

Getting started:

Before any analysis can be performed we need to get our data in order.

As we don't want to unnecessarily copy large amounts of data around, we can make a 'link' to the place where the original micrographs are on the M3 computer cluster.

So first we navigate to our directory where we will start the project (note from this point all commands to be typed in a terminal will be in 'commas')

```
' cd /scratch/"YOURPROJECTDIRECTOR" '
```

Now need to make a directory place the data, this will be called the working directory.

```
'mkdir apo'
```

```
'cd apo'
```

```
'mkdir Micrographs'
```

```
'cd Micrographs'
```

```
'ln -s /projects/cryoemfacility/hven0001/Cryoz2019/Micrographs/*.mrc .'
```

```
'cd ..'
```

```
'ln -s  
/projects/cryoemfacility/hven0001/Cryoz2019/Micrographs/FoilHole_21466324_Data_214  
61806_21461807_20180923_0905-gain-ref.MRC.'
```

Now we have a directory called Micrographs which contains 100 micrographs from an apo-ferritin data collection on the Titan Krios and we have a gain reference for the K2 detector for this collection in the root of the working directory.

As the M3 computer system uses a modular environment for applications we need to load the appropriate modules for the RELION workflow.

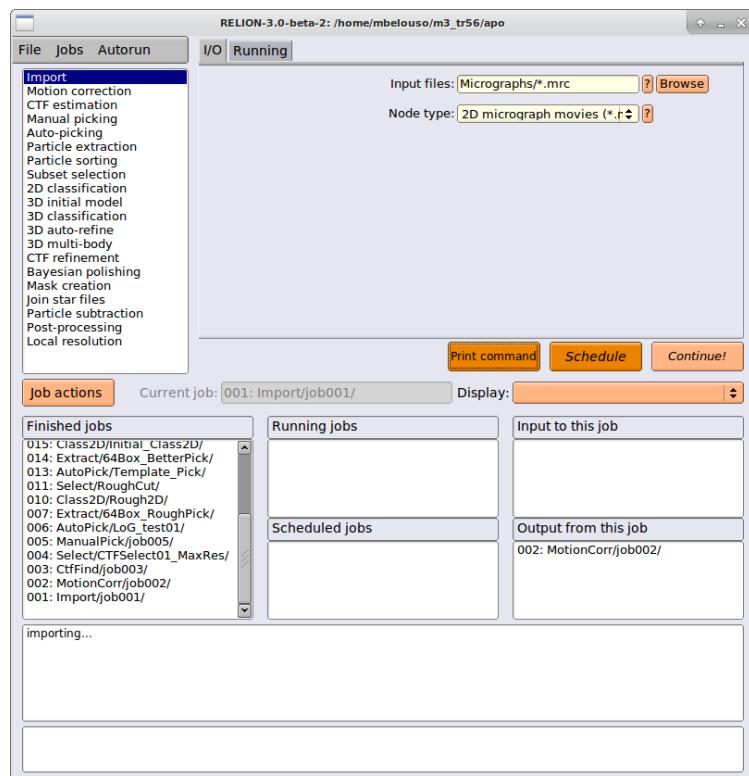
'module load relion/3.0-stable'

We are now ready to lauch RELION and this is done by:

'relion'

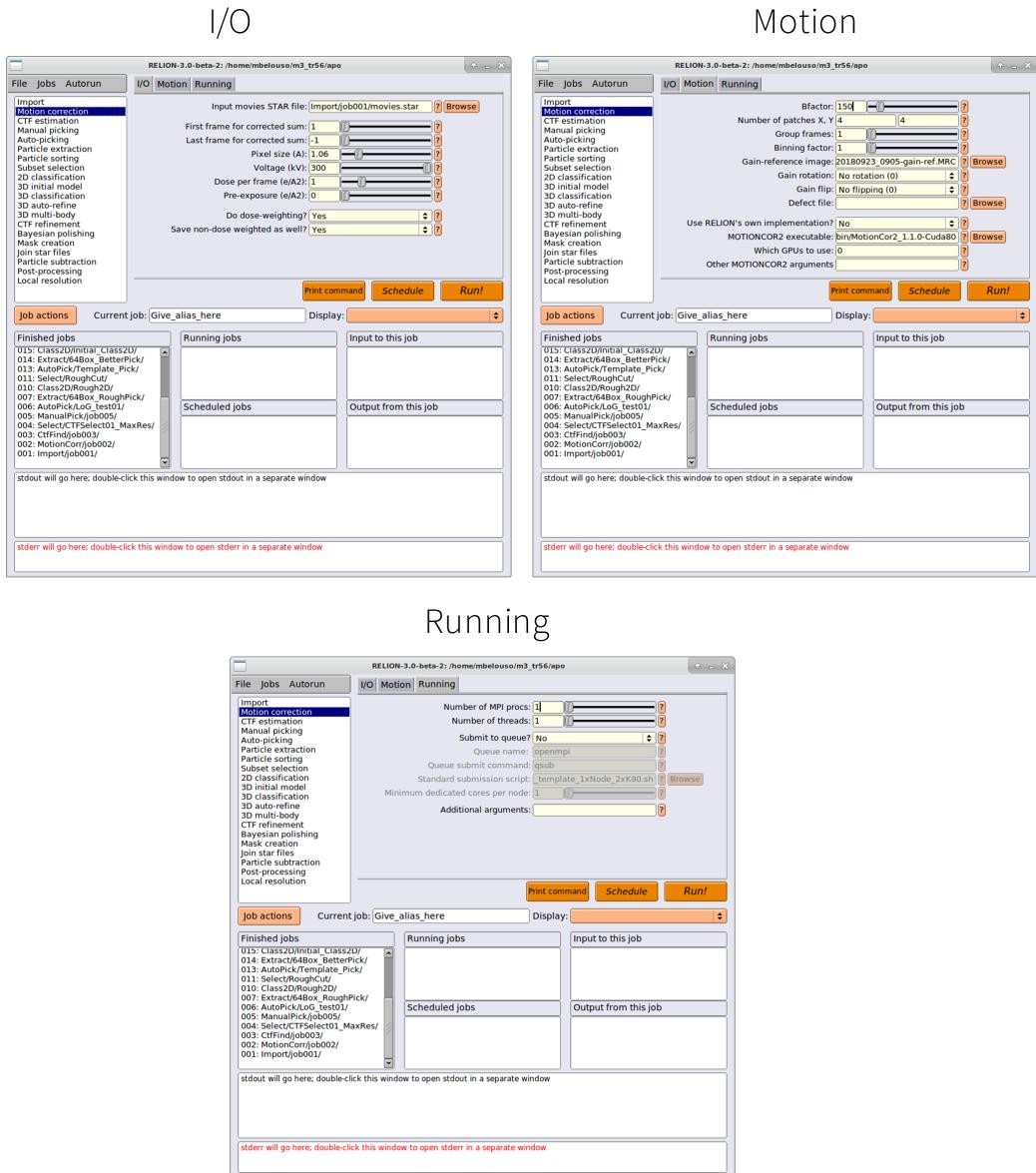
Press y to start a new project in your working directory:

Now that we have our data all ready to go and all organized we need to RELION where our data is and import the micrographs as 2D micrograph movies. Click on the Import and import all the mrc files from the Micrographs directory. In linux/unix/mac the * symbol represents a wild-card and will include all files that end with .mrc



Motion Correction:

To correct for electron beam induced motion we will use a program called UCSF MotionCor2, and luckily RELION is a ‘wrapper’ for this program so there is no need to leave the RELION environment to launch this program. So click on the next tab labelled Motion Correction:



We need to state the microscope parameters in the first window such as physical pixel size (sometime called apix or angpix or smpd in different programs) and the acceleration voltage of the microscope as well as the electron dose per movie frame. It is good to also save the non-doseweighted average as the next step which is CTF correction is more accurate on non-doseweighted averages. In the Motion tab we need to change a few values from the default such as the amount of patches (4x4 is a good place to start for

many projects, but if you have molecules with high contrast sometimes higher numbers give a more accurate motion estimation across the whole micrograph).

If we collected using super-resolution mode which is available on the Gatan K2 and K3 direct electron detectors at this point you would most likely ‘bin’ your data by a factor of 2, which will Fourier scale your micrographs to the desired amount.

The micrographs in this experiment were saved as un-gainreferenced (which saves on disk space), so we need to gain reference them, so you will need to point MotionCor to the gain reference.

