



High-Resolution cryoEM and single-particle-analysis

beta 2017/09/01

Short Tutorial

* a more comprehensive SPHIRE tutorial can be downloaded at: <http://www.sphere.mpg.de>

1. Download the tutorial data at:

https://ftp.gwdg.de/pub/misc/sphire/test_dataset/sphire_testdata_movies.tar

2. Extract the downloaded file. At the terminal type:

```
tar -xvf sphire_testdata_movies.tar
```

2. The project directory SPHIRE-demo will be created.

3. Change to this directory. At the terminal type:

```
cd SPHIRE-demo
```

4. Type `sphire &` to launch the SPHIRE GUI.

STEP 1. Register Settings.

The first step is to provide the project-wide constant-value parameters.

1. Press the Project Pictogram.

2. Provide following parameters:

Project Name: TcdA1

Micrograph Pixel Size [Å]: 1.14

CTF window size [pixels]: 512

Particle box size [pixels]: 352

Protein particle radius [pixels]: 145

Point group Symmetry: C5

Protein molecular mass [kDa]: 1400

Imaging Configurations: Titan Dortmund

3. Press the Register settings button.

STEP 2. Micrograph Movie alignment

The folder **Movies** contains **112** low-dose raw movies of TcdA1 in the file format mrc. To see the content of this folder, type:

```
ls Movies/*.mrc
```

The next step is to align these movies and apply dose weighting using unblur.

1. On the main window of the SPHIRE GUI, press the pictogram **Movie** on the left and then the button **Micrograph Movie alignment** (Utilities) in the middle.

Provide following parameters:

Unblur executable file:

[your_path_to_unblur_binary/unblur_openmp_7_17_15.exe](#)

Input Micrograph path pattern: Movies/TcdA1-*_frames.mrc

(select an image and replace the number with a wildcard)

Output directory: CorrectedSums

Summovie executable path:

[your_path_to_summovie_binary/sum_movie_openmp_7_17_15.exe](#)

Number of movie frames: 24

Microscope voltage [kV]: 300

Pre frame exposure [$e/\text{\AA}^2$): 2.5

Number of threads: 24

2. Provide parameters for the job submission:

Check the *submit job to queue* checkbox.

Job name: sxunblur

Submission command: qsub

Submission script template: msgui_qsub_dedicated24.sh

(we use this template script to submit a single process with 24 threads on our cluster using SGE. Various template files and instructions to prepare an

appropriate submission script for your own cluster environment, can be found here:

<http://sphire.mpg.de/wiki/doku.php?id=howto:submissions>

3. Press the *Run command* button.

4. Type `qstat` to check the status of the job (valid for SGE queues).

5. List the files in the project directory and sort them by time, with:

```
ls -lrt
```

6. To keep track of the progress of the `sxunblur` job, type:

```
tail -f sxunblur.ojob_id
```

7. Check the content of the `sxunblur` output folder.

```
ls -lrt CorrectedSums
```

STEP 3. Drift assessment

Now we will analyze the results of unblur and select the images with the lowest amount of drift, using SPHIRE's drift assessment tool.

1. On the main window of the SPHIRE GUI, press the pictogram **Movie** on the left and then the button **Drift assessment** in the middle.

Select the input shift list file

Input Shift list file: CorrectedSums/unblur_shiftfiles.txt

and press the **Run Command button** to open the drift assessment tool.

2. Check the *Overall Drift Histogram* Checkbox to open the plot for the overall drift.

3. Right click on the plot to set the discard threshold (red line) to 6 Å.

4. Press the *Register* button to register this threshold.

5. Click the Apply settings marked as criterion button

6. Type Tutorial as the prefix name for the output list files and click the select output directory and save selection button to specify an output directory with the name DriftAsses

Check the content of the output folder DriftAsses

STEP 4. CTF estimation

Now we will estimate the CTF of the selected motion corrected micrographs that are included in the list `Tutorial_selected.txt`. These are the selected micrographs from the previous step with the lowest amount of drift. Note that the CTF estimation will be performed exclusively on the **not** dose-weighted micrographs.

1. Go to the main window of the SPHIRE GUI and press the button CTER on the left and then the button CTF estimation in the middle.

