第 1 步. 使用 relion 估算 micrographs 的 CTF 参数，生成 micrograph_ctf.star

Step 1. Estimate CTF parameters for micrographs by RELION and generate *micrograph_ctf.star*.

第 2 步. 将 star 文件转换为 GisSPA 可读的 lst 文件

Step 2. Convert *.star* files to GisSPA-readable *.lst* files

```
relion2lst_dont_write_mics.py CtfFind/job005/micrographs_ctf.star --lst micrographs_ctf.lst
```

```
#LST
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00002.mrc   defocus=0.8401005371    dfdiff=0.0128565918     dfang=145.699879
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00003.mrc   defocus=0.8660711426    dfdiff=0.0188164551     dfang=171.893494
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00004.mrc   defocus=0.86874335935   dfdiff=0.01908408205    dfang=170.796783
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00005.mrc   defocus=0.88477197265   dfdiff=0.02731279295    dfang=166.863052
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00006.mrc   defocus=0.84074106445   dfdiff=0.01255209965    dfang=174.929901
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00007.mrc   defocus=0.84785268555   dfdiff=0.01629379885    dfang=157.232521
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00008.mrc   defocus=0.87532441405   dfdiff=0.02016367185    dfang=156.715553
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00009.mrc   defocus=0.90362680665   dfdiff=0.02053491215    dfang=157.642876
0       MotionCorr/job003/micrographs/Yt_105kx_1202_00010.mrc   defocus=0.84755024415   dfdiff=0.01133764645    dfang=140.132988
```

第 3 步. 为了节约计算时间，可以先对图进行压缩，根据像素大小可以选择 bin2

或者 bin4

Step 3. To improve computation efficiency, you can first compress the micrograph, and depending on the pixel size you can choose bin2 or bin4

```
cd MotionCorr/job003/micrographs/
```

```
mkdir ms2
```

```
foreach file(*.mrc)
e2proc2d.py $file ms2/$file --meanshrink=2
end
```

or

```
foreach file (*.mrc)
relion_image_handler --i $file --o ms2/$file --angpix 1.36 --rescale_angpix 2.72
end
```

替换 micrograph_ctf.lst 的内容:

Replace the path of micrograph:

```
sed -i 's/micrographs/ms2\/micrographs/g' micrographs_ctf.lst
```

```
#LST
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00002.mrc       defocus=0.8401005371    dfdiff=0.0128565918     dfang=145.699879
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00003.mrc       defocus=0.8660711426    dfdiff=0.0188164551     dfang=171.893494
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00004.mrc       defocus=0.86874335935   dfdiff=0.01908408205    dfang=170.796783
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00005.mrc       defocus=0.88477197265   dfdiff=0.02731279295    dfang=166.863052
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00006.mrc       defocus=0.84074106445   dfdiff=0.01255209965    dfang=174.929901
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00007.mrc       defocus=0.84785268555   dfdiff=0.01629379885    dfang=157.232521
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00008.mrc       defocus=0.87532441405   dfdiff=0.02016367185    dfang=156.715553
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00009.mrc       defocus=0.90362680665   dfdiff=0.02053491215    dfang=157.642876
0       MotionCorr/job003/ms2/micrographs/Yt_105kx_1202_00010.mrc       defocus=0.84755024415   dfdiff=0.01133764645    dfang=140.132988
```

第 4 步. 将三维模型 rescale 成与 micrograph_ctf.lst 相同的像素大小

Step 4. Rescale the 3D template to the same pixel size to the binned pixel size

```
relion_image_handler --i 3D_template.mrc --o 3D_template_rescaled.mrc --angpix ?? --rescale_angpix 2.72
```

第 5 步. 生成二维投影

Step 5. Generate 2D projections.

```
project3d 3D_template_rescaled.mrc out=2D_projections.hdf list=euler_C1_step5.lst
```

project3d 是 EMAN 软件包的一个用于创建二维投影的程序，下载 EMAN 即可使用（https://blake.bcm.edu/emanwiki/EMAN1）。用于投影的欧拉角文件可在测试数据中下载。

project3d is a program for creating 2D projections from the EMAN package, EMAN software package can be downloaded from https://blake.bcm.edu/emanwiki/EMAN1. The Euler file can be downloaded from the test Data dir.