As the P4 destops only have access to a single GPU you will have to tell MotionCor which one to use which is GPU id=0 (note for larger compute jobs with access to multiple GPUs you will have to state explicitly here which ones to use, for example if your computer or node has 4 GPUs and you want to split this job across all four you will have to add in the ‘Which GPUs to use: 0:1:2:3’

In the Running tab, we are only to use a single MPI process as we are only running on a single GPU, so we will keep the values at default.

NB: a useful command to put in the Additional arguments is: --only_do_unfinished as if you decide to add more micrographs later you don't have to motioncorrect them all again.

Once you press Run! This should take approximately 20 minutes. To check the output of this job you can press the Display tab and click on the logfile.pdf, which will supply useful information as to how the motion correction went. If all went well, there shouldn’t be any sharp spikes in Accumulated total motion. However in the real world not every micrograph is good as ice thickness varies a lot and there will certainly be a few poorly corrected micrographs. You can omit these at this stage, but generally it’s a good idea to omit micrographs after CTF estimation.

CTF Estimation:

CTF estimation is the next step, so the output from MotionCor goes straight into the CTF estimation. There are many programs to estimate the defocus parameters of each micrograph, but Gctf does an excellent job and executes very quickly. RELION also is acts as a wrapper for this program. The boxes below show how to fill out the tabs in RELION.

The figure consists of four side-by-side screenshots of the RELION software interface, version 3.0-beta-2, showing different tabs for CTF estimation:

- I/O Tab:** Shows input micrographs (corrected_micrographs.star) and various parameters for CTF estimation: Spherical aberration (mm), Magnified pixel size (Angstrom), and Amount of astigmatism (A).
- Searches Tab:** Shows search parameters: FFT box size (pix), Maximum resolution (A), Minimum defocus value (A), Maximum defocus value (A), Defocus step size (A), and Estimate phase shifts? (No).
- Gctf Tab:** Shows Gctf options: Use Gctf instead?, Gctf executable, Ignore 'Searches' parameters?, Perform equi-phase averaging?, Other Gctf options, and Which GPUs to use.
- Running Tab:** Shows running parameters: Number of MPI pros, Submit to queue?, Queue name, Queue submit command, Standard queue template, Minimum dedicated cores per node, and Additional arguments.

Each screenshot includes a "Print Command" button at the bottom right and a "Job actions" section with a "Current job: Give_alias_here" dropdown and a "Display:" button.

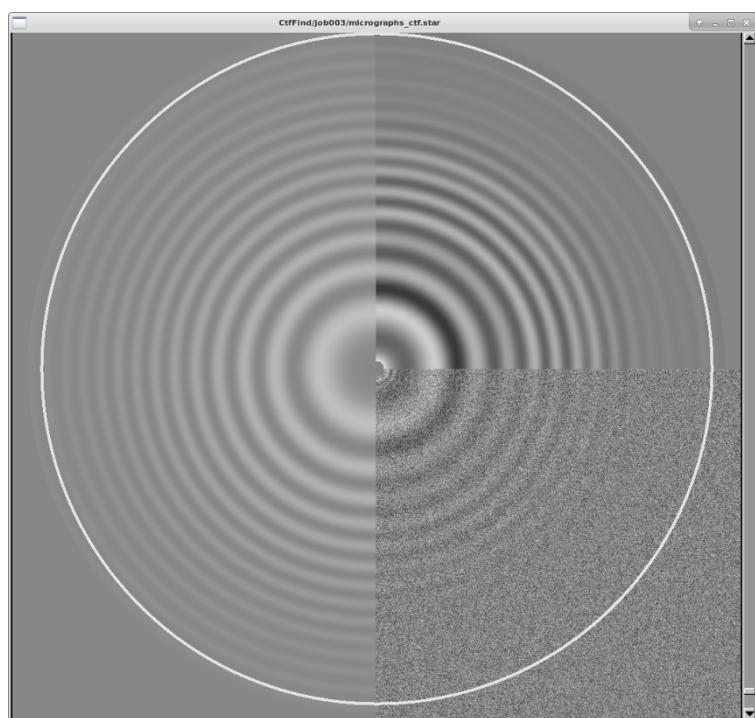
Changing the default values is helpful here. The amplitude contrast should be changes from 0.1 to 0.07 (higher values like 0.1 are more appropriate for large very high contrast molecules such as ribosomes, but protein only lower contrast molecules 0.07 is a good value). The amount of astigmatism should also probably be changed to be a bit higher, values from 100-800 are common.

In the searches tab, to get more accurate estimation a larger FFT box is recommended (it comes at the expense of calculation speed) and changing the min/max defocus values is a good idea too.

We are running GCTF on a single GPU, but the P4 GPUs on these nodes is capable of running two processes at once, so we will type 0:0 in the GPUs tab and tell GCTF not to ignore the ‘Searches’ parameters and equi-phase averaging generally yields better results.

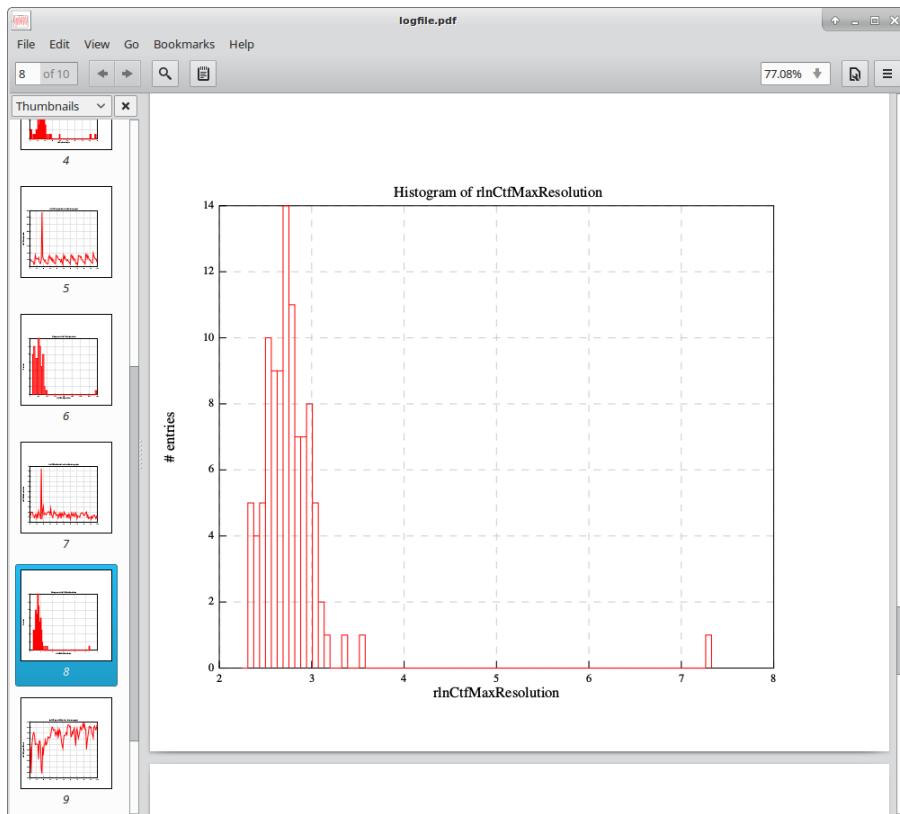
In the running tab we will run 2 MPI process to speed up the calculation.

Once Run! Is pressed, it should only take a few minutes to work through all the micrographs. To check the output of the Display tab will let you look at the power spectra of each micrograph (see below) and the fit that GCTF has calculated. This is a good diagnostic tool to check the program has done a good fit. The logfile.pdf also collated the data as a whole.



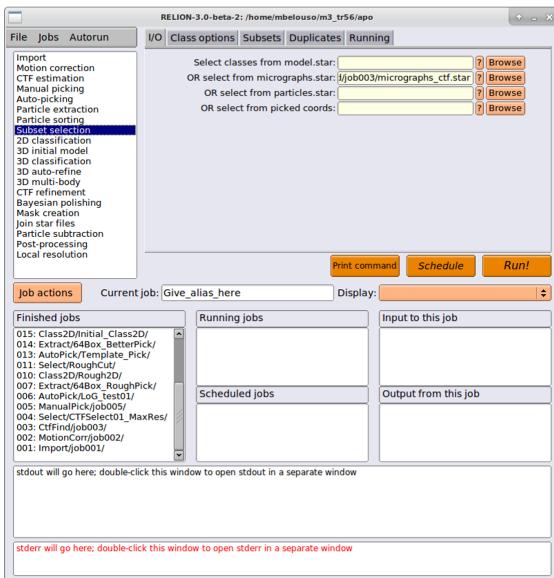
Selecting good micrographs:

As not all the micrographs are of excellent quality, after CTF estimation is a good time to remove any poor quality micrographs from further analysis. These outliers could be due to poor ice thickness or cracks in the grid or any of a number of other reasons. In this data set it was decided to only select micrographs that had a good correlation of thon rings in the power spectrum at high resolution. See below for the logfile output from CTF estimation. Note that in the histogram there are a few micrographs outside the gaussian distribution.

