## On-the-fly data processing with SCIPION 2.0



## CryoEM EMBO course Birkbeck Sept 2019

# Contents

Getting started							
1 Preprocessing							
	1.1	Creating a Project, importing movies	2				
	1.2	Beam-induced motion correction	5				
	1.3	Estimating CTF	5				
	1.4	Particle picking	7				
	1.5	Extracting particles	8				
<b>2</b>	$2\mathrm{D}$	classification and Initial volume	9				
	2.1	Reference-free 2D class averaging	9				
	2.2	Analyzing 2D results and creating subsets	11				
	2.3	De novo 3D model generation	12				
	2.4	Analyzing results, symmetrization	13				
3	3D	refinement	14				
	3.1	Running 3D auto-refine	14				
	3.2	Exercise: Improving obtained resolution	15				
	3.3	Exercise: Fitting an atomic structure	15				

## Getting started

### Software requirements

To follow this practical you will need to have SCIPION (de la Rosa-Trevín et al., 2016) properly installed on your system (version 2.0 or greater). During this tutorial we are going to use several CryoEM programs such as: Relion 3 (Kimanius et al., 2016), motioncor2 (Zheng et al., 2017), CTFFind4 (Rohou and Grigorieff, 2015) and Xmipp3 (de la Rosa-Trevín et al., 2013).

## Test data

In this tutorial we will use a subset of a dataset acquired by Julian Conrad at Swedish Cryo-EM Facility at SciLifeLab, Stockholm. The dataset was collected in a Titan Krios using Volta Phase Plate. The sample is Rubisco (from A. Thaliana), purified by Michael Hall, also microscopist at the Swedish Cryo-EM Facility, at the Umeå node.

During this document, we will refer to the path where the tutorial data is as **\$DATA/**. If you are running this tutorial in the EMBO course at Birkbeck 2019, then **\$DATA/=/i/embo2019/d/scipion/**.

## 1 Preprocessing

### 1.1 Creating a Project, importing movies

SCIPION will help you to get your processing organized by projects. SCIPION can be launched anywhere and then you can select the project you want to work on. Let's start by creating a new project by launching the following command from a terminal:

#### > scipion

After that, the window with all projects will be shown (it can be an empty list if there are no projects for this user). You can then click on Create Project button and enter the project name, for example: rubisco\_embo19 (as shown in Fig. 1). After clicking on Create, a new window will appear showing your new empty project. Don't worry, in the next step we will import a workflow template that will guide you through the processing pipeline of this practical.



Figure 1: Create Project dialog.

#### TIP

**\$HOME/ScipionUserData/projects** is the default location to store projects data, but this can be changed in a configuration file. Moreover, each project can be located in a different folder and in this case a link is created in **\$HOME/ScipionUserData/projects**. Projects are self-contained folders that can be moved from one computer to another. You can jump into the last opened project with command scipion project last



Figure 2: Project main GUI. Left panel displays a number of drop-down lists with processing tasks (protocols) that can be used (all protocols can be searched by using Ctrl + F and the search dialog). The top right panel displays the tree/sequence of protocols executed (runs) by user and their state: saved, running, finished or aborted. Bottom right panel displays information for the selected run, such as inputs and outputs, execution logs or documentation.

From the project GUI we can import a workflow template that will load all the steps and parameters that will guide us through this practical. To do that, you must go to the main menu: **Project Import workflow** and use the browser dialog to search for the workflow file. In our case, it should be **\$DATA/workflow-rubisco-embo19.json**. After selecting the file, click on the **Import** button and your project will have the workflow loaded, as shown in Figure 2.

#### TIP

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 SCIPION the *import* is almost the only place where the user needs to deal directly with files. In SCIPION each protocol has 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 pixel size). This information will not be requested later and should be properly propagated from one protocol to another.

To import movie files, double-click the import movies box in the workflow. 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 directory with movies (it can be found in **\$DATA/Movies**. You can either use the **Browse** icon to select the path or type it directly in the entry field. For this tutorial, the provided input movies has been already gain corrected, so we will let the related entries empty. For other cases, you can provide the gain and dark images as well, which will be propagated to other protocols that might need this information.

After selecting the path, we can press the **Execute** button and the box for this protocol should become yellow (running state) and then green (finished). Then, the summary tab will display some information such as the number of movies imported and the number of frames.

#### $\mathbf{TIP}$

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 into the project to avoid data duplication. Instead, soft links are created pointing to the files location.

