SPHIRE

SParx for **HI**gh **R**esolution **E**lectron Microscopy

structure determination with SPHIRE

a practical guide

Info?
Turn here!

SPHIRE 1.3 - September 2019

Written by Christos Gatsogiannis with input from the SPHIRE team; Latex document by Markus Stabrin

This guide contains a description of the general workflow of a cryo-EM project using SPHIRE 1.3. It is based on the GUI of SPHIRE and demonstrates how to obtain high-resolution 3D maps of macromolecular complexes from cryo-EM images. It will direct you through all steps of SPA based on a provided experimental data set. After completing the tutorial, you should be able to independently process your own data. In addition, almost every step contains a "**TIP**" section, with more advanced instructions that might help you to obtain the best results from your own data. Optional steps, performed outside the GUI, as well as known issues of the version 1.3 release are described within info boxes.

A detailed SPHIRE documentation is available at:

`http://sphire.mpg.de`

There is a mailing list for user support, discussions and future announcements.

You can subscribe at:

`https://listserv.gwdg.de/mailman/listinfo/sphire` and search the archives for your topic of interest.

Once you have subscribed, you can also use the address:

sphire@lists.mpg.de

to send e-mails to the SPHIRE team.

Contents

Software Installation

Install SPHIRE and MPI support

To follow this guide and run SPHIRE you will need to properly install SPHIRE, either on a Linux computing cluster or a Linux multi-core desktop. In both case an MPI installation is required. Download SPHIRE at

`http://sphire.mpg.de/wiki/doku.php?id=howto:download_latest`

and follow the instructions regarding SPHIRE and MPI installation. It should be noted that SPHIRE and **EMAN2** (Tang et al. 2007) are installed jointly, i.e., after installation of SPHIRE, **EMAN2** is also available and vice versa. Note that SPHIRE requires MPI installation to use the most advanced commands, while **EMAN2** does not. Both SPHIRE and **EMAN2** are issued under a joint BSD/GNU license (see the documentation) and are provided free of charge as a service to the scientific community.

To make sure the installation completed successfully, open a terminal, type ***sphire***, hit the **Return** key and check if the SPHIRE GUI pops up. To make sure the MPI installation finished successfully, type ***sp_sx.py*** in the terminal window and then type ***import mpi***. If there are no error messages, exit the interactive mode by typing ***quit***. Otherwise, it is advisable to contact either the local IT support or the SPHIRE team.

Install crYOLO

For particle selection from the digital micrographs, we will use the automated software **crYOLO**. Download **crYOLO** at

`http://sphire.mpg.de/wiki/doku.php?id=howto:download_latest`

and follow the instructions. If you do not have access to a NVIDIA GPU, you can always download the CPU version of the program. The CPU version of **crYOLO** will perform approximately 6x slower.

Installation of associated EM Software Packages

Alignment of direct detector movie frames is currently not included in SPHIRE. However, within the framework of the GUI we provide a wrapper as a utility to **Unblur** (Grant et al. 2015). Please download the program **Unblur**

(`http://grigoriefflab.janelia.org/unblur`, Grigorieff lab)

and follow the installation instructions.

For interactive visualization of the resulting structures, we use the molecular graphics program **UCSF CHIMERA** (Pettersen et al. 2004b) (`https://www.cgl.ucsf.edu/chimera/download.html`). An extensive tutorial to get familiar with the features of **UCSF CHIMERA** can be found here: `https://www.cgl.ucsf.edu/chimera/data/tutorials/groel/groel.html`

PROJECT

In this tutorial, we use the SPHIRE GUI to determine a 3D cryo-EM structure of the bacterial toxin component TcdA1 (Gatsogiannis et al. 2013) from a test dataset of 112 micrographs. The images were collected on a Cs-corrected FEI Titan Krios, operated at 300 kV and equipped with an XFEG and a Falcon II direct detector. The digital micrographs have a pixel size of 1.14 Å, and the particle is a pentameric assembly with C5 point group symmetry.

First, create a **project directory** named **SPHIRE-demo**.

```
mkdir SPHIRE-demo
```

The **project directory SPHIRE-demo** will be created. As it is necessary to **always start the SPHIRE GUI and run all project-associated commands from the project directory**, change to this directory:

```
cd SPHIRE-demo
```

Download the tutorial data. At the terminal type (one long command):

```
wget --no-check-certificate
https://ftp.gwdg.de/pub/misc/sphire/test_dataset/sphire_testdata_movies.tar
```

(Alternatively, you can use your file browser to download the data from here:

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

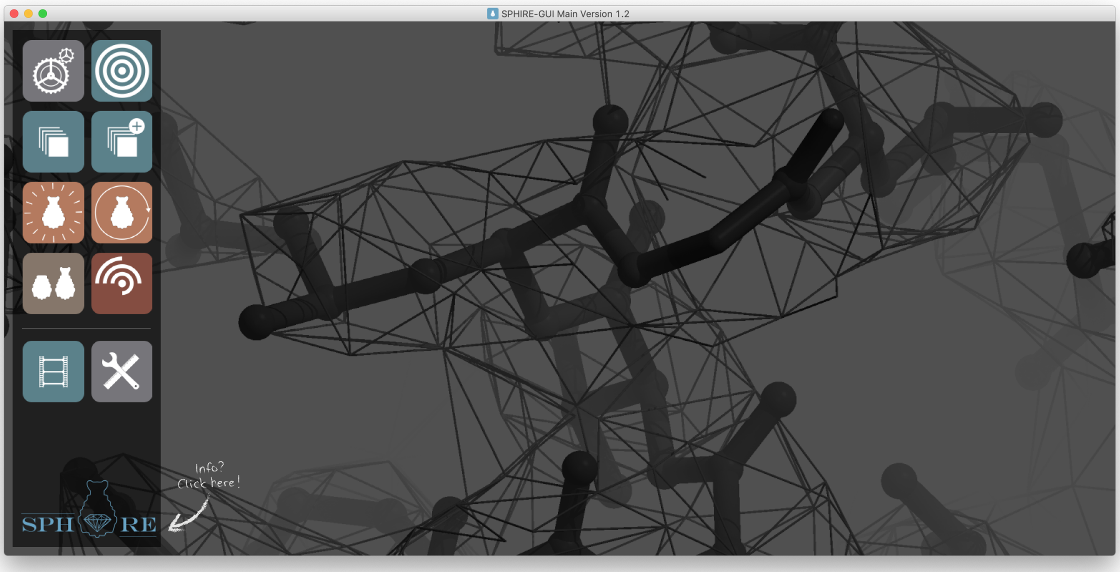
In order to extract the downloaded file, type:

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

To launch the SPHIRE GUI, type:

```
sphire
```

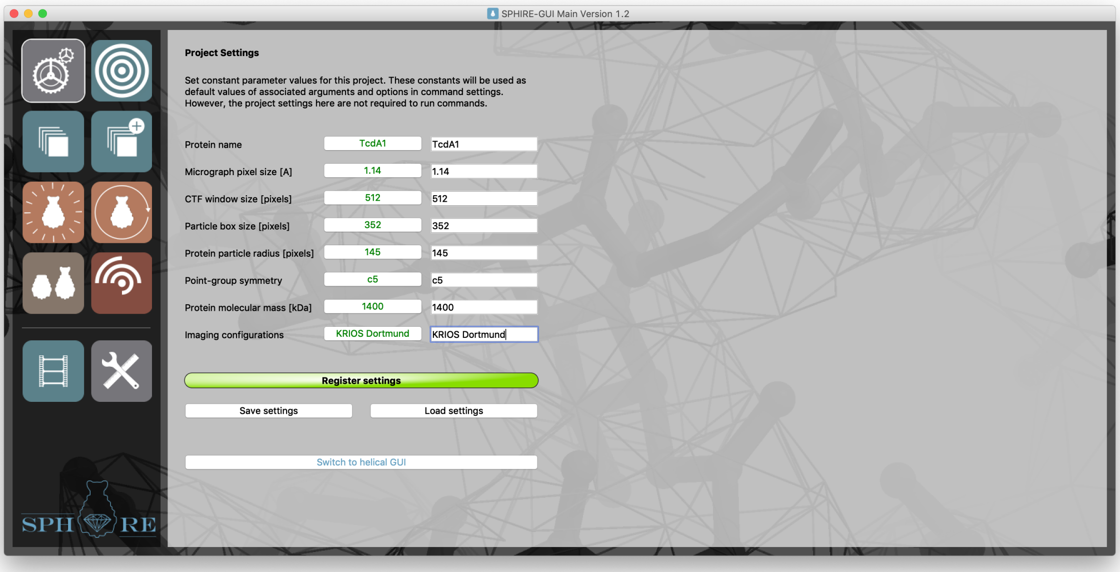
When SPHIRE is started for the first time, it will ask for permission to write a settings directory. Answer **y**. The main window will then appear.



The column on the left contains several pictograms that allow you to select a particular step of the single particle image analysis workflow. The first step is to provide the project-wide constant-value parameters. These constants will be used as default values for the subsequent steps of the processing workflow.

> **TIP:** *You can also skip registering these settings now, but in this case you will have to specify these standard parameters multiple times during the processing procedure.*

Click the ***PROJECT*** pictogram and fill out the following fields in the GUI interface that will appear:



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	KRIOS Dortmund	KRIOS Dortmund

NOTE: *Clicking the gray button, next to the respective input field, retrieves a default value. After the settings are registered here, they will become default values in the subsequent steps.*

- **Protein name:** TcdA1
- **Micrograph pixel size [A]:** 1.14

  TIP: *Do not forget to adjust the pixel size if you have already binned your micrographs. For example, micrographs recorded with the K2 in super-resolution mode have a very fine pixel size, and usually we bin them during the movie alignment before we start image processing in SPHIRE (see next section). If you already aligned your movies and you wish to bin all micrographs for example two times (ratio of new to old image size=0.5), you can use bash batch processing:*

  ```
  mkdir binned_images
  ```

  and afterwards (one long command)

  ```
  for file in $(ls *.mrc);do sp_process.py ${file} binned_images/${file}.mrc
  --changesize --ratio=0.5; done
  ```

  To bin a single micrograph, you can use the utility ***Change Size of Image or Volume***

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

  NOTE: *Should be slightly larger than particle size*

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

NOTE: *The particle box size should be at least 1.5 × to 2 × larger than the longest axis of the particle. The long axis of the tutorial particle is 25 nm (250 Å) and the pixel size is 1.14 Å. Thus, the long axis of the particle is 250 Å / 1.14 Å/pixel = 220 pixel. Here we set the box size to 352 pixel, which corresponds to 1.6 × the size of our particle.*

NOTE: *The window size has to be an even number due to Fast Fourier Transform (FFT) requirements of some programs.*

TIP: *If you recorded images at rather high defocus values (i.e., > 3 µm), you might want to consider using an even larger box size.*

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

  NOTE: *Adjust this parameter carefully, so the radius set here will correspond to at least half of the longest axis of the particle.*

  TIP: *This value will be used to create a circular mask for data processing, both in 2D as well as in 3D. Therefore, at the beginning of the project and especially if the center of the particle does not coincide with its center of mass, it might be preferable to use a slightly larger particle radius. For globular complexes that can be easily centered, such as ribosomes, you should use a rather tight radius closely corresponding to the particle radius right from the beginning.*

- **Point-group symmetry:** C5

  NOTE: *If the point group symmetry of the particle is uncertain, do not assume symmetry and enter C1.*

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

  NOTE: *It can be approximate. The molecular mass of the protein is used for validation purposes in some steps of data processing.*

- **Imaging configurations:** KRIOS Dortmund

  NOTE: *Your microscope name/configuration. It is optional and is kept for record-keeping only.*

Click the ***Register settings*** button, to register these values as defaults for all subsequent steps of the workflow, and then click the ***Save settings*** button, to save them in a text file. This text file may be useful for entering information into your lab book or can be used to reload the settings if necessary. A detailed description of the options is provided on our wiki page. Even if you do not save these settings now, registered parameters will be automatically displayed every time you start the GUI from the **project directory**.

Measuring the size of your particle

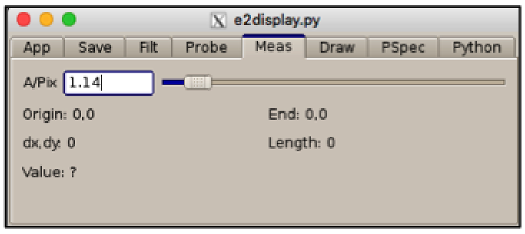
Before collecting high-resolution movies, you most probably screened the quality of your sample by negative stain EM and also possibly collected some cryo-EM overview images. You can use one of these images to measure the size of your particle using **e2display.py**.

1. Click the button ***UTILITIES*** on the left and then the Button ***Display Data*** in the middle to launch **e2display.py**.

2. Open an overview image with sufficient contrast.

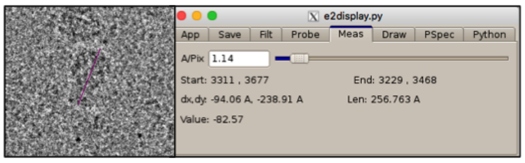
3. Click the **mouse wheel button** somewhere on the image. A pop-up window will appear.

4. Activate the ***Meas*** tab in the pop-up window, set the pixel size (A/Pix) and hit the **Return** key.



A/Pix 1.14
Origin: 0,0 End: 0,0
dx,dy: 0 Length: 0
Value: ?

5. Now identify the projection view of your particle with the largest size. Keep the **left mouse button** pressed and draw a line to measure the longest axis of the particle. The length of the line in Å will be reported in the e2display.py pop up window (**Len**).



A/Pix 1.14
Start: 3311 , 3677 End: 3229 , 3468
dx,dy: -94.06 A, -238.91 A Len: 256.763 A
Value: -82.57

6. To get the particle radius in pixels, divide the result by the pixel size:

$$\text{Len in pixels} = \frac{\text{Length in angstrom}}{\text{Pixel size}} = \frac{256}{1.14} \approx 112$$

Movie

Micrograph Movie Alignment

Modern direct detectors record high-resolution images as movie frames. Usually, we motion-correct the micrograph movies on the fly during image acquisition (using for example **tranSPHIRE**). If motion-correction has not been performed yet, the first step in the workflow is the alignment of these frames for each image to correct the overall motion. SPHIRE provides as a utility, a wrapper of the program **Unblur**, developed by the Grigorieff lab.

> **TIP:** *Not all microscopes are equipped with cameras with movie capabilities, and generally only high-resolution projects require recording movie frames. SPHIRE is very flexible in this respect and it is possible to skip certain steps of the workflow as long as the appropriate settings are entered. This is critical when importing data created with different software packages. Please read the TIP section for each step of the workflow carefully and follow the instructions that best fit your processing scenario.*

> **TIP:** *If you are not processing movies, you can directly copy the digital micrographs to a folder within the* ***project directory*** *and proceed directly to the* ***CTER*** *(CTF Estimation) Section of the Tutorial.*

> **TIP:** *If you already used* ***Unblur*** *or* ***MotionCor2*** *to align your movies, copy all results (including the logfiles) to the output folder* ***CorrectedSums*** *within the* ***project directory*** *and proceed to the next step.*

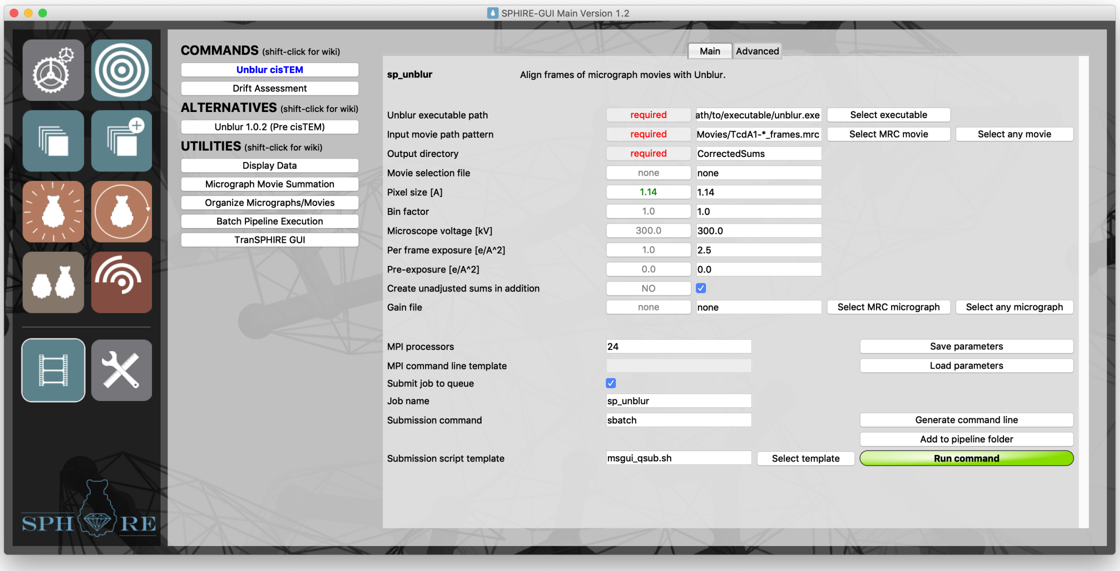
> **TIP:** *If you are processing* ***negative stain*** *data, you can directly proceed to the* ***WINDOW*** *(Particle Extraction) section of the Tutorial.*

During unpacking of the tutorial data, a folder called **Movies** was created within the **project directory**. To see the content of this folder, type:

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

This folder contains 112 low-dose movies of TcdA1 in the MRC file format.