Provide following parameters:

Input micrograph path pattern:

CorrectedSums/corrsum/TcdA1-*_frames_sum.mrc

Output directory: CTFest

Micrograph selection file: DriftAsses/Tutorial_selected.txt

Microscope spherical aberration (Cs): 0.01 (Cs corrected)

Microscope Voltage (kV): 300

Amplitude Contrast [%]: 10

Lowest frequency [1/Å]: 0.0285 (from 40 Å ...)

Highest Frequency [1/Å]: 0.285 (... to 4 Å)

MPI processors: 48

Submit jobs to queue: Checked!

Job name: sxcter

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

2. Press the Run Command button

3. Monitor the progress of the sxcter job.

4. Once the job has finished, check the content of the Output folder CTFest.

5. The CTF results are stored in the file `CTFest/partres.txt`

Type `cat CTFest/partres.txt` to check the content of the file.

STEP 5. CTF assesment

The CTF results can be analyzed using SPHIRE's CTF Assessment tool. Using this tool, multiple criteria can be used simultaneously, in order to remove outlier images that might have a negative impact on the final result.

Here, we will perform only a rather simplified screening by setting a threshold to discard outlier images with the lowest astigmatism frequency limit.

1. On the main window of SPHIRE, press the CTER button on the left and then the CTF assesment button in the middle.
2. Select the CTF parameter file partres.txt in the CTFFest folder and press the Run command button to launch the GUI tool.
3. Check the Sync. Sort Option on the Control Panel window.
4. Select the Astigm. Freq. Limit as parameter from the Histogram & Plot Settings Combo box.
5. Check the Histogram Plot window.
6. Left click on the Histogram window to set a low cut-off threshold (blue line) at 4 Å (corresponds to 0.285 [1/Å], you can also type this value at the respective interface), in order to discard micrographs below this limit.
7. Press the apply all thresholds button.
8. Control the number of unchecked micrographs.
9. Enter Tutorial in File Suffix and press the Save Selection Button.
10. Check the content of the folder CTFFest.

STEP 6. Particle Picking

The next step is to pick and extract particles from the **dose-weighted** micrographs. For this purpose we will use the gauss particle picker in e2boxer to pick the particles automatically.

1. First, move all discarded images (Drift & CTF assessment) into a different folder. At the terminal, type:

```
dir_path=CorrectedSums/corrsum_dose_filtered/;mkdir ${path}/DISCARDED
```

```
while read file; do mv ${dir_pathpath}/${file} ${dir_path}/DISCARDED/; done < CTFest/Tutorial_micrographs_discard.txt
```

```
while read file; do mv ${dir_path}/${file} ${dir_path}/DISCARDED/; done < DriftAsses/Tutorial_discarded.txt
```

2. Press the button Window on the left of the SPHIRE main window and then the Particle Coordinates button in the middle.

3. Select Micrograph TcdA1-0001_frames_sum.mrc and press the Run Command button to launch e2boxer.

4. Select the boxing tool Gauss Boxer.

5. Set Input and Output Pixel size to 1.14.

6. Check the Invert Contrast and uncheck the Use Variance Option.

7. Set Gauss Width Adjust to 1.222, Threshold low to 0.05, threshold high to 0.17 and press the Run button.

8. Control picking quality and adjust settings if necessary.

9. Once finished, press the Done button.

10. At the terminal, type:

```
e2boxer_old.py --indir CorrectedSums/corrsum_dose_filtered/*.mrc --write_dbbox -  
-gauss_autoboxer CorrectedSums/corrsum_dose_filtered/TcdA1-  
0001_frames_sum.mrc
```

in order to pick all images in the corrsum_dose_filtered directory automatically, using these settings. The coordinates are stored in the .box file format in your project directory.

11. Make a folder named Coordinates and move all .box files in this folder.

At the terminal, type:

```
mkdir Coordinates
```

```
mv *.box Coordinates/
```

STEP 7. Extract picked particles

1. Go back to the main window of the SPHIRE GUI and press the button window on the left and then the Particle Extraction button in the middle.

Provide following parameters:

Input Micrograph path pattern: CorrectedSums/corrsum_dose_filtered/TcdA1-*_frames_sum.mrc

Input Coordinates path pattern: CorrectedSums/corrsum_dose_filtered/TcdA1-*_frames_sum.box

CTF parameter source: CTFest/partres.txt

Output Directory: Particles

Micrograph selecting list: CTFest/Tutorial_micrographs_select.txt

Coordinate file format: eman1

Mpi Processors: 48

job name: sxwindow

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

2. Check the progress of the job.

3. Once the job has finished, combine the separate particle stacks for each micrograph into a single virtual dataset. Go back to the main window of the SPHIRE GUI and press the button window on the left and then the Particle Stack button in the middle.