### 1.2 Beam-induced motion correction

Aligning the individual frames of movies is necessary to correct for beam-induced image blurring and restore important high resolution information. In this practical we will use motioncor2 (Zheng et al., 2017) for movie alignment. For that, we just need to open the corresponding box and execute it, since all the parameters have been preloaded.

In general, one of the most important parameters at this step is to choose the frame range to use for aligning the frames and producing the final average micrograph. Additionally, motion correction protocols allow to apply dose weighting to each frame taking into account the accumulated dose. Another common option is the number of patches, in case that the protocol also supports local alignment (in this case we will use defaults  $5 \ge 5$  patches). Take you time to study the protocol form and use the ? button near any parameter to know what it means.

After launching the movie alignment protocol, we can go further and start the CTF estimation. Thanks to the stream processing capability in SCIPION, we don't need to wait until the first protocol finishes to start the second one in this case. Usually the motion correction step is the bottleneck for on-the-fly data processing.

#### ALTERNATIVES

- relion motioncor
- xmipp Optical Flow alignment
- grigorieff lab Unblur

#### 1.3 Estimating CTF

The next step is to estimate the CTFs (Contrast Transfer Functions) of the micrographs using *Gctf* (Zhang, 2016). This protocol estimates the PSD (Power Spectral Density) of the micrographs and the parameters of the CTF (defocus U, defocus V, defocus angle, etc.).

To estimate the CTF you will need to select the frequency region to be analyzed. The limiting frequencies must be such that all zeros of the CTF are contained within those frequencies. There is a wizard that helps to choose those frequencies. To see all available options, choose the *Advanced* expert level and click on the ? button for any specific parameter.

Like for every protocol, check the various entries and ask yourself what they mean. Over different datasets collected we realised that a CTF window for estimation of 1024 works better than any smaller one. Notice that this dataset was collected using a Volta Phase Plate (VPP). We do not suggest to use a VPP unless absolutely necessary (e.g. to visualise your particle). In this case we used it only to check the behaviour of our VPP. The phase shift has to be calculated. This is only a subset and images have a shift in between 0.2 pi (36°) and 0.8 pi (144°). The ideal shift for phase contrast is 0.5 pi (90°), but with the VPP we cannot exactly get this value and maintain it.

The CTFs of good micrographs typically have multiple concentric rings, 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) or rings that fade in a particular direction (drift). The output from any CTF estimation protocol is shown upon clicking on the Analyze Results button (Figure 3). To discard micrographs with bad CTF 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 good micrographs.



Figure 3: Results visualization after CTF estimation.

### ALTERNATIVES

• ctffind4

### 1.4 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 there are several integrated picking tools, allowing users to select the one which best fits their needs.

Here we will use Xmipp particle picking, that 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 be relaunched at any time. For this sample, we should set the box size of 200 pixels in the picking GUI.

In the manual/supervised step, we should start picking manually a few micrographs and then click the Activate Training button. It is recommended to pick manually in micrographs with junk, so the algorithm will "learn" to separate good particles from background/ice/aggregation. If the micrographs contains many particles, one can pick on a rectangular region that contains good and bad samples. Once the we are in the "Training mode", we can "correct" the classifier by adding missing particles or removing wrongly picked ones. We move to the following micrograph and we correct over proposed particles. After training with a few more micrographs, we can register the output coordinates by clicking on the Coordinates red button.

Then we can 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 picked coordinates and still have a chance to add/remove particles.

#### ALTERNATIVES

- Relion LoG / autopicking
- EMAN2 boxer / SPARX gaussian
- Gautomatch
- crYOLO

### **1.5** Extracting particles

Once we have a set of coordinates, we can proceed to particle extraction with Relion. We need to be careful with the options selected here, since this will affect later steps. The extract protocol (Figure 4) will allow us to extract, normalize and scale the picked particles, among other things.