In the main window of the SPHIRE GUI click the button ***MOVIE*** on the lower left corner and then the ***Unblur cisTEM*** button in the middle. An activated command button contains blue highlighted text.



sp_unblur Align frames of micrograph movies with Unblur.

Unblur executable path	required	ath/to/executable/unblur.exe
Input movie path pattern	required	Movies/TcdA1-*_frames.mrc
Output directory	required	CorrectedSums
Movie selection file	none	none
Pixel size [A]	1.14	1.14
Bin factor	1.0	1.0
Microscope voltage [kV]	300.0	300.0
Per frame exposure [e/A^2]	1.0	2.5
Pre-exposure [e/A^2]	0.0	0.0
Create unadjusted sums in addition	NO	✓
Gain file	none	none
MPI processors	24	
MPI command line template		
Submit job to queue	✓	
Job name	sp_unblur	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following input fields:

- **unblur executable path:** /path/to/executable/unblur.exe

  NOTE: *Type here the path to the* ***Unblur*** *executable file or click the* ***Select executable*** *button to use your file browser to select it.*

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

  NOTE: *Click the* ***Select MRC movie*** *button, choose* ***MRC (*.mrc)*** *as file type, and use the file browser to select a movie file. Then replace the variable part of the file name with the wildcard character "*". In the case of this tutorial dataset, the variable part of the file name is only the serial number (****TcdA1-0001_frames.mrc****).*

  IMPORTANT: All movie files in a project must contain the same number of frames and have the same pixel size

  TIP: *It is preferable, but not necessary, to use serial numbers in file names that have the same number of digits (e.g. use mic_0001.mrc, mic_0010.mrc instead of mic_1.mrc, mic_10.mrc).*

- **Output directory:** CorrectedSums
- **Movie selection file:** none
- **Pixel size [A]:** 1.14
- **Bin Factor:** 1.0

TIP: *If you wish to bin the resulting motion-corrected averages, type the respective binning factor here. Afterwards, adjust the **Micrograph Pixel Size** in the **General Project Settings** accordingly. For example, if your movies have a pixel size of 0.6Å/pix and you set here a bin factor of 2, you would have to change the **Micrograph Pixel Size** parameter in the general settings to 1.2 Å/pix.*

TIP: *Expert users can also click the **Advanced** button to display the advanced settings. However, the default values rarely need to be changed for standard projects, therefore these parameters will not be described in this tutorial.*

IMPORTANT: Dose weighting is activated by default and can be deactivated in the advanced settings

- **Microscope voltage [kV]:** 300.0
- **Per-frame Exposure [e/A^2]:** 2.5

NOTE: *Each movie of this dataset contains 24 frames with a total dose of 60 e/Å^2. Thus, the per-frame exposure is*

$$\frac{\textit{Total dose}}{\textit{Number of movie frames}} = \frac{60\,\mathrm{e}/\mathring{A}^2}{24} = 2.5\,\mathrm{e}/\mathring{A}^2 \,.$$

- **Pre-exposure [e/A^2]:** 0.0
- **Create unadjusted sums in addition:** YES

TIP: *Activate this option to additionally compute motion-corrected averages **without dose weighting**. These images will be required for the subsequent CTF estimation step.*

- **Gain File:** None

Specify the number of processors (we used 24 for this job) and submit the job to the queuing system of the cluster using an appropriate submission file (see below) by clicking the ***Run command*** button.

IMPORTANT: Number of processors should always be lower than the total number of micrographs.

A SGE template file can be found in your project directory (**msgui_qsub.sh**). You might need to configure or prepare a different template file for your own system. If you are not familiar with your cluster and its queuing system, you should seek help from the system IT administrator in order to prepare the correct file. Details about setting up the template file can be found on the SPHIRE wiki. (`http://sphire.mpg.de/wiki/doku.php?id=howto:submissions`) Monitor the progress of the job through the standard output. The name of the logfile containing the standard output depends on your submission file. At the terminal, type:

```
tail -f name_of_sp_unblur_logfile
```

```
main =>20.00% =>Elapsed time: 0.87min | Estimated total time: 4.35min | Time per micrograph:
0.87min/mic
```

On our Linux cluster, using 24 processors, the alignment of the 112 movies required about 6 minutes. Otherwise, if you need to save time, you can copy the pre-calculated results to your **project directory** and continue with those. To do so, type:

```
cp -Rp SphireDemoResults/CorrectedSums ./
```

To see the content of the CorrectedSums output folder, type:

```
ls CorrectedSums
```

The output folder contains following subfolders:

– **corrsum:** Contains the motion-corrected averages computed **without dose weighting**.

– **corrsum_dw:** Contains the motion-corrected, **dose-weighted** averages for all movie files

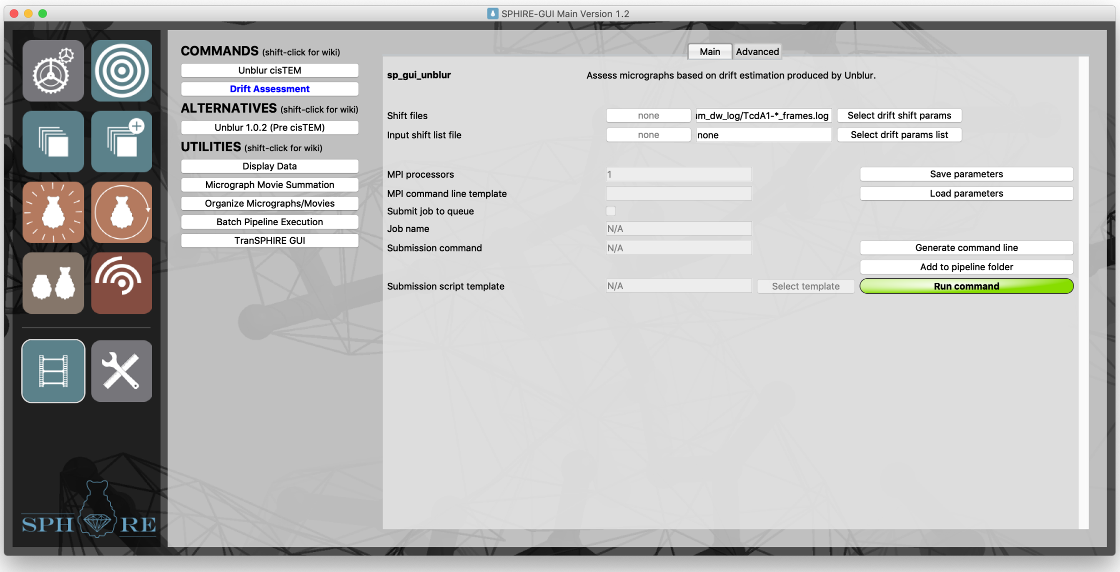
– **corrsum_dw_log and corrsum_log:** Contains the outputs of **Unblur**

The motion-corrected averages computed without dose weighting are necessary for CTF estimation, whereas the dose-weighted averages will be used for all other steps of the workflow.

Drift Assessment

Even after frame-alignment, some image averages still have a significant amount of drift (due to charging, holder instability or local grid damage) and some high-resolution information is lost. The next step is to identify and exclude these images from the subsequent steps of the workflow. For this purpose, we will analyze the results of **Unblur** using our **Drift Assessment GUI** tool (**sp_gui_unblur.py**) to select images with the lowest amount of drift.

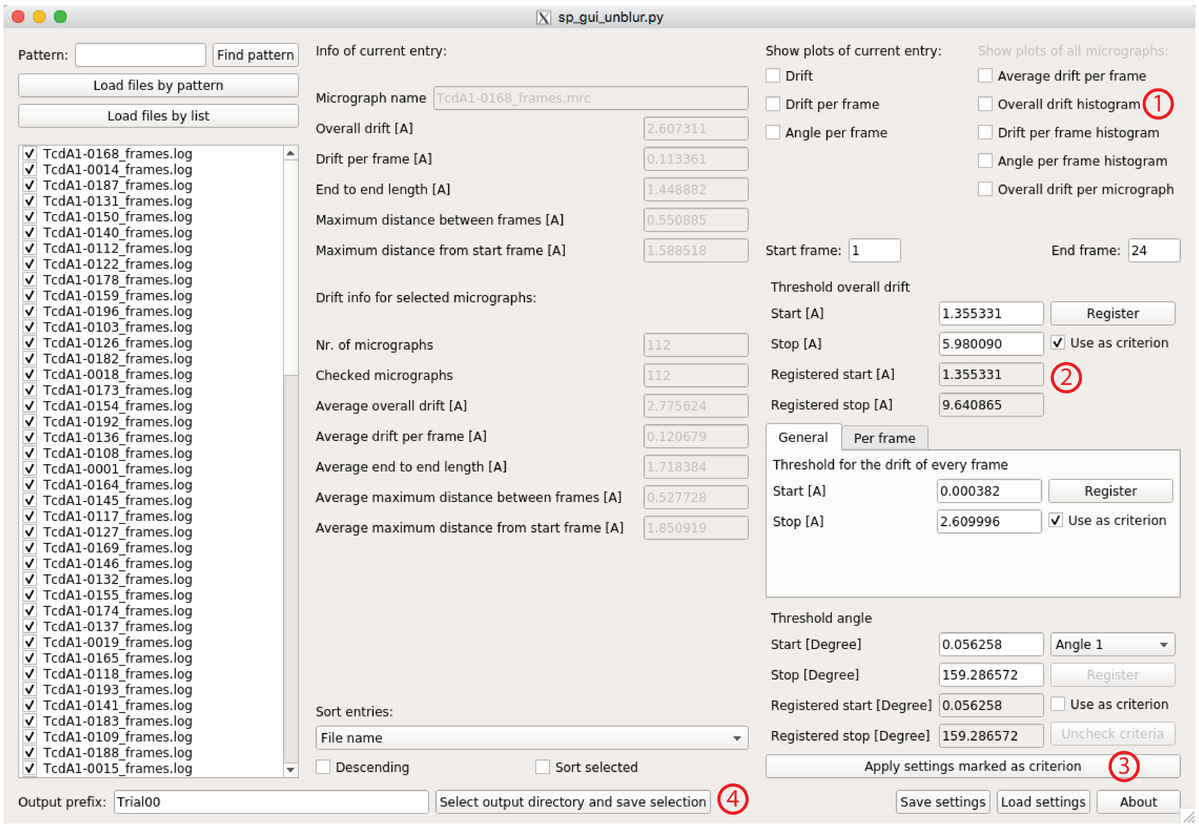
In the main window of the SPHIRE GUI first click the button ***MOVIE*** on the left and then the button ***Drift Assessment*** in the middle.


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Unblur cisTEM
Drift Assessment
ALTERNATIVES (shift-click for wiki)
Unblur 1.0.2 (Pre cisTEM)
UTILITIES (shift-click for wiki)
Display Data
Micrograph Movie Summation
Organize Micrographs/Movies
Batch Pipeline Execution
TranSPHIRE GUI
Main
Advanced
Select drift shift params
Select drift params list
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

sp_gui_unblur — Assess micrographs based on drift estimation produced by Unblur.

Shift files: none — ım_dw_log/TcdA1-*_frames.log
Input shift list file: none — none
MPI processors: 1
MPI command line template:
Submit job to queue: ☐
Job name: N/A
Submission command: N/A
Submission script template: N/A

Click the ***Select drift shift params*** button next to **Shift files,** and use the file browser to select one shift file (**CorrectedSums/corrsum_dw_log/TcdA1-0001_frames.log**). Replace the micrograph number with the wildcard "*" and then click the ***Run command*** button to open the GUI.



Pattern:

☑ TcdA1-0168_frames.log
☑ TcdA1-0014_frames.log
☑ TcdA1-0187_frames.log
☑ TcdA1-0131_frames.log
☑ TcdA1-0150_frames.log
☑ TcdA1-0140_frames.log
☑ TcdA1-0112_frames.log
☑ TcdA1-0122_frames.log
☑ TcdA1-0178_frames.log
☑ TcdA1-0159_frames.log
☑ TcdA1-0196_frames.log
☑ TcdA1-0103_frames.log
☑ TcdA1-0126_frames.log
☑ TcdA1-0182_frames.log
☑ TcdA1-0018_frames.log
☑ TcdA1-0173_frames.log
☑ TcdA1-0154_frames.log
☑ TcdA1-0192_frames.log
☑ TcdA1-0136_frames.log
☑ TcdA1-0108_frames.log
☑ TcdA1-0001_frames.log
☑ TcdA1-0164_frames.log
☑ TcdA1-0145_frames.log
☑ TcdA1-0117_frames.log
☑ TcdA1-0127_frames.log
☑ TcdA1-0169_frames.log
☑ TcdA1-0146_frames.log
☑ TcdA1-0132_frames.log
☑ TcdA1-0155_frames.log
☑ TcdA1-0174_frames.log
☑ TcdA1-0137_frames.log
☑ TcdA1-0019_frames.log
☑ TcdA1-0165_frames.log
☑ TcdA1-0118_frames.log
☑ TcdA1-0193_frames.log
☑ TcdA1-0141_frames.log
☑ TcdA1-0183_frames.log
☑ TcdA1-0109_frames.log
☑ TcdA1-0188_frames.log
☑ TcdA1-0015_frames.log

Info of current entry:

Micrograph name: TcdA1-0168_frames.mrc
Overall drift [A]: 2.607311
Drift per frame [A]: 0.113361
End to end length [A]: 1.448882
Maximum distance between frames [A]: 0.550885
Maximum distance from start frame [A]: 1.588518

Drift info for selected micrographs:

Nr. of micrographs: 112
Checked micrographs: 112
Average overall drift [A]: 2.775624
Average drift per frame [A]: 0.120679
Average end to end length [A]: 1.718384
Average maximum distance between frames [A]: 0.527728
Average maximum distance from start frame [A]: 1.850919

Sort entries: File name
☐ Descending ☐ Sort selected

Output prefix: Trial00

Show plots of current entry:
☐ Drift
☐ Drift per frame
☐ Angle per frame

☐ Average drift per frame
☐ Overall drift histogram
☐ Drift per frame histogram
☐ Angle per frame histogram
☐ Overall drift per micrograph

Start frame: 1 End frame: 24

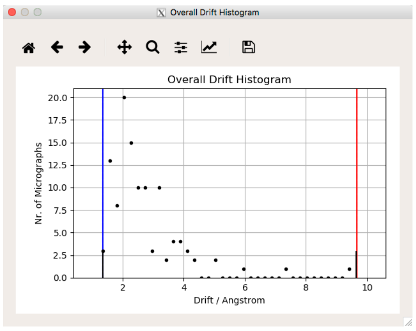
Threshold overall drift
Start [A]: 1.355331
Stop [A]: 5.980090 ☑ Use as criterion
Registered start [A]: 1.355331
Registered stop [A]: 9.640865

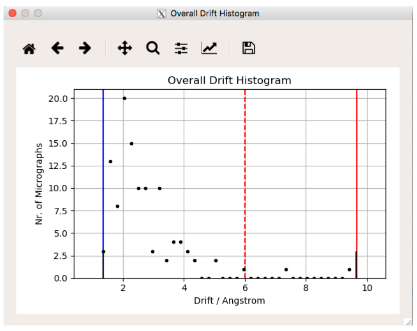
Threshold for the drift of every frame
Start [A]: 0.000382
Stop [A]: 2.609996 ☑ Use as criterion

Threshold angle
Start [Degree]: 0.056258 Angle 1
Stop [Degree]: 159.286572
Registered start [Degree]: 0.056258 ☐ Use as criterion
Registered stop [Degree]: 159.286572

*TIP: The **Drift Assessment GUI** also supports analysis of **MotionCor2** shift files. In this case, click the **Select drift shift params** button, and use the file browser to select one **MotionCor2** shift file (*-Full.log file), replace the variable file name with the wildtype character "*" and click the **Run command** button.*

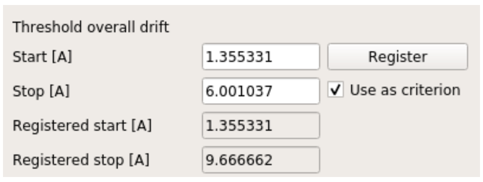
Check the ***Overall Drift Histogram*** (1) checkbox. On the plot for the overall drift (y-axis: number of micrographs, x-axis: overall drift in Å) you can set a threshold to discard micrographs with an overall drift higher than a specific value (red line).