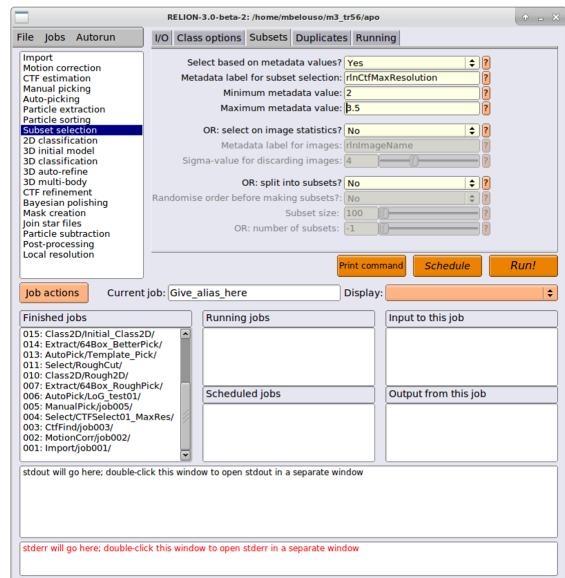


We will select based upon the flag: `rlnCtfMaxResolution`, however you can also run this job by manually going through all the power spectra and selecting the ones that fit well.

I/O



Subsets



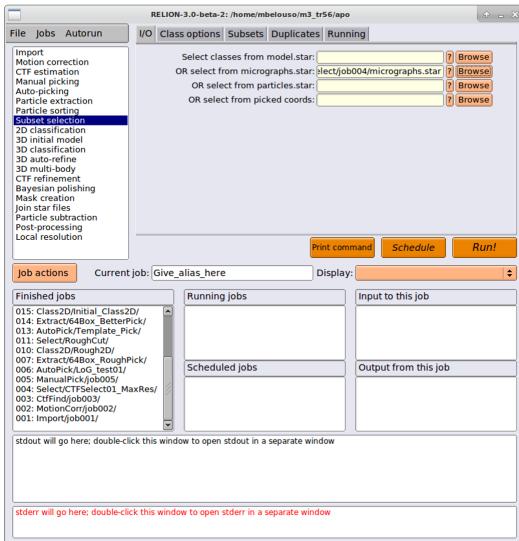
Particle Picking:

Now that we have a nicely curated set of motioncorrected micrographs we can begin the process picking the positions of the particles within these micrographs. There are many ways to approach this part of the data analysis and many excellent programs, but we will show a method that can yield results using the RELION workflow.

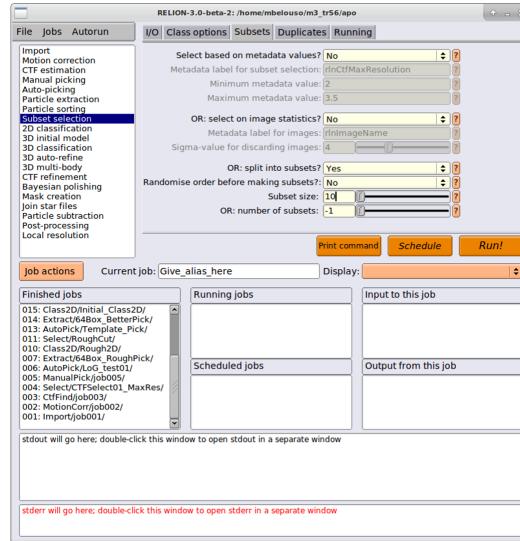
Firstly we want to make a subset of micrographs to pick initially, instead of picking the whole entire data set at once. While this is feasible with only 100 micrographs, most data collections will have significantly more micrographs and it is a waste of time and compute resources to do 1000's of micrographs in the first round of particle picking.

In the 'Subset Selection' tab we will point it to the selected micrographs from the last job we did, then use the 'Split into subsets' section to make a selection of 10 micrographs to initial pick and create 2D class averages from, see below:

I/O



Subsets

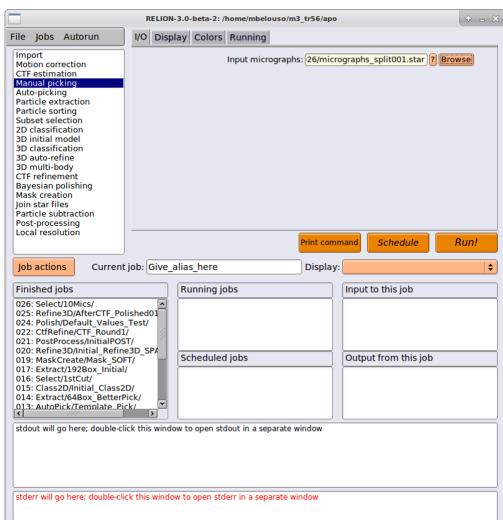


10 Micrograph subset selection

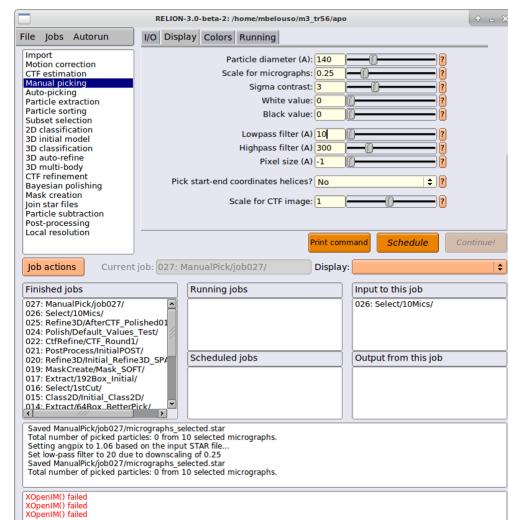
This will write out 10 different .star files each containing 10 micrographs (note: star files are a text file that RELION uses to keep track of data, initially they are quite simple just pointing RELION to where the micrographs are, but as each job progresses more and more data is added to them, such as defocus parameters and alignment angles).

The output of this subset selection will then be fed into a Manual Picking job, just select the first micrographs_split001.star as input.

