Single Particle Analysis using Relion / CCP-EM Doppio

Introduction

This tutorial will cover some aspects of the single particle analysis (SPA) processing pipeline with the aim of getting familiarity of using Relion 4.0 in the new CCP-EM Doppio GUI

- Getting started in Doppio
- Importing and preprocessing raw data
- Creating and optimizing a particle set
- De novo 3D model generation
- 3D classification
- 3D refinement
- Mask creation
- Post processing

The running parameters for this tutorial were tested on a system with a 1a RTX 4000 GPU and a 10 core CPU. These parameters may need to be adjusted based on your system. Additionally, some other job parameters are specific to your system. These are highlighted in yellow, and explanations are provided in the notes.

Job numbers are shown in the tutorial for various inputs e.g. **`AutoPick/job033/autopick.star`** please be aware that your job number may vary, for example if you try an extra step or have previous failed jobs. Be careful to select the correct job. You can use job aliases to help note key steps.

Installing software

This tutorial assumes Doppio and associated downstream programs are installed on your system. For info for setting this up see the Doppio user guide:

https://www.ccpem.ac.uk/docs/doppio/user_guide.html

The following programs need to be installed:

- Relion
- ctffind4
- Topaz

Topaz particle picking and Relion's 2D automated class selection program both need their own specific virtual environments to run in. The setup for this is detailed in the Doppio user guide.

Setting up a project

We will use a test data set on beta-galactosidase that was kindly given to us by Takayuki Kato from the Namba group at Osaka University, Japan. It was collected on a JEOL CRYO ARM 200 microscope. The full data set is also available at EMPIAR-10204.

The data can be downloaded and unpacked with the following commands in the terminal:

```
wget ftp://ftp.mrc-lmb.cam.ac.uk/pub/scheres/relion30_tutorial_data.tar

tar -xf relion30_tutorial_data.tar
```

This creates a directory **`relion30_tutorial`** with a directory called **`Movies`** that contains your test data.

Start Doppio

Doppio can be started as a standalone app or in a web browser.

Desktop application

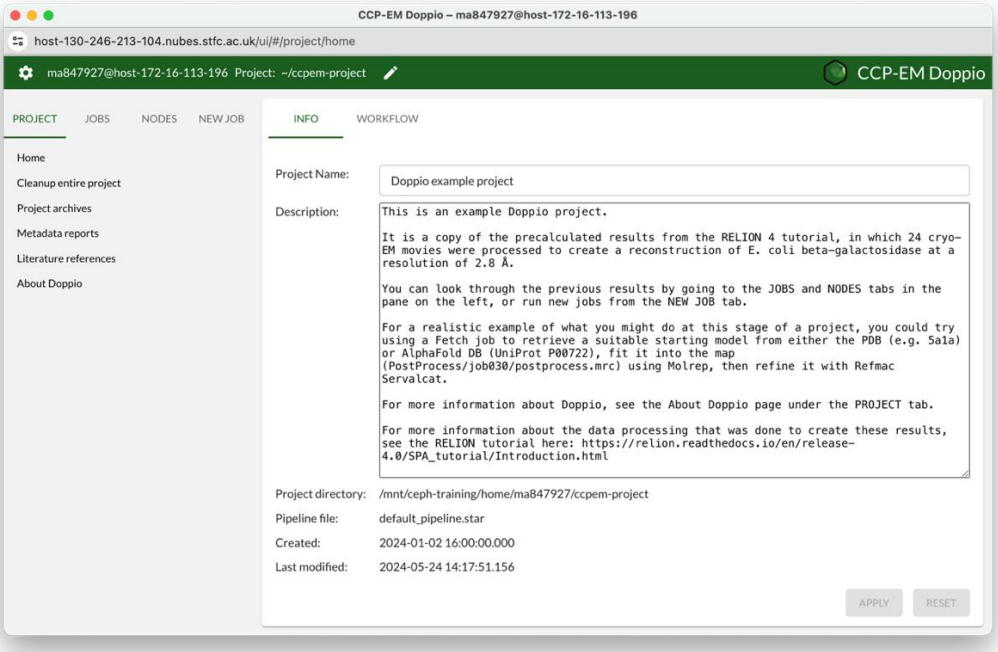
Run **`./doppio-desktop.AppImage`** to launch the desktop application

Web application

Navigate to the directory in which you extracted the application files in the previous section in the terminal

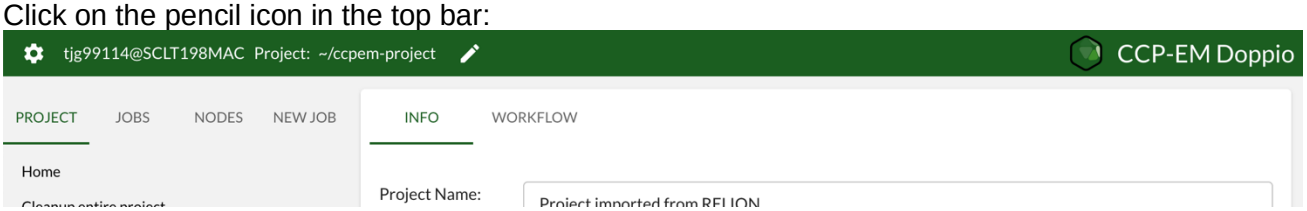
Run **`doppio-web/doppio-web`** to run the API and web application

The server will start and your web browser should open automatically to view Doppio

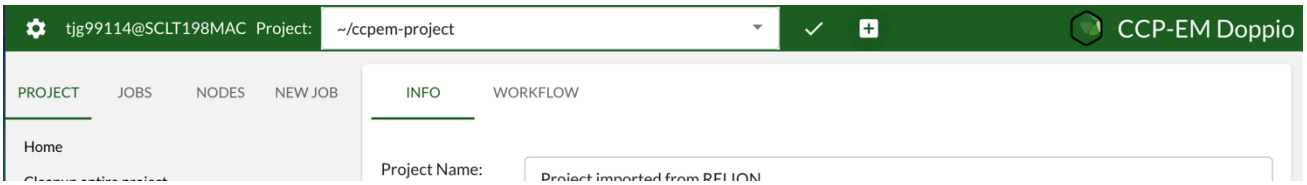


The program opens in a new blank project called ccpem-project. First let's create a project with the tutorial data.

