High-Resolution cryo-EM and single-particle-analysis

SPHIRE 1.2a

Short Tutorial

A more comprehensive full-length SPHIRE tutorial describing many more features can be downloaded at: http://www.sphire.mpg.de

STEP 0. Getting started

1. Open a terminal.

2. Go to the folder SphireDemoResults `cd SphireDemoResults`

3. Create a working directory `mkdir working_directory`

4. Change to this directory `cd working_directory`

5. Activate the SPHIRE environment

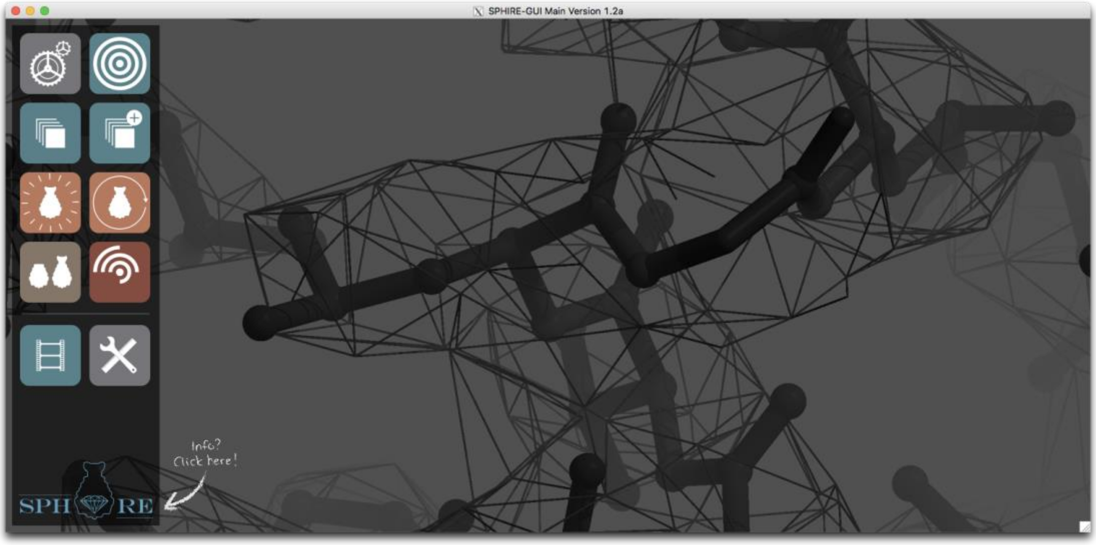
6. Launch the SPHIRE GUI `sphire`

7. Confirm that you want to create a new project directory

8. Suspend the GUI `Ctrl + z`

9. Then move the GUI to the background `bg`

Welcome to SPHIRE!

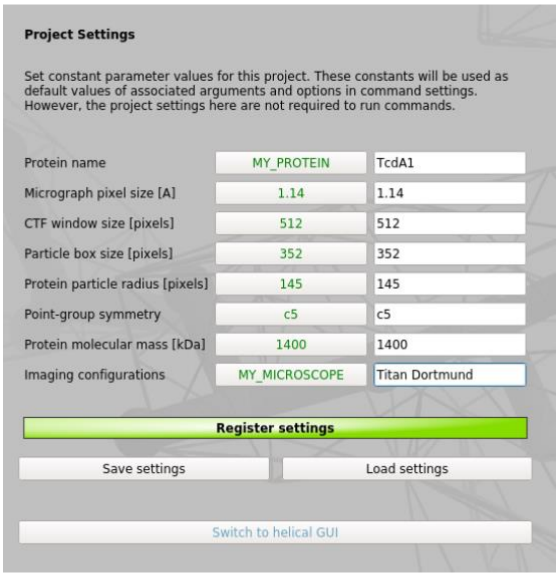


STEP 1. Register Settings

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

1. Press the Project pictogram.

2. Provide following parameters:



Project Settings

Set constant parameter values for this project. These constants will be used as default values of associated arguments and options in command settings. However, the project settings here are not required to run commands.

Protein name	MY_PROTEIN	TcdA1
Micrograph pixel size [A]	1.14	1.14
CTF window size [pixels]	512	512
Particle box size [pixels]	352	352
Protein particle radius [pixels]	145	145
Point-group symmetry	c5	c5
Protein molecular mass [kDa]	1400	1400
Imaging configurations	MY_MICROSCOPE	Titan Dortmund

Project name: **TcdA1**

Micrograph pixel size [Å]: **1.14**

CTF window size [pixels]: **512**

Particle box size [pixels]: **352**

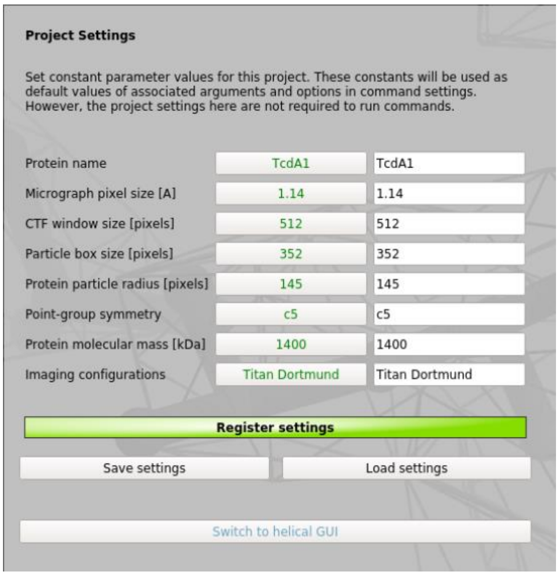
Protein particle radius [pixels]: **145**

Point-group symmetry: **c5**

Protein molecular mass [kDa]: **1400**

Imaging configurations: **Titan Dortmund**

3. Press the Register settings button.



Project Settings

Set constant parameter values for this project. These constants will be used as default values of associated arguments and options in command settings. However, the project settings here are not required to run commands.

Protein name	TcdA1	TcdA1
Micrograph pixel size [A]	1.14	1.14
CTF window size [pixels]	512	512
Particle box size [pixels]	352	352
Protein particle radius [pixels]	145	145
Point-group symmetry	c5	c5
Protein molecular mass [kDa]	1400	1400
Imaging configurations	Titan Dortmund	Titan Dortmund

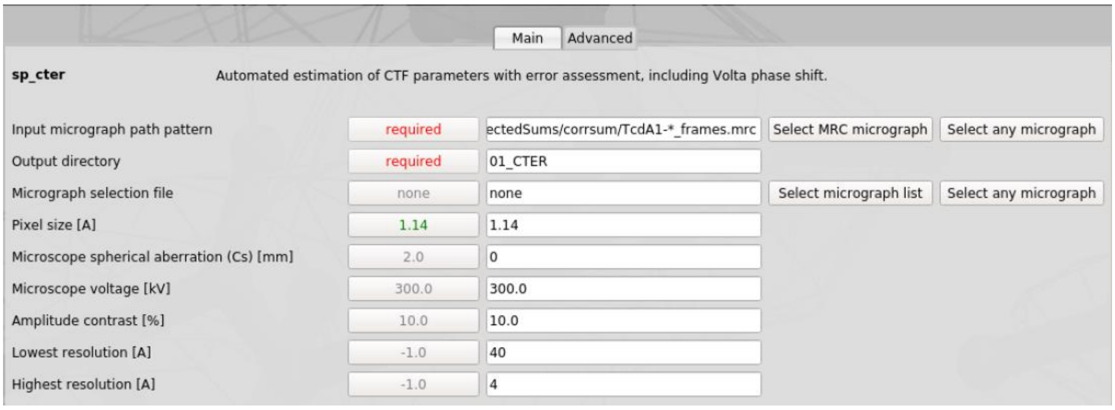
Registered settings will be filled in automatically in all windows and appear in green in the “restore default” button next to the related text field.

STEP 2. CTF Estimation

Now we will estimate the CTF of the motion corrected micrographs. 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:



Main | Advanced

sp_cter Automated estimation of CTF parameters with error assessment, including Volta phase shift.

Input micrograph path pattern	required	ectedSums/corrsum/TcdA1-*_frames.mrc	Select MRC micrograph	Select any micrograph
Output directory	required	01_CTER		
Micrograph selection file	none	none	Select micrograph list	Select any micrograph
Pixel size [A]	1.14	1.14		
Microscope spherical aberration (Cs) [mm]	2.0	0		
Microscope voltage [kV]	300.0	300.0		
Amplitude contrast [%]	10.0	10.0		
Lowest resolution [A]	-1.0	40		
Highest resolution [A]	-1.0	4		

Input micrograph path pattern: **CorrectedSums/corrsum/TcdA1-*_frames_sum.mrc**

Output directory: **01_CTER**

Micrograph selection file: **none**

Pixel size: **1.14** (registered setting)

Microscope spherical aberration (Cs) [mm]: **0** (Cs corrected)

Microscope voltage (kV): **300.0**

Amplitude contrast [%]: **10.0**

Lowest resoution [Å]: **40**

Highest resolution [Å]: **4**

Advanced Parameters:

Use PW spectrum: ✓

Calculate 2D power spectra: ✓

2. Press the **Run command** button.

3. Monitor the progress of the sp_cter job.

4. Once the job has finished, check the content of the output folder 01_CTER.

5. The CTF results are stored in the file 01_CTER/*partres.txt*.
Execute `cat 01_CTER/partres.txt` in the terminal or open the file with a text editor to check its content.

STEP 3. CTF Assessment

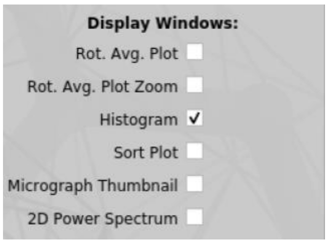
The CTF results can be analyzed using SPHIRE's CTF Assessment tool. Using this tool, multiple criteria can be used simultaneously, in order to remove outlier images that might have a negative impact on the final result.

Our data set is pre-cleaned, thus, we will only perform a simplified screening as proof of principle and discard the image with the highest defocus value.

1. On the main window of SPHIRE, press the CTER button on the left and then the CTF Assessment button in the middle.

2. Select the CTF parameter file 01_CTER_*/partres.txt* and press the Run command button to launch the GUI tool.

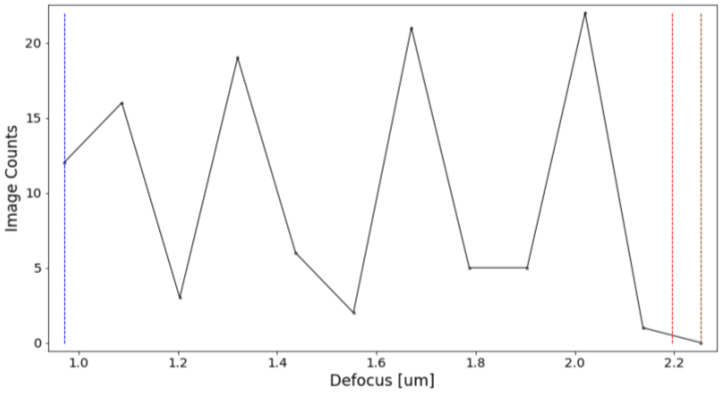
3. Uncheck Rot. Avg. Plot Zoom and Sort Plot to close plots we do not need at the moment



Display Windows:

Rot. Avg. Plot □
Rot. Avg. Plot Zoom □
Histogram ☑
Sort Plot □
Micrograph Thumbnail □
2D Power Spectrum □

4. Shift + left click on the histogram window to set the upper defocus limit (red line) to ~ 2.2 µm in order to discard micrographs above this limit.