In the tutorial dataset, the average overall drift is 2.77 Å (drift info for selected micrographs). Right click the plot to set the discard threshold to roughly 6 Å, in order to remove some outlier images.

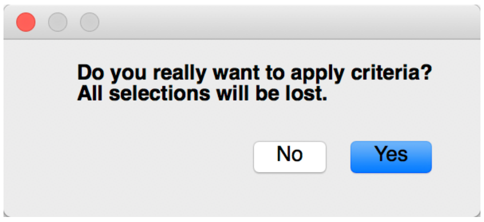
In order to register this threshold, click the ***Register*** button (2).



Threshold overall drift

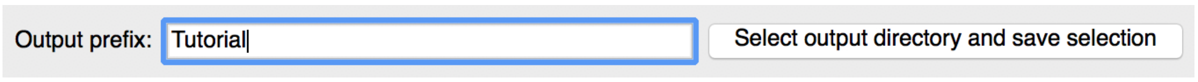
Start [A]	1.355331	Register
Stop [A]	6.001037	✓ Use as criterion
Registered start [A]	1.355331	
Registered stop [A]	9.666662	

Click ***Apply registered settings marked as criterion*** (3) and confirm the pop-up message box:



Do you really want to apply criteria?
All selections will be lost.

Now type **Tutorial** as the prefix name for the output list files and click the ***Select output directory and save selection*** button (4) to specify an output directory (make a new folder with the name **DriftAssess**)



Output prefix: Tutorial

Four text files are then written to the folder **DriftAssess** in the **project directory**.

- **Tutorial_selected.txt:** List of selected micrographs; 109 micrographs (97 %)
- **Tutorial_discarded.txt:** List of discarded micrographs; 3 micrographs (2 %)
- **Tutorial_shift_selected.txt:** Shifts of selected micrographs
- **Tutorial_shift_discarded.txt:** Shifts of discarded micrographs

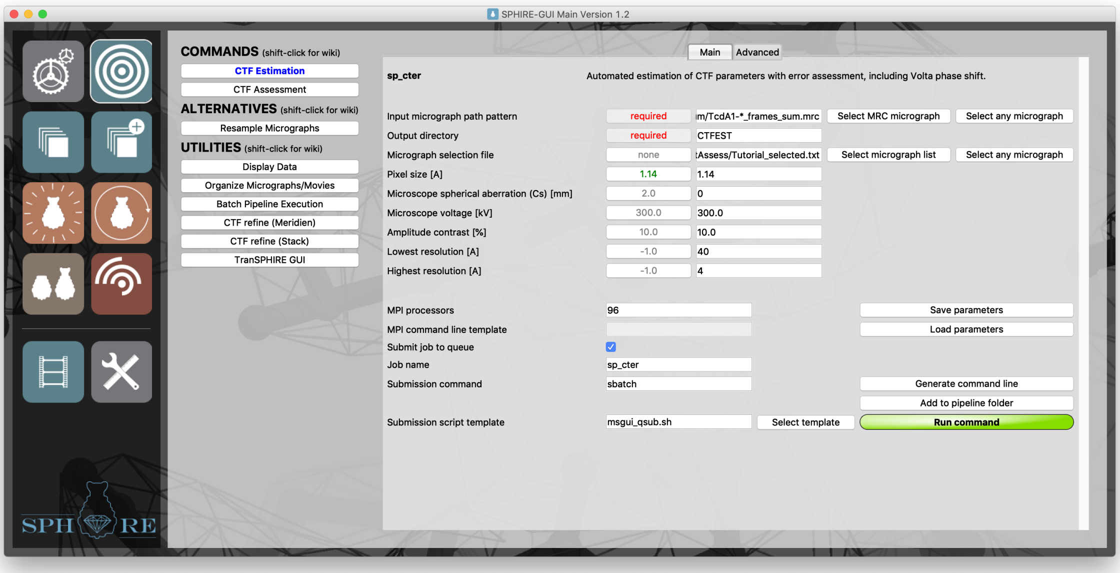
The average overall drift of the selected micrographs is now reduced to 2.64 Å.

TIP: *A more advanced usage of the **Drift Assessment GUI** is described on our wiki page. For example, if your dataset contains a sufficient number of micrographs, selecting the movies with the lowest drift rate during the first frames (drift per frame criterion) might also have a positive effect on the final resolution of the map.*

CTER

CTF Estimation

The next step is to estimate the CTF parameters of the selected motion-corrected micrographs that are included in the list **Tutorial_selected.txt**. Our CTF estimation tool is based on CTER (Automated estimation of CTF parameters with error assessment, (Penczek et al. 2014)). In addition to finding the parameters of the CTF, we will also perform extensive analysis of the quality of the fit and the reliability of individual parameters: defocus, astigmatism amplitude and angle, and, if selected, the phase shift. These error estimates allow you to automatically remove unreliable fits, which often are caused by the inferior quality of some micrographs. In doubtful cases the CTF assessment GUI makes it possible to review both the micrographs and CTF curves to gain more insight into the possible reasons of a poor fit. The most common causes are non-uniform micrograph fields (ice patches, contamination, mistakes with the electron dose) and/or defocus settings that are incompatible with other microscope settings or data sampling parameters (the so-called "Fourier aliasing" problems described in (Penczek et al. 2014)). Note the CTF estimation will be performed exclusively on the micrographs averaged **without** dose-weighting. In the main window of the SPHIRE GUI click the button ***CTER*** on the left and then the button ***CTF Estimation*** in the middle.



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

Input micrograph path pattern	required	m/TcdA1-*_frames_sum.mrc
Output directory	required	CTFEST
Micrograph selection file	none	tAssess/Tutorial_selected.txt
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
MPI processors		96
MPI command line template		
Submit job to queue		✓
Job name		sp_cter
Submission command		sbatch
Submission script template		msgui_qsub.sh

Fill out the following fields:

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

NOTE: *Click the* ***Select MRC micrograph*** *button, and use the file browser to select one of the motion-corrected averages computed* ***without*** *dose-weighting in the* ***corrsum*** *folder. Then replace the variable part of the file name (the serial number) with the wildcard character "*".*

- **Output directory:** CTFEST
- **Micrograph selection file:** DriftAssess/Tutorial_selected.txt

  NOTE: *Click the* ***Select micrograph list*** *button, and use the file browser to load the list of selected micrographs created in the previous step.*

- **Pixel size [A]:** 1.14
- **Microscope spherical aberration (Cs) [mm]:** 0

  NOTE: *The images of the tutorial dataset were collected on a Cs-corrected Titan Krios*

- **Microscope voltage [kV]:** 300.0
- **Amplitude contrast [%]:** 10

  TIP: *Amplitude contrast is typically in the range of 7 % to 14 %. 10 % yields good results for many projects in our lab.*

- **Lowest resolution [A]:** 40

  NOTE: *Low-resolution cut-off.*

- **Highest resolution [A]:** 4

  NOTE: *High-resolution cut-off.*

Advanced parameters:

- **Use PW Spectrum:** YES

IMPORTANT: Activate this option only if Thon rings are unambiguously visible beyond 4 Å in the power spectra of your micrographs.

Specify the number of processors (we used 96 for this job) and submit the job to the queuing system of the cluster using an appropriate submission file by clicking the ***Run command*** button.

IMPORTANT: Number of processors should be always lower than the total number of micrographs.

Monitor the progress of the job through the standard output. At the terminal, type:

```
tail -f name_of_sp_cter_logfile
```

```
Micrographs processed by main process (including percent of progress):

Processing CorrectedSums/corrsum/TcdA1-0001_frames_sum.mrc ---> 0.00%
```

On our cluster, this process finished after about 3 minutes. However, if you do not have enough time to wait for the results, copy our pre-calculated results to your project directory.

```
cp -r SphireDemoResults/CTFEST ./
```

Once the job has finished, the CTF results are stored in **CTFEST/partres.txt**. A detailed explanation of the outputs and the format of the **partres.txt** file can be found on the SPHIRE wiki. (`http://sphire.mpg.de/wiki/doku.php?id=pipeline:cter:sxcter`) and in the ***CTER*** paper (Penczek et al. 2014).

Due to internal randomization of the statistical resampling procedure used to compute the CTF parameters and their standard deviation, results of different runs will differ slightly.

TIP: *Open the output file (partres.txt) using a text editor and confirm that (a) the image defocus (column 1) is within the range used during data collection and that (b) its standard deviation (column 9) is low.*

TIP: *For CTF estimation of phase-plate data, activate the* ***Voltage Phase Plate Dataset*** *option found in the advanced settings and adjust the Defocus and phase shift search range and step according to your data collection settings.*

CTF Assessment

The CTF results are analyzed using the **CTF Assessment GUI (sp_gui_cter.py)**. This tool is used to assess the overall quality of the dataset and to identify and remove outliers that might have a negative impact on the final result. This is accomplished by evaluating errors of CTF parameters estimated from the collected micrographs. Micrographs for which the estimated CTF parameters are unreliable (i.e., have large errors) can be removed as they will not contribute any high-resolution information. The **CTF Assessment GUI** allows screening using multiple criteria simultaneously, which is particularly useful for the fast and effective analysis of large datasets.

Here are examples of possible micrograph selection criteria to employ:

⇒ Within a specific defocus range.

> **NOTE:** *If you aim to reconstruct a rather "large" particle, your dataset contains a sufficient number of images and image contrast is not an issue, then why would you keep far-from-focus images in your dataset? They will not contribute high resolution information to the project.*

⇒ With low standard deviations regarding the defocus.

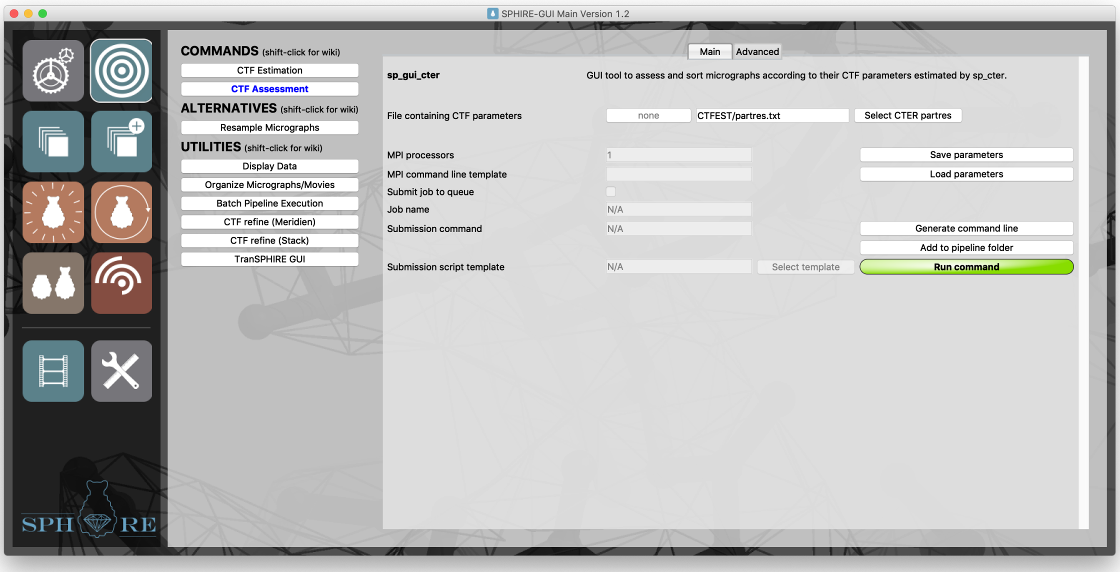
> **NOTE:** *A uniform defocus across the micrograph field is critical for near atomic resolution structure determination.*

⇒ With a certain frequency limit.

NOTE: *Micrographs with unreliable CTF parameters and large defocus gradients, will most likely have a negative impact on the quality of the final map.*

Elimination of low-quality micrographs reduces data processing time and improves the quality and resolution of the final structure.

In this tutorial, we will only perform a simplified screening by setting thresholds for the defocus value and the astigmatism frequency limit (1st and 16th column in **partres.txt,** respectively). Open the main window of SPHIRE GUI and click the button ***CTER*** on the left and then the button ***CTF Assessment*** in the middle.



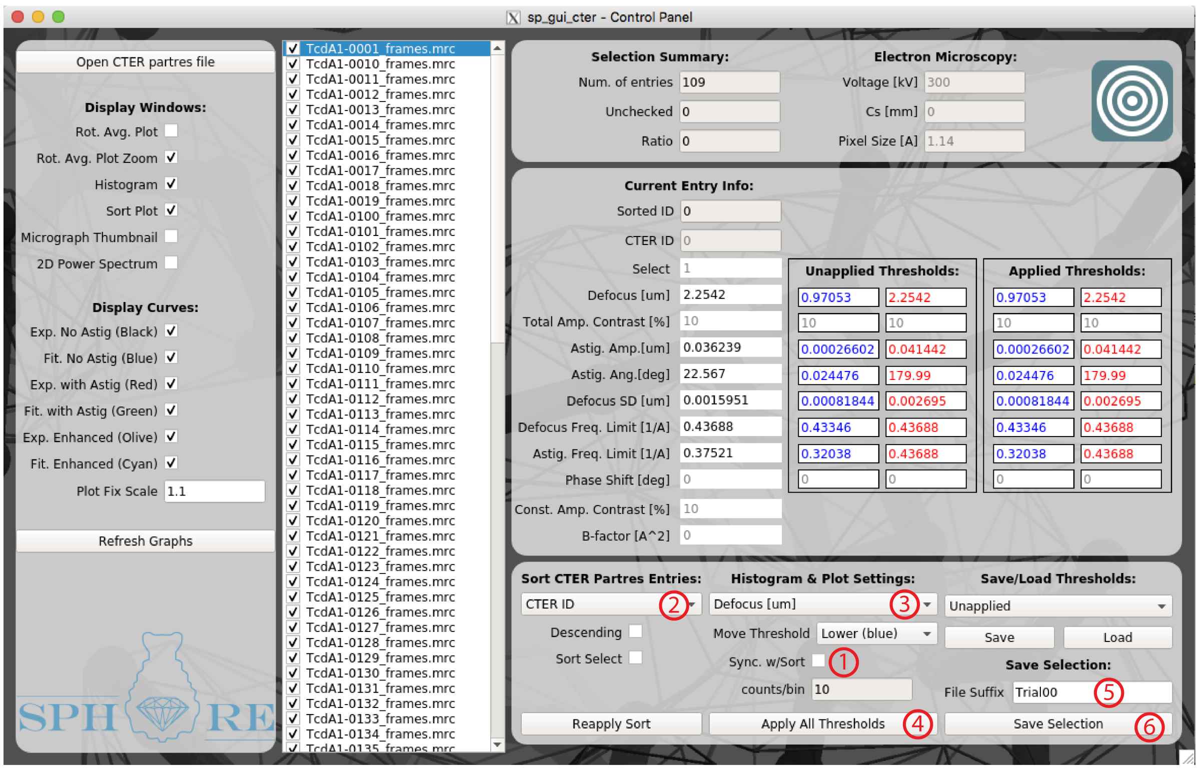
sp_gui_cter — GUI tool to assess and sort micrographs according to their CTF parameters estimated by sp_cter.