Click on the pencil icon in the top bar:

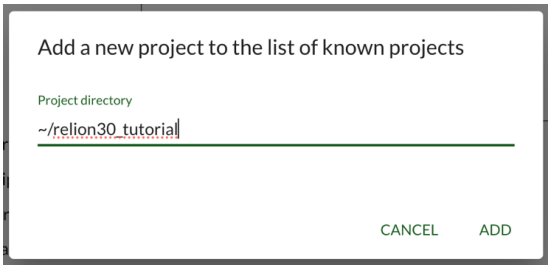


This will open project editing mode:



Click on the '**+**' icon to add a new project and type the path to the **`relion30_tutorial`** directory

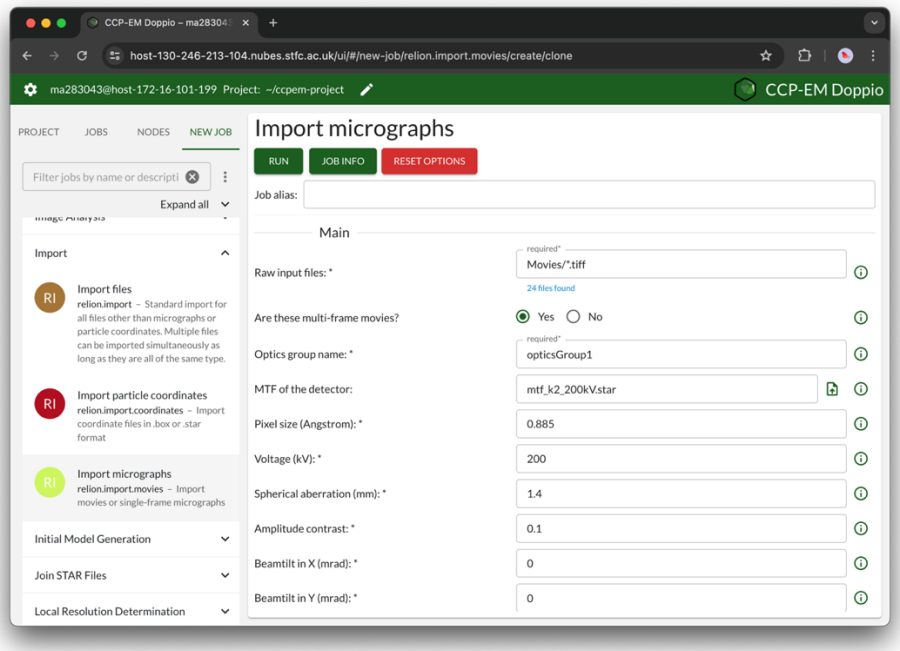
- The **project must use this name** as it needs access to the raw data
- If you are unsure of this path navigate into the **`relion30_tutorial`** directory in the terminal and use the command **`pwd`** to get it



The GUI will ask you to confirm creating a new project and then switch to your new project.

Importing the raw micrographs

The first step is to get the raw micrographs into the Doppio project. On the left panel select **NEW JOB.** Find the **Import** category and select an **Import Micrographs** job.



Update the following fields, parameters not listed can be left with the default values.

Raw input files: Movies/*.tiff
The GUI should show 24 movie files found

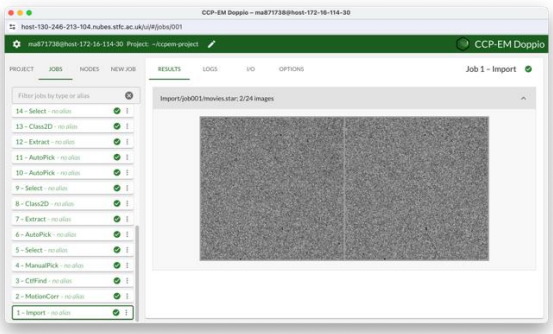
Are these multi-frame movies? Yes

MTF of the detector: <leave blank>

Pixel size (Angstrom): 0.885

Voltage (kV): 200

Spherical aberration (mm): 1.4
Press the **RUN** button to start the job. Instead let's look at the results of **Import/job001**. Click on it in the **JOBS** section of the left pane. Click on the **RESULTS** tab to show the results page:



This page is specifically tailored to each job type. For the **Import Micrographs** job it shows two sample micrograph movies, with all the frames summed. They don't look like much yet!

Pre-processing

The preprocessing steps correct beam-induced motion in the micrograph movies, create merged micrograph images, and calculates a contrast transfer function (CTF) for each micrograph.

Beam-induced motion correction

The Motion correction job implements RELION's own (CPU-based) implementation of the UCSF MOTIONCOR2 program for convenient whole-frame movie alignment.

In the **NEW JOB** panel go to the **Motion Correction** section and select a **RELION Motion Correction (RelionCorr)** job. Use the following running parameters, default values can be used for any not listed.

Input movies STAR file: Import/job001/movies.star

First frame for corrected sum: 1

Last frame for corrected sum: -1
A negative value here means to use all the frames

Write output in float16? Yes
This will save a factor of 2 in disk space compared to the default of writing in float32. Note that RELION and CCPEM will read float16 images, but other programs may not (yet) do so.

Bfactor: 150

Number of patches X: 5

Number of patches Y: 5

Group frames: 1

Save sum of power spectra? Yes

Sum power spectra every e/A2: 4

Binning factor: 1

Gain-reference image: Movies/gain.mrc

Gain rotation: No rotation (0)

Gain flip: No flipping (0)

Defect file: <leave blank>

Do dose-weighting? Yes

Save non-dose weighted as well? No
In some cases, non-dose-weighted micrographs give better CTF estimates. To save disk space, we're not using this option here as the data are very good anyway

Dose per frame (e/A2): 1.277

Pre-exposure (e/A2): 0

Running Options

Number of MPI procs: 1

Number of threads: 12
Hit Run and wait for the job to finish. It should take ~10 min to complete.

CTF Determination

Next, we will estimate the CTF parameters for each corrected micrograph. This job uses use Alexis Rohou and Niko Grigorieff's ctffind 4.1 to execute efficiently on the CPU.

Under **`Ctf Determination`** create a new **`RELION CTF Estimation (CTFFIND4)`** job:

`Input micrographs STAR file: MotionCorr/job002/corrected_micrographs.star`

`Use micrograph without dose-weighting? No`
These may have better Thon rings than the dose-weighted ones, but we decided in the previous step not to write these out

`Estimate phase shifts? No`
This is only useful for phase-plate data

`Amount of astigmatism (A): 100`
Assuming your microscope was reasonably well aligned, this value will be suitable for many data sets
`FFT box size (pix): 512`

`Minimum resolution (A): 30`

`Maximum resolution (A): 5`

`Minimum defocus value (A): 5000`

`Maximum defocus value (A): 50000`

`Defocus step size (A): 500`

`Estimate CTF on window size (pix): -1`

`Use power spectra from MotionCorr job? Yes`

`Running Options`

`Number of MPI procs: 6`

`MPI run command: ccpem-mpirun -n XXXmpinodesXXX`

Particle picking

This next set of steps will produce individual particle images from the micrographs. We will pick some particles from a small sample of the micrographs using a reference-free method. This particle set will then be cleaned up by 2D classification and used to train a model for the more advanced Topaz picker.