5. Press the Apply All Thresholds button and confirm selection.

6. Have a look at the number of unchecked micrographs.

7. Enter Tutorial in File Suffix and press the Save Selection Button.

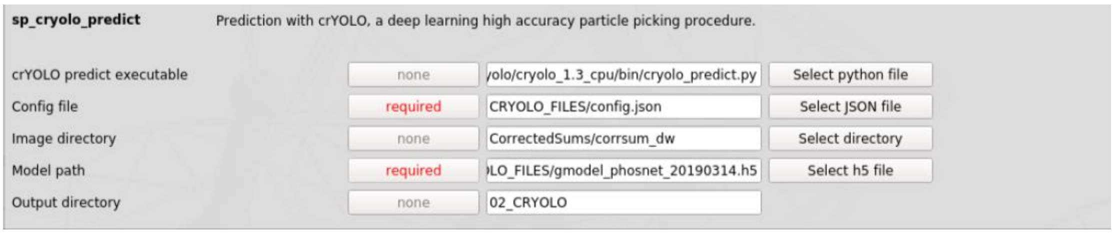
8. Check the output of CTF Assessment in the 01_CTER folder.

STEP 4. Particle Picking

The next step is to pick and extract particles from the **dose-weighted** micrographs. For this purpose, we will select particles automatically using our neuronal network based particle picker crYOLO and the generalized model.

1. Go to the main window of the SPHIRE GUI and press the button Window on the left and then the button *crYOLO - predict* in the middle.

Provide following parameters:



sp_cryolo_predict — Prediction with crYOLO, a deep learning high accuracy particle picking procedure.

crYOLO predict executable	none	yolo/cryolo_1.3_cpu/bin/cryolo_predict.py	Select python file
Config file	required	CRYOLO_FILES/config.json	Select JSON file
Image directory	none	CorrectedSums/corrsum_dw	Select directory
Model path	required	LO_FILES/gmodel_phosnet_20190314.h5	Select h5 file
Output directory	none	02_CRYOLO	

crYOLO predict executable: ***"Path to crYOLO executable/cryolo_predict.py"***

Config file: **CRYOLO_FILES/config.json**

Image directory: **CorrectedSums/corrsum_dw**

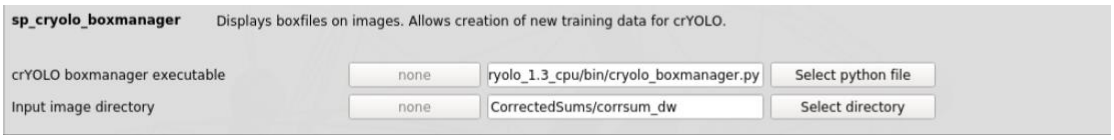
Model path: **CRYOLO_FILES/gmodel_phosnet_20190314.h5**

Output directory: **02_CRYOLO**

2. Press the **Run command** button.

3. Monitor the progress of the crYOLO job.

4. Once the job finished, use the cryolo_box_manager to display the picking results. In the main window of the SPHIRE GUI click the button Window on the left and then the button crYOLO - boxmanager in the middle (Utilities). Fill out the following fields:



sp_cryolo_boxmanager — Displays boxfiles on images. Allows creation of new training data for crYOLO.

crYOLO boxmanager executable	none	ryolo_1.3_cpu/bin/cryolo_boxmanager.py	Select python file
Input image directory	none	CorrectedSums/corrsum_dw	Select directory

crYOLO boxmanager executable: ***"Path to crYOLO executable/cryolo_boxmanager.py"***

Input image directory: **CorrectedSums/corrsum_dw**

And press the **Run command** button to open the box manager.

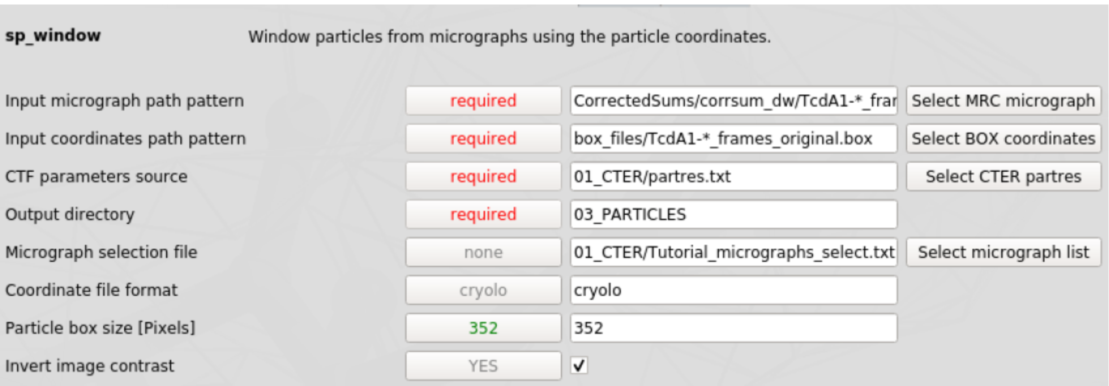
5. Click on the File and then select the Import box files option. Use the file browser to select the CBOX folder within the folder 02_CRYOLO to load the coordinates for each micrograph. Within the box manager you can **adjust the confidence threshold** to optimize picking results.

STEP 5. Particle Extraction

Due to limited computational resources, we will not use all boxes crYOLO picked but a **smaller subset**.

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:



sp_window Window particles from micrographs using the particle coordinates.

Input micrograph path pattern	required	CorrectedSums/corrsum_dw/TcdA1-*_frar	Select MRC micrograph
Input coordinates path pattern	required	box_files/TcdA1-*_frames_original.box	Select BOX coordinates
CTF parameters source	required	01_CTER/partres.txt	Select CTER partres
Output directory	required	03_PARTICLES	
Micrograph selection file	none	01_CTER/Tutorial_micrographs_select.txt	Select micrograph list
Coordinate file format	cryolo	cryolo	
Particle box size [Pixels]	352	352	
Invert image contrast	YES	☑	

Input micrograph path pattern: **CorrectedSums/corrsum_dw/TcdA1-*_frames.mrc**

Input coordinates path pattern: **box_files/TcdA1-*_frames_original.box** (subset)

CTF parameter source: **01_CTER/partres.txt**

Output directory: **03_PARTICLES**

Micrograph selection file: **01_CTER/Tutorial_micrographs_select.txt**

Coordinate file format: **cryolo**

Particle box size [Pixels]: ***352***

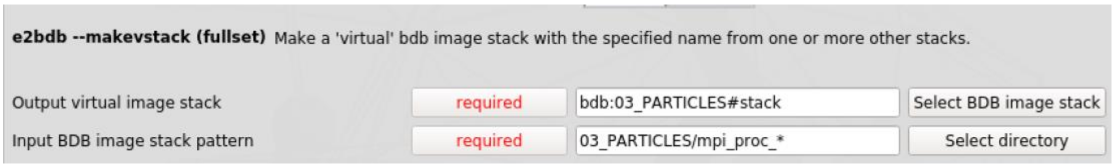
Invert contrast: ✓

And press the **Run command** button.

2. Monitor the progress of the sp_window job.

3. When the job is finished, create a bdb stack containing all extracted particles by pressing the Window button on the left and Particle Stack in the middle.

Provide following parameters:



e2bdb --makevstack (fullset) Make a 'virtual' bdb image stack with the specified name from one or more other stacks.

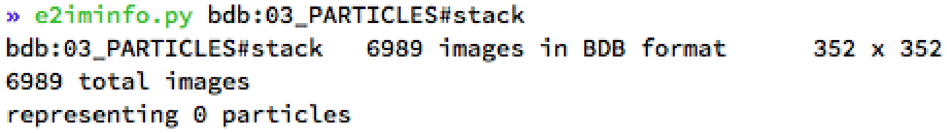
Output virtual image stack	required	bdb:03_PARTICLES#stack	Select BDB image stack
Input BDB image stack pattern	required	03_PARTICLES/mpi_proc_*	Select directory

Output virtual image stack: **bdb:03_Particles#stack**

Input BDB image stack pattern: **03_Particles/mpi_proc_***

4. Afterwards, check the total number of particles in the resulting stack. Therefore, execute the following command in the terminal

```
e2iminfo.py bdb:03_PARTICLES#stack
```



```
» e2iminfo.py bdb:03_PARTICLES#stack
bdb:03_PARTICLES#stack   6989 images in BDB format      352 x 352
6989 total images
representing 0 particles
```

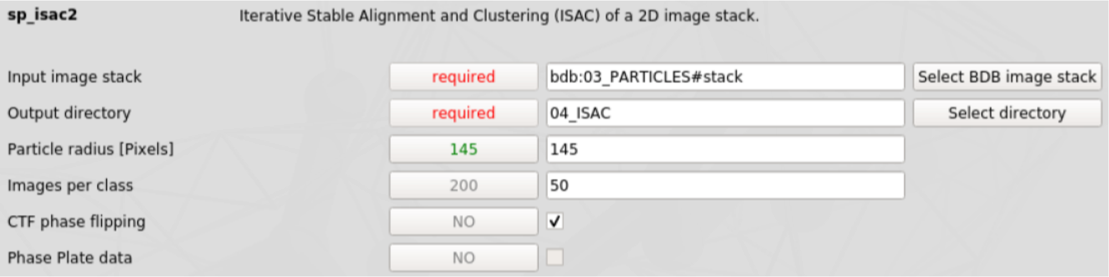
5. You can display the virtual stack using the utility *Display Data.*

STEP 6. ISAC – 2D Classification

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

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

Provide following parameters:



sp_isac2 Iterative Stable Alignment and Clustering (ISAC) of a 2D image stack.

Input image stack	required	bdb:03_PARTICLES#stack	Select BDB image stack
Output directory	required	04_ISAC	Select directory
Particle radius [Pixels]	145	145	
Images per class	200	50	
CTF phase flipping	NO	☑	
Phase Plate data	NO	□	

Input image stack: **bdb:03_PARTICLES#stack**

Output directory: **04_ISAC**

Particle radius [Pixels]: **145**

Images per class: **50**

CTF phase flipping: ✓

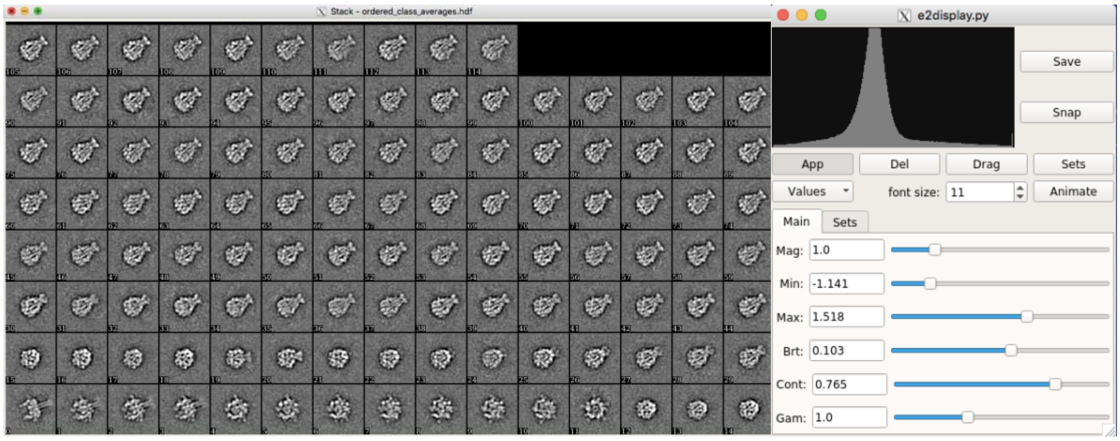
Advanced Parameters:

Minimum size of reproducible classes: **30**

And press the **Run command** button.