I/O



Display

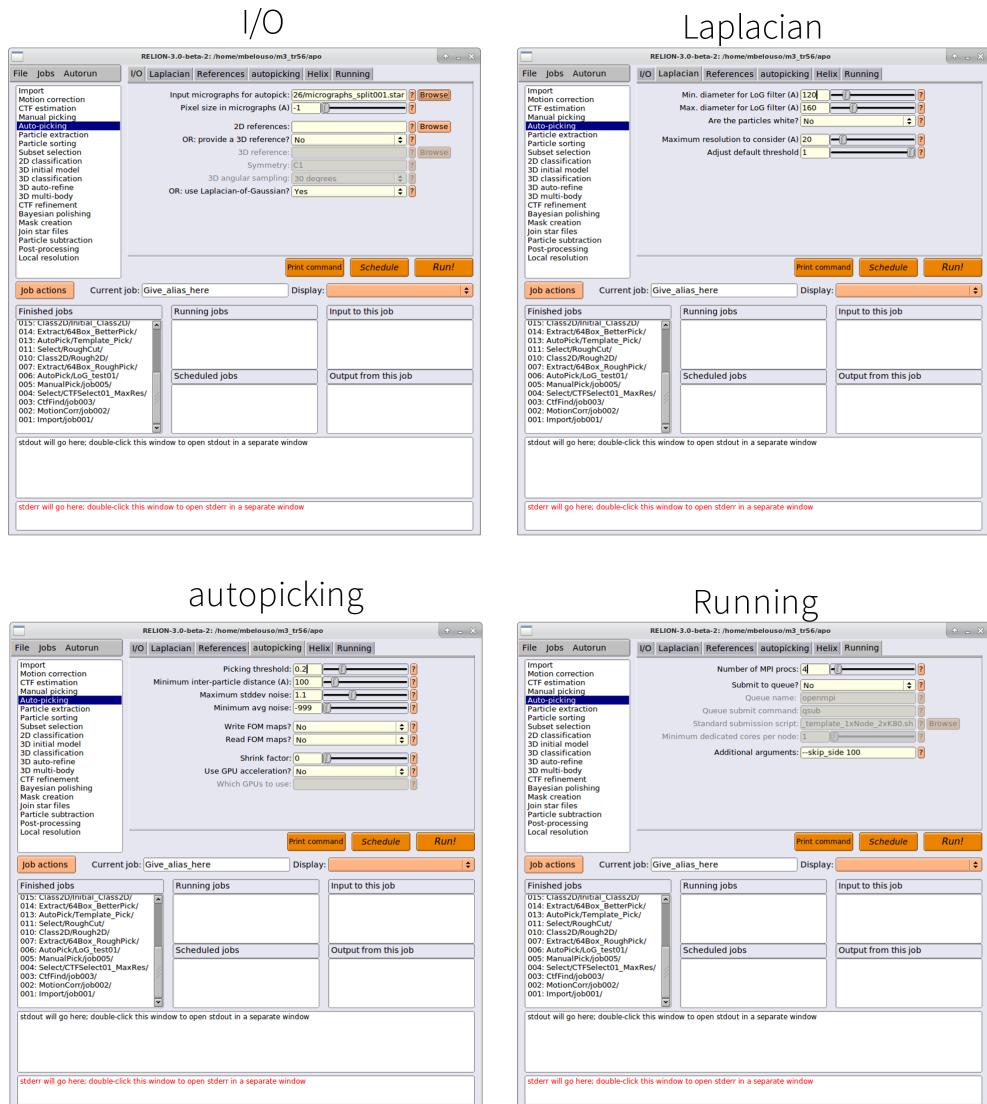


Manual Picking Settings

It is often good to adjust the low and high pass filters until you can clearly see your particles in the micrographs and at this stage also entering in the expected size of your molecule as a diameter in Angstrom.

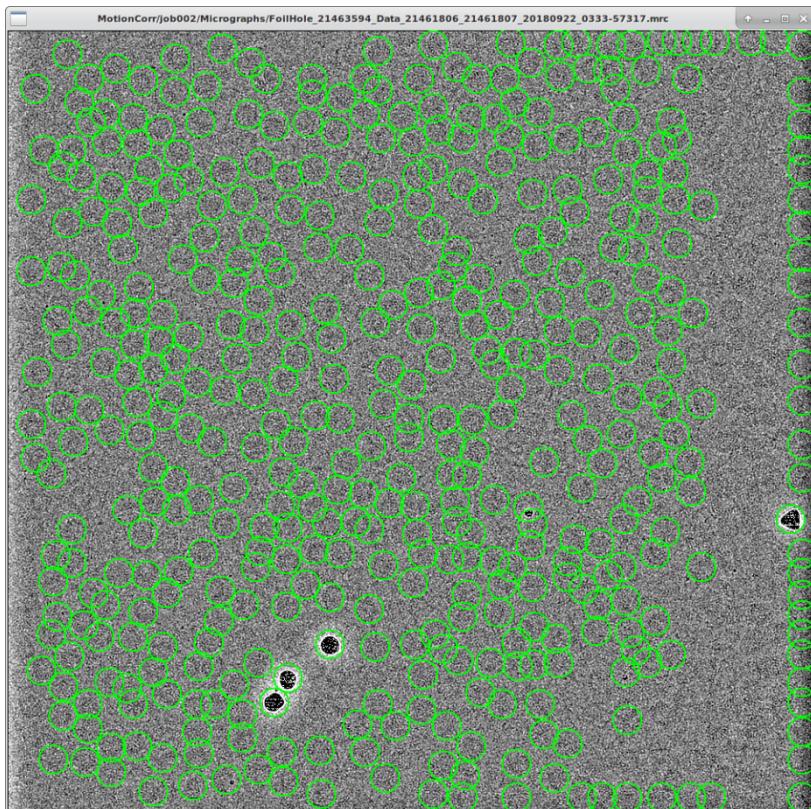
Laplacian of Gaussian Picker

The Laplacian of Gaussian (LoG) picker in RELION picks objects on the micrograph which have a Fourier Peak of a certain size. It is a crude method for particle picking, but is often useful if you don't have any previous references to help pick or you don't the tedious task of manually picking your data by hand.



LoG picking a subset of 10 micrographs

The output from this will be a set of coordinates corresponding to Fourier peaks on your micrographs that may/may not be your particles. See below (as some ice contaminants were also picked). However the next step would be extract these particles from the 10 micrographs and subject them to 2D classification. Changing a few settings like the 'Picking threshold' and the 'Adjust default threshold' will allow the picker to be more or less selective. Often multiple attempts at this job will be required to get the right picking conditions.

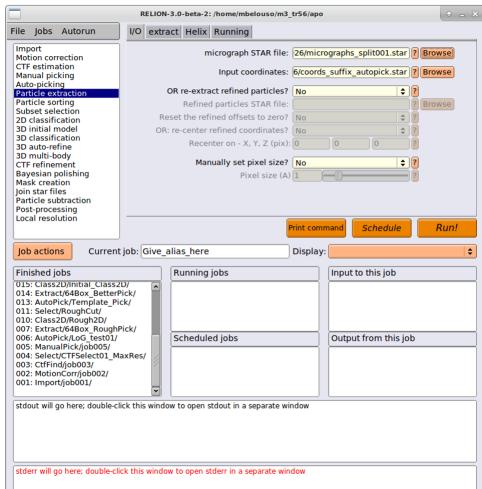


Example of picked coordinates from RELION LoG picker

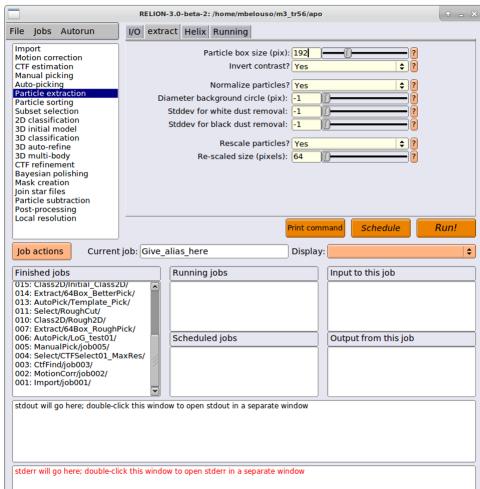
We will now extract the particles using the following in the Particle Extraction Tab (See below). The input will be the 10 micrograph subset as well as the output from the LoG picker. At this stage we will also Fourier Crop the particles to a smaller box size than the native micrograph so we can speed up calculation times (calculation speed is proportional to the square of the box size, so smaller boxes yield faster run times).

Also it is important to note at this stage that we have to make some choices as the appropriate box size. Firstly the particle is roughly 125 Å in diameter so the box size needs to be in the vicinity of ~1.5 times the particle size so as to encapsulate the CTF and enable a high quality reconstruction. It is also important that the box sizes be an even number and better if that even number has a large amount of factors, this speeds up the Fourier calculations and leads to faster run times.

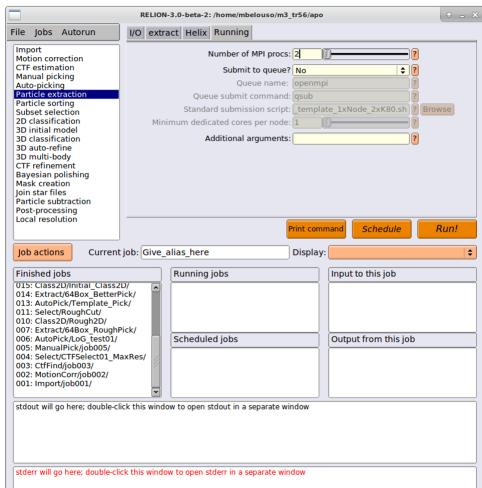
I/O



Extract



Running



Particle Extraction + Rescaling the particle dimensions

This will run quickly and the output of this job will feed into the first 2D classification (see below for setup). At this stage it is often useful to ‘Ignore CTFs until first peak’, which prevents lower resolution features of the particles driving alignments, which has the benefit of separating class averages a bit more effectively. Also a choice has to be made of how many classes to calculate, higher numbers lead to longer calculation times, but may be useful for separating particles and a mask diameter has to be chosen. We have suggested a slightly larger diameter than the molecule (which is common if you are uncertain as to the nature of your sample), but if you are sure you can use a tighter circular mask of ~135 Å for apo-ferritin.