Set following parameters:

Output virtual image stack: bdb:Particles/stack

Input BDB image stack pattern: Particles/mpi_proc_*

and press the Run Command button.

4. Once the job finished, check the total number of particles in the resulting dataset. At the terminal, type:

```
e2iminfo.py bdb:Particles/stack
```

5. Display the virtual stack using the utility Display Data.

STEP 8. ISAC – 2D Classification

We will use the ISAC approach to obtain validated 2D class-averages from this dataset.

1. Press the ISAC button on the left and then the ISAC- 2D Clustering button in the middle.

Provide following parameters at the respective GUI interface:

Input image stack: *bdb:Particles/stack*

Output directory: *Class2D*

Images per Class: *100*

Phase-Flip: *Yes*

MPI processors: *59*

Submit jobs to queue: *Checked!*

Job name: *sxisac2*

Submission command: *qsub*

Submission script template: *msgui_qsub_fillup.sh*

and press the Run Command button.

2. Check the progress of the job

3. Once the job has finished, display the resulting 2D class averages (Class2D/class_averages.hdf and Class2D/ordered_class_averages.hdf) using the utility Display Data.

4. Delete “bad” class averages from the file class_averages.hdf with the Display Data Utility and store the remaining class averages into a new file named Class2D/best.hdf.

Does this dataset contain a single population of particles?

5. Type:

`e2iminfo.py Class2D/best.hdf` to check the number of averages in this file.

STEP 9. Beautifier – 2D Sharpening

The particles of the dataset have been resized during ISAC (radius from 145 to 29 pixels, box size from 352 to 76 pixels). We will now use SPHIRE's Beautifier tool to resample these class averages to their original pixel size, refine and sharpen them, in order to reveal high-resolution features.

1. Press the ISAC button on the left and then the Beautifier button in the middle.

Provide following parameters at the respective GUI interface:

Original image stack: `bdb:Particles/stack`

Isac Output directory: `Class2D`

Output directory: `Class2D-beaut`

Particle Radius: `145`

MPI processors: `59`

Submit jobs to queue: `Checked!`

Job name: `sxcompute_sxisac_avg`

Submission command: `qsub`

Submission script template: `msgui_qsub_fillup.sh`

and press the Run Command button.

2. Display the 'beautified' 2D class averages (`Class2D-beaut/ordered_class_averages.hdf`) using the *Display Data utility*.

[*We use this tool to quickly check suitability of the dataset for high-resolution structural analysis!*](#)

STEP 10. VIPER – Initial model

We will now use RVIPER, to determine a reproducible and validated initial model at intermediate resolution, using the small subset of class averages produced by ISAC.

1. Press the VIPER button on the left and then the Initial 3D Model – RVIPER button in the middle.

Provide following parameters at the respective GUI interface:

Input images stack: *Class2D/best.hdf*

Output directory: VIPER

Target Particle Radius: 29

Symmetry: c5

RVIPER iterations: 10

Restarting iteration: 0

Advanced Parameter: Minimum VIPER runs: 9

MPI processors: 48

Submit jobs to queue: Checked!

Job name: srxvipер

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

and press the Run Command button.

2. Check the progress of the job.

3. Display the final VIPER volume (VIPER/main00?/average_volume.hdf) using UCSF Chimera.

4. Compare the volume with the available x-ray structure (pdb-ID: 1VW1)

[Does the volume show the right-handedness? If this is not the case, type:

`e2proc3d.py VIPER/main?/?/average_volume.hdf`

`VIPER/main?/?/average_volume_flip.hdf --process xform.flip:axis=z`

in order to flip the volume. Open then the resulting volume in Chimera]

STEP 11. Prepare reference for 3D refinement

As mentioned before, the particles of the dataset have been resized during ISAC.

To check the shrink ratio used to downscale the images before ISAC, type:

```
cat Class2D/README_shrink_ratio.txt
```

Now we will upscale the VIPER model to match the original pixel size.