2. Monitor the progress of the sp_isac2 job.

3. Display the resulting 2D class averages 04_ISAC/ordered_class_averages.hdf using the utility *Display Data.*



font size: 11

Mag: 1.0

Min: -1.141

Max: 1.518

Brt: 0.103

Cont: 0.765

Gam: 1.0

Does the data set contain contamination or different populations of particles?

Delete “bad” class averages (middle mouse button, press Del and then click on classes you want to delete) and store the remaining class averages into a new file named 04_ISAC*/best.hdf* (save button).

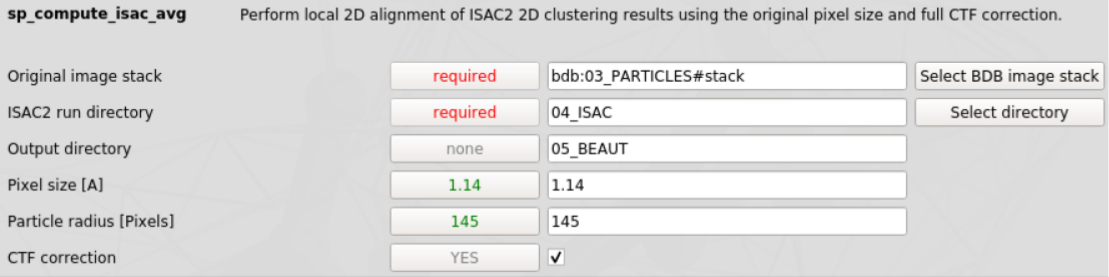
4. Check the number of remaining “good” classes by executing `e2iminfo.py 04_ISAC/best.hdf` in the terminal.

STEP 6b. Beautifier – High resolution classes for publication

Particles are automatically downsized to 76 pixels within ISAC to speed up calculations. One can already see fine details in those classes, however, for publications one might want to compute full size class averages using the Beautifier to recover high resolution features.

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

Provide following parameters:



sp_compute_isac_avg Perform local 2D alignment of ISAC2 2D clustering results using the original pixel size and full CTF correction.

Original image stack	required	bdb:03_PARTICLES#stack	Select BDB image stack
ISAC2 run directory	required	04_ISAC	Select directory
Output directory	none	05_BEAUT	
Pixel size [A]	1.14	1.14	
Particle radius [Pixels]	145	145	
CTF correction	YES	☑	

Original image stack: **bdb:03_PARTICLES#stack**

ISAC2 run directory: **04_ISAC**

Output directory: **05_BEAUT**

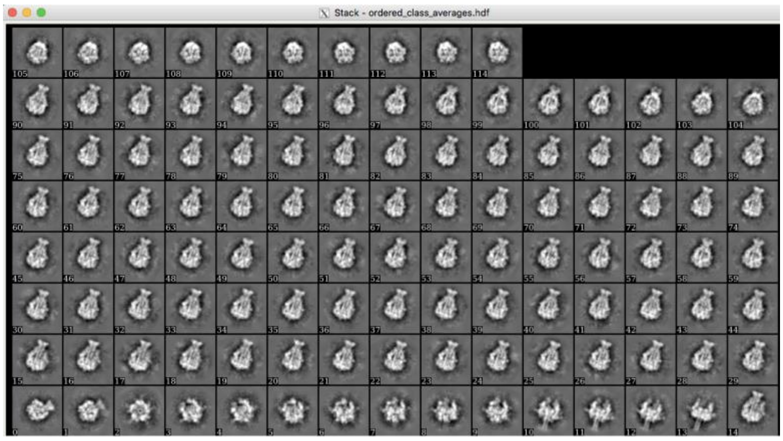
Pixel size [A]: **1.14**

Particle radius [Pixels]: **145**

CTF correction: ✓

And press the **Run command** button.

2. Display the resulting 2D class averages 05_BEAUT/ordered_class_averages.hdf using the utility *Display Data.*

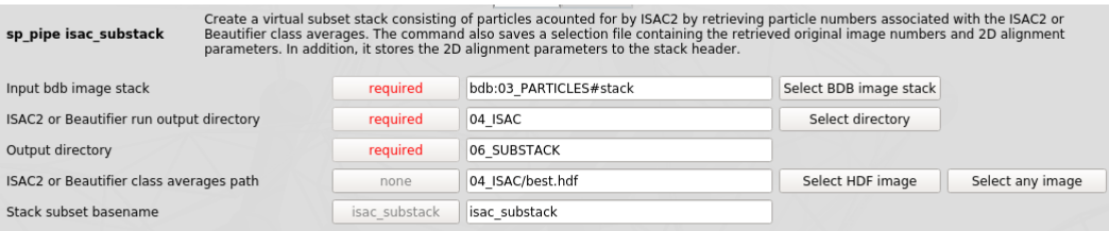


STEP 7. Creation of particle sub stack for high-resolution refinement

We found that our data set contains some particles of a second conformation and removed corresponding classes. Now, we create a sub stack of all particles belonging to "good" ISAC class averages using ISAC2 Stack Subset and will subsequently use it for the high-resolution 3D refinement

1. Press the ISAC button on the left and then the ISAC2 Stack Subset button in the middle.

Provide following parameters:



sp_pipe isac_substack Create a virtual subset stack consisting of particles acounted for by ISAC2 by retrieving particle numbers associated with the ISAC2 or Beautifier class averages. The command also saves a selection file containing the retrieved original image numbers and 2D alignment parameters. In addition, it stores the 2D alignment parameters to the stack header.

Input bdb image stack: required bdb:03_PARTICLES#stack
ISAC2 or Beautifier run output directory: required 04_ISAC
Output directory: required 06_SUBSTACK
ISAC2 or Beautifier class averages path: none 04_ISAC/best.hdf
Stack subset basename: isac_substack isac_substack

Input bdb image stack: **bdb:03_PARTICLES#stack**

ISAC2 or Beautifier run output directory: **04_ISAC**

Output directory: **06_SUBSTACK**

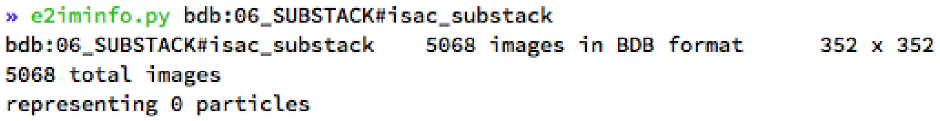
ISAC2 or Beautifier class averages path: **04_ISAC/best.hdf**

Stack subset basename: **isac_substack**

And press the **Run command** button.

2. Afterwards, check the total number of particles in the resulting stack. Therefore, execute the following command in the terminal

```
e2iminfo.py bdb:06_SUBSTACK#isac_substack
```



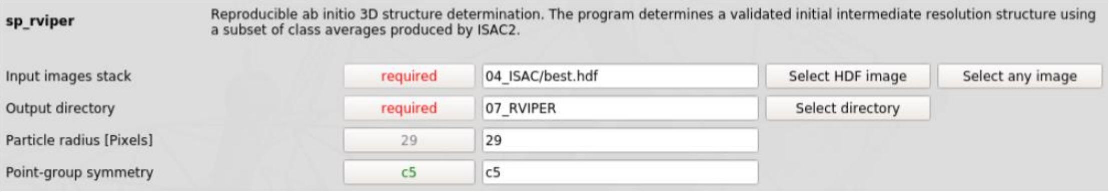
```
» e2iminfo.py bdb:06_SUBSTACK#isac_substack
bdb:06_SUBSTACK#isac_substack    5068 images in BDB format     352 x 352
5068 total images
representing 0 particles
```

STEP 8. R-VIPER – Initial model generation

A good initial model is crucial for a successful 3D refinement. We will use R-VIPER to determine a reproducible and validated initial model at intermediate resolution. Therefore, we will use the "good" class averages as input taking advantage of their high signal to noise ratio.

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

Provide following parameters:



sp_rviper — Reproducible ab initio 3D structure determination. The program determines a validated initial intermediate resolution structure using a subset of class averages produced by ISAC2.

Input images stack	required	04_ISAC/best.hdf	Select HDF image	Select any image
Output directory	required	07_RVIPER	Select directory	
Particle radius [Pixels]	29	29		
Point-group symmetry	c5	c5		

Input image stack: **04_ISAC/best.hdf**

Output directory: **07_RVIPER**

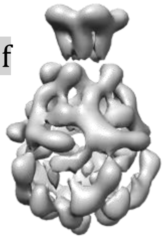
Particle radius [Pixels]: **29** (downsized radius used by ISAC, default is 29 pixels)

Point-group symmetry: **c5**

And press the **Run command** button.

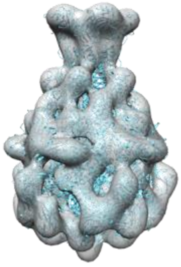
2. Monitor the progress of the sp_rviper job.

3. Display the final VIPER volume 07_RVIPER/average_volume_00*.hdf using **UCSF Chimera**.



4. Compare the volume with the available x-ray structure

- Press File ➜ Fetch by ID ➜ Fetch **PDB: 1VW1**
- Press Favorites ➜ Volume Viewer ➜ Features ➜ Activate Coordinates ➜ Change the **voxel size to 5.7** (= 1.14 / 0.2 – shrink ratio within ISAC)
- Favorites ➜ Model Panel ➜ activate 1VW1 only ➜ dock roughly into density using left and middle mouse button
- Press Tools ➜ Volume data ➜ Fit in map ➜ Fit PDB into density
- Change the opacity of your volume by clicking the color panel in Volume Viewer ➜ Opacity ✓ and the lower A value



Does the volume show the right-handedness?

You can change the handedness by

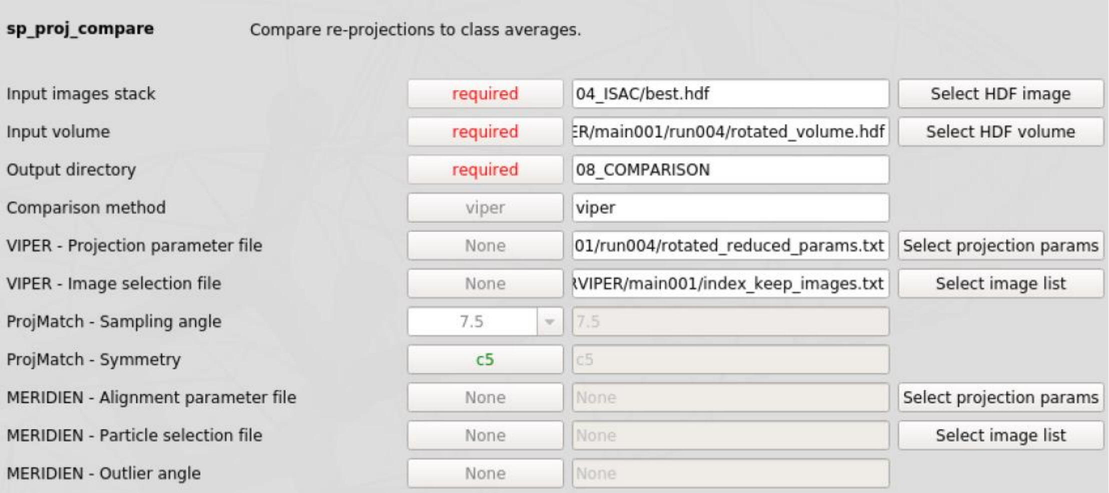
- Executing vop zflip #0 in the command line of **UCFS Chimera**
- Memorize if your handedness is correct or not for future processing steps

STEP 9. Comparison of 2D classes with the initial model

As the success of a 3D refinement strongly depends on the initial model provided, one should in general carefully check the generated volume. Although, RVIPER creates a reproducible and validated initial model, one might still want to compare projections of this volume with the original classes using Compare Re-projections.