第 6 步. 写 config 文件。config 文件可在测试数据中下载，里面包含一些固定参数和可调参数

Step 6. Write config file, including some optional parameters.

```
input           = micrographs_ctf.lst
template        = 2D_projections.hdf
eulerfile       = euler_C1_step5.lst
angpix          = 2.72
phistep         = 3
kk              = 3
energy          = 300
cs              = 2.7
Highres         = 8
Lowres          = 400
diameter        = 160

norm_type       = 1
invert          = 1
threshold       = 6.2
output          = Output/test.lst
first           = 0
last            = 1
window_size     = 512
GPU_ID          = 0
phase_flip      = 1
overlap         = 160
```

input = input micrograph lstfile with ctf information
template = input 2D projections templates in .hdf format
eulerfile = euler file with euler values
angpix = input pixel size in angstroms

phistep = inplane rotation sampling
kk = overlapping density parameter, default is 3.
energy = accerlerating voltage in kV.
cs = spherical aberration in um.
Highres = high resolution cut
Lowres = low resolution cut
diameter = target diameter in pixels Optional Parameters.
norm_type = CCG norm optional parameter, 1 for global normalization, 0 for per-window normalization
invert = whether to invert contrast in micrograph, 1 for yes, 0 for no.
threshold = cc threshold value, only output score beyond this value
output = output lstfile filename
first = the first image id to process.
last = the last image id to process.
window_size = the window size which is splitted from raw IMG, which must satisfy window_size % 32 == 0.
GPU_ID = ID of GPU device, only support single ID.
phase_flip = Whether do filtering on images, operation(1) or not(0), in case of input being filtered already).
overlap = size of overlap between different windows.

第 7 步. 运行程序进行颗粒探测

Step 7. Run program and do target detection.

```
~/GisSPA-main/main config
```

第 8 步. 合并重复探测的颗粒

Step 8. Merge duplicated particles

```
remove_repeat_particles_from_list.py Output/test.lst 1 4 8 Output/test_merge.lst
```

remove_repeat_particles_from_list.py
<input1> <number of images> <euler_thres> <output>
<input1> lst filename generated from target detection
<number of images> the number of micrographs used for detection in this input lst file
* * the threshold of coordinate (x,y) range used to merge detected targets
<euler_thres> the threshold of Euler angle (x,y) range used to merge detected targets

第 9 步. 将合并后的颗粒文件格式转换为 relion 可读的 star 文件

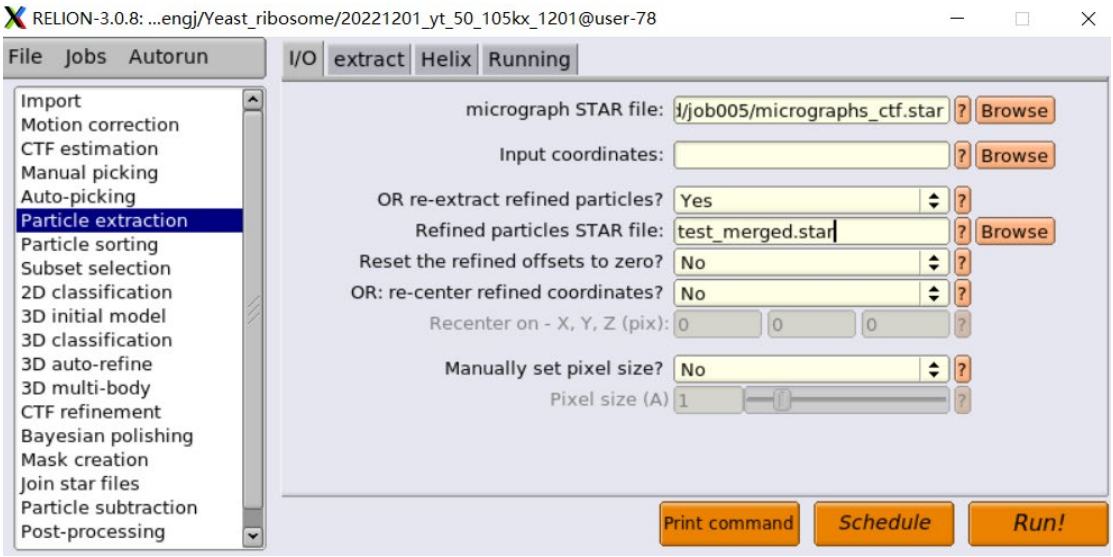
Step 9. Convert the merged lst file to the RELION-readable star file

```
convert_my-lst_to_relion_mic.py Output/test_merge.lst 2 1.36 test_merged.star MotionCorr/job003/micrographs/
```