File containing CTF parameters: none, CTFEST/partres.txt
MPI processors: 1
MPI command line template:
Submit job to queue:
Job name: N/A
Submission command: N/A
Submission script template: N/A

Click the ***Select CTER partres*** button, and use the file browser to select the CTF parameter file **partres.txt** in the **CTFEST** folder and click the ***Run command*** button to launch the GUI tool. Following windows will pop-up:

- **Control Panel**
- **Histogram**
- **Sort Plot**
- **Fit Plot**

This is the **Control Panel** window:



sp_gui_cter - Control Panel

Open CTER partres file

Display Windows:

Rot. Avg. Plot	☐
Rot. Avg. Plot Zoom	✓
Histogram	✓
Sort Plot	✓
Micrograph Thumbnail	☐
2D Power Spectrum	☐

Display Curves:

Exp. No Astig (Black)	✓
Fit. No Astig (Blue)	✓
Exp. with Astig (Red)	✓
Fit. with Astig (Green)	✓
Exp. Enhanced (Olive)	✓
Fit. Enhanced (Cyan)	✓
Plot Fix Scale	1.1

Refresh Graphs

✓ TcdA1-0001_frames.mrc
✓ TcdA1-0010_frames.mrc
✓ TcdA1-0011_frames.mrc
✓ TcdA1-0012_frames.mrc
✓ TcdA1-0013_frames.mrc
✓ TcdA1-0014_frames.mrc
✓ TcdA1-0015_frames.mrc
✓ TcdA1-0016_frames.mrc
✓ TcdA1-0017_frames.mrc
✓ TcdA1-0018_frames.mrc
✓ TcdA1-0019_frames.mrc
✓ TcdA1-0100_frames.mrc
✓ TcdA1-0101_frames.mrc
✓ TcdA1-0102_frames.mrc
✓ TcdA1-0103_frames.mrc
✓ TcdA1-0104_frames.mrc
✓ TcdA1-0105_frames.mrc
✓ TcdA1-0106_frames.mrc
✓ TcdA1-0107_frames.mrc
✓ TcdA1-0108_frames.mrc
✓ TcdA1-0109_frames.mrc
✓ TcdA1-0110_frames.mrc
✓ TcdA1-0111_frames.mrc
✓ TcdA1-0112_frames.mrc
✓ TcdA1-0113_frames.mrc
✓ TcdA1-0114_frames.mrc
✓ TcdA1-0115_frames.mrc
✓ TcdA1-0116_frames.mrc
✓ TcdA1-0117_frames.mrc
✓ TcdA1-0118_frames.mrc
✓ TcdA1-0119_frames.mrc
✓ TcdA1-0120_frames.mrc
✓ TcdA1-0121_frames.mrc
✓ TcdA1-0122_frames.mrc
✓ TcdA1-0123_frames.mrc
✓ TcdA1-0124_frames.mrc
✓ TcdA1-0125_frames.mrc
✓ TcdA1-0126_frames.mrc
✓ TcdA1-0127_frames.mrc
✓ TcdA1-0128_frames.mrc
✓ TcdA1-0129_frames.mrc
✓ TcdA1-0130_frames.mrc
✓ TcdA1-0131_frames.mrc
✓ TcdA1-0132_frames.mrc
✓ TcdA1-0133_frames.mrc
✓ TcdA1-0134_frames.mrc
✓ TcdA1-0135_frames.mrc

Selection Summary:

Num. of entries	109
Unchecked	0
Ratio	0

Electron Microscopy:

Voltage [kV]	300
Cs [mm]	0
Pixel Size [A]	1.14

Current Entry Info:

		Unapplied Thresholds:		Applied Thresholds:	
Sorted ID	0				
CTER ID	0				
Select	1				
Defocus [um]	2.2542	0.97053	2.2542	0.97053	2.2542
Total Amp. Contrast [%]	10	10	10	10	10
Astig. Amp.[um]	0.036239	0.00026602	0.041442	0.00026602	0.041442
Astig. Ang.[deg]	22.567	0.024476	179.99	0.024476	179.99
Defocus SD [um]	0.0015951	0.00081844	0.002695	0.00081844	0.002695
Defocus Freq. Limit [1/A]	0.43688	0.43346	0.43688	0.43346	0.43688
Astig. Freq. Limit [1/A]	0.37521	0.32038	0.43688	0.32038	0.43688
Phase Shift [deg]	0	0	0	0	0
Const. Amp. Contrast [%]	10				
B-factor [A^2]	0				

Sort CTER Partres Entries: CTER ID (2); Descending ☐; Sort Select ☐

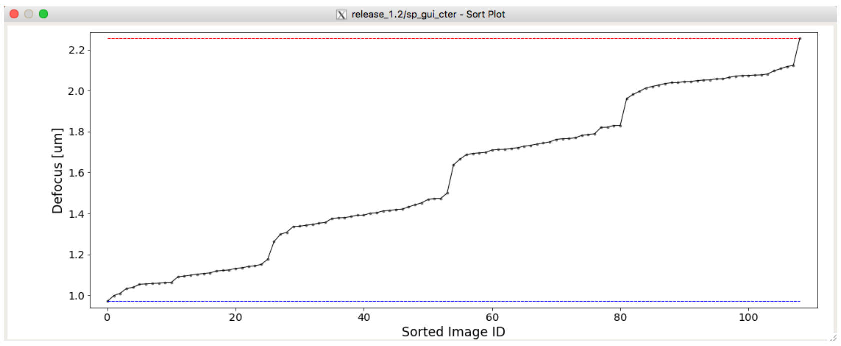
Histogram & Plot Settings: Defocus [um] (3); Move Threshold: Lower (blue); Sync. w/Sort ☐ (1); counts/bin: 10

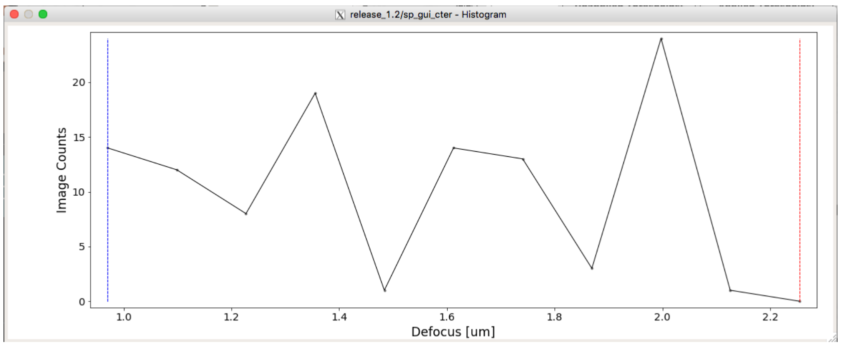
Save/Load Thresholds: Unapplied; Save; Load

Save Selection: File Suffix: Trial00 (5)

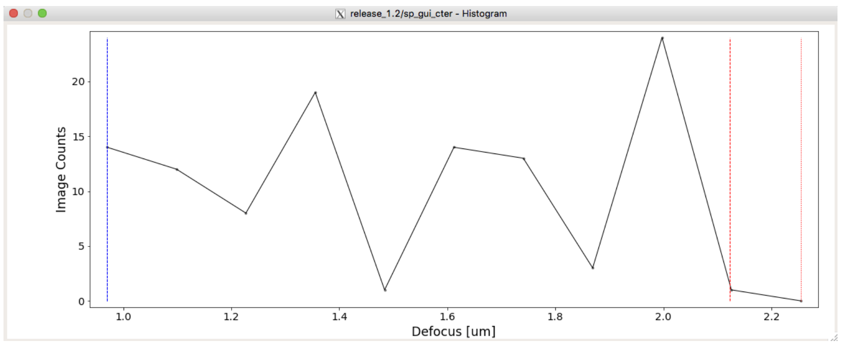
Reapply Sort — Apply All Thresholds (4) — Save Selection (6)

First, check the **Sync. Sort** option (1). This will synchronize the selection of parameters in **Sort CTER Partres Entries** (2) and **Histogram & Plot Settings** (3) and their respective windows. Now we will analyze the defocus distribution of this tutorial dataset. Select **Defocus [um]** from the combo box (3) and check the **Sort Plot** and **Histogram** windows.





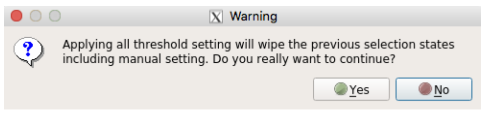
From both plots, we can see that the defocus of this dataset ranges from 0.97 µm to 2.25 µm. To demonstrate the functionality of this tool, we will now discard all micrographs with a defocus value > 2.12 µm by setting the respective threshold. Left click the Histogram window while clicking the SHIFT button to set the threshold at 2.13 µm and a red dashed line will appear at the respective position. Alternatively, you can set this threshold directly at the respective input field in the **Control Panel** window.



TIP: *You can also set a low cutoff-threshold in order to discard micrographs with a defocus lower than a specific value. In this case **left click** the **Histogram** window at the respective value and a blue dashed line will appear at the respective position.*

NOTE: *The **Sort plot** window supports the same mouse controls. Usually, we use this approach to identify defocus "outliers" and/or select micrographs within a specific search range.*

In order to apply the user-defined defocus threshold, click the ***Apply All Thresholds*** (4) button. A message dialog will pop up. Click **Yes** to apply the threshold.

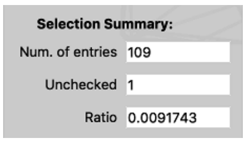

Warning

Applying all threshold setting will wipe the previous selection states including manual setting. Do you really want to continue?

Yes No

NOTE: *The respective outliers have now been unchecked in the micrograph list.*

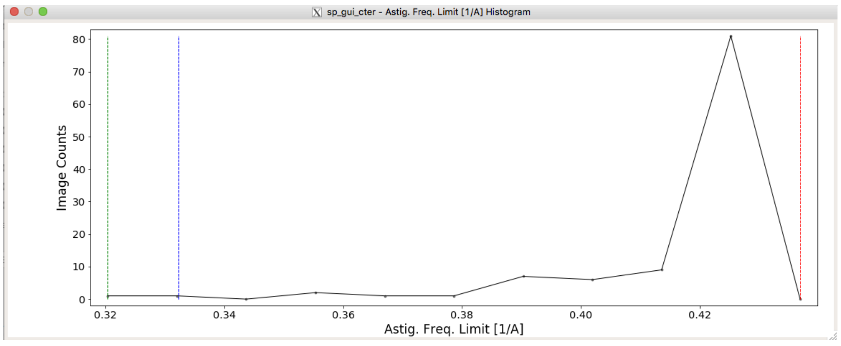
You can see the number and ratio of the unchecked images in the **Selection Summary** (7).


Selection Summary:

Num. of entries	109
Unchecked	1
Ratio	0.0091743

TIP: *You can also visually examine unchecked micrographs before discarding them. To do so, activate the* ***Sort Select*** *checkbox under* ***Sort CTER Partres Entries****. All unchecked micrographs will now be placed at the top of the micrograph list.* ***Left click*** *an unchecked entry in the list to display the graphs and parameter values and click on the* ***Micrograph Thumbnail Checkbox (Display Windows)*** *to display the selected micrograph. In the* ***Histogram*** *and* ***Sort Plot*** *windows, the green line indicates the value of the selected entry. If you want to keep the selected micrograph,* ***check*** *the checkbox next to the file name in the micrograph list.*

Now, we will use a second selection criterion to discard micrographs: **Astigm. Freq. Limit [1/A]**. This parameter provides a good criterion to identify micrographs that have little high-resolution information content. In the control panel, instead of **Defocus [um]**, select the **Astigm. Freq. Limit [1/A]** parameter from the combo box (3) and check again the **Histogram** and **Sort Plot** windows. Using the procedure described above, set a low cut-off threshold at about 0.33 1/Å (=3 Å) and click again the **Apply All Thresholds** button (4) (the higher the limit of the astigmatism, the better the quality of the micrograph; thus, we want now to discard micrographs that have a limit below this threshold). After applying this threshold, the low value cut-off (blue dashed line) should be at the following position in the Histogram plot:



NOTE: *The value of 0.33 1/Å corresponds to a resolution limit of $^1/_{0.33}$ Å = 3 Å. Thus, in this case we will discard all micrographs that have an astigmatism resolution limit > 3 Å.*

TIP: *You can also use other parameters to screen micrographs. However, from our experience* ***Astigm. Freq. Limit [1/A]*** *is the most convenient criterion to identify bad micrographs. Thus, shifting the mean value of the* ***Astigm. Freq. Limit [1/A]*** *of your dataset to a higher value is likely to improve the final outcome. Moreover, if you would prefer to discard micrographs with high amount of astigmatism, we recommend using the* ***Astig. Amp. [um]*** *criterion. However, as long as Thon rings in images extend sufficiently far and the astigmatism estimations are precise, astigmatism is not per se detrimental to high-resolution work. Finally, you should always remove micrographs whose CTF parameters have unusually high errors, meaning high standard deviations (SD of parameters).*

At this point the number of unchecked micrographs increased from 1 to 2. Enter “Tutorial” in **File Suffix** (5) under **Save Selection:** in the **Control Panel** and click the ***Save Selection*** button (6). A message dialog will pop up and inform you that following files are saved:

- **Tutorial_micrographs_select.txt:** List of selected Micrographs
- **Tutorial_ micrographs_discard.txt:** List of discarded micrographs.
- **Tutorial_partres_select.txt:** CTF parameters of the selected micrographs.
- **Tutorial_partres_discard.txt:** CTF parameters of discarded micrographs.
- **Tutorial_thresholds.txt:** Applied thresholds.

These files are stored in the location of the input CTF parameter file **partres.txt** in the folder **CTFEST**.

TIP: *After this first selection step with the* ***CTF Assessment GUI****, it is possible to perform a second round of screening if necessary. For this purpose, click the* ***Open CTER partres file*** *button (8) and load the CTF parameter file of the selected micrographs (****Tutorial_partres_select.txt****).*

Window

Particle Coordinates

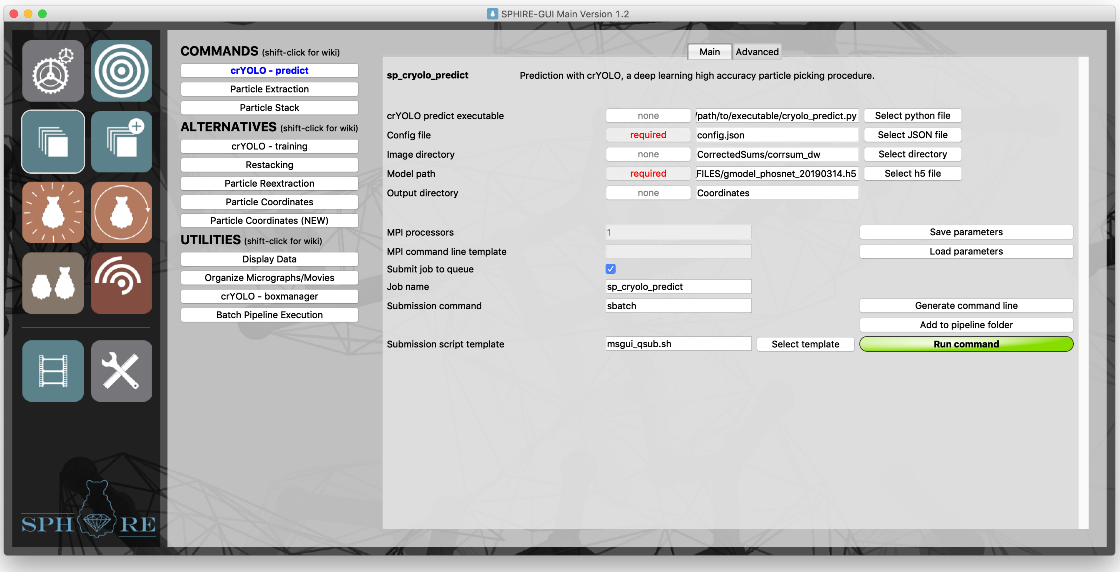
We will pick particles automatically using the **crYOLO** tool and store their coordinates. **crYOLO** is a “smart” particle picker based on convolutional neural networks, which utilizes the YOLO deep learning object detection approach. You can train **crYOLO** yourself, or you can use a general **crYOLO** model. In order to train **crYOLO** you will need to pick several micrographs manually to create a model which is specific for your data. Alternatively, you can use the general SPHIRE network model instead, which was trained on more than 50 datasets and has already learned a sufficient set of generic features and allows **crYOLO** to pick the most datasets in a fully automated manner with high precision, with no need for any additional manual training. Here we will perform reference-free particle picking with **crYOLO** in a fully-automated manner using the general model.

TIP: *If your particle displays a “non-standard” shape not present in the general model and/or you wish to perform a “digital purification” during picking (e.g. train **crYOLO** to skip a certain particle population), it will be necessary to pick some micrographs manually and create your own training model.*

An extensive tutorial to create your own training data and to get familiar with **all features** of **crYOLO** can be found here: `http://sphire.mpg.de/wiki/doku.php?id=pipeline:window:cryolo`

IMPORTANT: For negative stain datasets, the standard general model will not work. Please download, and use the negative stain general model, instead. `https://sphire.mpg.de/wiki/doku.php?id=downloads:cryolo_1#for_negative_stain_images`

Now, in the main window of the SPHIRE GUI first click the button ***WINDOW*** on the left and then the ***crYOLO - predict*** button in the middle.