1. Press the VIPER button on the left and then the Compare Re-projections button in the middle.

Provide following parameters:



sp_proj_compare Compare re-projections to class averages.

Input images stack	required	04_ISAC/best.hdf	Select HDF image
Input volume	required	R/main001/run004/rotated_volume.hdf	Select HDF volume
Output directory	required	08_COMPARISON	
Comparison method	viper	viper	
VIPER - Projection parameter file	None	01/run004/rotated_reduced_params.txt	Select projection params
VIPER - Image selection file	None	VIPER/main001/index_keep_images.txt	Select image list
ProjMatch - Sampling angle	7.5	7.5	
ProjMatch - Symmetry	c5	c5	
MERIDIEN - Alignment parameter file	None	None	Select projection params
MERIDIEN - Particle selection file	None	None	Select image list
MERIDIEN - Outlier angle	None	None	

Input image stack: **04_ISAC/best.hdf**

Input volume: **07_RVIPER/main001/run00*/rotated_volume.hdf** (use last run folder)

Output directory: **08_COMPARISON**

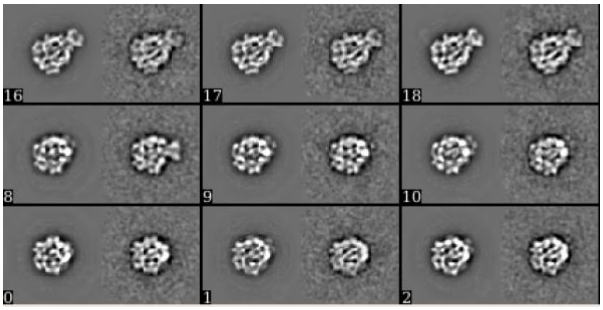
Comparison method: **viper**

VIPER – Projection parameter file: **07_RVIPER/main001/run00*/rotated_reduced_params.txt**

VIPER – Image selection file: **07_RVIPER/main001/index_keep_images.txt**

And press the **Run command** button.

2. Display the resulting comparison of 2D projections and 2D class averages 08_COMPARISON/comp-proj-reproj.hdf using the utility *Display Data.*

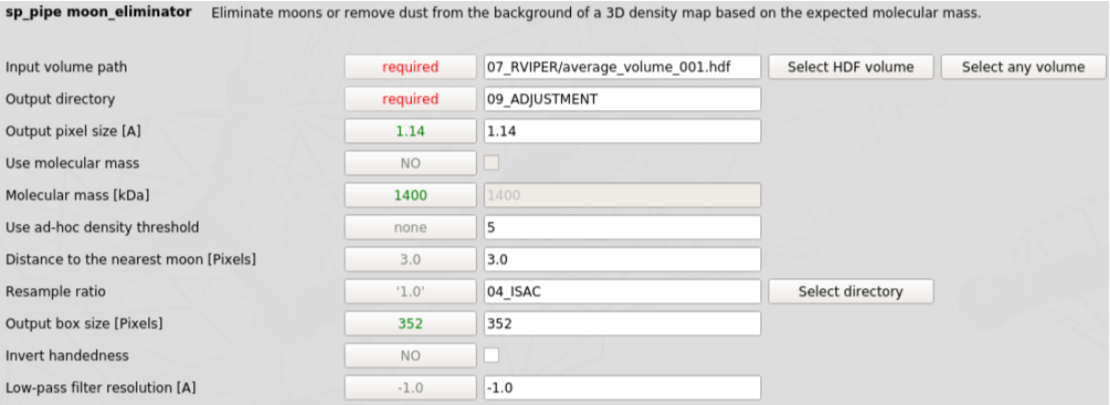


STEP 10. Rescaling of the initial reference

As mentioned before, particles have been resized during ISAC. Thus, the **initial RVIPER model** we generated from "good" class averages **needs to be rescaled** to match the original pixel size using the Volume Adjustment tool.

1. Press the VIPER button on the left and then the Volume Adjustment button in the middle.

Provide following parameters:



sp_pipe moon_eliminator Eliminate moons or remove dust from the background of a 3D density map based on the expected molecular mass.

Input volume path	required	07_RVIPER/average_volume_001.hdf	Select HDF volume Select any volume
Output directory	required	09_ADJUSTMENT	
Output pixel size [A]	1.14	1.14	
Use molecular mass	NO	□	
Molecular mass [kDa]	1400	1400	
Use ad-hoc density threshold	none	5	
Distance to the nearest moon [Pixels]	3.0	3.0	
Resample ratio	'1.0'	04_ISAC	Select directory
Output box size [Pixels]	352	352	
Invert handedness	NO	□	
Low-pass filter resolution [A]	-1.0	-1.0	

Input volume path: **07_RVIPER/average_volume_00*.hdf**

Output directory: **09_ADJUSTMENT**

Output pixel size [A]: **1.14**

Use ad-hoc density threshold: **5** (check in Chimera)

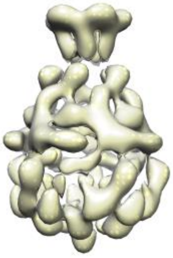
Distance to nearest moon [Pixels]: **3**

Resample ratio: **04_ISAC**

Output box size [Pixels]: **352**

Invert handedness: **If your handedness was wrong, tick this option ✓**

Low-pass filter resolution [A]: **-1**



And press the **Run command** button.

2. When the job is finished, open the resulting volume 09_ADJUSTMENT/vol3d_ref_moon_eliminated.hdf in **UCFS Chimera**.

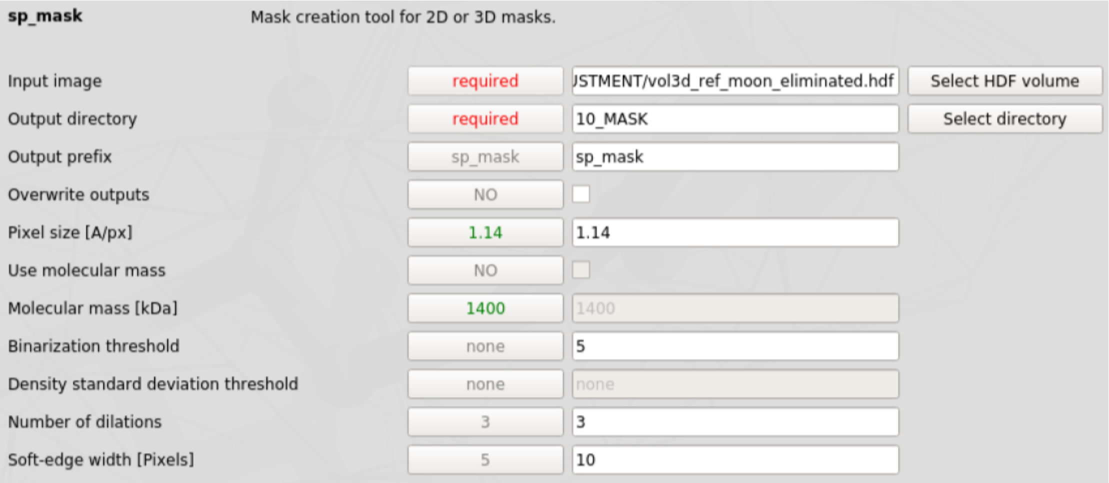
3. Also open the original RVIPER model and enter the correct voxel sizes (1.14 Å for the adjusted volume and 5.7 Å for the original RVIPER model). Finally, fit one of the maps into the other. If your handedness was correct, the volumes should be almost identical.

STEP 11. Preparing a mask for 3D refinement

Now we will use the resampled VIPER 3D volume to create a mask for the subsequent high resolution 3D refinement using the Masking tool.

1. Press the VIPER button on the left and then the Masking button in the middle.

Provide following parameters:



sp_mask Mask creation tool for 2D or 3D masks.

Input image	required	JSTMENT/vol3d_ref_moon_eliminated.hdf	Select HDF volume
Output directory	required	10_MASK	Select directory
Output prefix	sp_mask	sp_mask	
Overwrite outputs	NO	□	
Pixel size [A/px]	1.14	1.14	
Use molecular mass	NO	□	
Molecular mass [kDa]	1400	1400	
Binarization threshold	none	5	
Density standard deviation threshold	none	none	
Number of dilations	3	3	
Soft-edge width [Pixels]	5	10	

Input image: **09_ADJUSTMENT/vol3d_ref_moon_eliminated.hdf**

Output directory: **10_MASK**

Output prefix: **sp_mask**

Pixel size [A]: **1.14**

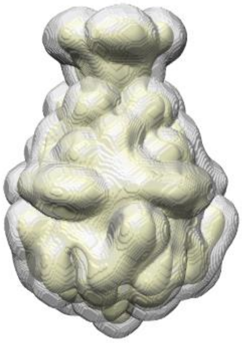
Binarization threshold: **5** (check in Chimera)

Number of dilations: **3** (each dilation corresponds to a binary extension by ~ 2 pixels)

Soft-edge width [Pixels]: **10**

And press the **Run command** button

2. Display the resulting 3D mask 10_MASK/sp_mask_mask.hdf and the rescaled reference 09_ADJUSTMENT/vol3d_ref_moon_eliminated.hdf with **Chimera**. The mask should enclose the complete volume even at low thresholds.

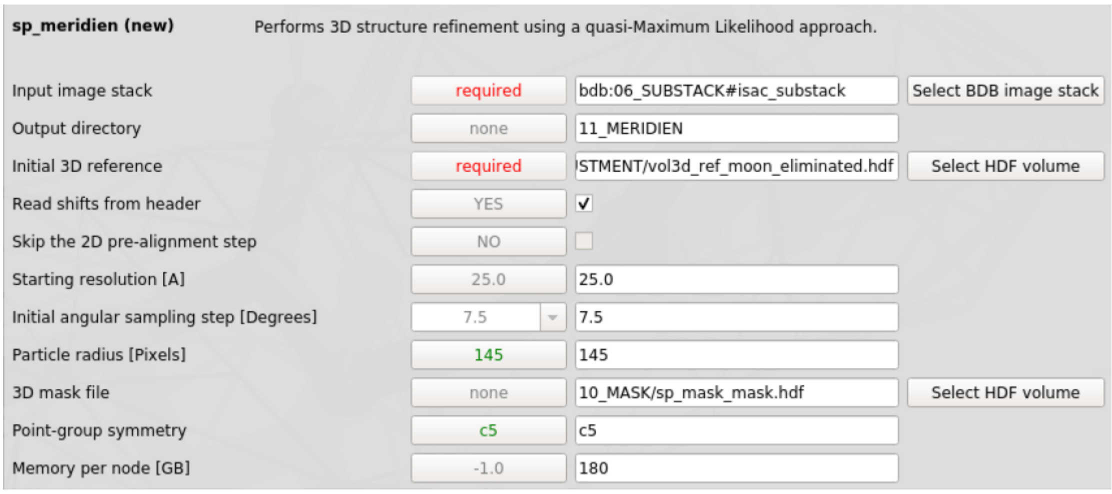


STEP 12. Meridien- High Resolution 3D refinement

Within this step we will create a high resolution density map from the sub stack of “good” particles we created in step 7 using Meridien – 3D Refinement. As a reference we will use the adjusted volume from RVIPER and the corresponding mask.