convert_my-lst_to_relion_mic.py
<input lstfile> <scale factor> <pixel size> <out star file> <micrographs dir>
<input lstfile> merged lst file
<scale factor> binning factor when target detection, 2 for bin2
<pixel size> the un-binned pixel size of original micrograph
<out star file> filename of output star file
<micrographs dir> the directory of original micrographs.

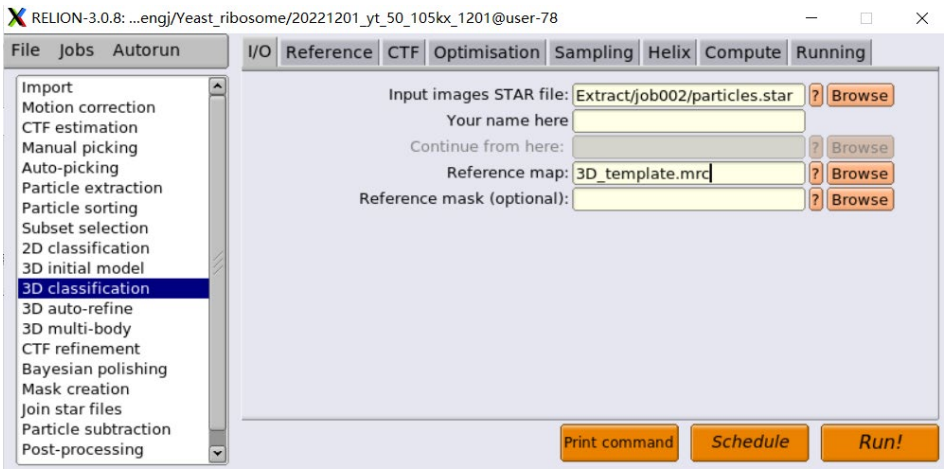
第 10 步. 提取颗粒。使用 relion 的 re-extract 功能从 micrograph 中提取颗粒

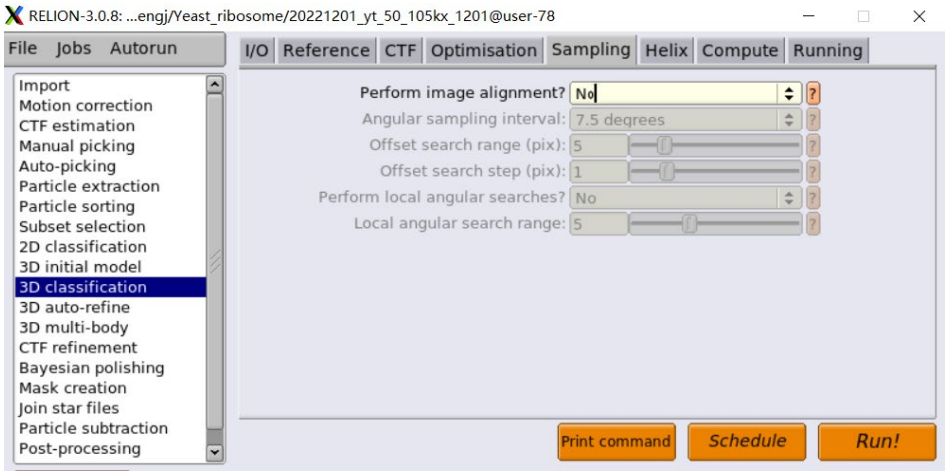
Step 10. Particle extraction by RELION.