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

crYOLO predict executable	none	/path/to/executable/cryolo_predict.py
Config file	required	config.json
Image directory	none	CorrectedSums/corrsum_dw
Model path	required	FILES/gmodel_phosnet_20190314.h5
Output directory	none	Coordinates
MPI processors	1	
MPI command line template		
Submit job to queue	✓	
Job name	sp_cryolo_predict	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following fields:

- **crYOLO predict executable:** /path/to/executable/cryolo_predict.py

  NOTE: *Type here the path to the* ***crYOLO_predict.py*** *executable file or click the* ***Select python file*** *button to use your file browser to select it.*

- **Config file:** config.json

  NOTE: *Click the* ***Select JSON file*** *button, and use the file browser to load the* ***crYOLO*** *configuration file*

  TIP: *When you process your own data, it is necessary to edit the config.json file and set the value in the "anchors" field to your desired box size (pixels). This size has to completely enclose the longest axis of your particle.*

- **Image Directory:** CorrectedSums/corrsum_dw

  NOTE: *Click the* ***Select directory*** *button, and use the file browser to choose the directory containing the* ***dose-weighted*** *motion-corrected averages.*

- **Model path**: CRYOLO_FILES/gmodel_phosnet_201900909.h5

  NOTE: *Click the* ***Select h5 file*** *button. and use the file browser to choose the general training model.*

  TIP: *Please use the latest version of the general model, since we regularly add additional handpicked datasets.* `https://sphire.mpg.de/wiki/doku.php?id=downloads:cryolo_1#for_cryo_images_low-pass_filtered`

- **Output Directory:** Coordinates

and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. Monitor the progress of the job through the standard output. At the terminal, type:

```
tail -f name_of_sp_cryolo_predict_logfile
```

TIP: ***CrYOLO*** *filters the digital micrographs prior particle selection to a resolution of 30 Å.*

```
Filtering CorrectedSums/corrsum_dw/TcdA1-0191_frames.mrc

134 particles are found in tmp_filtered/CorrectedSums/corrsum_dw/TcdA1-0126_frames.mrc ( 0 %
)
```

On our cluster, this process finished on a single processor after about 9 minutes. However, if you do not have enough time to wait for the results, simply copy our pre-calculated results to your project.

```
cp -r SphireDemoResults/Coordinates ./
```

The output folder **Coordinates** within the project directory will include following subfolders:

- **EMAN:** this folder includes a particle coordinate file for every micrograph in the EMAN1 .box file format.
- **STAR:** here you can find the coordinate files in the STAR file format.
- **CBOX:** The format here is similar to EMAN1's .box file format, but these files contain an additional column for the confidence score for each particle during picking. You can use these scores to fine-tune your particle picking (see description below).

Now we will use **crYOLO**'s boxmanager to display the picking results. In the main window of the SPHIRE GUI, the button ***Window*** on the left and then the button ***crYOLO - boxmanager*** in the middle (under the section Utilities). Fill out the following fields:

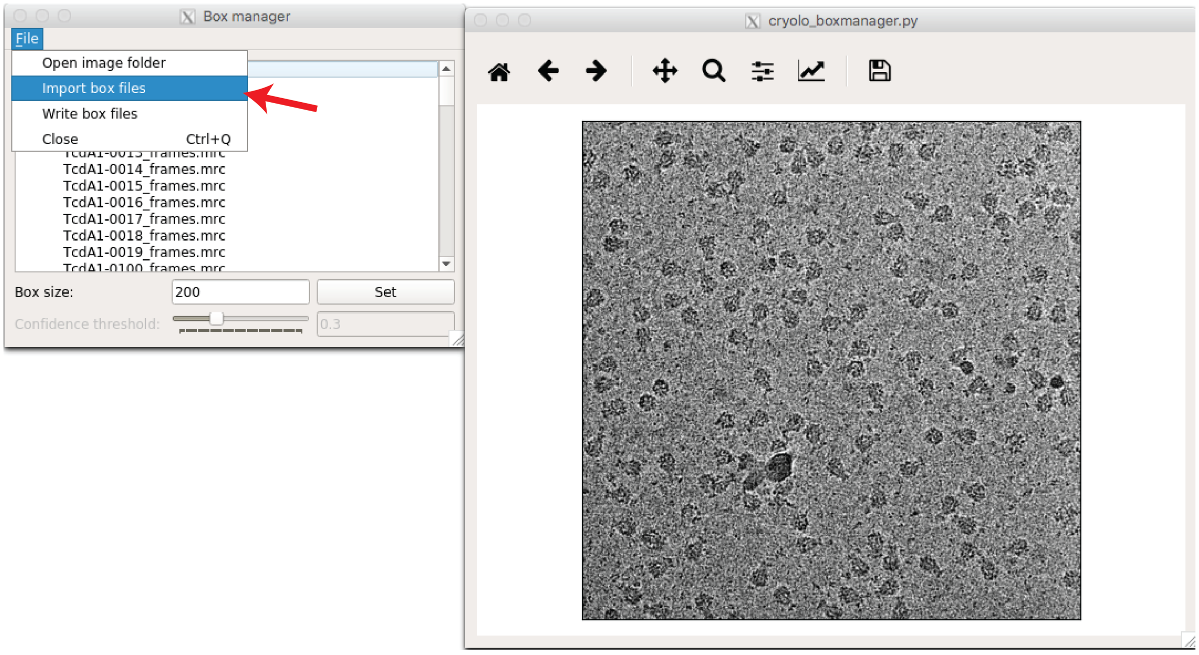
- **crYOLO boxmanager executable:** /path/to/executable/cryolo_boxmanager.py

  NOTE: *Type here the path to the* ***crYOLO_boxmanager.py*** *executable file or click the* ***Select python file*** *button to use the file browser to select it.*

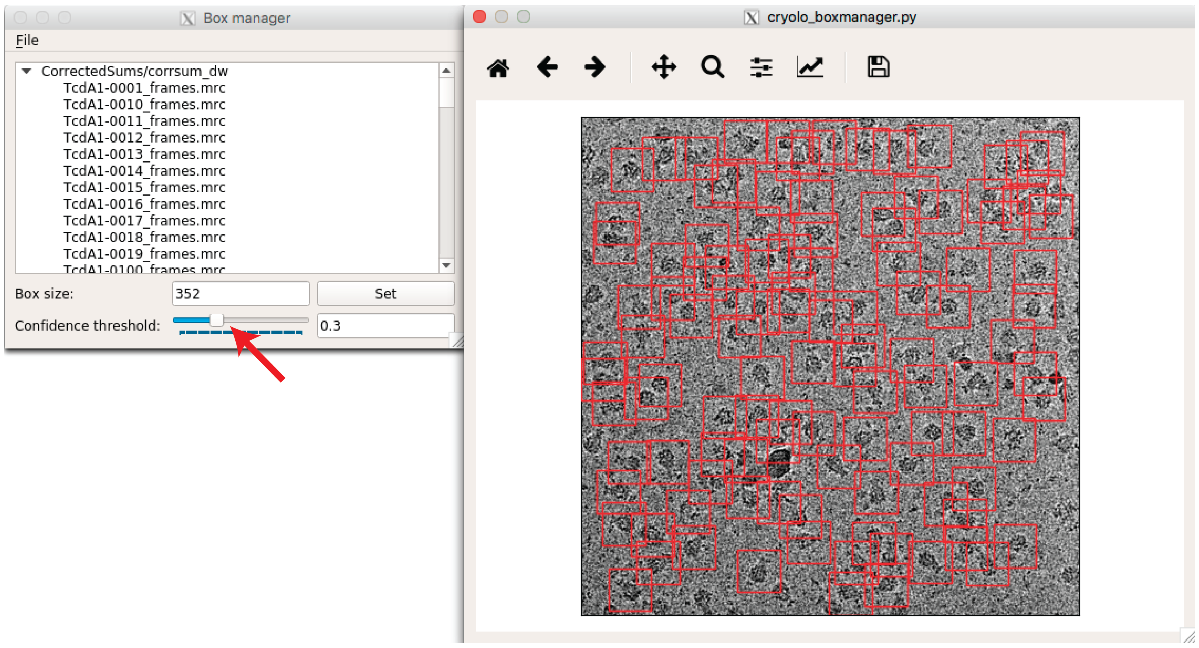
- **Input image directory:** CorrectedSums/corrsum_dw/

  NOTE: *Click the* ***Select directory*** *button, and use the file browser to select the directory containing the* ***dose-weighted*** *motion-corrected micrographs*

Click the ***Run command*** button to launch the **crYOLO box manager** GUI tool. Click on the **File** tab and then select the **Import box files** option. Use the file browser to select the **CBOX** folder within the **Coordinates** directory to load the coordinates for each micrograph.



Here it is possible to adjust the confidence threshold using the slider button and the live preview in order to filter your particle boxes and to exclude (or include) additional particles.



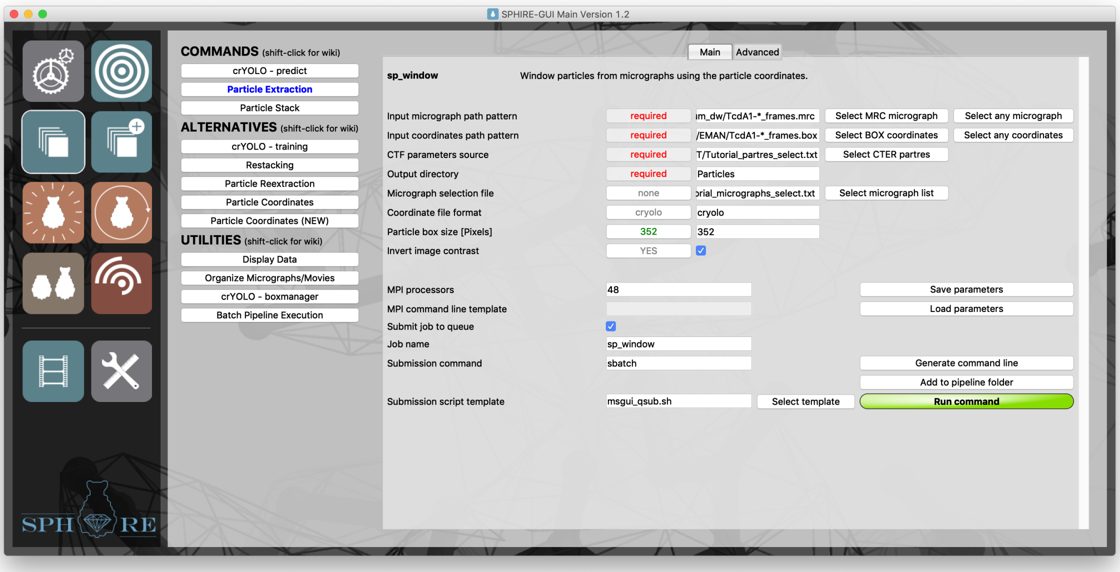
TIP: *An animated GIF showing how to filter particle boxes using the* ***crYOLO*** *box manager can be found on the* ***SPHIRE*** *wiki:* `http://sphire.mpg.de/wiki/lib/exe/detail.php?id=pipeline%3Awindow%3Acryolo\&media=pipeline:window:ezgif-1-3b966b0324d1.gif`

For this dataset, particle picking with the default parameters worked well, and therefore it is not necessary to fine-tune the confidence threshold or repeat the picking procedure using a hand-picked training model.

TIP: *If you picked your micrographs with a different particle picking utility, copy the coordinate files to the* ***Coordinates*** *folder and continue from there. It is not necessary to use the tutorial naming convention. We prefer using the* ***EMAN1*** *coordinate file format (.box), but SPHIRE also supports coordinates from* ***EMAN2*** *and* ***SPIDER*** *with the .json and .spi extension, respectively. If* ***RELION*** *coordinate files are available (.star), you can easily convert them to the* ***EMAN1*** *file format (.box) using the* ***sp_relion2sphire*** *utility. For this purpose, in the main window of the |||SPHIRE GUI||| click the pictogram* ***UTILITIES*** *on the left and then the* ***RELION to SPHIRE conversion*** *button in the middle. Detailed instructions are provided in the* ***Import a star file page*** *of the wiki.* `http://sphire.mpg.de/wiki/doku.php?id=howto:relion2sphire`

Particle Extraction

In the main window of the SPHIRE GUI, click the button ***WINDOW*** on the left and then the **Particle Extraction** button in the middle.



sp_window Window particles from micrographs using the particle coordinates.

Input micrograph path pattern	required	ım_dw/TcdA1-*_frames.mrc
Input coordinates path pattern	required	/EMAN/TcdA1-*_frames.box
CTF parameters source	required	T/Tutorial_partres_select.txt
Output directory	required	Particles
Micrograph selection file	none	ɔrial_micrographs_select.txt
Coordinate file format	cryolo	cryolo
Particle box size [Pixels]	352	352
Invert image contrast	YES	✓
MPI processors		48
MPI command line template		
Submit job to queue		✓
Job name		sp_window
Submission command		sbatch
Submission script template		msgui_qsub.sh

Fill out the following input fields at the GUI interface:

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

*NOTE: Click the **Select MRC micrograph** button, and use the file browser to select one of the micrographs in the **CorrectedSums/corrsum_dw** folder. Replace the variable part of the file name with the wildcard character "*" (e.g. from **TcdA1-0010_frames_sum.mrc** to **TcdA1-*_frames_sum.mrc**).*

*TIP: If the micrographs are from negative stain and the particle coordinates are available, one can start directly with particle extraction and skip all previous steps. In this case, insert **the/path/to/your/micrographs/*.mrc** here instead.*

- **Input coordinates path pattern:** Coordinates/EMAN/TcdA1-*_frames.box

  *NOTE: Click the **Select BOX coordinates** button, and use the file browser to select one of the coordinate files in the **Coordinates/EMAN** folder. Replace the variable part of the file name with the wildcard character "*" (i.e., from **TcdA1-0010_frames_sum.box** to **TcdA1-*_frames_sum.box**). In this demo, the folder **Coordinates/EMAN** contains 101 .box files.*

  *TIP: It is not necessary to use the same root names for micrographs and coordinates files. However, the variable part defined by the wildcard character **needs to be** identical for the association of the **input micrograph** and **coordinate file**.*

- **CTF parameters source:** CTFEST/Tutorial_partres_select.txt

  *NOTE: Click the **Select CTER partres** button, and use the file browser to select the **tutorial_partres_select.txt** file in the **CTFEST** folder. This file contains the CTF parameters of the **selected** micrographs. The CTF information will be transferred to the header of all extracted particles from their respective micrograph.*

  TIP: If you process negative stain data, do not provide the CTF parameter source file here, but instead type the pixel size of your micrographs. In this case, the program will include an "ideal" CTF (no modulation of the signal; 90° phase shift; defocus 0 µm) to the header of the extracted particles.

- **Output directory:** Particles
- **Micrograph selection file:** CTFEST/Tutorial_micrographs_select.txt

  *NOTE: Click the **select micrograph list** button, and use the file browser to select the **tutorial_micrographs_select.txt** file (list of selected micrographs) in the **CTFEST** folder.*

  TIP: If you process negative stain data and/or skipped the micrograph selection steps, leave this field empty.

- **Coordinate file format:** cryolo

  *NOTE: Additional supported file formats are **EMAN1** (.box), **EMAN2** (.json), and **SPIDER** (.spi).*

- **Particle box size [Pixels]:** 352
- **Invert image contrast:** YES

  NOTE: *For typical cryo-EM data (particles appear dark on a bright background), leave this flag as it is. The program will invert the contrast of your cryo-EM images automatically.*

  NOTE: *If your particles appear bright on a dark background (e.g. negative stain or inverted cryo-EM images), deactivate this flag.*

Specify the number of processors (we used 48) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. Monitor the progress of the job through the standard output.

```
TcdA1-0010_frames.mrc ---> 0.00%

TcdA1-0011_frames.mrc ---> 50.00%

Summary of micrograph level processing... Valid : 107

Processed : 107

Rejected by no coordinates entries : 0

Global summary of coordinates level processing...

Detected : 12658

Processed : 12658

Rejected by out of boundary : 0

Please execute from the command line : e2bdb.py Particles/mpi_proc_* --makevstack=bdb:Particles#data

DONE
```

However, if there is no time to wait for the results, copy our precalculated results to your **project directory**.

```
cp -Rp SphireDemoResults/Particles ./
```

Once the job has finished, the extracted particles are stored in the folder Particles. For each processed micrograph, extracted particle images are stored in an independent BDB data file. On our cluster, this step completed in about 1 minute.

Particle Stack

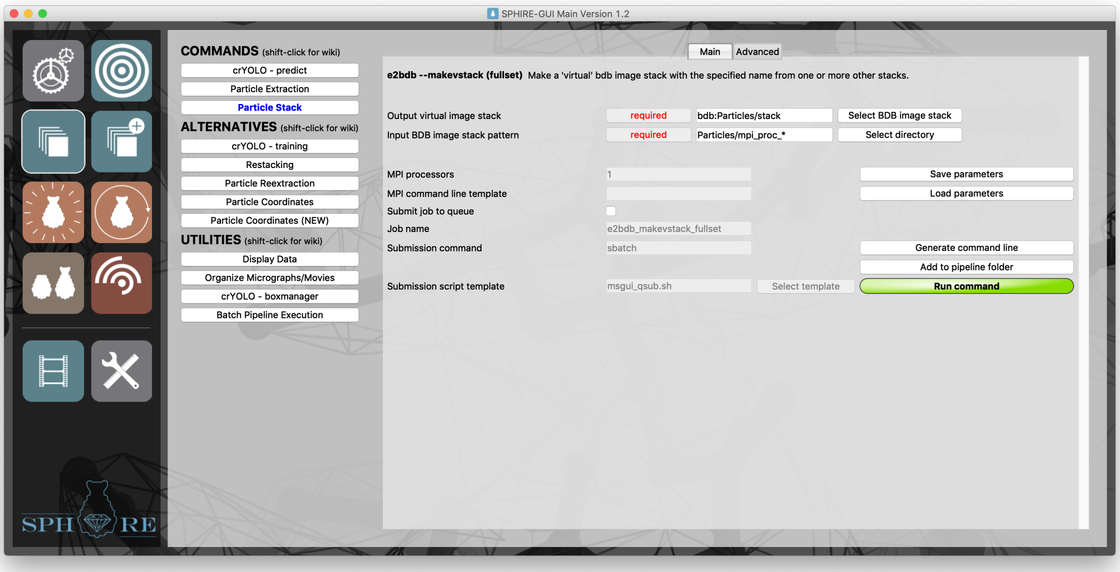
The next step is to combine the individual per-micrograph stacks into a single "virtual" particle stack.