1. Press the MERIDIEN button on the left and then the 3D Refinement button in the middle.

Provide following parameters:



sp_meridien (new) Performs 3D structure refinement using a quasi-Maximum Likelihood approach.

Input image stack	required	bdb:06_SUBSTACK#isac_substack	Select BDB image stack
Output directory	none	11_MERIDIEN	
Initial 3D reference	required	STMENT/vol3d_ref_moon_eliminated.hdf	Select HDF volume
Read shifts from header	YES	☑	
Skip the 2D pre-alignment step	NO	□	
Starting resolution [A]	25.0	25.0	
Initial angular sampling step [Degrees]	7.5	7.5	
Particle radius [Pixels]	145	145	
3D mask file	none	10_MASK/sp_mask_mask.hdf	Select HDF volume
Point-group symmetry	c5	c5	
Memory per node [GB]	-1.0	180	

Input image stack:_ **bdb:06_SUBSTACK#isac_substack**

Output directory: **11_MERIDIEN**

Initial 3D reference: **09_ADJUSTMENT/vol3d_ref_moon_eliminated.hdf**

Read shifts from header: ✓

Starting resolution [A]: **25.0**

Initial angular sampling [˚]: **7.5**

Particles radius [Pixels]: **145**

3D mask file: **10_MASK/sp_mask_mask.hdf**

Point-group symmetry: **c5**

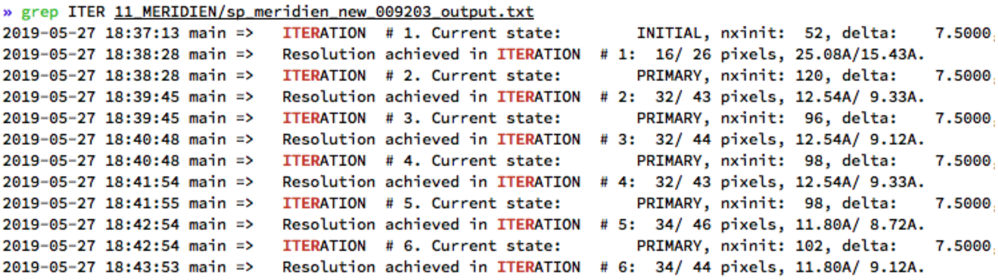
Memory per node [GB]: Needs to be adjusted to available hardware

And press the **Run command** button

2. Monitor the progress of the sp_meridien_new job.

You can quickly check the resolution and current status per iteration executing the following command in the terminal

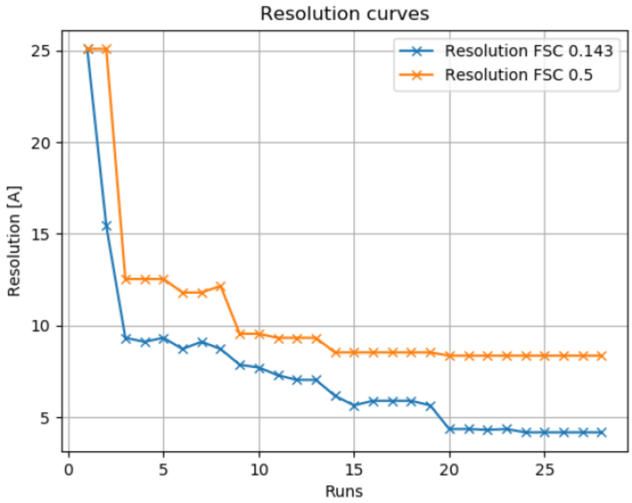
```
grep ITER “meridien_logfile”
```



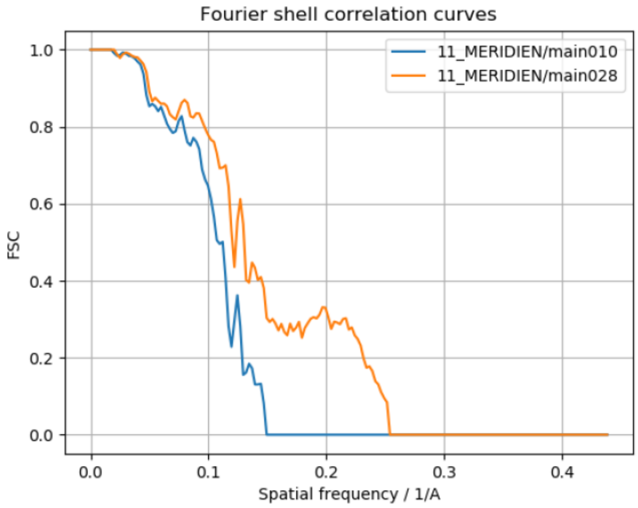
```
» grep ITER 11_MERIDIEN/sp_meridien_new_009203_output.txt
2019-05-27 18:37:13 main =>   ITERATION  # 1. Current state:       INITIAL, nxinit:  52, delta:   7.5000
2019-05-27 18:38:28 main =>   Resolution achieved in ITERATION  # 1:  16/ 26 pixels, 25.08A/15.43A.
2019-05-27 18:38:28 main =>   ITERATION  # 2. Current state:       PRIMARY, nxinit: 120, delta:   7.5000
2019-05-27 18:39:45 main =>   Resolution achieved in ITERATION  # 2:  32/ 43 pixels, 12.54A/ 9.33A.
2019-05-27 18:39:45 main =>   ITERATION  # 3. Current state:       PRIMARY, nxinit:  96, delta:   7.5000
2019-05-27 18:40:48 main =>   Resolution achieved in ITERATION  # 3:  32/ 44 pixels, 12.54A/ 9.12A.
2019-05-27 18:40:48 main =>   ITERATION  # 4. Current state:       PRIMARY, nxinit:  98, delta:   7.5000
2019-05-27 18:41:54 main =>   Resolution achieved in ITERATION  # 4:  32/ 43 pixels, 12.54A/ 9.33A.
2019-05-27 18:41:55 main =>   ITERATION  # 5. Current state:       PRIMARY, nxinit:  98, delta:   7.5000
2019-05-27 18:42:54 main =>   Resolution achieved in ITERATION  # 5:  34/ 46 pixels, 11.80A/ 8.72A.
2019-05-27 18:42:54 main =>   ITERATION  # 6. Current state:       PRIMARY, nxinit: 102, delta:   7.5000
2019-05-27 18:43:53 main =>   Resolution achieved in ITERATION  # 6:  34/ 44 pixels, 11.80A/ 9.12A.
```

3. To monitor the progress of resolution during the refinement press the MERIDIEN button on the left and then the 3D Refinement Assessment button in the middle and press the **Run command** button.

Select the Refinement directory 11_MERIDIEN to display a plot showing the progression of resolution (FSC 0.143 & 0.5) from iteration to iteration (runs).

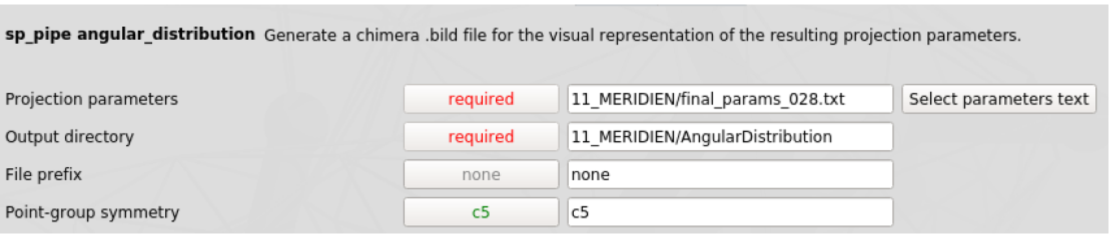


You can also select specific iteration(s) in the main window of the 3D Refinement Assessment tool (click the arrow and activate the check box) to display the FSC between the two half maps for this particular iteration of the refinement (Plot New ➜ FSC Plot). Note that the reported resolution during the refinement is underestimated.



6. Once the job has finished, press the MERIDIEN button on the left and then the Angular Distribution button in the middle to calculate a visualization of the final angular distribution.

Provide following parameters:



sp_pipe angular_distribution Generate a chimera .bild file for the visual representation of the resulting projection parameters.

Projection parameters	required	11_MERIDIEN/final_params_028.txt	Select parameters text
Output directory	required	11_MERIDIEN/AngularDistribution	
File prefix	none	none	
Point-group symmetry	c5	c5	

Projection parameters: **11_MERIDIEN/final_params_0*.txt**

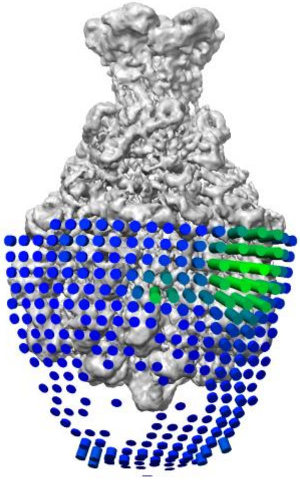
Output directory: **11_MERIDIEN/AngularDistribution**

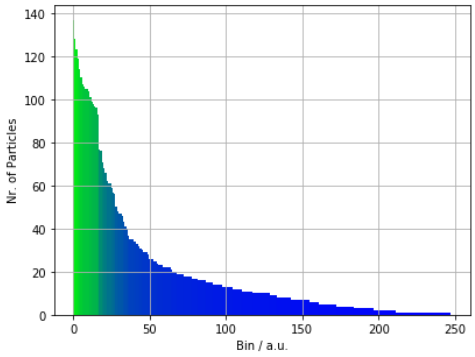
And press the **Run command** button.

7. Once the job has finished, open one of the final unfiltered half volumes 11_MERIDIEN/vol_0_unfil_0*.hdf and the angular distribution file 11_MERIDIEN/AngularDistribution/final_params_0*.bild in **Chimera.**

The color code for the 3D angular distribution can be read from the corresponding histogram 11_MERIDIEN/AngularDistribution/final_params_0*.png.

Due to the C5 symmetry of our protein, the angular distribution is limited to the unique fraction of the 3D sphere. The ideal case would be an even coverage of all possible projection angles. In our case we have more side views, which is in perfect agreement with the 2D classes we got earlier. Strongly preferred orientations can impede a successful high-resolution 3D refinement.