R	Protocol Run: ProtRelionExtractParticles	~ ^ ⊗	R	Proto	col Run: ProtRelionExtractParticles	~ ^ 😣
RELION	Protocol: relion - particles extraction	saved CrCite BHelp	RELION	Protocol: relion - partic	eles extraction	saved Cite OHelp
Run			Run			
Run name	relion - particles extraction binne 🖋 Comment we extract binned particles in ord	1	Run name re	elion - particles extraction binne	<ul> <li>Comment we extract binned particles in ord</li> </ul>	8
Run mode	Continue Restart      Host localhost		Run mode	Continue      Restart	Host localhost	
Parallel	MPI 8 Ø Use queue? • Yes O No	0	Parallel	MPI 8	Use queue? · Yes · No	0
	Wait for	0			Wait for	0
Input Prepr	ocess Streaming		Input Preprocess Streaming			
Input			Preprocess			
	Input coordinates xmipp3 - auto-picking (step 2).outputCoordinates	Q 🗎 👁 😡		Invert contrast?	? ⊛ Yes ⊜ No	0
	Micrographs source Same as picking Souther	Ø		Normalize particles?	? ● Yes ◯ No	0
Input micrographs motioncorr - movie alignment (copy).outputMicrographsDoseWeighted Q 🗎 👁 🕑				ground circle before scaling (px	) -1	0
	CTF estimation gctf - ctf estimation.outputCTF	Q 🗎 😡		Stddev for white dust removal:	-1.0	0
	Particle box size (px) 300	0		Stddev for black dust removal:	-1.0	0
	Rescale particles?   • Yes   No	0				1
	Re-scaled size (px) 100	Ø				
	🗙 Close 🖺 Save	e Execute			🗶 Close 🖺 Sav	ve 👯 Execute

Figure 4: 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 classification and it expects particles to be white over black. We have choose a box size of 300 pixels and will re-scale the images to 100 to speed up further computations.

#### TIP

At any time, coordinates of selected particles can be extracted using the scipion - extract coordinates protocol. Moreover, the protocol scipion - assign alignment can also be used in order to save coordinates and apply alignment parameters previously calculated.

## 2 2D classification and Initial volume

### 2.1 Reference-free 2D class averaging

The reference-free 2D class averaging in Relion is a great tool to throw away bad particles. 2D classification is also a quite useful way to check that different views of the molecule are present in the dataset during on-the-fly data processing. 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. Open the relion - 2D classification box check the parameters.

Relion 2D classification protocol form has several tabs that you need to go through before launching the job.

In the CTF tab, set:

- **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 classes: 100
- 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).
- 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 10-15A.

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. 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. Relion3 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 Relion3 tutorial.

### 2.2 Analyzing 2D results and creating subsets

It is possible to analyze the results while the job is still running or when it finishes, click on the Analyze Results button. By default, the last iteration is displayed, but you can select any previous one(s).

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 and sorted in reverse order of the number of particles assigned to each class. From this view you can easily group particles from different classes and create a new set of particles to be used in further steps.
- 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.
- Show \*\_optimiser.star file: Display the \*\_optimiser.star file that contains some general information about the refinement process. From this view, you



can easily open Relion star files in a new window.

Figure 5: Output classes displayed as Scipion classification. From this view good classes can be selected to create a subset of particles assigned to them. This action is registered in SCIPION as a 'user interaction' and stored in the pipeline as another run (boxes in the chart) that is connected to the classification run.

When the 2D classification job finishes, display the resulting 2D classes and select the subset of particles from the GUI by clicking in the Particles button. This will create a new subset of particles from the selected classes. This subset will be used in the next step of generating the 3D initial model. (Note: the box in the subset selection in the template is just for reference and should not be executed).

#### 2.3De novo 3D model generation

Relion has implemented a Stochastic Gradient Descent (SGD) algorithm to generate a 3D initial model de novo from the 2D particles. From version 3.0, this implementation very closely follows the approach of the cryoSPARC (Punjani et al., 2017). Provided you have a reasonable distribution of viewing directions, and your data were good enough to yield detailed class averages in 2D classification , this algorithm is very likely to yield a suitable, low-resolution model that can subsequently be used for 3D classification or 3D auto-refinement .

We can execute this box by opening 3D Initial volume relion - 3D initial model. Following is a summary of selected parameters for this case:

In the *Input* tab:

- **Input particles**: Select the resulting subset of particles after 2D classification results
- Particle mask diameter: 220

In the *Optimisation* tab:

• Number of classes: 1

Just one in this case to speed-up computing, sometimes can be useful to specify more than one)

• Symmetry: c1

If you don't know what symmetry is, it is best to start with a C1 reconstruction.