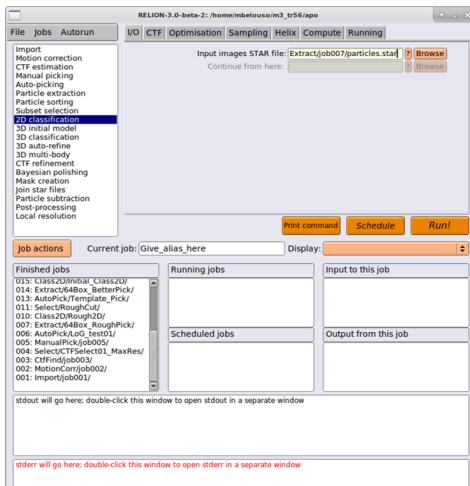
Limiting the resolution of alignments is also a useful tool for initial classification which is found in the Limit resolution E-step box (values from 8-15 Å is a good range to explore).

This 2D classification calculation is able to be accelerated by GPUs so in the compute tab, enabling GPU support should be set. Unlike MotionCor and GCTF, you will not need to tell RELION which GPUs to use, it will determine your hardware configuration automatically.

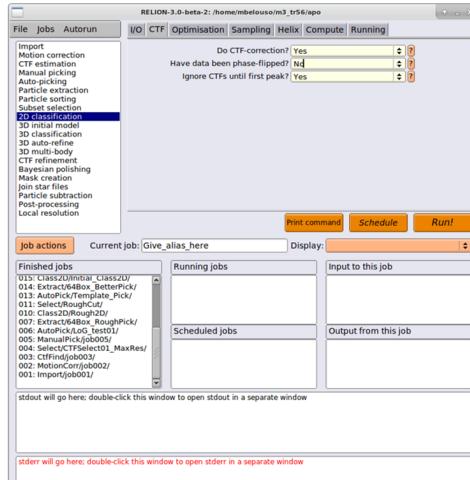
In the Running tab it is important for all further RELION jobs to consider carefully what to use for MPI ranks and threads.

For 2D/3D and 3D-autorefine, you must have n+1 MPI ranks. So for example on the P4 desktop sessions you will want to run 2 MPI ranks on the single GPU, so you need set the MPI tasks to 3 (2 running on the GPU and 1 master process that is responsible for keeping the whole calculation in check and synchronised). The threads is how many CPUs are assigned to a single MPI task. For 2D generally only one thread is useful per task, but for 3D having more per task is useful so as to enable more efficient useage of the computers main memory.

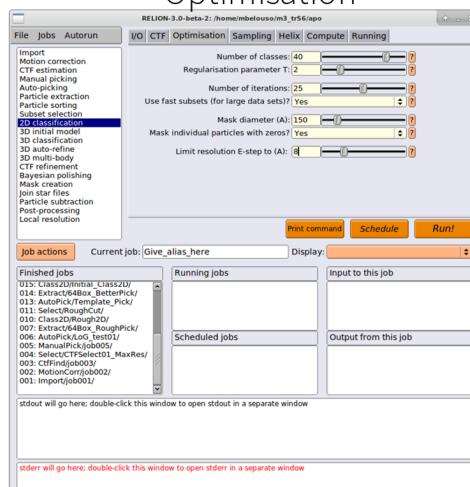
I/O



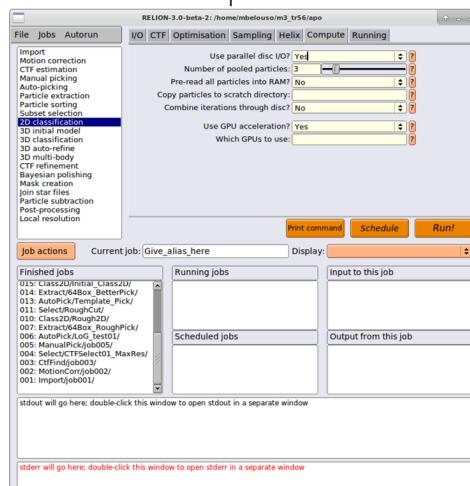
CTF



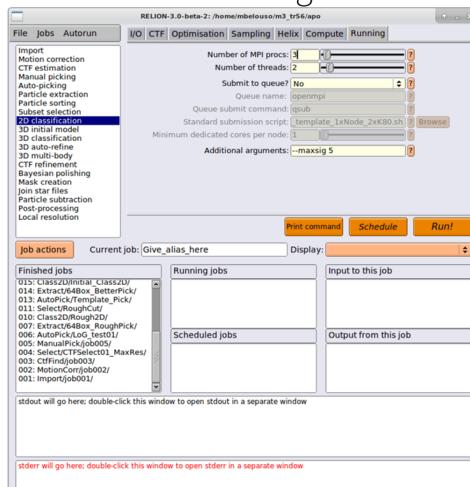
Optimisation



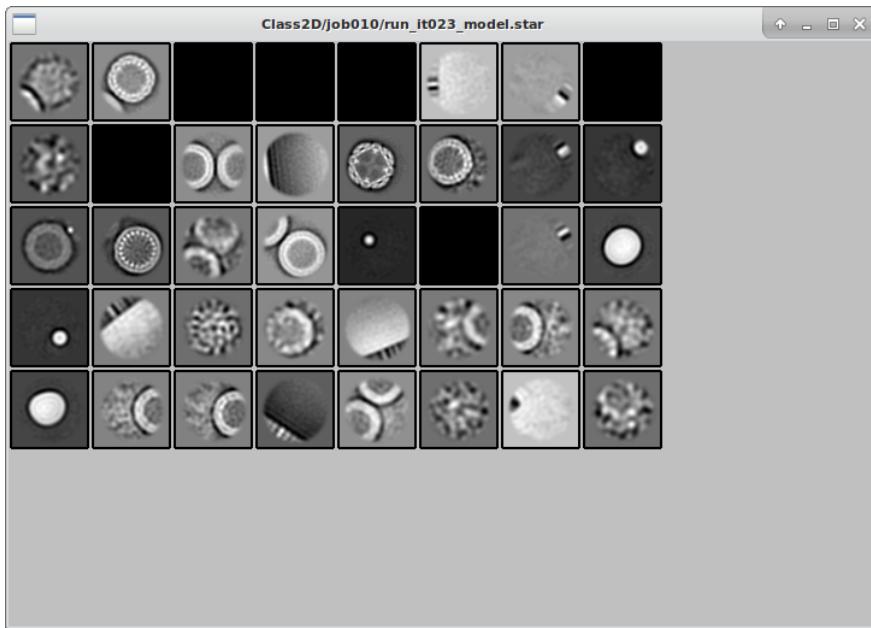
Compute



Running



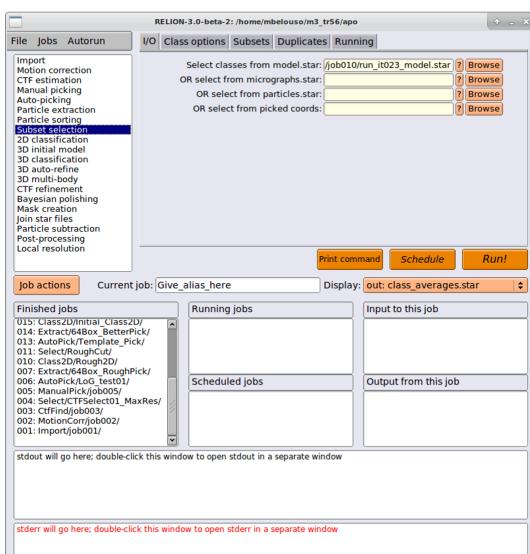
Initial 2D classification from LoG Picker