IMPORTANT: Skip this step if you used only a single processor to extract particles.

We call this stack "virtual" since, taking advantage of **EMAN2**'s BDB format, it contains **only** metadata (header information associated with images, such as pixel size, particle micrograph origin, any alignment parameters) and links to the original images. In this way we minimize disk space usage while affording great flexibility with rapid creation of particle data stacks as necessary in subsequent data processing steps.

TIP: *BDB files reside in separate directories, always named* ***EMAN2DB****.*
An ***EMAN2DB*** *directory that is not linked to any other BDB directory (through the virtual stack mechanism) can be easily archived, copied and transferred. However, directories linked through the virtual stack mechanism have to be copied jointly while preserving relative directory structure. It is very important not to modify* ***EMAN2DB*** *directories content in any way as this usually results in irreversible loss of information.*

In the main window of the SPHIRE GUI click the button ***WINDOW*** on the left and then the ***Particle Stack*** button in the middle.



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:Particles/stack
Input BDB image stack pattern	required	Particles/mpi_proc_*
MPI processors		1
MPI command line template		
Submit job to queue		☐
Job name		e2bdb_makevstack_fullset
Submission command		sbatch
Submission script template		msgui_qsub.sh

Fill out the following input fields:

- **Output virtual image stack:** bdb:Particles/stack

  NOTE: *This defines the* ***Particles*** *folder as destination for the virtual stack.*

- **Input BDB image stack pattern:** Particles/mpi_proc_*

  NOTE: *Click the* ***Select directory*** *button, and use the file browser to select folder* ***Particles*** *and then one of the* ***mpi_proc*** *folders, which includes particle stacks for micrograph files extracted from a specific processor. Replace the variable part of the name of the folder by the wildcard character "*" (i.e.,* ***mpi_proc_000*** *to* ***mpi_proc_****).

Click the ***Run command*** button to create the stack and monitor the progress of this job through the standard output. This step is usually finished in a few seconds.

After the job is finished, you can verify the number of particles in the resulting stack using **EMAN2**'s **e2iminfo.py** command.

At the terminal, type:

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

Depending on the settings you used during particle picking (confidence threshold), the dataset processed here should contain about 8500 to 12000 single particles. The stack in the pre-calculated results contains 12658 particles.

You can also display the virtual stack using the utility **Display Data**. For this purpose, in the main window of the SPHIRE GUI click the ***Display Data*** button in the middle, and load the stack. However, please keep in mind that the display of a very large particle stack might result in a memory crash.

ISAC

2D Clustering

The 2D classification is one of the crucial steps of the workflow. It will quickly give you a good idea about the quality and heterogeneity of the data and is a great tool to sort out bad particles. Thus, it is a key step towards a high-resolution structure determination. In other packages, this is usually done via computationally expensive 3D classifications, which usually have to be performed multiple times. In contrast, we prefer to conduct 3D analysis using datasets that were pre-cleaned in 2D and, as a result, we usually need to perform heterogeneity analysis in 3D only once.

SPHIRE uses the **ISAC** method (Yang et al. 2012) to perform 2D alignment and clustering in order to determine validated and homogeneous subsets of images with minimal human intervention. Due to the high number of reproducibility tests during the validation procedure, **ISAC** used to be (pre-2016 version) very time- and memory-consuming. The current, optimized **ISAC2** version delivers results for data sets of up to 700000 and it is 10 × to 15 × faster than the pre-2016 version.

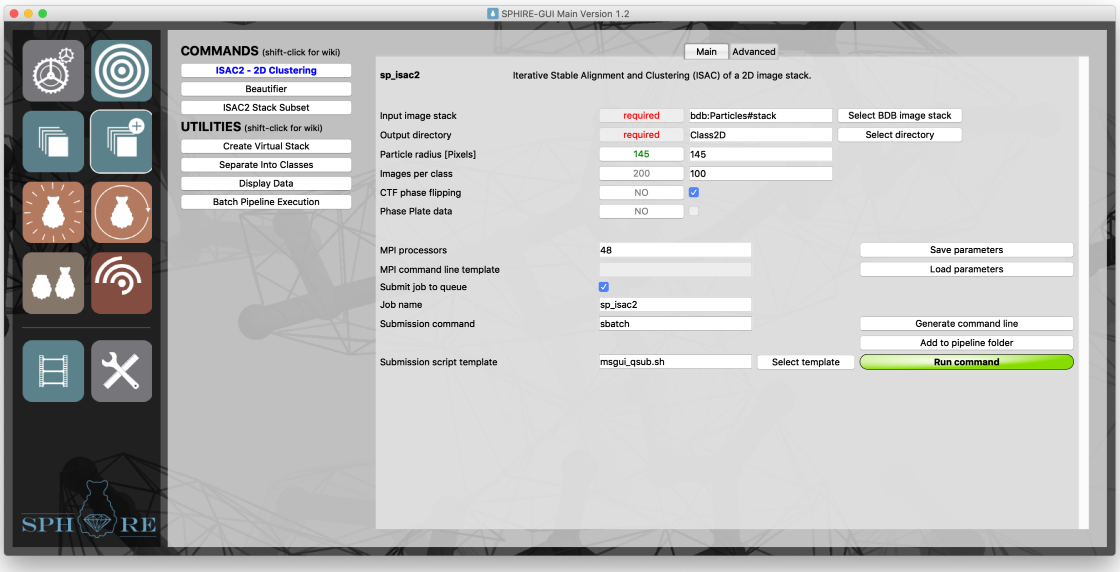
The ISAC program will:

⇒ Phase-flip the 2D images in the stack, if requested.

⇒ Establish initial translation parameters using a reference-free approach, if requested.

⇒ Perform equal size K-means clustering, run a group stability test and output stable class averages. Images that cannot be accounted for by the current classes are set aside. The determined class averages and their associated images, called “accounted for”, are returned as output. The remaining, “unaccounted for” images, are forwarded to the next round of **ISAC** clustering (called generation). The program will terminate once no further stable class averages can be produced. Finally, the particle members of stable output classes are returned as “Processed”, while any particles that could not be accounted for are returned as “NotProcessed”.

In the main window of the SPHIRE GUI click button **ISAC2** on the left and then the ***ISAC2 - 2D Clustering*** button in the middle.



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

Input image stack	required	bdb:Particles#stack
Output directory	required	Class2D
Particle radius [Pixels]	145	145
Images per class	200	100
CTF phase flipping	NO	✓
Phase Plate data	NO	
MPI processors	48	
MPI command line template		
Submit job to queue	✓	
Job name	sp_isac2	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following input fields:

- **Input image stack:** bdb:Particles#stack

  NOTE: *Click the **Select BDB image stack** button, and use the file browser to select the virtual stack created in the previous step (containing all single particles in the project).*

  TIP: *In principle, you can also directly load any type of single particle dataset that was created by a different program, as long as it was converted to **EMAN2**'s BDB or HDF file format (preferably BDB), and if the CTF information is stored in the header of particles in the expected format. In particular, **RELION** data files (stacks and metainformation stored in .star files) can be directly converted using the **sp_relion2sphire.py** utility). To access this tool, in the main window of the SPHIRE GUI first click the pictogram **UTILITIES** on the left and then the **RELION to SPHIRE conversion** button in the middle. Detailed instructions are provided in the **Import a star file** page of the wiki.* `http://sphire.mpg.de/wiki/doku.php?id=howto:relion2sphire`.

- **Output directory:** Class2D
- **Particle radius [Pixels]:** 145
- **Images per class:** 100

  NOTE: *Our dataset contains about 12628 particles, thus by setting this parameter to 100, the expected number of 2D averages will be about*

$$\textit{Number of classes} = \frac{\textit{Total Number of Particles}}{\textit{particles per class}} = \frac{12628}{100} \approx 126\,.$$

TIP: *Proper setting of the desired number of particles per class requires some experience and testing multiple **ISAC2** runs with different settings might be necessary in order to obtain optimal results. 100 to 200 particles per class is a good starting point for cryo-EM datasets of high quality containing about 50.000 to 100.000 particles. For early exploratory analysis of very large, autopicked datasets you may want to consider using a larger number (i.e., 500 to 2000) of particles per class. Moreover, depending on the appearance of the resulting class averages, you may need to adjust this value accordingly. For example, if the resulting averages are excessively noisy, it often helps to repeat **ISAC2** run using a larger number of particles per class. However, this is always at the expense of the amount of detail in the final averages: The larger the number of members per averages, the worse the amount of discernible details in the resulting averages. Note that the computational time of **ISAC2** is a steeply raising function of the number of class averages and to obtain high-quality homogeneous (i.e., with possibly few members) class averages, significant processing resources have to be invested. It should be also emphasized that in the next step of ab initio structure determination (**VIPER**) you will need at least 100 to 150 high quality class averages to produce a reliable 3D model.*

TIP: *The final number of particles per class and the number of classes are determined by the algorithm itself based on its ability to align them consistently and thus form stable class averages. The number does not depend on the number of input images but rather on their quality and heterogeneity of the set, i.e., the number of distinct views, different conformational states, etc. Setting the number too high (typically more than a few hundred) may result in heterogeneous (a.k.a. "fake") class averages and/or the rejection of rare views by the program.*

TIP: *Use 50 to 150 members per class for negative stain datasets of 5000 to 15000 particles.*

- **CTF phase flipping:** YES

  TIP: *Deactivate this flag for negative stain data.*

- **Phase Plate Data:** NO

  NOTE: *Activate this option if you process Phase Plate Data.*

  Listed below are advanced options, please do not change the default values unless you understand their meaning well.

Advanced parameters:

- **Pixel error threshold [Pixels]:** 0.7

NOTE: *__ISAC2__ evaluates quality of class averages by performing multiple reference-free ab initio alignments of images within a class. Orientation parameters of individual images determined in these independent runs are compared, and their dispersion is reported as the __pixel error__ of a given class (or image). Class averages whose __pixel errors__ exceed the pre-set threshold are discarded and their image members are set aside for processing in the subsequent generation. __The pixel error threshold__ is thus one of the most important __ISAC2__ parameters.*

TIP: *For __ISAC2__ runs with large datasets of excellent quality and low number of particles per class (e.g. 100), one should use the rather conservative default value for this threshold (0.7). With increased number of particles per class or if you expect a high degree of flexibility in your particle, you should set a higher threshold value (for example 1.7). Otherwise, __ISAC2__ might discard too many particles.*

- **Target particle radius [Pixels]:** 29

  NOTE: *Leave the default value.*

- **Target particle image size [Pixels]:** 76

  NOTE: *Leave the default value.*

  NOTE: *To speed up the process, the particles of the input dataset are resized to match the user-provided particle radius and image size. The default and thoroughly tested values are 29 pixels radius and the box size 76 pixels, respectively. Thus, the resulting class averages will have a new (usually larger) pixel size and most probably high-resolution features (such as secondary structure elements) will not be visible. The higher-resolution information is not lost though and will be restored later in the subsequent Beautifier step. The conversion formula is:*

$$\textit{Target particle radius} = \frac{\textit{Target particle image size} \times \textit{Original particle radius}}{\textit{Original particle image size}}$$

  NOTE: *Computational time will increase exponentially with increased target particle image size.*

Specify the number of processors (48 in our case) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. Monitor progress of the job through the standard output. On our cluster, this job running on 48 processors finished after about 21 minutes.

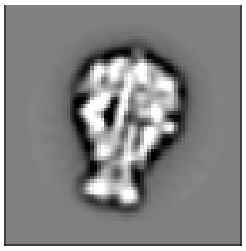
However, if you do not have enough time to wait for the results, copy our precalculated results to your **project directory**.

```
cp -Rp SphireDemoResults/Class2D ./
```

Once the job has finished, the Class2D folder will contain the following folders and files:

– **2dalignment folder:** Contains the results of the (optional) reference-free prealignment.

NOTE: *Display the total-average of your dataset after the pre-alignment (**Class2D/2dalignment/aqfinal.hdf**) using the **Display Data** utility.*



TIP: *Confirm that the 2D total-average is properly centered and that the circular 2D mask covers the entire particle. If this is not the case, increase the particle radius (project-wide parameter) and rerun **ISAC2**.*

– **class_averages.hdf:** Contains the final stable class averages.

TIP: *If the file is not produced or contains only few class averages, it means that given the quality of the data the value of **Pixel error threshold [pixels]** parameter was set too low and thus too many class averages were eliminated during the stability tests. In this case, you can try to increase the value of **Pixel error threshold [pixels]** and run **ISAC2** again. If this does not help either, consider recording a dataset with better contrast and/or improve the quality of picking. In most cases it is not possible to determine a reliable structure from a dataset that does not yield good (i.e., with distinct features) 2D class averages. In the case of a dataset containing many images of contamination (e.g. if you set a rather low confidence threshold during autopicking with **crYOLO**), a second **ISAC2** run might be necessary to obtain the best results.*

– **processed_images.txt:** Contains the particle IDs of the members of the stable class averages.

NOTE: *At the terminal, type:*

```
wc -l Class2D/processed_images.txt
```

This command will count the number of lines in the specified file. This corresponds to the total number of particles assigned to stable class averages. For the precalculated results this file has 11516 entries.

– **not_processed_images.txt:** Contains the particle IDs of the particles not assigned to stable class averages.

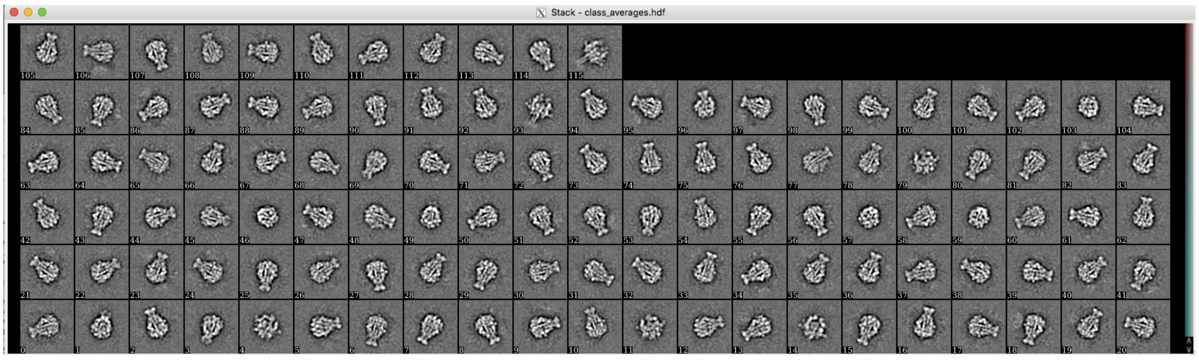
NOTE: *At the terminal, type:*

```
wc -l Class2D/not_processed_images.txt
```

*The printed number corresponds to the total number of particles not assigned to class averages. For the precalculated results **ISAC2** discarded 1142 "bad" particles.*

***NOTE:** If the **ISAC2** run completed at least one main iteration but crashed afterwards for some reason, it is possible to restart calculations using the "Continue mode". Open the **ISAC2 - 2D Clustering** page of the SPHIRE GUI and enter the exact same settings you used for the unfinished **ISAC2** run. Then first click the "advanced settings" button, set the "Restart Run" parameter at the bottom to 0 and then click the **Run command** button to submit the job. The program will identify the last fully completed main iteration and continue from there.*

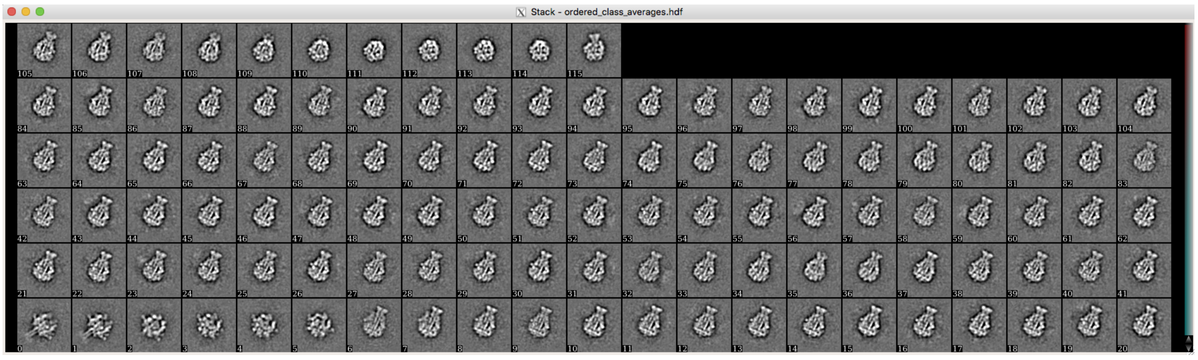
Display the final stable class averages (**Class2D/class_averages.hdf**) using the ***Display Data*** utility.



