form.](image)

Figure 38: Relion 3D auto-refine form.

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

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

The analyse Result Form is basically the same as Relion 3D classification Analyse Result, as shown in Figure 39, although has in addition a FSC plot.

![Figure 39: Analyse Result of Relion 3D auto-refine.](image)

The following options are available:

- **Particles angular assignment**: This option will show the SetOfParticles in table mode showing its Matrix Transformations (it is the standard representation of the angular assignment in Scipion).
• **Show *_.optimiser.star file**: Already explained above in Relion 3D classification protocol.

• **Volume to visualize**: You may select which 3D map to show among half1, half2 or final.

• **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 2D plot or using Chimera with red spheres.

• **Display SSNR plots**: Display signal to noise ratio plots.

• **Display resolution plots (FSC)**: Display fourier shell correlation plots.

### References