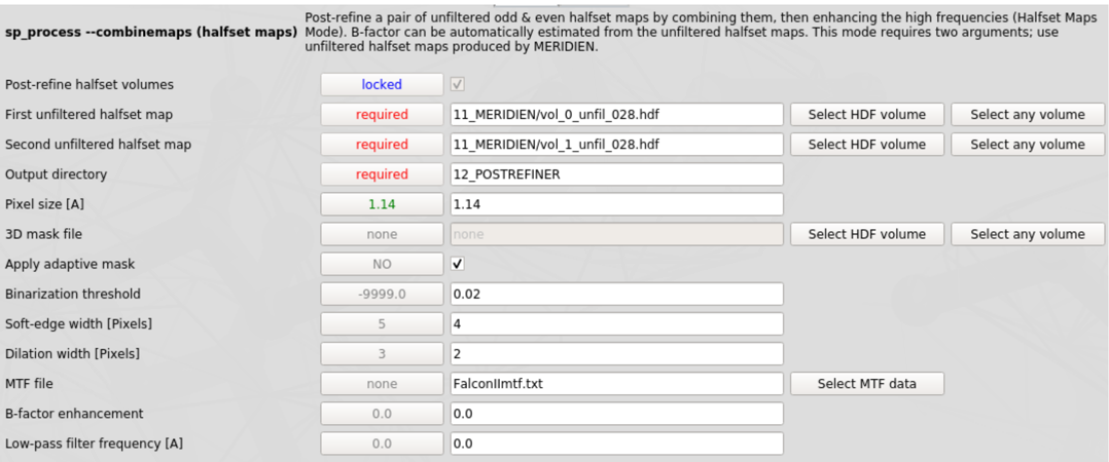


STEP 13. PostRefiner - Sharpening and FSC curve

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

1. Press the MERIDIEN button on the left and then the PostRefiner button in the middle.

Provide following parameters:



sp_process --combinemaps (halfset maps) Post-refine a pair of unfiltered odd & even halfset maps by combining them, then enhancing the high frequencies (Halfset Maps Mode). B-factor can be automatically estimated from the unfiltered halfset maps. This mode requires two arguments; use unfiltered halfset maps produced by MERIDIEN.

Post-refine halfset volumes	locked	☑		
First unfiltered halfset map	required	11_MERIDIEN/vol_0_unfil_028.hdf	Select HDF volume	Select any volume
Second unfiltered halfset map	required	11_MERIDIEN/vol_1_unfil_028.hdf	Select HDF volume	Select any volume
Output directory	required	12_POSTREFINER		
Pixel size [A]	1.14	1.14		
3D mask file	none	none	Select HDF volume	Select any volume
Apply adaptive mask	NO	☑		
Binarization threshold	-9999.0	0.02		
Soft-edge width [Pixels]	5	4		
Dilation width [Pixels]	3	2		
MTF file	none	FalconIImtf.txt	Select MTF data	
B-factor enhancement	0.0	0.0		
Low-pass filter frequency [A]	0.0	0.0		

First unfiltered halfset map: **11_MERIDIEN/vol_0_unfil_0*.hdf**

Second unfiltered halfset map: **11_MERIDIEN/vol_1_unfil_0*.hdf**

Output directory: **12_POSTREFINER**

Pixel size [A]: **1.14**

Apply adaptive mask: ✓

Binarization threshold: **0.02** (check in Chimera)

Soft-edge width [Pixels]: **4**

Dilation width [Pixels]: **2** (each dilation corresponds to a binary extension by ~ 2 pixels)

MTF file: **FalconIImtf.txt**

B-factor enhancement: **0.0** (automatic)

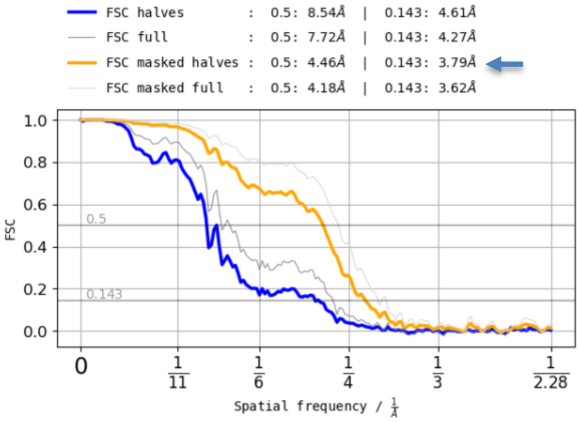
Low-pass filter frequency [A]: **0.0** (to nominal resolution FSC@0.143)

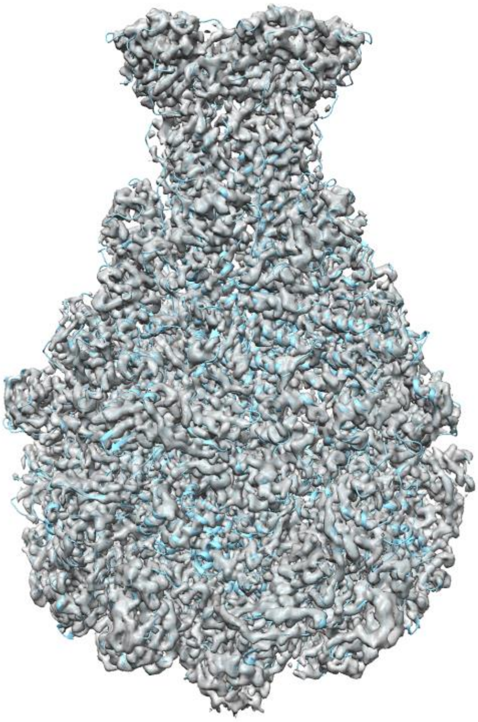
And press the **Run command** button.

2. You can monitor the progress by executing the following command in the terminal `tail -f 12_POSTREFINER/log.txt`.

3. Check the FSC curves 12_POSTREFINER/fsc.png for your final resolution (FSC masked halves) and to see if they look healthy (fall smoothly to 0 and fluctuate around this value) and

4. Open the final volume 12_POSTREFINER/vol_combined.hdf in **Chimera** and compare it to the x-ray structure **PDB: 1VW1** (and the initial model).





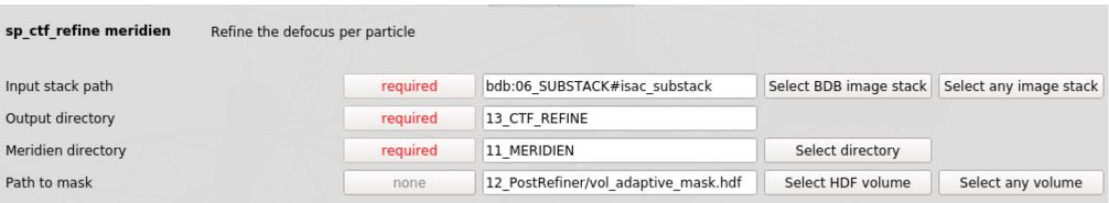
Congratulations! You produced your first near-atomic resolution reconstruction with SPHIRE!

STEP 14. CTF refinement – Per particle defocus

In step 2 we estimated the CTF per micrograph to enable the correction for its effect. Assuming one defocus value for all particles on one micrograph is, however, only an approximation. In reality, particles have different defocus values due to different z-heights in the thin vitrified ice layer. Therefore, we now calculate per particle CTF parameters using CTF refine.

1. Press the MERIDIEN button on the left and then the CTF refine (Meridien) button in the middle (Utilities).

Provide following parameters:



sp_ctf_refine meridien   Refine the defocus per particle

Input stack path	required	bdb:06_SUBSTACK#isac_substack	Select BDB image stack	Select any image stack
Output directory	required	13_CTF_REFINE		
Meridien directory	required	11_MERIDIEN	Select directory	
Path to mask	none	12_PostRefiner/vol_adaptive_mask.hdf	Select HDF volume	Select any volume

Input stack path: **bdb:06_SUBSTACK#isac_substack**

Output directory: **13_CTF_REFINE**

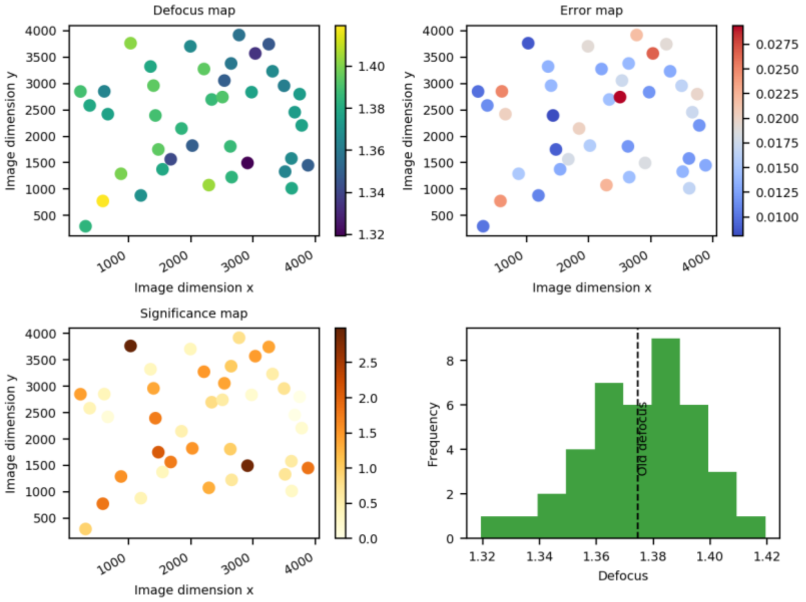
Meridien directory: **11_MERIDIEN**

Path to mask: **12_PostRefiner/vol_adaptive_mask.hdf**

And press the **Run command** button.

2. Monitor the progress of the sp_ctf_refine_meridien job.

3. The output is a folder 13_CTF_REFINE/statistics containing text and image files to assess the performance of the CTF refinement and a virtual stack with per particle defocus values bdb:13_CTF_REFINE/ctf_refined.

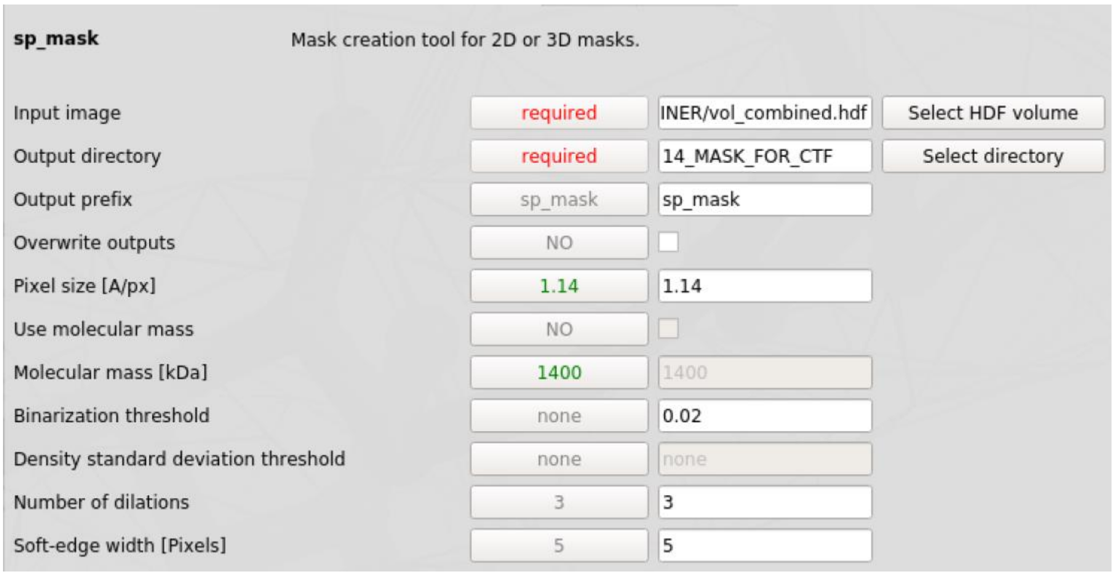


STEP 15. Preparing a mask from the sharpened volume

In the next step we will rerun the 3D refinement using the CTF refined stack (step 14) as input and our sharpened map (step 13) as initial reference. In step 11 we already created a 3D mask from our initial RVIPER model. As it is always a good idea to use a mask that matches the initial reference, we will create a new mask using the Masking tool which is based on the sharpened map.