Usually, there is not need to change the default values in the SGD tab, but for getting result faster, we will reduce the number of iterations:

- Number of initial iterations: 25
- Number of in-between iterations: 100
- Number of initial iterations: 25

On the *Compute* tab, optimise things for your system. You may well be able to pre-read the few thousand particles into RAM again. GPU acceleration will also yield speedups, though multiple maximisation steps during each iteration will slow things down compared to standard 2D or 3D refinements or classifications.

#### 2.4 Analyzing results, symmetrization

In SCIPION, you can click on the Analyze Results button and check how the generate volume looks like. You can display it by slices or in UCSF Chimera. Many other options are available similarly to the 3D classification or refinement jobs.

If looking at the volume you recognise additional point group symmetry at this point, then you will need to align the symmetry axes with the main X,Y,Z axes of the coordinate system, according to Relion's conventions. Relion 3.0 contains a new program to facilitate this, that we have conveniently wrap into the relion - symmetrize volume protocol (internally using the programs: relion\_align\_symmetry and relion\_image\_handler). Open that protocol and provide the following parameters:

- Input volume: Select the volume generated in the initial model job
- Symmetry: d4

After this job, you can visualize the resulting volumes as usual by slices or in Chimera.

## 3 3D refinement

### 3.1 Running 3D auto-refine

After having an initial volume, we could either go into 3D classification or refinement. Since this tutorial data does not contain 3D heterogeneity, we will jump directly into 3D refinement. In other cases, it might help to do 3D classification before to clean more the good set of particles that could go to high resolution.

- Input particles: Particles output from the initial volume job.
- Particle mask diameter: 220
- Input volume(s): The volume that after the symmetrization job.
- Symmetry: d4
- Initial low-pass filter: 30

We typically start auto-refinements from low-pass filtered maps to prevent bias towards high-frequency components in the map, and to maintain the "goldstandard" of completely independent refinements at resolutions higher than the initial one.

### 3.2 Exercise: Improving obtained resolution

At this point, if everything went well with the **relion - 3D autorefine** job, we should have a refined map between 6 and 7 Å. Is this the best resolution that we could achieve with this data? What processing strategy could we following in order to obtain a higher resolution map?



- scipion extract coordinates
- relion extract particles
- relion autorefine
- relion create 3D mask
- relion postprocess

### 3.3 Exercise: Fitting an atomic structure

It is possible to use the pdb 5iu0 crystal structure and fit it into your final map. You will first need to save a single polypeptide chain, fit it into your EM map and then symmetrise.

## References

- de la Rosa-Trevín, J., Otón, J., Marabini, R., Zaldívar, A., Vargas, J., Carazo, J., and Sorzano, C. (2013). Xmipp 3.0: An improved software suite for image processing in electron microscopy. J. Struc. Biol., 184(2):321 – 328.
- de la Rosa-Trevín, J., Quintana, A., del Cano, L., Zaldívar, A., Foche, I., Gutiérrez, J., Gómez-Blanco, J., Burguet-Castell, J., Cuenca-Alba, J., Abrishami, V., Vargas, J., Otón, J., Sharov, G., Vilas, J., Navas, J., Conesa, P., Kazemi, M., Marabini, R., Sorzano, C., and Carazo, J. (2016). Scipion: A software framework toward integration, reproducibility and validation in 3d electron microscopy. J. Struc. Biol., 195(1):93 99.

- Kimanius, D., Forsberg, B., Scheres, S. H. W., and Lindahl, E. (2016). Accelerated cryo-em structure determination with parallelisation using gpus in RELION-2. *eLife*, 5:e18722.
- Punjani, A., Rubinstein, J. L., Fleet, D. J., and Brubaker, M. A. (2017). cryosparc: algorithms for rapid unsupervised cryo-em structure determination. *Nature Meth*ods, 14:290. Article.
- Rohou, A. and Grigorieff, N. (2015). CTFFIND4: Fast and accurate defocus estimation from electron micrographs. J. Struc. Biol., 192(2):216–221.
- Scheres, S. H. W. (2012). RELION: Implementation of a bayesian approach to cryo-EM structure determination. J. Struc. Biol., 180(3):519 – 530.
- Zhang, K. (2016). Gctf: Real-time CTF determination and correction. J. Struc. Biol., 193(1):1–12.
- Zheng, S., Palovcak, E., Armache, J.-P., Verba, K., Cheng, Y., and Agard, D. (2017). Motioncor2: anisotropic correction of beam-induced motion for improved cryoelectron microscopy. *Nature methods*, 14(4):331–332.