1. Press the Change Size of VIPER Model in the middle of the SPHIRE GUI.

Provide following parameters at the respective GUI interface:

Input Viper Model: *VIPER/main00?/average_volume.hdf (or flip volume)*

Output Resized Viper Model: VIPER/tmp.hdf

Ratio of new to old image size: 5

Submit jobs to queue: Checked!

Job name: sxprocess_changesize

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

and press the Run Command button.

2. Check the progress of the job.

3. Now we will pad the resulting volume to match the original box size.

Press the Window VIPER Model button in the middle of the SPHIRE GUI.

Specify following parameters:

Window to specified size [Pixels]: 352

Input Volume: VIPER/tmp.hdf

Output windowed volume: VIPER/ref.hdf

Submit jobs to queue: Checked!

Job name: e2proc3d_clip

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

and press the Run Command button.

4. Check the progress of the job.
5. Open the resulting volume in Chimera.

Step 12. Prepare mask for 3D refinement

Now we will use the resampled VIPER 3D volume to prepare a mask for the subsequent high resolution 3D refinement, with the tool 3D adaptive mask.

1. Press the Adaptive 3D Mask button in the middle (Utilities).

Adjust following settings:

Input Volume: VIPER/ref.hdf

Output mask: VIPER/mask.hdf

Binarization threshold: 13

(Use Chimera to find a suitable binarization threshold for this volume!)

Submit jobs to queue: Checked!

Job name: sxprocess_adaptive_mask

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

and press the Run Command button.

2. Check the progress of the job.

3. Display the resulting 3D mask with Chimera.

STEP 13. Prepare stack for 3D refinement

During 2D Classification, ISAC eliminated most of the junk particles of the original dataset. In addition, we removed manually some class averages corresponding to a second conformation. Thus, now we can create a “clean” dataset that will include only the particle members of the chosen, final, stable and reproducible ISAC class averages and will subsequently then use it for the high-resolution 3D refinement.

1. Go back to the main window of the SPHIRE GUI and press the “ISAC” button on the left and then the Create Stack Subset Button in the middle.

Adjust following parameters:

Input BDB image stack: bdb:Particles#stack

ISAC class average file path: Class2D/best.hdf

Output directory: Substack

and press the Run command button.

This will create a virtual “clean” stain containing only the selected ISAC class members.

2. Check the progress of the job. Once the job has finished, check the number of particles of the clean dataset

`e2iminfo.py bdb:Substack#isac_substack`

How many particles were eliminated in total?

STEP 14. Meridien- High Resolution 3D refinement

1. Go back to the main window of the SPHIRE GUI and press the button “Meridien” on the left and then the 3D Refinement Button in the middle.

Enter following parameters:

Input image stack: bdb:Substack#isac_substack

Output Directory: Refine3D

Initial Reference: ref.hdf

3D mask: VIPER/mask.hdf

Starting Resolution: 18

Initial angular sampling: 7.5

Memory per node: 64

MPI processors: 96

Submit jobs to queue: Checked!

Job name: sxmeridien

Submission command: qsub

Submission script template: msgui_ppn24.sh

And press the Run command button.

2. Monitor the progress of resolution during the refinement.

Go back to the main window of the SPHIRE GUI and press the button “Meridien” on the left and then the 3D Refinement Assessment Button in the middle. Press the Run Command button.

Select the Refinement directory: Refine3D.

The resolution overview plot shows the progress in resolution (FSC @0.143) from iteration to iteration.

You can also select a specific iteration in the main window of the 3D Refinement Assessment Tool (activate check box) and then display the FSC between the two half maps for this particular round of the refinement (Plot → New FSC Plot).

Note that the reported resolution during the refinement is rather underestimated.

3. Once the job has finished, open one of the final unfiltered half volumes (Refine3D/vol_0_unfil.hdf) in Chimera.

5. Calculate the angular distribution of your dataset.

Press the Angular Distribution button (Utilities) in the middle of the SPHIRE GUI.

Adjust following parameters:

Alignment parameter file: Refine3D/final_params_(nr_of_final_iteration).txt

Submit jobs to queue: Checked!

Job name: sxprocess_angular_distribution

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run command button.

6. Once the job has finished, open the angular distribution file (Refine3D/distribution bild) in Chimera.

STEP 15. Sharpening

We will now compute the final sharpened reconstruction from the two half-volumes and report the final resolution. The final map will be masked and filtered according to its nominal resolution according to [FSC@0.143](#).

Go back to the main window of the SPHIRE GUI and press the button Sharpening in the middle.