Get a subset of the micrographs

Under the **`Subset Selection`** category select the `Random` **`Sample from Starfile`** job and run it with the following parameters:

`Sample from micrographs file: CtfFind/job003/micrographs_ctf.star`

Number of samples: 5

LoG Picking

We will now use a template-free auto-picking procedure based on a Laplacian-of-Gaussian (LoG) filter to select an initial set of particles. These particles will then be used in a 2D classification job to generate 2D class averages

In **Automated Particle Picking** select a **RELION Autopick LoG (Single Particle)** job.

Input micrographs for autopick: Select/job004/selected_micrographs.star

Pixel size in micrographs (A): -1
The pixel size will be set automatically from the information in the input STAR file

Min. diameter for LoG filter (A): 150

Max. diameter for LoG filter (A): 180

Are the particles white? No

Maximum resolution to consider (A: 20

Adjust default threshold (stddev): 0

Upper threshold (stddev): 5

Write FOM maps? No

Read FOM maps? No

Running options

Number of MPI procs: 6

MPI run command: ccpem-mpirun -n XXXmpinodesXXX

Run the job and check the results. It should have picked ~250 particles per micrograph for ~1200 in total

Particle Extraction

Next, we must extract the individual particle images from the micrographs. They will be down sampled to speed up later processing steps.

In **Extract Particles** select a **RELION Extract Particles (Single Particle)** job.

micrograph STAR file: CtfFind/job003/micrographs_ctf.star

Input coordinates: AutoPick/job005/autopick.star

Extraction Options

Particle box size (pix): 256

Invert contrast? Yes

Normalize particles? Yes

Diameter background circle (pix): 200

Stddev for white dust removal: -1

Stddev for black dust removal: -1

Rescale particles? Yes

Re-scaled size (pixels): 64

Write output in float16? Yes

Running options

Number of MPI procs: 6

MPI run command: ccpem-mpirun -n XXXmpinodesXXX

2D Classification

We almost always use reference-free 2D class averaging to throw away bad particles. Because bad particles do not average well together, they often go to relatively small classes that yield ugly 2D class averages. Throwing those away then becomes an efficient way of cleaning up your data.

Under **2D Classification** select a new **RELION Class2D (EM, Single Particle)** job.

Input images STAR file: Extract/job006/particles.star

Number of iterations: 25

Do CTF-correction? Yes
We will perform full phase+amplitude correction inside the Bayesian framework

Ignore CTFs until first peak? No
This option is occasionally useful, when amplitude correction gives spuriously strong low-resolution components, and all particles get classified together in very few, fuzzy classes.

Number of classes: 50
For cryo-EM data we like to use on average at least approximately 100 particles per class. However, with this small number of particles, we have observed a better separation into different classes by relaxing these numbers. Possibly, always having a minimum of 50 classes is not a bad idea.

Regularisation parameter T: 2
For the exact definition of T, please refer to Scheres, JMB, 2012. For cryo-EM 2D classification we typically use values of T=2-3, and for 3D classification values of 3-4. For negative stain sometimes slightly lower values are better. In general, if your class averages appear very 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 diameter (A): 200
This mask will be applied to all 2D class averages. It will also be used to remove solvent noise and neighbouring particles in the corner of the particle images. On one hand, you want to keep the diameter small, as too much noisy solvent and neighbouring particles may interfere with alignment. On the other hand, you want to make sure the diameter is larger than the longest dimension of your particles, as you do not want to clip off any signal from the class averages.

Mask individual 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. Difficult classifications, i.e. with very noisy data, often benefit from limiting the resolution.

Center class averages? Yes
This will re-center all class average images every iteration based on their center of mass. This is useful for their subsequent use in template-based auto-picking, but also for the automated 2D class average image selection in the next section

Perform image alignment? Yes

In-plane angular sampling: 6

Offset search range (pix): 5

Offset search step (pix): 1

Allow coarser sampling? No

Number of pooled particles: 30

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>

Running options

Number of MPI procs: 5

Number of threads: 6

MPI run command: ccpem-mpirun -n XXXmpinodesXXX

2D class selection

Option 1 – automated selection

Next, we will select the good 2D classes using an automated procedure based on a neural network that was trained on thousands of 2D class averages. This requires Relion's 2D class auto selection to

Under **Automated Class Selection** select a new **RELION Automatic 2D Class Selection** job

Select classes from optimiser.star: Class2D/job007/run_it025_optimiser.star

Minimum threshold for auto-selection: 0.24

Python executable: /opt/relion_2d_autoselect/bin/python
This must be set to the python installation in the specific virtual environment for Relion class 2D automated class selection, specific to your system. See the Doppio documentation for how to set this up or use option 2 below if it is unavailable.

Once the job has finished, examine the results. You can hover the mouse over the rejected classes to see their threshold scores. If you are not satisfied with the results, you can run the job again with a modified threshold.

Option 2 – Manual selection

Alternativly, classes can be selected manually. This option cannot be used if you are running doppio on a remote machine.

Under **Subset Selection** select a new **RELION subset selection (interactive)** job:

Select classes from optimiser.star: Class2D/job090/run_it007_optimiser.star

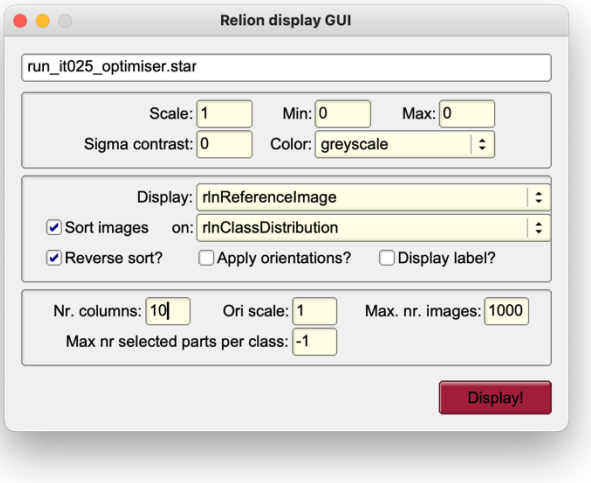
Re-center the class averages? Yes

Regroup the particles? Yes

Approximate nr of groups: 20

Pixel size before extraction (A) -1

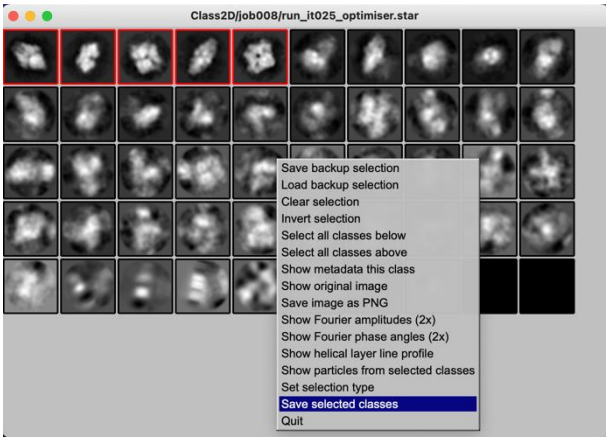
When you press run this will launch the Relion manual selection GUI on your local machine.