1. Press the MERIDIEN button on the left and then the Masking button in the middle (Utilities).

Provide following parameters:



sp_mask Mask creation tool for 2D or 3D masks.

Input image	required	INER/vol_combined.hdf	Select HDF volume
Output directory	required	14_MASK_FOR_CTF	Select directory
Output prefix	sp_mask	sp_mask	
Overwrite outputs	NO	□	
Pixel size [A/px]	1.14	1.14	
Use molecular mass	NO	□	
Molecular mass [kDa]	1400	1400	
Binarization threshold	none	0.02	
Density standard deviation threshold	none	none	
Number of dilations	3	3	
Soft-edge width [Pixels]	5	5	

Input image: **12_POSTREFINER/vol_combined.hdf**

Output directory: **14_MASK_FOR_CTF**

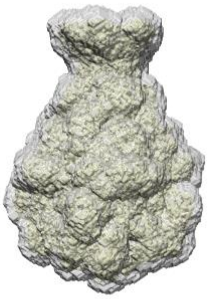
Output prefix: **sp_mask**

Pixel size [A]: **1.14**

Binarization threshold: **0.02** (check in Chimera)

Number of dilations: **3**

Soft-edge width [Pixels]: **5**



And press the **Run command** button

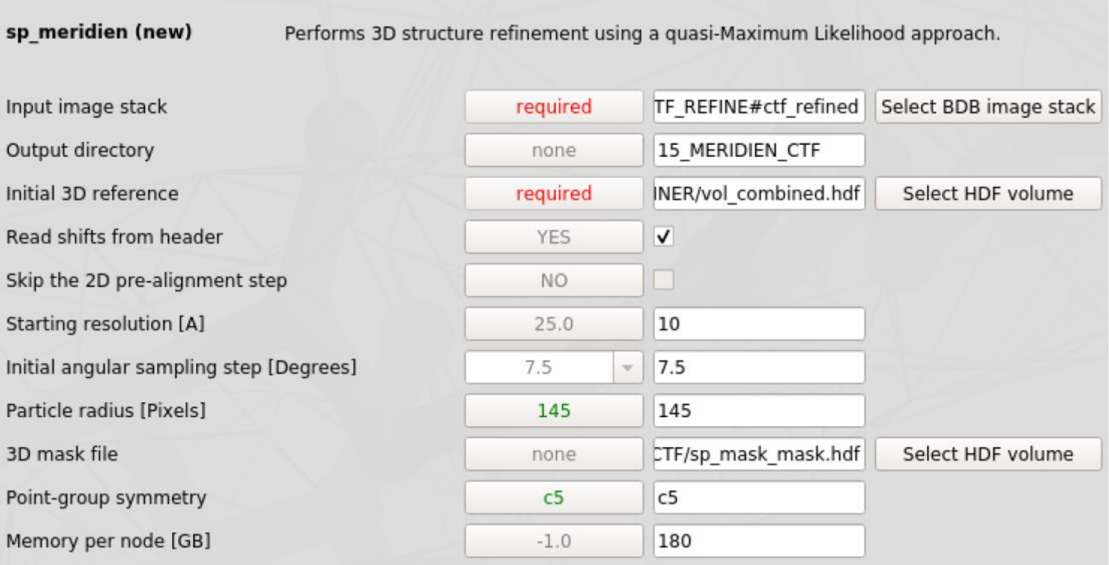
2. Display the resulting 3D mask 14_MASK_FOR_CTF/sp_mask_mask.hdf and the sharpened map 12_POSTREFINER/vol_combined.hdf with **Chimera**.
The mask should enclose the complete volume even at low thresholds.

STEP 16. High Resolution 3D refinement of CTF refined particles

In this step we rerun Meridien – 3D Refinement on the CTF refined particle stack using the sharpened map as a reference and the corresponding mask.

1. Press the MERIDIEN button on the left and then the 3D Refinement button in the middle.

Provide following parameters:



sp_meridien (new) Performs 3D structure refinement using a quasi-Maximum Likelihood approach.

Input image stack	required	TF_REFINE#ctf_refined	Select BDB image stack
Output directory	none	15_MERIDIEN_CTF	
Initial 3D reference	required	INER/vol_combined.hdf	Select HDF volume
Read shifts from header	YES	☑	
Skip the 2D pre-alignment step	NO	□	
Starting resolution [A]	25.0	10	
Initial angular sampling step [Degrees]	7.5	7.5	
Particle radius [Pixels]	145	145	
3D mask file	none	CTF/sp_mask_mask.hdf	Select HDF volume
Point-group symmetry	c5	c5	
Memory per node [GB]	-1.0	180	

Input image stack: **bdb:13_CTF_REFINE#ctf_refined**

Output directory: **15_MERIDIEN_CTF**

Initial 3D reference: **12_POSTREFINER/vol_combined.hdf**

Read shifts from header: ✓

Starting resolution [A]: **10.0**

Initial angular sampling [˚]: **7.5**

Particles radius [Pixels]: **145**

3D mask file: **14_MASK_FOR_CTF/sp_mask_mask.hdf**

Point-group symmetry: **c5**

Memory per node [GB]: Needs to be adjusted to available hardware

And press the **Run command** button

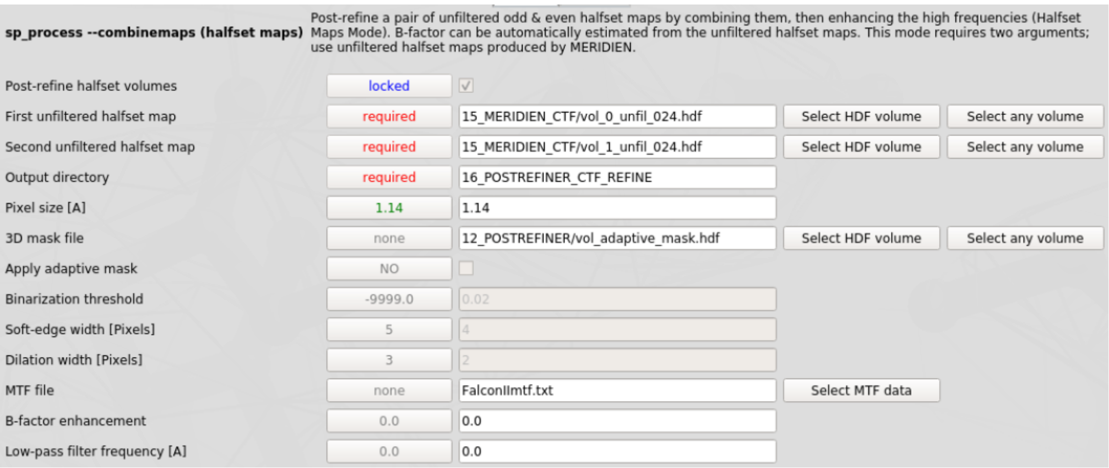
2. Monitor the progress of the sp_meridien_new job and analyze and display results as described in step 12.

STEP 16. Final PostRefiner

We will now compute the final sharpened reconstruction using PostRefiner (see step 13 for details)

1. Press the MERIDIEN button on the left and then the PostRefiner button in the middle.

Provide following parameters:



sp_process --combinemaps (halfset maps) Post-refine a pair of unfiltered odd & even halfset maps by combining them, then enhancing the high frequencies (Halfset Maps Mode). B-factor can be automatically estimated from the unfiltered halfset maps. This mode requires two arguments; use unfiltered halfset maps produced by MERIDIEN.

Post-refine halfset volumes	locked	☑		
First unfiltered halfset map	required	15_MERIDIEN_CTF/vol_0_unfil_024.hdf	Select HDF volume	Select any volume
Second unfiltered halfset map	required	15_MERIDIEN_CTF/vol_1_unfil_024.hdf	Select HDF volume	Select any volume
Output directory	required	16_POSTREFINER_CTF_REFINE		
Pixel size [A]	1.14	1.14		
3D mask file	none	12_POSTREFINER/vol_adaptive_mask.hdf	Select HDF volume	Select any volume
Apply adaptive mask	NO	□		
Binarization threshold	-9999.0	0.02		
Soft-edge width [Pixels]	5	4		
Dilation width [Pixels]	3	2		
MTF file	none	FalconIImtf.txt	Select MTF data	
B-factor enhancement	0.0	0.0		
Low-pass filter frequency [A]	0.0	0.0		

First unfiltered halfset map: **15_MERIDIEN_CTF/vol_0_unfil_0*.hdf**

Second unfiltered halfset map: **15_MERIDIEN_CTF/vol_1_unfil_0*.hdf**

Output directory: **16_POSTREFINER_CTF_REFINE**

Pixel size [A]: **1.14**

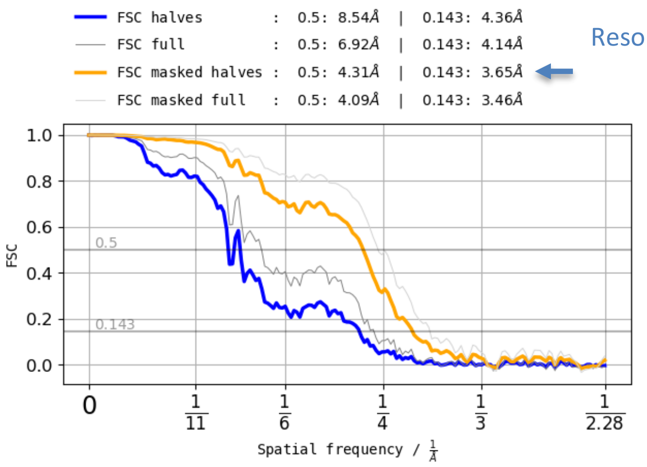
3D mask file: **12_POSTREFINER/vol_adaptive_mask.hdf**

MTF file: **FalconIImtf.txt**

B-factor enhancement: **0.0** (automatic)

Low-pass filter frequency [A]: **0.0** (to nominal resolution FSC@0.143)

And press the **Run command** button.