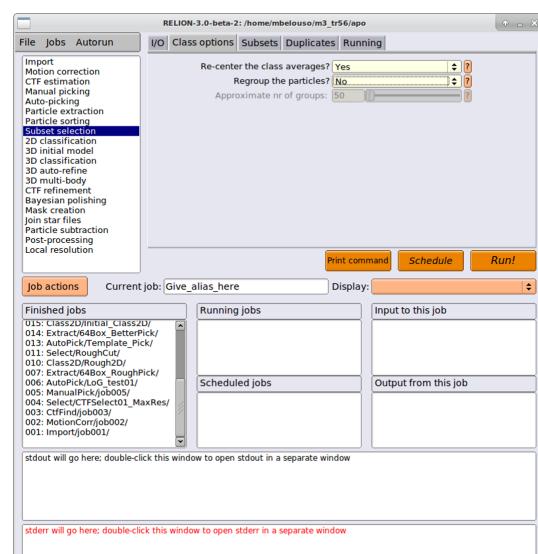
Output from Initial 2D Classification

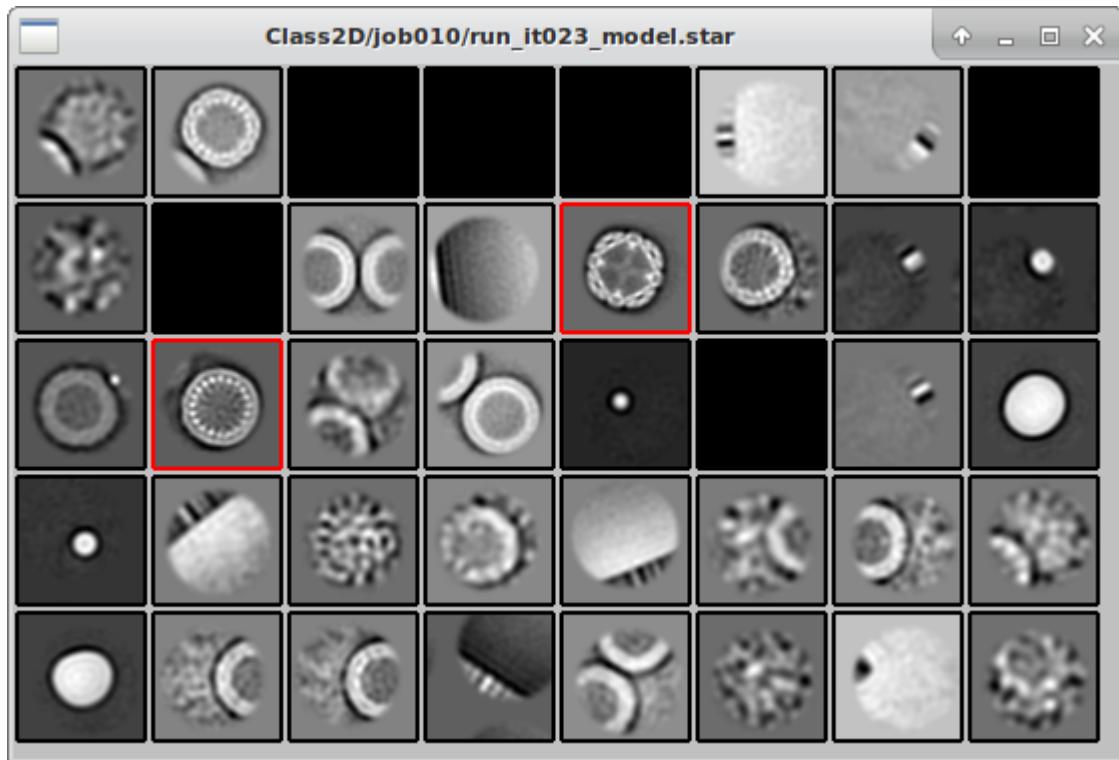
So now we have our first 2D classes we will select just two distinctly different classes to be used as templates to more accurately pick the particles from the whole dataset. This will be done using again the subset selection tab. The input will be the output of the Class2D and we will want to recenter the classes in the class options tab. The input will be the output of the Class2D and we will want to recenter the classes in the class options tab. When Run! Is pressed you will be able to see all the classes and select the ones you like with the left mouse and to save your selection you right click and select ‘save selected classes’

I/O



Class options





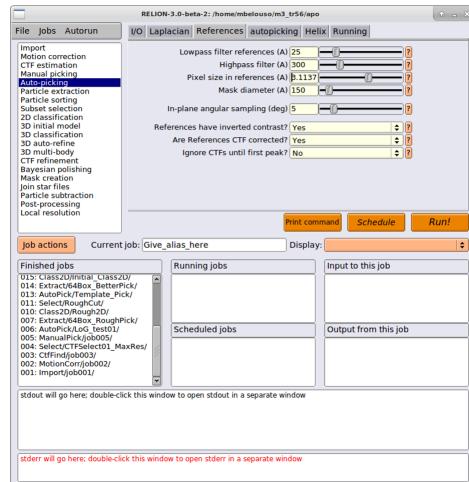
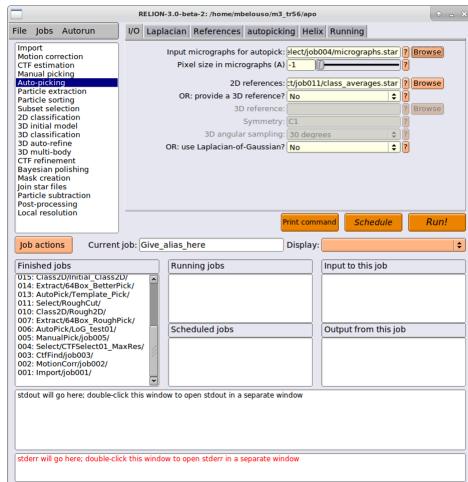
The two classes that were selected are highlighted in RED.

Template picker in RELION

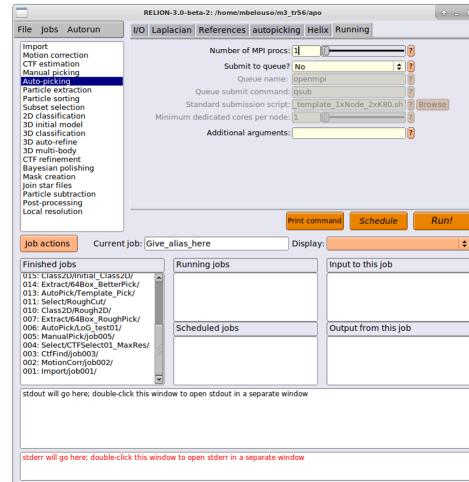
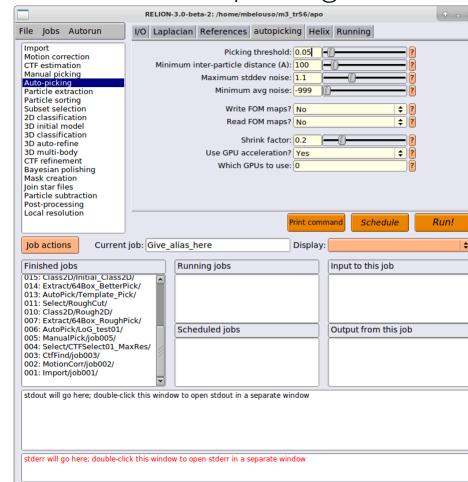
After selecting the appropriate templates from the first Class2D we can use these for Fourier template matching using RELION. This is a more accurate particle picker and will yield better results in terms of accurately picking the center of mass of the particle and discerning your particle from contaminants. We will use this to pick particles from the whole data set. See below for the optimal settings. Again this is a GPU accelerated process and as we only have a single GPU we will run a single process on that, but be aware that if you have a bigger system you can task multiple GPUs to get through the data quicker.

It is also important to note that the templates don't have to be the sample pixel size as the micrographs as is the case in this experiment, you just have to take note of the pixel size of your templates and enter it in the 'Pixel size in references' box.

I/O



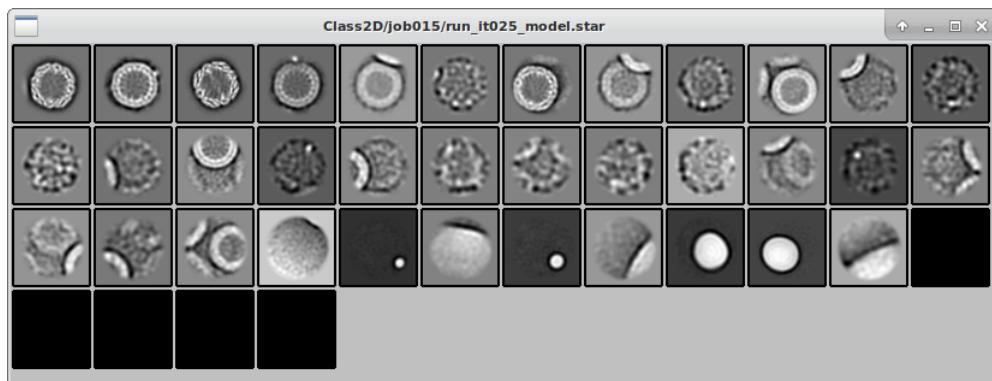
autopicking



References

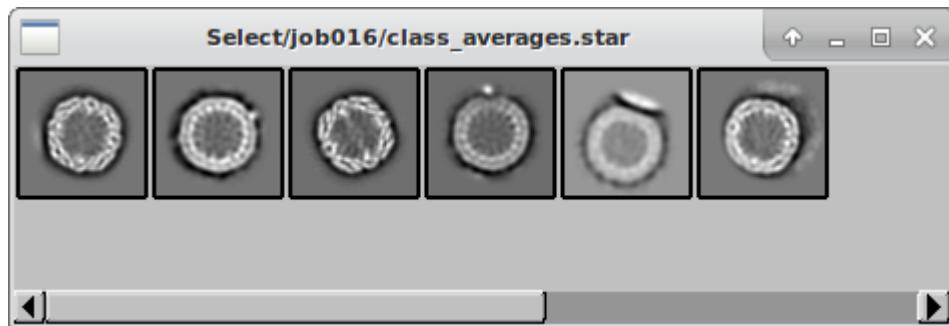
Template picker settings for RELION on whole data set

The output of this job can now be used to **extract particles** from the whole data set for another round of **2D classification**. Similar settings will be used as the ones above. See below for the output of the 2D classification of the whole data set.