Check the **Sort images?** and **Reverse Sort?** boxes and press **Display!**.

This will launch a second window with the class selector.

In the class view left click on the classes you wish to select and then right click to and select **Save selected classes** in the menu.



Finally, close both Relion GUI windows to finish the job.

Re-training the TOPAZ neural network

Now we will use the good classes to train a Topaz neural network for much improved particle picking.

Under **Automated Particle Picking** select a new **RELION Topaz Training** job.

Input micrographs for autopick: Select/job004/selected_micrographs.star

Pixel size in micrographs (A): -1

Shrink factor: 0

OR train on a set of particles? Yes

Particles STAR file for training: Select/job008/particles.star
This will be the same job name regardless of which class selection method you chose above.

Particle diameter (A): 180

Nr of particles per micrograph: 300

Which GPUs to use: 1

Running options
Number of MPI procs: 1

Particle picking with Topaz

Now we will pick particles from all the micrographs with the Topaz picker

From **Automated Particle Picking** select a new **RELION Autopick Topaz (Single Particle)** job

Input micrographs for autopick: CtfFind/job003/micrographs_ctf.star
This is the CTF info for all micrographs from the precalculated results

Pixel size in micrographs (A): -1
A negative value here means the pixel size willbe read from the file

Shrink factor: 0

Particle diameter (A): 180

Nr of particles per micrograph: 300

Trained topaz model: AutoPick/job009/model_epoch10.sav
This is the trained model from the precalculated results

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>

Running Options

Number of MPI procs: 1

Examine the results for this job. Compare them to the results of **AutoPick/job005** (the LoG picking). Look how many more particles have been found!

Extracting the Full Particle Set

We will now extract the full particle set from all the micrographs. The particles will again be down sampled for faster processing. In **Extract Particles** select a **Relion Extract Particles (Single Particle)** job.

micrograph STAR file: CtfFind/job003/micrographs_ctf.star

Input coordinates: AutoPick/job010/autopick.star

Extraction Options

Particle box size (pix): 256

Invert contrast? Yes

Normalize particles? Yes

Diameter background circle (pix): 200

Stddev for white dust removal: -1

Stddev for black dust removal: -1

Rescale particles? Yes

Re-scaled size (pixels): 64

Write output in float16? Yes

Running options

Number of MPI procs: 6

MPI run command: ccpem-mpirun -n XXXmpinodesXXX

Cleaning up the particle set

We will next run another round of 2D classification and 2D class selection to improve the particle set. This time we will use the gradient based 2D classification algorithm

From **2D Classification** select a new **RELION Class2D (VDAM, Single Particle)** job.

Input images STAR file: Extract/job011/particles.star

Number of VDAM mini-batches: 100

Do CTF-correction? Yes

Ignore CTFs until first peak? No

Number of classes: 100

Regularisation parameter T: 2

Mask diameter (A): 200

Mask individual particles with zeros? Yes

Limit resolution E-step to (A): -1

Center class averages? Yes

Perform image alignment? Yes

In-plane angular sampling: 6

Offset search range (pix): 5

Offset search step (pix): 1

Allow coarser sampling? No

Number of pooled particles: 30

Use parallel disc I/O? Yes

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>

Running options

Number of threads: 12

After the 2D classification job has finished run Automated 2D class selection.

Select the particles from the good classes

This uses Relion's automated 2D class selection on the new full data set. Alternatively yu can do another manual class selection as in the previous 2D class selection job.

Under **Automated Class Selection** select a new **RELION Automatic 2D Class Selection** job

Select classes from optimiser.star: Class2D/job012/run_it0100_optimiser.star

Minimum threshold for auto-selection: 0.20

Python executable: /opt/relion_cl2d_autoselect/bin/python

As before, once the job has finished, examine the results. You can hover the mouse over the rejected classes to see their threshold scores. If you are not satisfied with the results, you can run the job again with a modified threshold.

De novo 3D model generation

Relion-4.0 uses a gradient-driven algorithm to generate a de novo 3D initial model from the 2D particles. As of release 4.0, this algorithm is different from the SGD algorithm in the CryoSPARC program. Provided you have a reasonable distribution of viewing directions, and your data is good enough to yield detailed class averages in 2D classification, this algorithm is likely to yield a suitable, low-resolution model that can subsequently be used for 3D classification or 3D auto-refine.

Running the job

Select the **New Job** tab and find the **RELION initial model generation** job type in **Initial Model Generation**.

Use the following parameters to run a new initial model creation job. Parameters that are not specified can be left with the default value.

Input images STAR file: Select/job013/particles.star
Input the particles selection Node from the 2D classification selection task.

Number of VDAM mini-batches: 100
The algorithm will loop over mini-batches, which contain only hundreds to thousands of particles.

Number of classes: 1
Sometimes, using more than one class may help in providing a 'sink' for suboptimal particles that may still exist in the data set. In this case, which is quite homogeneous, a single class should work just fine.

Symmetry: D2

Run in C1 and apply symmetry later?: Yes
The actual refinement will be run in C1, which has been observed to converge better than performing it in higher symmetry groups. After the refinement, the relion_align_symmetry program is run to automatically detect the symmetry axes and the symmetry will be applied.

Mask diameter (A): 200

Flatten and enforce non-negative solvent: Yes

Initial angular sampling: 15 degrees
The default angular and offset samplings should be fine for most cases, perhaps except for highly symmetric particles like viruses, which may require finer samplings.

Offset search range (pix): 6

Offset search step (pix): 2

Compute Options:

Use parallel disc I/O?: Yes

Number of pooled particles: 30

Skip padding?: Yes

Skip gridding?: Yes

Pre-read all particles into RAM?: No
For small datasets / computers with large RAM allocation this can be faster however we will write the particles to a scratch disk instead, see below.