Set following Parameters:

First unfiltered half-volume: Refine3D/vol_0_unfil.hdf

Second unfiltered half-volume: Refine3D/vol_1_unfil.hdf

MTF file: FalconIImtf.txt

Output file: vol_postrefine_masked.hdf

B-factor lower limit: 10

Apply adaptive mask: Check!

Job name: sxprocess_postprocess

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run command button.

2. Monitor the progress of the job through the logfile log.txt

```
tail -f log.txt
```

[What is the resolution of the final volume?](#)

3. Check the final volume in Chimera.

[Congratulations! You produced your first near-atomic resolution reconstruction with SPHIRE!](#)

4. Create a folder named Sharpening.

```
mkdir sharpening
```

Type this command to move the sharpening results to this folder:

```
mv vol* fsc.txt log.txt guinierlines.txt Sharpening/
```

STEP 16. 3D Variability

Now we will calculate a 3D variability map using the dataset and the final orientation parameters. Regions with higher variability are less reliable and the data may require further analysis.

1. First, we will import the projection parameters of the final refinement round (with the highest resolution) to the header of our “clean” particle stack. At the terminal, type:

```
sxheader.py bdb:Substack#isac_substack --params=xform.projection  
--import=Refine3D/final_params_iterationnr.txt
```

2. Now, we will symmetrize the projection parameters, in order to cover the entire 3D angular space for the calculation of the 3D variability field.

Go back to the main window of the SPHIRE GUI and press the button *Sort3D* on the left and then the *3D Variability Preprocess* button in the middle.

Enter following parameters:

Input image stack: bdb:Substack#isac_substack

Point Group Symmetry: c5

Submit jobs to queue: Checked!

Job name: sx3dvariability_symmetrize

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run command button.

3. Check the progress of the job.

Once the job has finished, the symmetrized stack is stored in file

`bdb:sdata.`

[How many particles does this dataset include?](#)

4. Go back to the main window of the SPHIRE GUI and press the button *3D Variability estimation* in the middle.

Provide following parameters to the respective input fields:

Input image stack: bdb:sdata

Output 3D average: 3davg.hdf

Output 3D variability: 3dvar.hdf

Use CTF: checked

Lowpass-filter frequency: 0.076

Lowpass-filter fall-off: 0.1

MPI processors: 59

Submit jobs to queue: Checked!

Job name: sx3dvariability

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run command button.

5. Monitor the progress of the job.

6. Once the job has finished, open the average (3davg.hdf) and variance (3dvar.hdf) volume with Chimera.

7. Use Chimera to apply a Gaussian filter to the volume.

8. Make a folder named Variability. Type this command to move the variability analysis results to this folder:

```
mv 3d* Variability/
```

The uniform variability map and the high-resolution achieved in the 3D refinement from a rather low number of particles, suggest that our tutorial particle is rather rigid. Nevertheless, we will continue with a 3D heterogeneity analysis to demonstrate the usage of the program.

STEP 17. 3D Sorting

We will now perform a 3D classification using the SORT3D approach.

1. Go back to the main window of the GUI and press the button 3D Clustering – SORT3D in the middle and adjust the following settings:

Enter following parameters:

Input 3D Refinement directory: Refine3D

Output Directory: Sort3D

3D mask: Sharpening/vol_adaptive_mask.hdf

Focus 3D mask: none

Use CTF: Check!

Images per Group: 4750

Initial image size for sorting: 124

Smallest Group Size: 200

Chunk name for 1st subset: Refine3D/main023/chunk_0_023.txt

Chunk name for 2nd subset: Refine3D/main023/chunk_1_023.txt

MPI processors: 96

Submit jobs to queue: Checked!

Job name: sxsort3d_new

Submission command: qsub

Submission script template: msgui_qsub_ppn12_dedicated2.sh

And press the Run command button.

2. Monitor the progress of the job.

3. Once the job has finished, open the resulting volumes (vol0.hdf, vol1.hdf) with Chimera and compare them with the Morph_map Utility.

4. Check the number of particles in each population and the total number of accounted particles. At the terminal type:

```
wc -l Sort3D/generation000/stable_class*.txt
```

```
wc -l Sort3D/generation000/Unaccounted.txt
```

How many particles did sort3d remove?