Output of Class2D from whole data set picked with the RELION template picker

We will again use the subset selection tool to select the best classes for refinement. Below is an example of the classes that were selected.



Initial Model generation:

To begin refining our 2D projections into 3D space we need an initial model. There are lots of excellent pieces of software that can achieve this. To name a few would be CryoSPARC, SIMPLE 3.0, EMAN 2.2, cisTEM 1.0. However in this tutorial we will use RELION whose initial model generation uses a similar principle to the cryoSPARC software.

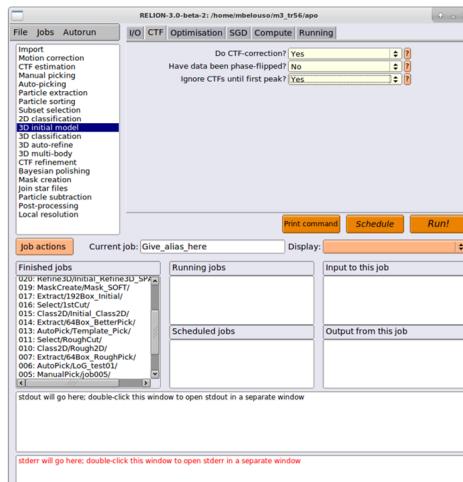
We will change a few of default parameters, both to reflect our molecule and hardware as well as change some parameters to enable a slightly faster calculation of an intitial model. It also must be noted that for Refine3D to work, you do not need a high resolution model, just one that reflects the general topology of your molecule. That is to say time is better spent in AutoRefine3D than in intitial model generation in most cases.

Also you will note that we refine the initial model with C1 point group symmetry even though apo-ferritin is an octahedrally symmetric molecule. This will be generally the case for most symmetric molecules, where you will enforce symmetry only once you are certain of it.

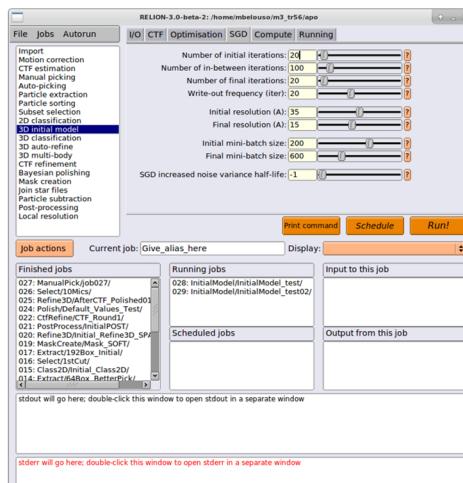
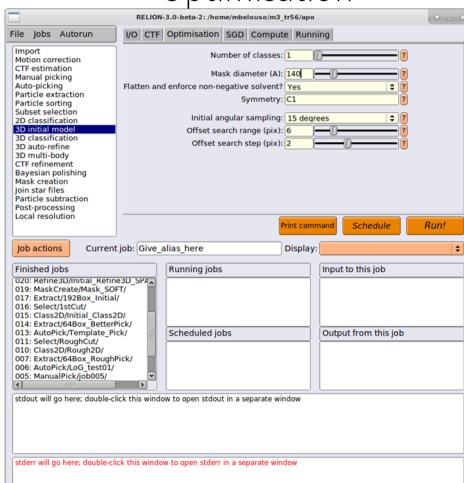
I/O



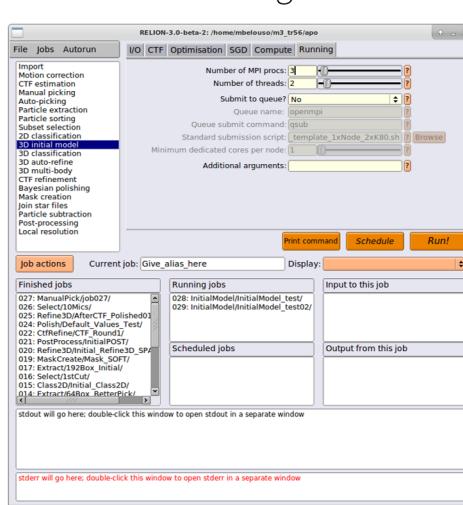
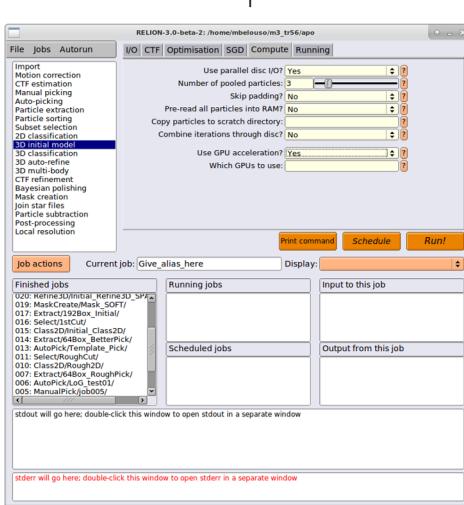
CTF



Optimisation



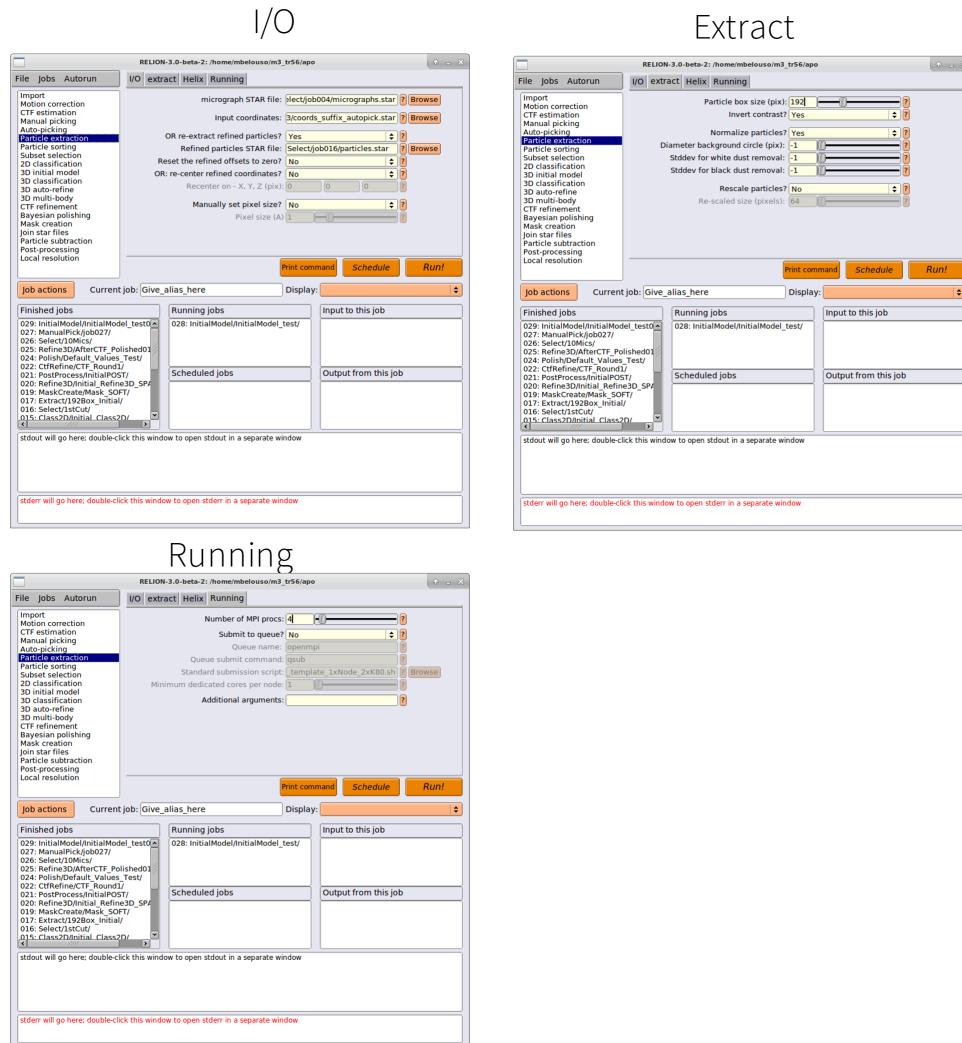
Compute



Initial model generation using Stochastic Gradient Descent as implemented in RELION

3D Autorefine

In this tutorial we will immediately go to 3D autorefine, but in other cases you may want to explore the structural heterogeneity of your sample by rounds of 3D classification. Firstly we will re-extract the good subset of particles we identified in Class2D at their native pixel size using the Particle Extraction tab:



Re-extraction of particles belonging to Good 2D classes at native pixel size

Before we begin any refinements we need to rescale and align our initial model to the correct point group symmetry. This is achieved through the command line with the following commands:

```
'relion_align_symmetry --sym O --i InitialModel/job0##/run_it140_class001.mrc --o 64box_O_Aligned.mrc'      (## will be the job number that your initial model job was run)
```

This will align your initial model to the octahedral point group symmetry axis that RELION uses.