Copy particles to scratch directory: <leave blank>
N.B. This can be useful if you don't have enough RAM.

Combine iterations through disc? Yes

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>
If your computer has only one GPU this can be left blank. If multiple GPUs are present they can be specified via a comma separated list of GPU IDs e.g. 0,1,2,3.

Running Options

Number of threads: 8

Click **RUN** to start the job. Using the settings above, this job took ~3 minutes on our system. Your initial model generation job will be run as job 32.

Analysing the results

When the job is complete you can visualise the output map from the **Results** tab in the Doppio interface by clicking on the Initial model panel for **initial_model.mrc**. This map should have been symmetrised. Visually confirm that the map has the expected shape and symmetry. The resolution of the map will be low, but the general shape should be correct.

Unsupervised 3D classification

All data sets are heterogeneous! The question is how much you are willing to tolerate. Relion's 3D multi-reference refinement procedure provides a powerful unsupervised 3D classification approach.

Running the job

In the **3D Classification** Section of **New Job** select the **RELION 3D classification (single particle)**. Setup with the following parameters:

Input images STAR file: Select/job013/particles.star

Reference map: InitialModel/job014/initial_model.mrc
Use the initial model job you ran previously.

Reference mask (optional): <leave blank>
This is the place where we for example provided large/small-subunit masks for our focussed ribosome refinements. If left empty, a spherical mask with the particle diameter given below will be used. This introduces the least bias into the classification.

Ref. map is on absolute greyscale: Yes
Given that this map was reconstructed from this data set, it is already on the correct greyscale. Any map that is not reconstructed from the same data in relion should probably be considered as not being on the correct greyscale.

Initial low-pass filter (A): 50
One should NOT use high-resolution starting models as they may introduce bias into the refinement process. As also explained in (Scheres 2010)], one should filter the initial map as much as one can. For ribosome we often use 70 Å, for smaller particles we typically use values of 40-60 Å.

Symmetry: C1
Although we know that this sample has D2 symmetry, it is often a good idea to perform an initial classification without any symmetry, so bad particles, which are not symmetric, can get separated from proper ones, and the symmetry can be verified in the reconstructed maps.

Do CTF correction? Yes

Ignore CTFs until first peak? No
Only use this option if you also did so in the 2D classification job that you used to create the references.

Number of classes: 4
Using more classes will divide the data set into more subsets, potentially describing more variability. The computational costs scales linearly with the number of classes, both in terms of CPU time and required computer memory.

Regularisation parameter T: 4
For the exact definition of T, please refer to Scheres, 2012a (A Bayesian view...). For cryo-EM 2D classification we typically use values of T=1-2, and for 3D classification values of 2-4. 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.

Number of iterations: 25
We typically do not change this.

Mask diameter (A): 200
Just use the same value as we did before in the 2D classification job-type.

Mask individual 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 10-15Å anyway.

Angular sampling interval: 7.5 degrees

Offset search range (pix): 5

Offset search step (pix): 1

Perform local angular searches? No

Allow coarser sampling? No
The above are all set to the default values which rarely change except for large and highly symmetric particles, like icosahedral viruses, where 3.7 degrees angular sampling is typically used.

Compute options

Use parallel disc I/O? Yes

Number of pooled particles:30

Skip padding? No

Skip gridding? Yes

Pre-read all particles into RAM? No
Again, this is only possible if the data set is small and/or you have a large amount of memory.

Copy particles to scratch directory: <leave blank>
N.B. please enter your personal workshop computer ID here e.g. /home/to659912/relion.

Combine iterations through disc? Yes

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>
If your computer has only one GPU this can be left blank. If multiple GPUs are present they can be specified via a comma separated list of GPU IDs e.g. 0,1,2,3.

Number of threads: 6
3D classification takes more memory than 2D classification, so often more threads are used. However, in this case the images are rather small and RAM-shortage may not be such a big issue.

Number of MPI procs: 3

MPI run command: ccpem-mpirun -n XXXmpinodesXXX
We are using the pre-built mpi libraries that come with the CCP-EM software suite. Make sure "`mpirun`" is replaced with "`ccpem-mpirun`".

Click **RUN** to start the job. Using the settings above, this job took ~20 minutes on the STFC VMs.

Once completed in the RESULTS panel you can view the particle class distribution and the 3D maps for each 3D class. From the distribution plots see how many particles have been selected for each class and if the selection has converged. The 3D map(s) of the major class(es) allow you to check they have the expected shape.

Selecting good particles for further processing

When you are ready to choose your class(es) launch **3D class auto selection** via the **NEW JOB** menu.

Input 3D classification particles file: Class3D/job015/run_it25_data.star

This job selects the particles from highest resolution 3D class. In most cases this will give the best result. After the job has completed compare the selected class to the classes in the previous 3D classification job; did the job select the best 3D class?

High-resolution 3D refinement

Once a subset of sufficient homogeneity has been selected, one may use the 3D auto-refine procedure in relion to refine this subset to high resolution in a fully automated manner. This procedure employs the so-called gold-standard way to calculate Fourier Shell Correlation (FSC) from independently refined half-reconstructions to estimate resolution, so that self-enhancing overfitting may be avoided (Scheres & Chen, 2012). Combined with a procedure to estimate the accuracy of the angular assignments (Scheres, 2012) it automatically determines when a refinement has converged. Therefore, this procedure requires very little user input, i.e. it remains objective, and has been observed to yield excellent maps for many data sets. Another advantage is that one typically only needs to run it once, as there are hardly any parameters to optimize.

Particle re-extraction

In the earlier steps of this tutorial the extracted particles were down sampled so processing steps would run more quickly. This is possible because these early steps do not require very high resolution information. Before we start our high-resolution refinement, we should first re-extract our current set of selected particles with less down-scaling, so that we can potentially go to higher resolution.

Select a **RELION extract particles** job from the **Extract Particles** section and use the following parameters:

Inputs

Micrograph STAR file: CtfFind/job003/micrographs_ctf.star
This is from the precalculated results

Re-extract refined particles from a STAR file: Select/job016/selected_particles.star
Use the output from your class3D auto selection previous job.

Extraction Options

Particle box size (pix): 360
We will use a larger box, so that de-localised CTF signals can be better modelled. This is important for subsequent CTF refinement.

Invert contrast? Yes
This will make white particles.

Normalize particles? Yes
We always normalize with Relion.

Rescale particles: Yes

Re-scaled size (pixels): 256
*To prevent working with very large images, let's down-sample to a pixel size of 360*0.885/256=1.244Å. This will limit our maximum achievable resolution to 2.5 Å (i.e. 2xNyquist), which is probably enough for such a small data set. N.B. this should always be an even number!*

Write output in float16? Yes

If set to Yes, this program will write output images in float16 MRC format. This will save a factor of two in disk space compared to the default of writing in float32.