The high quality of this dataset is reflected by the impressive quality of the 2D class averages. As mentioned above, in order to speed up the process, the particles of the input dataset are resized to a large pixel size and high-resolution features are not visible. However, the stable 2D class averages will be further processed in the subsequent **Beautifier** step and high-resolution information will be restored.

To facilitate a better comparison between the stable class averages, **ISAC2** realigns them relative to each other and orders them based on pair-wise similarity. The result is stored in the file: **Class2D/ordered_class_averages.hdf**

Display the ordered and aligned final stable class averages
(**Class2D/ordered_class_averages.hdf**) using the **Display Data** utility.



Known issue in the release

We optimized the parallelization of **ISAC2** and the current **ISAC2** version can now handle large datasets. On our cluster with 128 GB of RAM and 24 cores per node, we managed to successfully process datasets with up to 750000 particles. However, larger datasets or large datasets of lower quality might still fail due to insufficient memory. The suggested workaround is to split the data into subsets: Run **ISAC2** for each subset separately, as described above, and then combine the results. For example, to split a dataset in 4 subsets type at the terminal (it is one long line command):

```
 n=4; for i in $(seq 0 $((n-1))); do e2bdb.py bdb:Particles#stack
--makevstack=bdb:Particles#stack_${i} --step=${i},${n}; done
```

The stack bdb:Particles#stack_0 contains every fourth particle of the original stack starting from index 0, the stack bdb:Particles#stack_1 contains now every fourth particle of the original stack starting from index 1 and so on.
This command creates 4 virtual particle stacks. You can easily change the number of substacks by changing the 4 in the beginning of the command to the desired value.
To combine the clean stacks obtained from 4 runs of **ISAC2** into a single virtual stack, use the following before proceeding to the next step VIPER (it is one long line command):

```
 e2bdb.py bdb:Particles#isac_substack_1 bdb:Particles#isac_substack_2
 bdb:Particles#isac_substack_3 bdb:Particles#isac_substack_4
--makevstack=bdb:Particles#stack_all
```

If you do not have enough time to perform the described steps, copy our precalculated **ISAC2** results and the resulting clean dataset to the **project directory**.

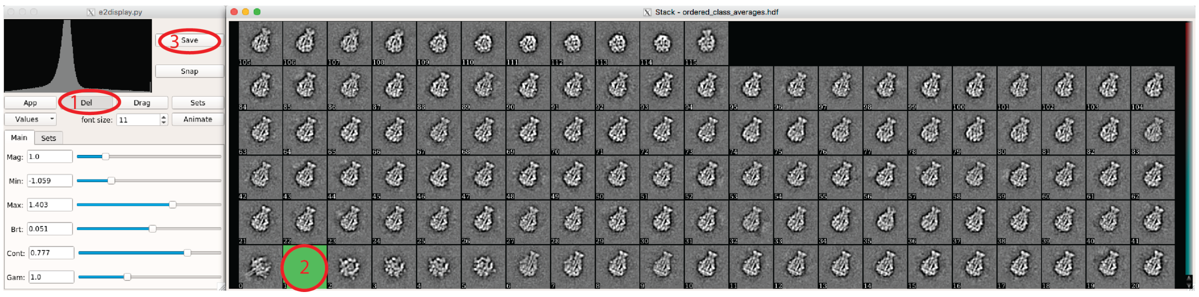
```
cp -Rp SphireDemoResults/Class2D ./
cp -Rp SphireDemoResults/Particles ./
```

Most class averages show the typical pineapple-like shape of the side-view of the prepore state of the TcdA1 toxin (Gatsogiannis et al. 2013). However, several class averages (Nr. 0-5 in the precalculated results) differ considerably; they show an umbrella-like shape and correspond to the pore state of the complex (Gatsogiannis et al. 2013; Gatsogiannis et al. 2016). Thus, the present dataset includes a mixture of two populations corresponding to two different conformational states of the same complex in a ratio of approximately 1:20.

We will now delete these class averages using the ***Display Data*** Utility. For this purpose, in the main window of the SPHIRE GUI, click the **Display Data** button under **UTILITIES** and load the file **Class2D/ordered_class_averages.hdf**.

TIP: *Usually, at this step we only delete "junk" class averages (e.g. blurry class averages, ice contamination, carbon edges etc.) and then let **RVIPER** identify and exclude outliers during the initial model generation procedure. However, in this the case the structural differences are very obvious, and therefore we will delete these classes right away.*

To delete "bad" class averages or outliers, press the mouse wheel button somewhere on the graphics window and activate the ***Del*** button (1), in the pop-up window.



font size: 11
Mag: 1.0
Min: -1.059
Max: 1.403
Brt: 0.051
Cont: 0.777
Gam: 1.0

Next, select the “bad” class averages by clicking on the respective images (2) and click the ***Save*** button (3) to save the remaining images under a new name: **Class2D/best.hdf**.

> **TIP:** *In the pop-up graphics window, you can also click the **Values** button and select the **n_objects** parameter in order to display the number of images for each class average.*

These class averages (**Class2D/best.hdf**) will be used to generate an initial 3D model with **RVIPER**.

Beautifier

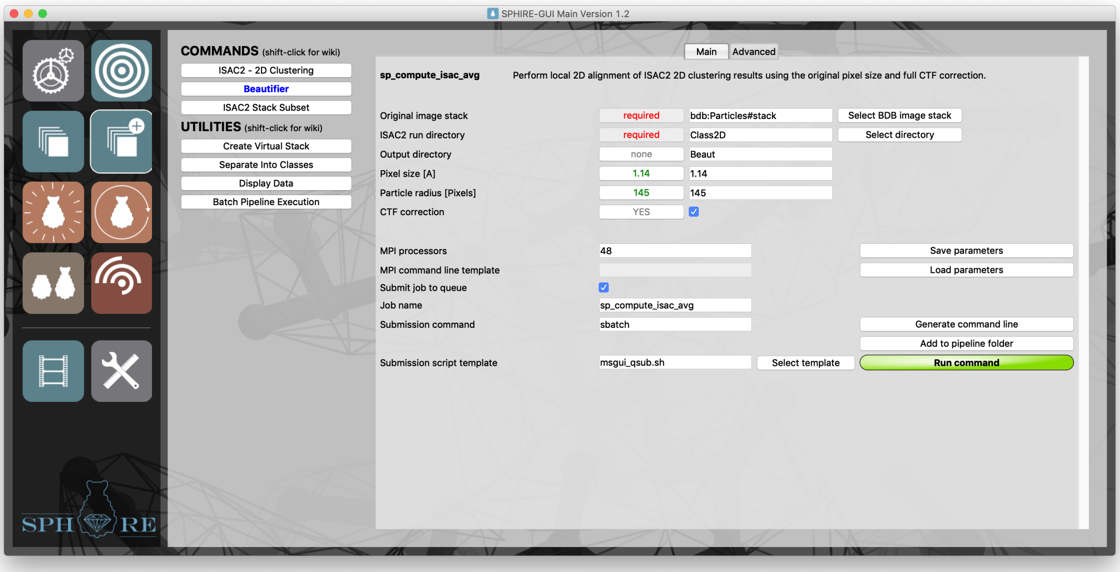
In order to speed up calculations, we resized the particles of the input dataset during **ISAC2** to a smaller box and a larger pixel size. Most importantly, we did not apply full CTF correction; instead, images were only phase-flipped. Therefore, in most cases, because these operations remove high-resolution information from the data, possible fine details (such as secondary structure elements) will not be visible in the **ISAC2** 2D class averages.

The **Beautifier** tool extracts the members of each **ISAC2** class average from the original, full-size dataset, performs local 2D refinement and applies full CTF correction. Afterwards, the full-size class averages are aligned and ordered by pair-wise similarity. It is also possible to request a low-pass filtration to suppress high-resolution noise. Thus, **Beautifier** tool provides full-size aligned averages computed at fully attained resolution.

> **TIP:** *High-resolution datasets will in most cases yield “beautified” class averages with distinct high-resolution features. Thus, the results of the current processing step provide a good checkpoint to verify that the project has the potential for high-resolution 3D structure determination.*

> **TIP:** *It is not necessary to use the Beautifier tool for negative stain data.*

In the main window of the SPHIRE GUI, click the ***ISAC2*** button on the left and then the ***Beautifier*** button in the middle.



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:Particles#stack
ISAC2 run directory	required	Class2D
Output directory	none	Beaut
Pixel size [A]	1.14	1.14
Particle radius [Pixels]	145	145
CTF correction	YES	✓
MPI processors	48	
MPI command line template		
Submit job to queue	✓	
Job name	sp_compute_isac_avg	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following input fields of the GUI:

- **Original image stack:** bdb:Particles#stack

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select the virtual stack containing all particles in the* ***Particles*** *folder. You* ***have to*** *select the stack used as input to perform the 2D Classification with* ***ISAC2****.*

- **ISAC2 run directory:** Class2D

  NOTE: *Click the* ***Select directory*** *button, and use the file browser to select the* ***ISAC2*** *output directory.*

- **Output directory:** Beaut
- **Particle radius [Pixels]:** 145

  NOTE: *You must use the original particle radius (145) and* ***not*** *the target particle radius used during* ***ISAC2*** *(29).*

- **CTF correction:** YES

  NOTE: *Activate the checkbox to apply full CTF correction. Do not activate if you are processing negative stain data.*

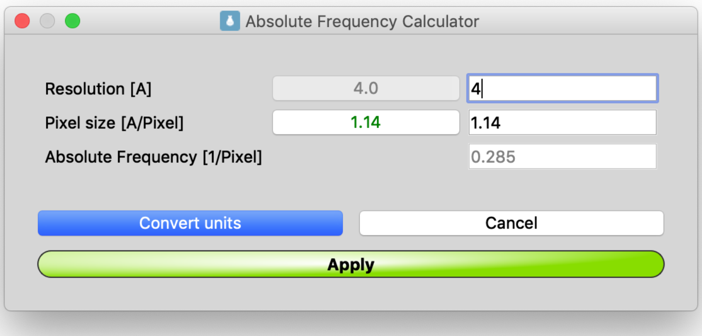
Advanced parameters:

- **Local alignment:** YES

  NOTE: *Activate the checkbox to perform a local alignment of the* ***ISAC2*** *averages. The "beautified" class averages will improve, but also the computational time will increase as well.*

- **Low-pass filter cutoff resolution [1/Pixel]:** 0.285

  NOTE: *Here you can specify, a cutoff for the low-pass filter to be applied (in absolute frequency units [1/Pixel]). You can also use our calculator to convert resolution in Å to absolute frequency.*



Absolute Frequency Calculator

Resolution [A]	4.0	4
Pixel size [A/Pixel]	1.14	1.14
Absolute Frequency [1/Pixel]		0.285

*To do so, click the **Use resolution[A]** button; in the pop-up window set the resolution to 6 Å, and then click the **Convert units** button. A resolution of 6 Å corresponds to an absolute frequency of 0.19 with a given pixel size of 1.14 Å.*

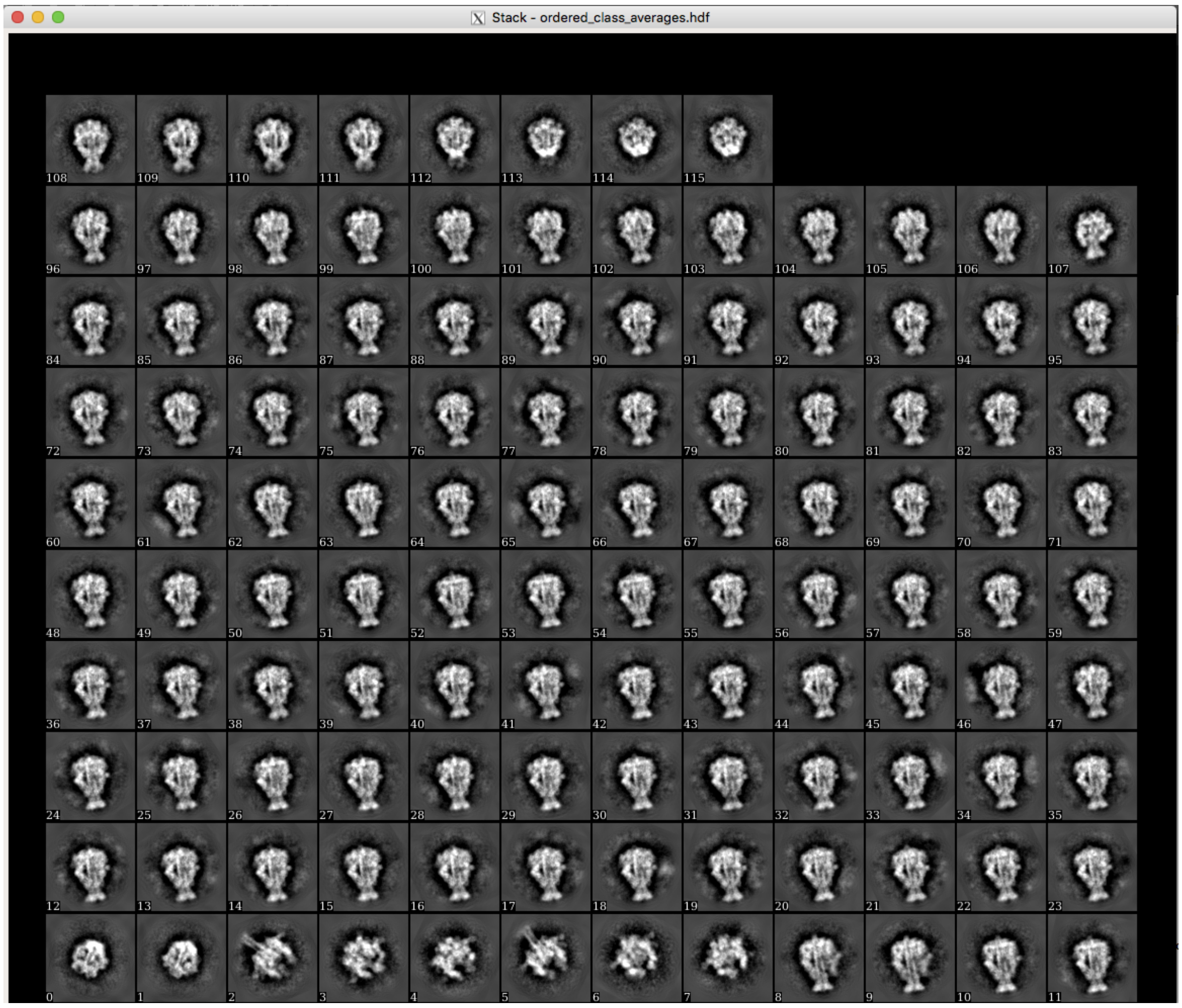
The conversion formula is:

$$\mathit{Absolute\ frequency} = \frac{\mathit{Resolution}}{\mathit{pixel\ size}}$$

*Now click the **Apply** button to set this value in the low-pass filter frequency input field.*

NOTE: *According to the above setting, the "beautified" class averages will be low pass filtered to 6 Å [Absolute frequency 0.19]. We prefer to apply this filter, since the class averages contain only a few members (100 members/group). If you do not wish to low-pass filter the averages, use the default value of -1 instead (low-pass filter will not be applied). Setting this parameter to 0 will low-pass filter to the resolution automatically determined from the **ISAC2** averages **before** the local refinement; setting it to -2, the program will low-pass filter to the resolution automatically determined from the **ISAC2** averages **after** the local refinement.*

Specify the number of processors (for this job, we used 48) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. On our cluster this job took about 7 minutes using 48 processors. Display the final stable, resized, refined, power-spectrum adjusted and CTF-corrected and ordered class averages (**Beaut/ordered_class_averages.hdf**) using the **Display Data** utility.



Click the **mouse wheel button** while pointing somewhere on the graphics window and adjust the **brightness** and the **contrast** at the **e2display.py** pop-up window. The high quality of this dataset is reflected again in the high level of detail of the beautified 2D class averages. Delete the class averages of the pore state of the complex, as described in the previous section, and store the remaining images in a new file named "**Beaut/ best.hdf**".