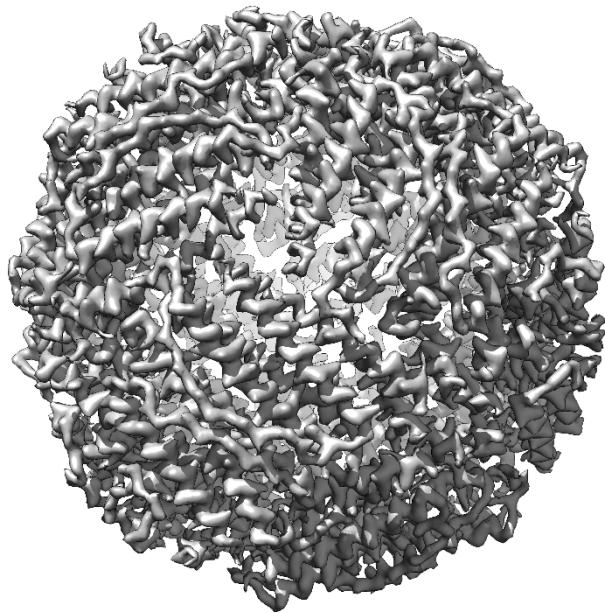
```
'relion_image_handler --rescale_angpix 1.06 --i 64box_O_Aligned.mrc --o startmodel_192box.mrc'
```

This will rescale your initial model to the native pixel size. Now in the root directory of your working folder you will have a file called “startmodel_192box.mrc” which will be your starting point for 3D-refinement.

At this point you may want to consider making a mask for 3-D refinement, which is a good idea if your particles are quite close together, however in this tutorial we will use a circular mask. See below for the setting up of 3D-autorefinement.

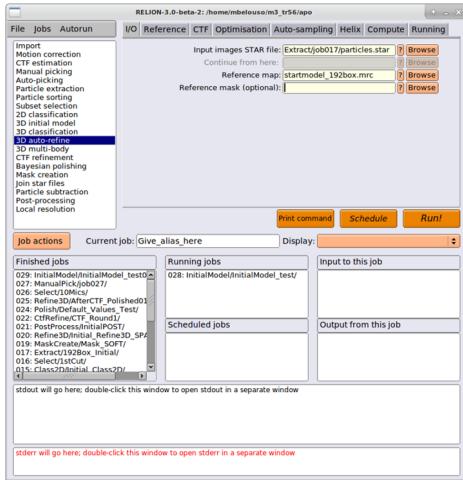
There are many settings in this routine. Again its GPU accelerated so ensure that that is enabled and that you have changed a few of settings as noted below. In general for high symmetry sometimes using a finer angular sampling is helpful where as at lower symmetries sometimes even starting with a 15 deg angular sampling can yield more accurate results. If your system has access to a fast local SSD for caching the particle stack, this should also be entered, but is not the case for the tutorial. Also take note for larger molecules with bigger box sizes, most GPUs will not allow more than one MPI task per GPU as you will run out of Video memory and the job will crash.

At the end of this step you should have a lovely high resolution map that looks like the figure below.

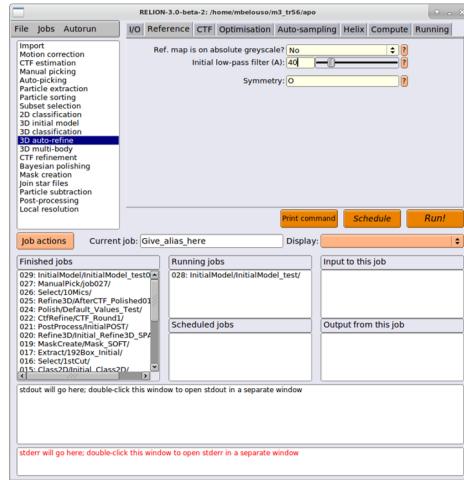


Initial output of RELION 3D-autorefine a ~2.5 Å map of apo-ferritin

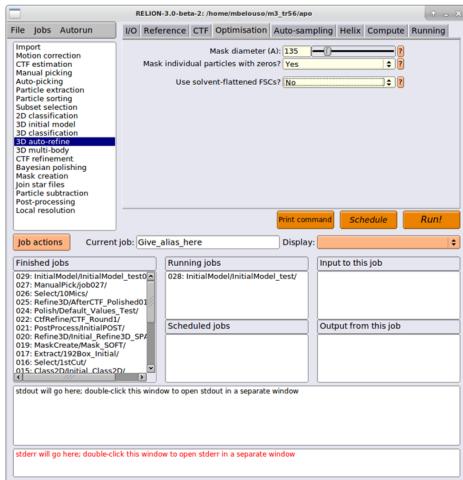
I/O



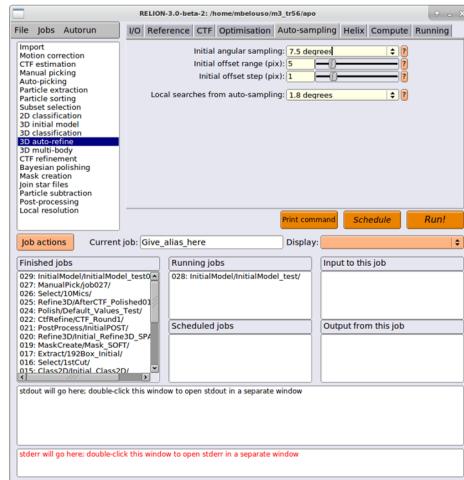
Reference



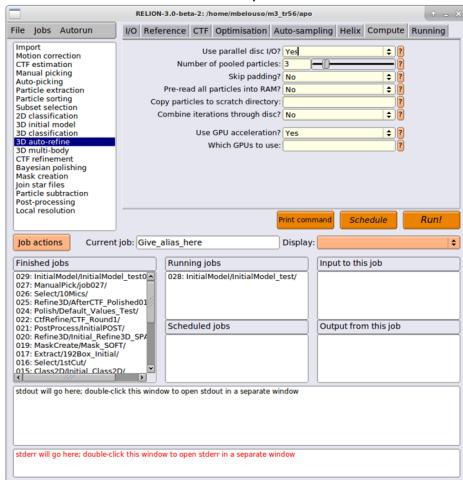
Optimisation



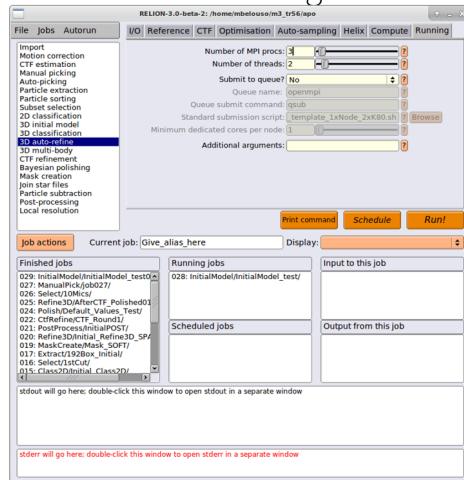
Autosampling



Compute



Running



Settings for Initial Refine3D

Visualising your Results:

A convient and powerful tool for map visulisation and manipulation is UCSF chimera. To use it on your P4 desktops you will type this into a terminal:

‘module load chimera/1.13’

Followed by:

‘vglrun chimera’

This will load chimera, then you can navigate to the Refine3D folder and open up the file called ‘run_class001.mrc’

At this point you may want to consider making a mask so you can post-process your map and calculate the gold-standard FSC, a means to estimate the resolution of your map.