Re-extraction options:

Reset the refined offsets to zero? No
This would discard the translational offsets from the previous classification runs.

Re-center refined coordinates? Yes
This will re-center all the particles according to the aligned offsets from the 3D classification job above.

Re-center on X-coordinate (in pix): 0
We want to keep the centre of the molecule in the middle of the box.

Re-center on X-coordinate (in pix): 0
We want to keep the centre of the molecule in the middle of the box.

Re-center on X-coordinate (in pix): 0
We want to keep the centre of the molecule in the middle of the box.

Running Options

Number of MPI procs: 6

Click **RUN** to perform the extraction.

Rescaling the map

Because we have changed the size and pixel size of the extracted particles, we now need to re-scale the best map obtained so far from the **Class3D** job so the map and particles have the same box and pixle size.

Under **Map Utilities** create a new **Change map box and pixel** size job:

Input map: Select/job016/selected_class.mrc
This is the best class that was selected from Class3D/job015 by 3D class auto selection in Select/job016.

Rebox the map? Yes

New box size (px): 256

Rescale the map? Yes

Rescaled pixel size: 1.244

Fix disparities in pixel size precision? Yes

Click **RUN** to start the job. You can check the map once it's completed to ensure the correct map was re-scaled.

Running the auto-refine job

From **3D Refinement** Start the **RELION 3D auto-refine (single particle)** job and enter the following:

Input images STAR file: Extract/job017/particles.star

The re-extracted particles with a larger box and smaller pixel size.

Reference map: ReboxRescale/job018/selected_class_reboxed_rescaled.mrc
Use the reboxed and rescaled map.

Reference mask (optional): <leave blank>

Ref. map is on absolute greyscale? No
Because of the different normalisation of down-scaled images, the rescaled map is no longer on the correct absolute grey scale. Setting this option to No is therefore important and will correct the greyscale in the first iteration of the refinement.

Initial low-pass filter (A): 50
We typically start auto-refinements from low-pass filtered maps to prevent bias towards high frequency components in the map, and to maintain the gold-standard of completely independent refinements at resolutions higher than the initial one.

Symmetry: D2
We now aim for high-resolution refinement, so imposing symmetry will effectively quadruple the number of particles.

Do CTF correction? Yes

Ignore CTFs until first peak? No

Mask diameter (A): 200

Mask individual particles with zeros? Yes

Angular sampling interval: 7.5 degrees

Local searches from auto-sampling: 1.8 degrees
The orientational sampling 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 default angular sampling of 7.5 degrees, and local searches from a sampling of 1.8 degrees. Only for higher symmetry refinements, we use 3.7 degrees sampling and perform local searches from 0.9 degrees.

Use finer angular sampling faster? Yes
This will be more aggressive in proceeding with iterations of finer angular sampling faster and therefore speed up the calculations. You might want to check that you're not losing resolution for this in the later stages of your own processing, but during the initial stages it often does not matter much.

Compute Options:

Pre-read all particles into RAM? No

Copy particles to scratch directory: <leave blank>

Combine iterations through disc? Yes

Use GPU acceleration? Yes

Which GPUs to use: <leave blank>

Number of MPI procs: 3

Number of threads: 6
As the MPI nodes are divided between one master (who does nothing else than bossing the others around) and two sets of slaves who do all the work on the two half-sets, it is most efficient to use an odd number of MPI processors, and the minimum number of MPI processes for 3D auto-refine jobs is 3. Memory requirements may increase significantly at the final iteration, as all frequencies until Nyquist will be considered, so for larger sized boxes than the ones in this test data set you may want to run with as many threads as you have cores on your cluster nodes.

MPI run command: ccpem-mpirun -n XXXmpinodesXXX
We are using the pre-built mpi libraries that come with the CCP-EM software suite. Make sure "`mpirun`" is replaced with "`ccpem-mpirun`".

On the STFC VM we use 1 GPU, 3 MPI processes (one master, two for the half sets) and as we have 12 CPUs 6 threads (6 per half set). This should take ~20 minutes. Results may vary on your system.

Analysing the 3D refinement

You can see the final 3D map and resolution in the **RESULTS** tab. The map should exceed ~4 Å.

In the **I/O** tab you can see **run_half1_class001_unfil.mrc** which is the first half map from the final iteration of refinement. Also present is **run_class001.mrc** which is the final or full map made from combining both half maps.

Look at the **LOGS** tab and note that the automated increase in angular sampling is an important aspect of the auto-refine procedure. It is based on signal-to-noise considerations that are explained in Scheres, 2012 (Implementation of a Bayesian…), to estimate the accuracy of the angular and translational assignments. The program will not use finer angular and translational sampling rates than it deems necessary (because it would not improve the results). The estimated accuracies and employed sampling rates, together with current resolution estimates are all stored in the **_optimiser.star** and **_model.star** files and written in the **run.out** file displayed here.

In the last iteration the two independent half-reconstructions are joined together, the resolution will typically improve significantly in the last iteration. Because the program will use all data out to Nyquist frequency, this iteration also requires more memory and CPU.

Mask creation & Post processing

After performing a 3D auto-refinement, the map needs to be sharpened. Also, the gold-standard FSC curves inside the auto-refine procedures only use unmasked maps (unless you've used the option **Use solvent-flattened FSCs**). This means that the actual resolution is under-estimated during the actual refinement, because noise in the solvent region will lower the FSC curve. Relion's procedure for B-factor sharpening and calculating masked FSC curves (Chen et al, 2013) is called post-processing. First however, we'll need to make a mask to define where the protein ends and the solvent region starts. This is done using the Mask Creation job-type.

Making a mask

In the **Mask Creation** category select **RELION mask create** job and set the following parameters:

Input 3D map: Refine3D/job019/run_class001.mrc
Use the full map from your Refine3D job.

Lowpass filter map (A):15
A 15 Å low-pass filter seems to be a good choice for smooth solvent masks for many proteins.

Pixel size (A): -1
This value will be taken automatically from the header of the input map.

Initial binarisation threshold: 0.005
This should be a threshold at which rendering of the low-pass filtered map in e.g. chimera shows no noisy spots outside the protein area. Move the threshold up and down to find a suitable spot.

Extend binary map this many pixels: 0
The threshold above is used to generate a black-and-white mask. The white volume in this map will be grown this many pixels in all directions. Use this to make your initial binary mask less tight.

