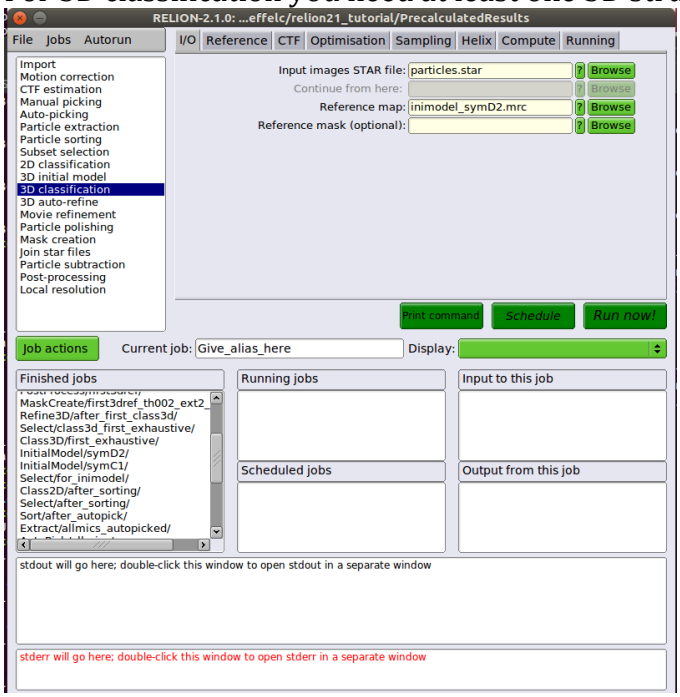
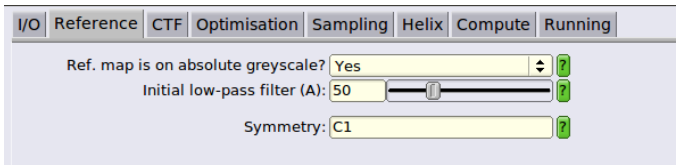
3D classification in Relion 2.1

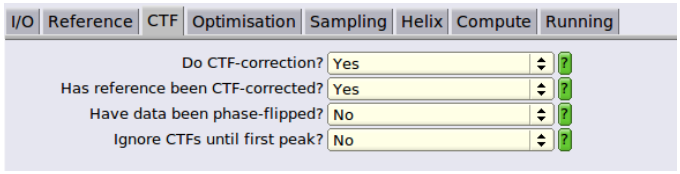
For 3D classification you need at least one 3D structure and particles as an input.



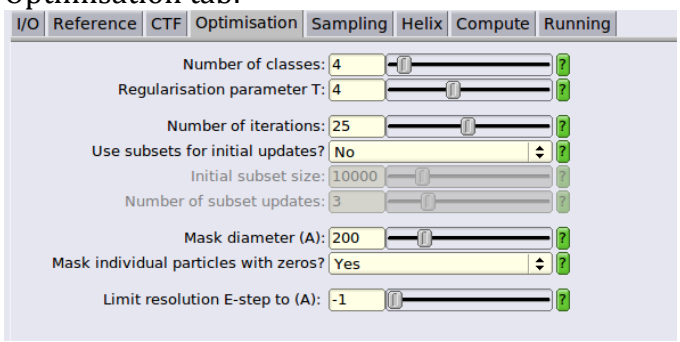
Reference tab:



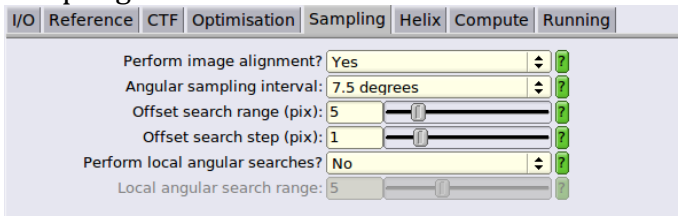
CTF tab:



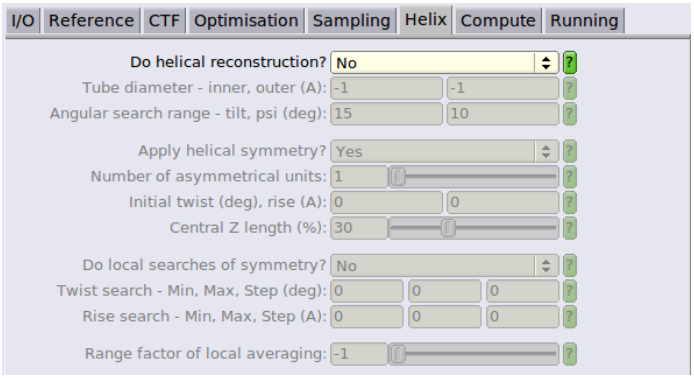
Optimisation tab:



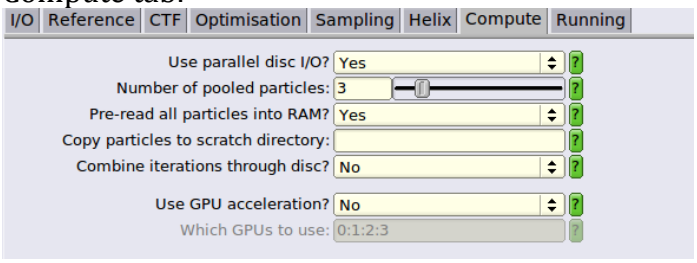
Sampling tab:



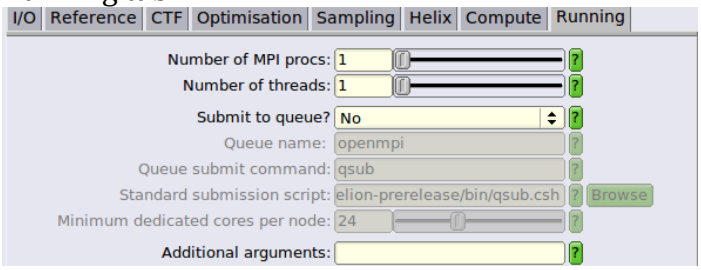
Helix tab:



Compute tab:



Running tab:



Now you could press run.
If the calculations take too long you can instead copy the precalculated run to your running directory and use these results:
cp –r PrecalculatedResults/Class3D/job018 Class3D/job018

Analysing the results and selection of the best 3D classes

There are several ways to analyze your 3D classes:
1. The Relion gui:
Choose the **Class3D/job018/run_it025_model.star** file in the **Subset selection** panel and display the classes as sections through the middle of the structures:

You can select the best classes and save the particles corresponding to these classes for your further calculations.

2. the statistics in your run_it025_model.star file:
in the terminal type
more Class3D/job018/run_it025_model.star

```
data_model_general

_rlnReferenceDimensionality                 3
_rlnDataDimensionality                      2
_rlnOriginalImageSize                     100
_rlnCurrentResolution                  14.750000
_rlnCurrentImageSize                       68
_rlnPaddingFactor                       2.000000
_rlnIsHelix                                0
_rlnFourierSpaceInterpolator               1
_rlnMinRadiusNnInterpolation              10
_rlnPixelSize                           3.540000
_rlnNrClasses                              4
_rlnNrBodies                               1
_rlnNrGroups                              15
_rlnTau2FudgeFactor                     4.000000
_rlnNormCorrectionAverage               0.510559
_rlnSigmaOffsets                        1.833866
_rlnOrientationalPriorMode                 0
_rlnSigmaPriorRotAngle                  0.000000
_rlnSigmaPriorTiltAngle                 0.000000
_rlnSigmaPriorPsiAngle                  0.000000
_rlnLogLikelihood                    1.347089e+08
_rlnAveragePmax                         0.901647


data_model_classes

loop_
_rlnReferenceImage #1
_rlnClassDistribution #2
_rlnAccuracyRotations #3
_rlnAccuracyTranslations #4
_rlnEstimatedResolution #5
_rlnOverallFourierCompleteness #6
Class3D/job018/run_it025_class001.mrc  0.080344  3.815000  0.767000  27.230769  0.999997
Class3D/job018/run_it025_class002.mrc  0.507154  2.412000  0.442000  14.750000  0.999020
Class3D/job018/run_it025_class003.mrc  0.234911  3.080000  0.561000  19.666667  0.999275
Class3D/job018/run_it025_class004.mrc  0.177591  3.305000  0.603000  25.285714  0.999998
.
.
.
```

the yellow marked column shows you how the particles are distributed between the classes. The remaining columns tell you something about the accuracy of alignment of the particles in the different classes, the estimated resolution of the classes and their 3D completeness (preferential orientation).
Also the value of `rlnLogLikelihood` should become smaller each iteration.

3. using Chimera
Open the 4 classes in chimera using the command line:
chimera Class3D/job018/run_it025_class*.mrc

You can estimate which classes are the best in the easiest way by looking at them.