5. Check the final resolution of each 3D volume. At the terminal, type:

```
grep low_pass sxsort3d_output
```

Due to the overall similarity, we will merge both populations and perform a local refinement with all particles that were accounted during Sort3D.

For this purpose, at the terminal type:

```
cat Sort3D/generation000/stable_class0.txt  
Sort3D/generation000/stable_class1.txt > Sort3D/Sort3D_Accounted.txt
```

Now we will extract these particles and create a “clean” dataset to perform another 3D refinement, starting from a better reference and using a more precise 3D mask.

To create the “clean” dataset, at the terminal type:

```
e2bdb.py bdb:Substack/isac_substack --list=Sort3D/Sort3D_Accounted.txt --  
makevstack=bdb:Substack-afterSort3D/sort3d-substack
```

STEP 18. 3D Refinement of the “clean” dataset

Go back to the main window of the SPHIRE GUI and press the button “Meridien” on the left and then the 3D Refinement Button in the middle.

Enter following parameters:

Input image stack: bdb:Substack-afterSort3D#sort3d-substack

Output Directory: Refine3D-afterSort3D

Initial Reference: Sharpening/vol_postrefine_masked.hdf

3D mask: Sharpening/vol_adaptive_mask.hdf

Starting Resolution: 6

Initial angular sampling: 3.75

Memory per node: 128

MPI processors: 192

Submit jobs to queue: Checked!

Job name: sxmeridien

Submission command: qsub

Submission script template: msgui_ppn24.sh

And press the Run command button.

Check the progress of resolution during the refinement with the 3D Refinement Assessment tool.

STEP 19. Sharpening

We will now compute the final sharpened reconstruction from the resulting half-volumes and report the final resolution. The final map will be then masked, but this time, we will **not** filter the volume according to its FSC.

Go back to the main window of the SPHIRE GUI and press the button Sharpening in the middle.

Set following Parameters:

First unfiltered half-volume: Refine3D-afterSort3D/vol_1_unfil.hdf

Second unfiltered half-volume: Refine3D-afterSort3D /vol_0_unfil.hdf

MTF file: FalconIImtf.txt

Low pass filter Frequency: -1

Output file: vol_postrefine_masked.hdf

B-factor high-limit: 0

Apply adaptive mask: Check!

Job name: sxprocess_postprocess

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run command button.

2. Monitor the progress of the job through the logfile log.txt

```
tail -f log.txt
```

What is the resolution of the volume?

3. Check the final volume in Chimera.

4. Create a folder named Sharpening-subset-refinement. Type this command to move the sharpening results to this folder:

```
mv vol* fsc.txt log.txt guinierlines.txt Sharpening-subset-Refinement/
```

STEP 20. Local resolution and filtering.

Now we will calculate the local resolution of the final map and filter it accordingly.

1. Go back to the main window of the SPHIRE GUI and press the button LocalRes on the left and then the Local Resolution button in the middle.

Provide following parameters:

First half-volume: Refine3D-afterSort3D/vol_0_unfil.hdf

Second half-volume: vol_1_unfil

3D Mask: Sharpening-subset-Refinement/vol_adaptive_mask.hdf

Output volume: local-res.hdf

FSC window size: 7

Fourier Shell step size: 2

Resolution cut-off: 0.5

Overall resolution: 0.329

MPI processors: 59

Job name: sxlocres

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the Run Command button.

2. Once the program has finished, check the sharpened, unfiltered volume (Sharpening-subset-refinement/vol_postrefine_masked.hdf) and the local resolution map (local-res.hdf) in Chimera .

3. Now we will filter the sharpened, unfiltered volume (Sharpening-local-refinement/vol_postrefine_masked.hdf) according to its local resolution.

Go back to the main window of the SPHIRE GUI and press the button LocalRes on the left and then the 3D Local Filter button in the middle.

Provide following parameters:

Input volume: Sharpening-subset-Refinement/vol_postrefine_masked.hdf

Local resolution file: local-res.hdf

3D Mask: Sharpening-subset-Refinement/vol_adaptive_mask.hdf

Output volume: FINAL-VOLUME.hdf

MPI processors: 59

Job name: sxfilterlocal

Submission command: qsub

Submission script template: msgui_qsub_fillup.sh

And press the *Run Command* button.

4. Once the job has finished, check the final Volume in Chimera. Compare the volume with the available x-ray structure, at regions of different local resolution (pdb-ID: 1VW1).