Add a soft-edge of this many pixels: 6
This will put a cosine-shaped soft edge on your masks. This is important, as the correction procedure that measures the effect of the mask on the FSC curve may be quite sensitive to too sharp masks. As the mask generation is relatively quick, we often play with the mask parameters to get the best resolution estimate.

Running options:

Number of threads: 12
This will speed up the calculation.

This should take <2 mins to run. When the job is complete click on the **RESULTS** tab. You can see the mask overlaid on the input map. Make sure the mask encapsulates the entire structure but does not leave a lot of solvent inside the mask. Also ensure the mask does not have the high-resolution features present in the input map. You repeat with different settings until you are happy.

Post-processing

In **Map Postprocessing** select a new **RELION Postprocessing** job and set the following:

One of the 2 unfiltered half-maps: Refine3D/job019/run_half1_class001_unfil.mrc

Solvent mask: MaskCreate/job020/mask.mrc

Calibrated pixel size (A):1.244
Sometimes you find out when you start building a model that what you thought was the correct pixel size, in fact was off by several percent. Inside relion, everything up until this point was still consistent. so you do not need to re-refine your map and/or re-classify your data. All you need to do is provide the correct pixel size here for your correct map and final resolution estimation.

Estimate B-factor automatically: Yes
This procedure is based on the classic 2003 Rosenthal and Henderson paper, and will need the final resolution to extend significantly beyond 10 Å. If your map does not reach that resolution, you may want to use your own ad-hoc B-factor instead.

Lowest resolution for auto-B fit (A): 10
This is usually not changed.

Use your own B-factor? No

MTF of the detector (STAR file): Movies/mtf_k2_200kV.star
This file came from the microscope. NOTE: you cannot use the file browser to input this file as it looks in your local machine rather than the remote virtual machine where the project is running.

Original detector pixel size: 0.885
This is the original pixel size (in Angstroms) in the raw (non-super-resolution!) micrographs.

Skip FSC-weighting? No
This option is sometimes useful to analyse regions of the map in which the resolution extends beyond the overall resolution of the map. This is not the case now.

Run the job (no need for a cluster, as this job will run very quickly) and look at the RESULTS tab. You'll see the improvement in resolution estimate from masking out the noisy solvent regions in the two half maps. You can also see this improvement in FSC plot. The resolution estimate is based on the phase-randomization procedure as published previously (Chen et al, 2013). Make sure that the FSC of the phase-randomised maps (the red curve) is more-or-less zero at the estimated resolution of the postprocessed map. If it is not, then your mask is too sharp or has too many details. In that case use a stronger low-pass filter and/or a wider and softer mask in the **Mask creation** step above and repeat the postprocessing.

Further processing

In the typical workflow we would do several further steps to maximise the quality of the map produced from your data. These include:

- CTF and aberration refinement
- Per-particle defocus estimation
- Bayesian polishing

CTF and aberration refinement

Next, we'll use the CTF refinement job-type to estimate the asymmetrical and symmetrical aberrations in the dataset; whether there is any anisotropic magnification; and we'll re-estimate per-particle defocus values for the entire data set. Running this job-type can lead to further improvements in resolution at a relatively minor computational cost, but it all depends on how flat your ice was (for per-particle defocus estimates), and how well you had aligned your scope (for the aberrations). It runs from a previous 3D auto-refine job as well as a corresponding Post-processing job. Let's start with the higher-order aberrations, to see whether this data suffered from beamtilt or trefoil (which are asymmetric aberrations), or from tetrafoil or an error in spherical aberration (which are symmetric aberrations).

Higher Order Abberations

Under **`CTF Refinement`** Select a new **`RELION Ctf Refinement`** job

`Particles (from Refine3D): Refine3D/job019/run_class001.mrc`

`Postprocess STAR file: Postprocess/job021/postprocess.star`

`Minimum resolution for fits (A): 30`

`Estimate beamtilt? Yes`

`Also estimate trefoil? Yes`

`Estimate 4th order aberrations? Yes`

`Perform CTF parameter fitting? No`

`Fit defocus? No`

`Fit astigmatism? No`

`Fit B-factor? No`

`Fit phase-shift? No`

`Number of MPI procs: 1`

`Number of threads: 12`

`Submit to queue? No`

`MPI run command: ccpem-mpirun -n XXXmpinodesXXX`

You'll see that this data suffered from some beamtilt: one side of the asymmetrical aberration images is blue, whereas the other side is red. There was also a small error in the spherical aberration, as the

symmetrical aberration image shows a significant, circularly symmetric difference (the image is blue at higher spatial frequencies, i.e. away from the center of the image). Importantly, for both the asymmetric and the symmetric aberaations, the model seems to capture the aberrations well.

If the data had suffered from trefoil, then the asymmetric aberration plot would have shown 3-fold symmetric blue/red deviations. If the data had suffered from tetrafoil, then the symmetric aberration plot would have shown 4-fold symmetric blue/red deviations. Examples of those are shown in the supplement of our 2019 publication on Tau filaments from the brain of individuals with chronic traumatic encephalopathy (CTE).

Anisotropic Magnification Correction

Next, let's see whether these data suffer from anisotropic magnification. Under **`Ctf Refinement`** select a new **`Anisotrpoic Magnification Correction`** job

`Particles (from Refine3D): CtfRefinement/job022/particles_ctf_refine.star`

`Postprocess STAR file: PostProcess/job021/postprocess.star`

`Minimum resolution for fits (A): 30`

`Running options`
`Number of MPI procs: 1`
`MPI run command: ccpem-mpirun -n XXXmpinodesXXX`
`Number of threads: 12`

Per-particle defocus values

Under **`CTF Refinement`** Select a new **`RELION Ctf Refinement`** job.

`Particles (from Refine3D): CtfRefine/job023/run_class001.mrc`
The name of the parameter specifies 'from Refine3D' but we want to use the particles from the previous anisotropic magnification correction job.