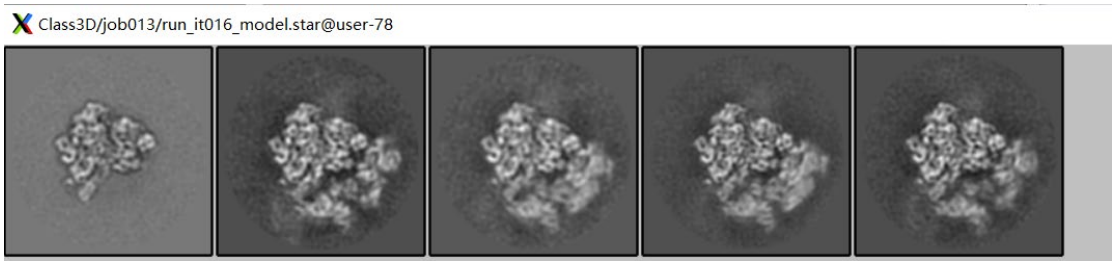


第 11 步. 三维分类

Step 11. 3D classification







后续的操作和分析都与单颗粒的优化类似，但是必须是 local refinement，一定不能跑全局 refinement ！!!

The subsequent operations and analysis are similar to the single particle analysis, but only local refinement is allowed, never try global refinement !!!

如果需要消除模型偏差，可使用 jalign 计算颗粒与重构之间的相位残差，计算的频率范围为：低频为 8 埃，高频由重构的分辨率决定。jalign 是 JSPR 软件包的一个程序，JSPR 下载链接为：https://jiang.bio.purdue.edu/jspr/

In case of model bias, the phase residuals between the particles and the reconstruction can be calculated using jalign for a frequency range of 8Å for low frequency cut and high frequency cut is determined by the resolution of the reconstruction. *jalign* is a program from the JSPR package, the JSPR can be downloaded at: https://jiang.bio.purdue.edu/jspr/

(1) 首先将 run_data.star 转换为 lst 格式的文件，注意需要将 subset 1 和 subset 2 分开

(1) First convert run_data.star to a file in lst format, note that subset 1 and subset 2 need to be separated

```
relion_star_handler --i run_data.star --o run_data_subset1.star --select rlnRandomSubset --minval 0.9 --maxval 1.1
relion_star_handler --i run_data.star --o run_data_subset2.star --select rlnRandomSubset --minval 1.9 --maxval 2.1

relion2lst_dont_write_particles.py run_data_subset1.star --lst run_data_subset1.lst --boxsize ???

relion2lst_dont_write_particles.py run_data_subset2.star --lst run_data_subset2.lst --boxsize ???
```

（2）计算相位残差

（2）calculate phase residuals

```
$PATH/jalign run_data_subset2.lst run_class001.mrc out.lst --first $firstparticleID --last $lastparticleID
 --apix $pixelsize --aligner refineMicrographDefocus:batchsize=1:defocusrange=0.001:maskradius=$targetradius
:masksoft=-4:precision=0.001:stepdefocus=0.001:useOrigDefocus=1:verbose=1 --cmp phaseResRange:
ampweight=0:ctfampweight=1:lowRes=8:highRes=$resolution:abs=0 --batchsize -1 --force 1
```

（3）选择合适的阈值，保留相关颗粒后重新转回 relion 格式

（3）Select particles with score above one threshold and re-convert it to star format

```
delete_bad_score.py $jspr_scored_lstfile $score_threshold $output_filename

jspr_refine_2_relion_class3d.py $selected_particles_lst $box_size $pixel_size $output_star_file 0/1 0/1/2
```