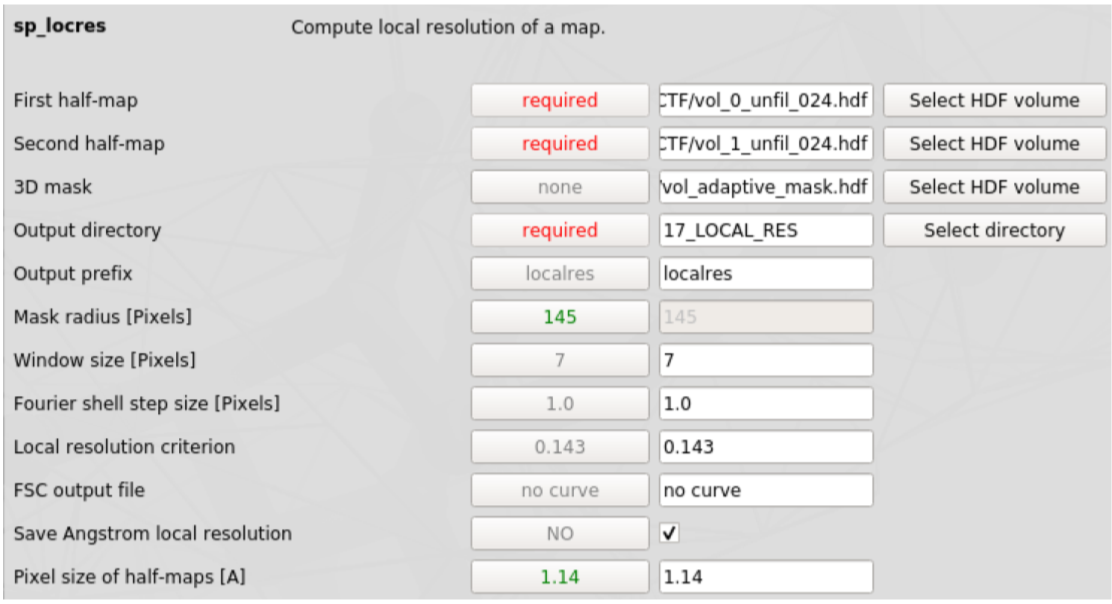
Resolution improved by 0.14 Å

STEP 17. Local Resolution and Local Filter (optional)

Within PostRefiner we filter our sharpened map to the nominal resolution based on the FSC@0.143 criterion. The resolution is however not uniform across the protein but varies. Thus, we will now calculate the Local Resolution and filter our map accordingly using 3D Local Filter.

1. Press the LOCALRES button on the left and then the Local Resolution button in the middle.

Provide following parameters:



sp_locres Compute local resolution of a map.

First half-map	required	CTF/vol_0_unfil_024.hdf	Select HDF volume
Second half-map	required	CTF/vol_1_unfil_024.hdf	Select HDF volume
3D mask	none	vol_adaptive_mask.hdf	Select HDF volume
Output directory	required	17_LOCAL_RES	Select directory
Output prefix	localres	localres	
Mask radius [Pixels]	145	145	
Window size [Pixels]	7	7	
Fourier shell step size [Pixels]	1.0	1.0	
Local resolution criterion	0.143	0.143	
FSC output file	no curve	no curve	
Save Angstrom local resolution	NO	✓	
Pixel size of half-maps [A]	1.14	1.14	

First half-map: **15_MERIDIEN_CTF/vol_0_unfil_0*.hdf**

Second half-map: **15_MERIDIEN_CTF/vol_1_unfil_0*.hdf**

3D mask: **12_POSTREFINER/vol_adaptive_mask.hdf**

Output directory: **17_LOCAL_RES**

Output prefix: **localres**

Window size [Pixels]: **7**

Fourier shell step size [Pixels]: **1.0**

Local resolution criterion: **0.143**

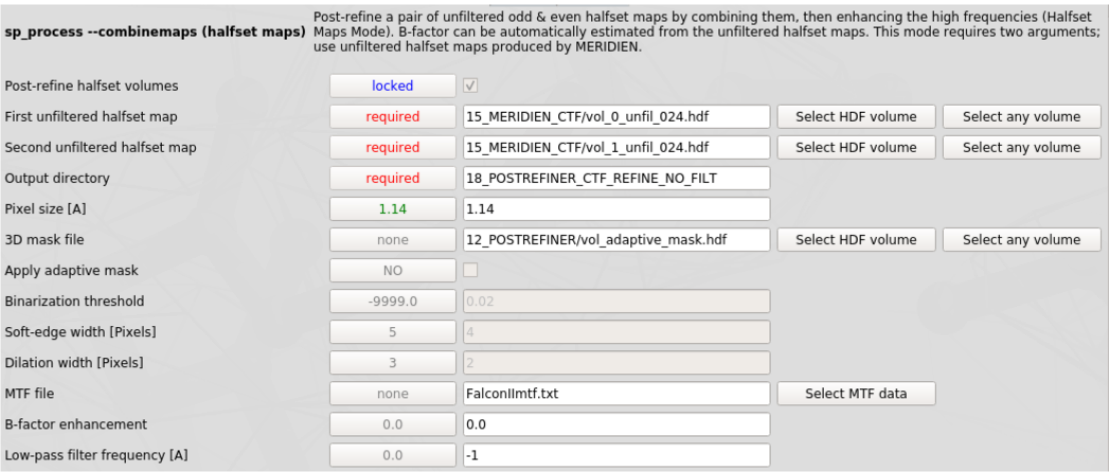
FSC output file: **no curve**

Save Angstrom local resolution: ✓

Pixel size of half-maps [A]: **1.14**

And press the **Run command** button.

2. To filter the final map to local resolution using 3D Local Filter, we need to combine and sharpen half maps without filtering them. Therefore, rerun PostRefiner as described in step 16 but change parameters as indicated below.



sp_process --combinemaps (halfset maps) Post-refine a pair of unfiltered odd & even halfset maps by combining them, then enhancing the high frequencies (Halfset Maps Mode). B-factor can be automatically estimated from the unfiltered halfset maps. This mode requires two arguments; use unfiltered halfset maps produced by MERIDIEN.

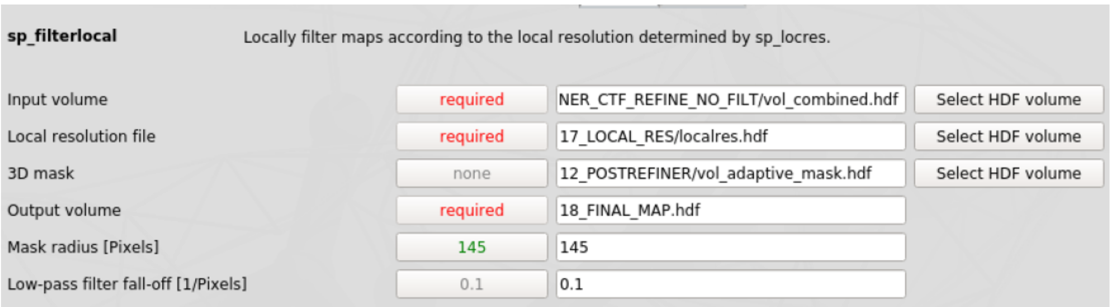
Post-refine halfset volumes	locked	☑		
First unfiltered halfset map	required	15_MERIDIEN_CTF/vol_0_unfil_024.hdf	Select HDF volume	Select any volume
Second unfiltered halfset map	required	15_MERIDIEN_CTF/vol_1_unfil_024.hdf	Select HDF volume	Select any volume
Output directory	required	18_POSTREFINER_CTF_REFINE_NO_FILT		
Pixel size [A]	1.14	1.14		
3D mask file	none	12_POSTREFINER/vol_adaptive_mask.hdf	Select HDF volume	Select any volume
Apply adaptive mask	NO	□		
Binarization threshold	-9999.0	0.02		
Soft-edge width [Pixels]	5	4		
Dilation width [Pixels]	3	2		
MTF file	none	FalconIImtf.txt	Select MTF data	
B-factor enhancement	0.0	0.0		
Low-pass filter frequency [A]	0.0	-1		

Output directory: **18_POSTREFINER_CTF_REFINE_NO_FILT**

Low-pass filter frequency [A]: **-1** (no low pass filter applied)

3. Press the LOCALRES button on the left and then the 3D Local Filter button in the middle.

Provide following parameters:



sp_filterlocal Locally filter maps according to the local resolution determined by sp_locres.

Input volume	required	NER_CTF_REFINE_NO_FILT/vol_combined.hdf	Select HDF volume
Local resolution file	required	17_LOCAL_RES/localres.hdf	Select HDF volume
3D mask	none	12_POSTREFINER/vol_adaptive_mask.hdf	Select HDF volume
Output volume	required	18_FINAL_MAP.hdf	
Mask radius [Pixels]	145	145	
Low-pass filter fall-off [1/Pixels]	0.1	0.1	

Input volume: **18_POSTREFINER_CTF_REFINE_NO_FILT /vol_combined hdf**

Local resolution file: **17_LOCAL_RES /localres.hdf**

3D mask: **12_POSTREFINER/vol_adaptive_mask.hdf**

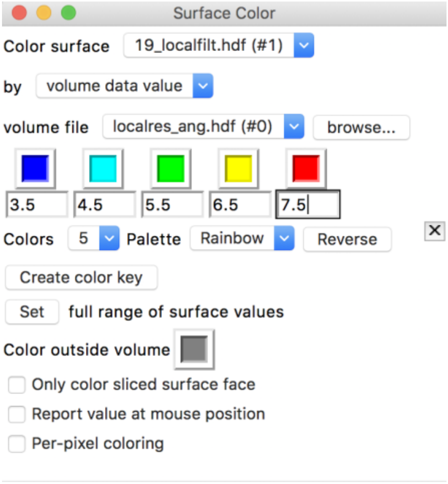
Output volume: **17_LOCAL_RES /19_localfilt.hdf**

Mask radius [Pixels]: **145**

Low-pass filter fall-off [1/Pixels]: **0.1**

And press the **Run command** button.

4. Open the locally filtered map 17_LOCAL_RES/19_localfilt.hdf and the local resolution file 17_LOCAL_RES/localres_ang.hdf in **Chimera**. Finally, color the final map by its local resolution by clicking Tools ➜ Volume Data ➜ Surface Color



Surface Color

Color surface 19_localfilt.hdf (#1)

by volume data value

volume file localres_ang.hdf (#0) browse...

🟦	🟦	🟩	🟨	🟥
3.5	4.5	5.5	6.5	7.5

Colors 5 Palette Rainbow Reverse

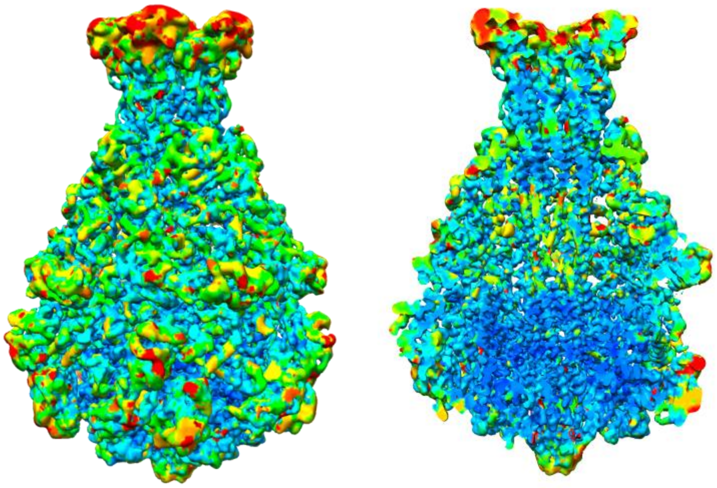
Set full range of surface values

Color outside volume

□ Only color sliced surface face

□ Report value at mouse position

□ Per-pixel coloring



Congratulations!
You successfully finished the SPHIRE short tutorial.

If you are interested in our other features and more advanced tips please download our full-length tutorial at: http://www.sphire.mpg.de