`Postprocess STAR file: Postprocess/job021/postprocess.star`

`Minimum resolution for fits (A): 30`

`Estimate beamtilt? No`

`Also estimate trefoil? No`

`Estimate 4th order aberrations? No`

`Perform CTF parameter fitting? Yes`

`Fit defocus? Per-particle`

`Fit astigmatism? Per-micrograph`

`Fit B-factor? No`

`Fit phase-shift? No`

`Number of MPI procs: 1`

Number of threads: 12

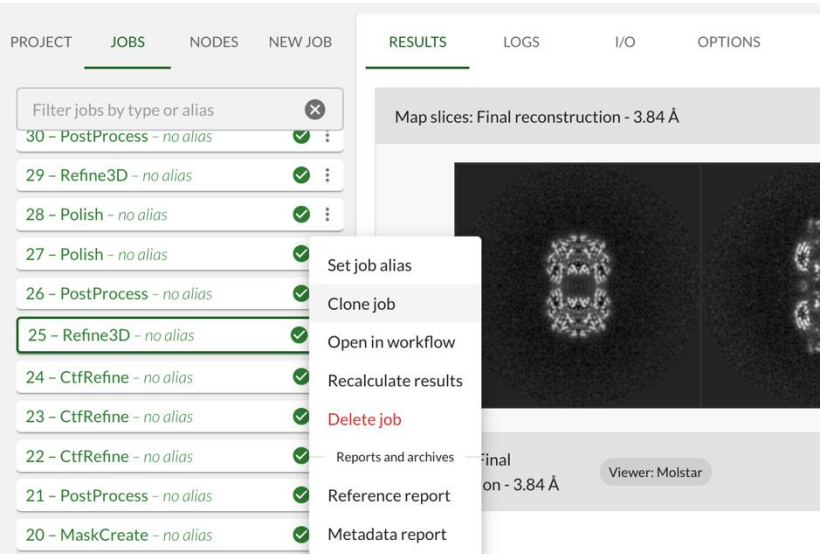
MPI run command: ccpem-mpirun -n XXXmpinodesXXX

Per-particle defocus values are plotted by colour for each micrograph in the logfile.pdf. Can you spot micrographs with a tilted ice layer?

3D Refinement with CTF Refined Particles

It is probably a good idea to re-run 3D auto-refine and Post-processing at this stage, so we can confirm that the new particle STAR file actually gives better results.

Lets make life easy. In the lefthand window go to job **Refine3D/job046/** and select **Clone Job** in the context menu.



This creates a new job with the same parameters as the original job. Change two of them:

Input images STAR file: CtfRefine/job024/particles_ctf_refine.star

Reference map: Refine3D/job019/run_class001.mrc

Run the Job and then do another post processing job to see how the CTF refinement has improved the reconstruction.

Clone **PostProcess/job021/** and change just the input file:

One of the 2 unfiltered half-maps: Refine3D/job025/run_half1_class001_unfil.mrc

Bayesian polishing

Relion also implements a Bayesian approach to per-particle, reference-based beam-induced motion correction. This approach aims to optimise a regularised likelihood, which allows us to associate with each hypothetical set of particle trajectories a prior likelihood that favors spatially coherent and temporally smooth motion without imposing any hard constraints. The smoothness prior term requires three parameters that describe the statistics of the observed motion. To estimate the prior that yields the best motion tracks for this particular data set, we can first run the program in ‘training mode’. Once the estimates have been obtained, one can then run the program again to fit tracks for

the motion of all particles in the data set and to produce adequately weighted averages of the aligned movie frames.

Polishing training

This step calculates the optimal parameters for particle polishing.

Under **Particle Polishing** select a new **RELION Bayesian Polishing – Training** job:

Micrographs (from MotionCorr): MotionCorr/job002/corrected_micrographs.star

Particles (from Refine3D or CtfRefine): Refine3D/job025/run_data.star
These particles will be polished.

Postprocess STAR file: PostProcess/job026/postprocess.star
The mask and FSC curve from this job will be used in the polishing procedure.

First movie frame: 1

Last movie frame: -1
(Some people throw away the first or last frames from their movies. Note that this is not recommended when performing Bayesian polishing in relion. The B-factor weighting of the movie frames will automatically optimise the signal-to-noise ratio in the shiny particles, so it is best to include all movie frames.)

Fraction of Fourier pixels for testing: 0.5

Use this many particles: 3500
That's almost all we have anyway. Note that the more particles, the more RAM this program will take. If you run out of memory, try training with fewer particles. Using much fewer than 4000 particles is not recommended.

Particle Polishing

We will perform the particles polishing step using pre-calculated parameters

Under **Particle Polishing** select a new **RELION Bayesian Polishing** job:

Micrographs (from MotionCorr): MotionCorr/job002/corrected_micrographs.star
From the pre-calcalculated results

Particles (from Refine3D or CtfRefine): Refine3D/job025/run_data.star
These particles will be polished.

Postprocess STAR file: PostProcess/job026/postprocess.star
The mask and FSC curve from this job will be used in the polishing procedure.

First movie frame: 1

Last movie frame: -1

Extraction size (pix in unbinned movie): 360

Re-scaled size (pixels): 256

Optimised parameter file: Polish/job027/opt_params_all_groups.txt
The optimised parameters from the pre-calculated data

OR use your own parameters? No

Number of MPI procs: 1

Number of threads: 12

Finally run one more set of 3D reconstruction and postprocessing jobs by cloning jobs **Refine3D/job025** and **Postprocess/job026** and using the polished results as inputs.

Relion References:

Rosenthal P and Henderson R. Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. Journal of Molecular Biology,333(4):721–745, October 2003.

Scheres SHW. Classification of Structural Heterogeneity by Maximum-Likelihood Methods. In Cryo-EM, Part B: 3-D Reconstruction, volume 482 of Methods in Enzymology, pages 295–320. 2010.

Scheres SHW. A Bayesian view on cryo-EM structure determination. Journal of Molecular Biology, 415(2):406–418, January 2012. doi:10.1016/j.jmb.2011.11.010.

Scheres SHW. RELION: Implementation of a Bayesian approach to cryo-EM structure determination. Journal of Structural Biology, 180(3):519–530, December 2012. doi:10.1016/j.jsb.2012.09.006.

Scheres SHW and Chen S. Prevention of overfitting in cryo-EM structure determination. Nature methods, 9(9):853–854, September 2012. doi:10.1038/nmeth.2115.

Shaoxia Chen, Greg McMullan, Abdul R. Faruqi, Garib N. Murshudov, Judith M. Short, Sjors H. W. Scheres, and Richard Henderson. High-resolution noise substitution to measure overfitting and validate resolution in 3d structure determination by single particle electron cryomicroscopy. Ultramicroscopy, 135:24–35, December 2013. doi:10.1016/j.ultramic.2013.06.004

Refmac-Servalcat references:

Yamashita, K, Palmer, C M, Burnley, T, Murshudov, G. N. Cryo-EM single particle structure refinement and map calculation using Servalcat. Acta Cryst D77, 1282-129, 2021.

Current approaches for the fitting and refinement of atomic models into cryo-EM maps using CCP-EM. Nicholls, RA, Tykac M, Kovalevskiy, O, & Murshudov, GN Acta Cryst D74, 492-505, 2018.

CCP-EM reference:

Burnley, T, Palmer, CM & Winn M. Recent developments in the CCP-EM software suite. Acta Cryst D73, 469-47, 2017.