> **TIP:** *Direct visualization of secondary structure elements in 2D class averages requires a sufficiently large dataset and a class-size of 1000 to 2000 members.*

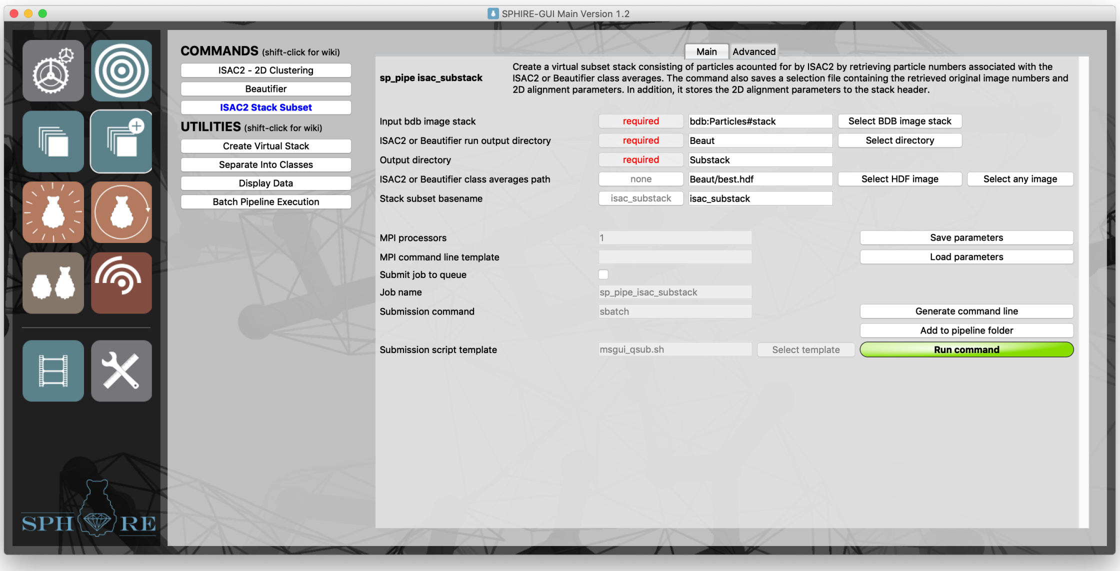
Beautified class averages can also be used to calculate an initial model using **RVIPER**. In several cases we used them to obtain sub-nanometer structures, and thus better initial models. However, running time of **RVIPER** increases significantly with the box size. Therefore, we generally recommend using the downscaled **ISAC2** class averages as input for **RVIPER** instead.

ISAC2 Stack Subset

The initial model that we will calculate in the next step (**RVIPER**) will be refined against the particle members of the selected class averages (the "clean" stack). This will be done in our **MERIDIEN** 3D structure refinement.

Now, we will create a virtual "clean" particle stack that will contain only the members of the selected class averages (**Beaut/best.hdf**).

It should be emphasized that the **Beautifier** tool performs a local 2D refinement of the particle members of the beautified, full resolution class averages and centering is in general improved. The resulting orientation parameters can be passed later to the 3D **MERIDIEN** refinement helping to accomplish higher resolution of the final map at a shorter time of calculations. Therefore, we store in the header of the particles of the "clean" stack the 2D alignment parameters. In the main window of the SPHIRE GUI, click the button ***ISAC2*** on the left and then the ***ISAC2 Stack Subset*** button in the middle.



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:Particles#stack
ISAC2 or Beautifier run output directory	required	Beaut
Output directory	required	Substack
ISAC2 or Beautifier class averages path	none	Beaut/best.hdf
Stack subset basename	isac_substack	isac_substack
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	sp_pipe_isac_substack	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Next, fill out the following input fields:

- **Input BDB image stack:** bdb:Particles#stack

  NOTE: *Click the **Select BDB image stack** button, and use the file browser to select in the **Particles** folder the virtual stack that contains all particles in the **Particles** folder.*

- **ISAC or Beautifier run output directory:** Beaut

NOTE: *Click the **Select directory** button, and use the file browser to select the output directory of the **Beautifier** run. From this directory the program will import the 2D alignment parameters provided by the **Beautifier** local refinement and store them to the header of the extracted particles.*

TIP: *If you skipped the **Beautifier** step, you should select the **ISAC2** output directory (**Class2D**) instead. In this case the 2D parameters stored in the header of the resulting "clean" stack will be imported from the **ISAC2** 2D alignment parameters. Keep in mind that since **ISAC2** performs the alignment and clustering on downscaled images, the x-, y-shifts will be less precise, and the 3D refinement will not start from the best possible centering parameters and might need more iterations to converge.*

- **Output directory:** Substack
- **ISAC2 or Beautifier class averages path:** Beaut/best.hdf

  NOTE: *Click the **Select HDF image** button, and use the file browser to select the file that contains the selected class averages.*

- **Stack subset basename:** isac_substack

  NOTE: *Particles that are members of selected class averages will be stored in a virtual BDB file named **isac_substack** in the specified output directory (**bdb:Substack#isac_substack**).*

Click the ***Run command*** button and check the output of the job.

At the terminal, type:

```
tail -f name_of_sp_pipe_logfile

Found 11516 entries in Beaut/ali2d_local_params.txt

Detected 110 ISAC class averages in Beaut/best.hdf

Extracted 10949 ISAC class members from Beaut/best.hdf
```

The "clean" stack contains 10949 particles. In the case of this high-quality dataset, we eliminated about 16 % of particles.

TIP: *For most of our datasets the number of eliminated particles is usually much higher (20 % to 30 %). Sometimes the number of accounted for particles does not even reach 50 %, but it is nevertheless preferable to work with a smaller but cleaner dataset.*

TIP: *If the dataset contains populations of particles whose shape differ significantly (e.g. different oligomers or proteins) and which are easily recognizable, we recommend storing the respective class averages and their members in different files and performing the subsequent steps of structure determination for each subset independently.*

VIPER

Initial 3D Model - RVIPER

The **RVIPER** program is designed to determine a reproducible and validated initial model at intermediate resolution using a small subset of class averages produced by **ISAC2**.

First, check the number of your selected **ISAC2** class averages.

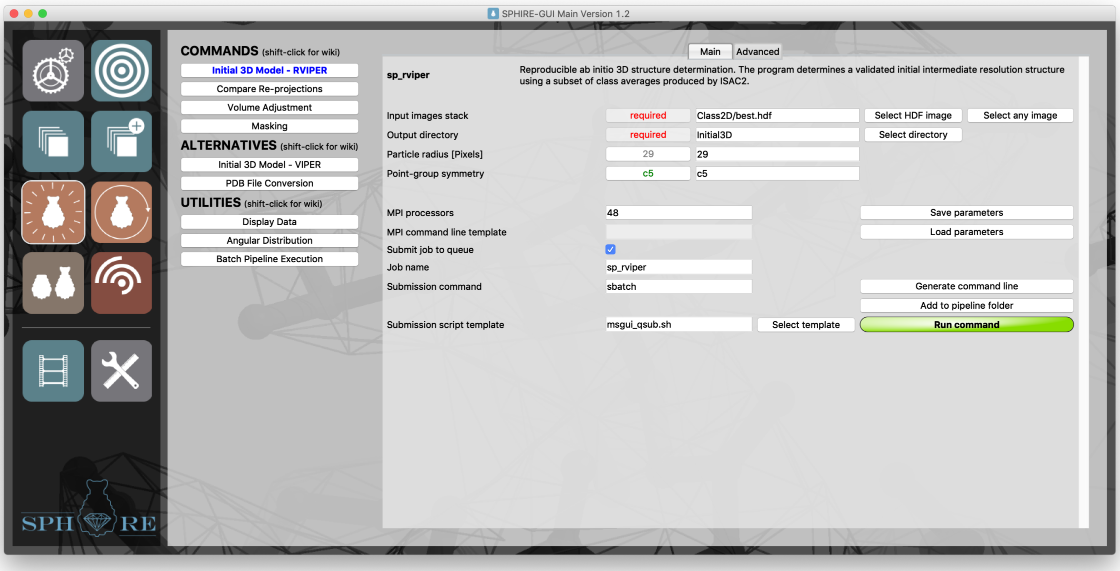
```
e2iminfo.py Class2D/best.hdf
```

In general, about 150 high quality class averages are sufficient to obtain a validated initial structure in a reasonable amount of time. Although **RVIPER** would likely provide better results with more than 150 averages, computational time would increase significantly as well and in most cases there would be no significant improvement of the resulting model.

After running **ISAC2**, we deleted the averages of the second population. The output file **best.hdf**, contains only 110 class averages (the exact number of class averages can slightly differ due to the internal randomization used during the initialization of the **ISAC2** clustering process). However, the test complex has C5 symmetry and therefore the available number of class averages should be sufficient to calculate a reliable 3D model with **RVIPER**.

> **TIP:** *In order to run **RVIPER** successfully for cryo data it is necessary to have at least 60 to 80 high quality averages with about 100 to 200 members each (for negative stain about 100 members). Be sure that the input class averages include as many orientations of the particle, as possible. It will not be possible to calculate a structure from images dominated by one (or a few) preferred orientation(s). If class averages contain only few members or have a low signal to noise ratio, you might need to use more than 150 class averages for **RVIPER**, particularly in the case of asymmetric complexes.*

In the main window of the SPHIRE GUI, click button **VIPER** on the left and then the Initial ***Initial3D Model - RVIPER*** button in the middle.



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	Class2D/best.hdf
Output directory	required	Initial3D
Particle radius [Pixels]	29	29
Point-group symmetry	c5	c5
MPI processors	48	
MPI command line template		
Submit job to queue	✓	
Job name	sp_rviper	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following input fields:

- **Input images stack:** Class2D/best.hdf
- **Output directory:** Initial3D
- **Target particle radius [Pixels]:** 29

  IMPORTANT: Use the same target particle radius as the one used for **ISAC2**.
- **Point-group symmetry:** C5

  TIP: *Use C1 if the symmetry of the complex is not known.*

Advanced parameters:

- **Restarting iteration:** 0

  NOTE: *If your **RVIPER** run crashed for some reason, set this parameter to 0 and resubmit the job. The program will then identify the latest fully completed iteration and continue from there.*
- **Eliminate disconnected regions:** 1400,1.14

NOTE: *Enter the approximate molecular weight of the complex (kDa) and the pixel size (Å), separated by a comma. Based on these values the program will compute, after each iteration, the threshold at which the candidate structure occupies the expected volume (in voxels). Any disconnected pieces that will appear at this threshold will be eliminated and the corrected 3D structure will be used as reference for the next iteration. The intention is to force the program to favour candidate structures that have "structural integrity", as reasonably expected for physically plausible macromolecular complexes.*

NOTE: *If the molecular weight is underestimated or there is strong complex flexibility the program may eliminate valid regions of the structure, a problem particularly common with negative stain data; In this case, try to increase the molecular weight value and restart the program. If in doubt, one can also perform two runs of* ***RVIPER****: one with elimination of disconnected regions, one without and afterwards compare the results.*

TIP: *To suppress the above functionality, enter "none" into the input field.*

Specify the number of processors (for this job we used 48) and submit the job to the queuing system of the cluster using the appropriate submission script and by clicking the ***Run command*** button. Monitor the progress of the job through the standard output. On our cluster this job required about 27 minutes to complete.

NOTE: *Note, that the number of processors needs to be a multiple of the advanced settings option* ***GA population size*** *(default 4).*

Known issue

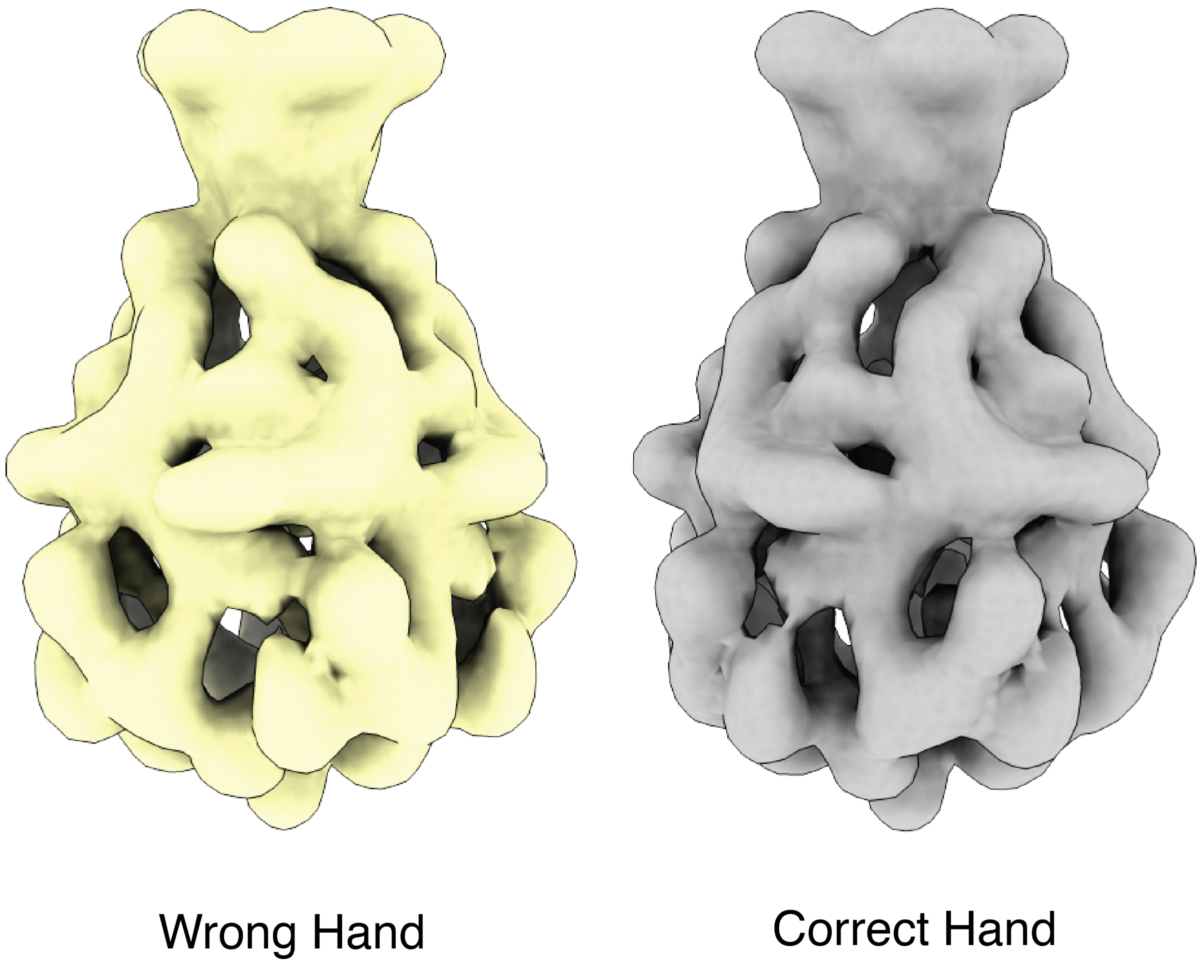
The program might crash if the number of processors exceeds the number of class averages.

RVIPER (**R**eproducible **VIPER**) performs multiple **VIPER** runs (*ab initio* 3D structure determination using a small set of ISAC2 class averages) and thus calculates multiple initial models. It performs a reproducibility test, removes unstable projections and produces a single validated initial model.

After the program finished, output **mainITERNR** folders are placed in the **Initial3D** output. The number of **mainITERNR** folders corresponds to the number of **RVIPER** iterations. Folders named **runITERNR** within each main folder contain the results of each independent **VIPER** run. In our case there is only one such folder: **main001**. The final reproducible structure is **Initial3D /average_volume_001.hdf**.

TIP: *If the internal stability criterion is not met after 10 independent* ***VIPER*** *runs, the program will stop (it will not continue to the final iteration) and no average structure will be created. However, we still recommend careful examination of the output structures from all independent runs (i.e.,* ***volf.hdf*** *and* ***refvol2.hdf*** *in the subdirectories* ***main001/runITERNR****) as some of them might fit the input data sufficiently well and thus might serve as suitable* *initial models in the subsequent 3D **MERIDIEN** structure refinements. Nevertheless, it should be emphasized that these structures are not validated and therefore their quality has to be assessed visually. In particular their structural integrity, i.e., presence of "moons" (significant large densities) floating around the main body of the structure can indicate problems.*

Keep in mind that one cannot determine the hand of a 3D structure based on the set of its 2D projections alone. Therefore, **RVIPER** might have produced an enantiomer (mirror structure). There is no way to establish correct handedness of low-resolution models short of performing tilt experiments. In our case the hand of the test structure is known and therefore the handedness of the map obtained from the current **RVIPER** run can be set correctly already at this point. For this purpose, display the **RVIPER** map using the molecular graphics program **UCSF Chimera** (Pettersen et al. 2004a).

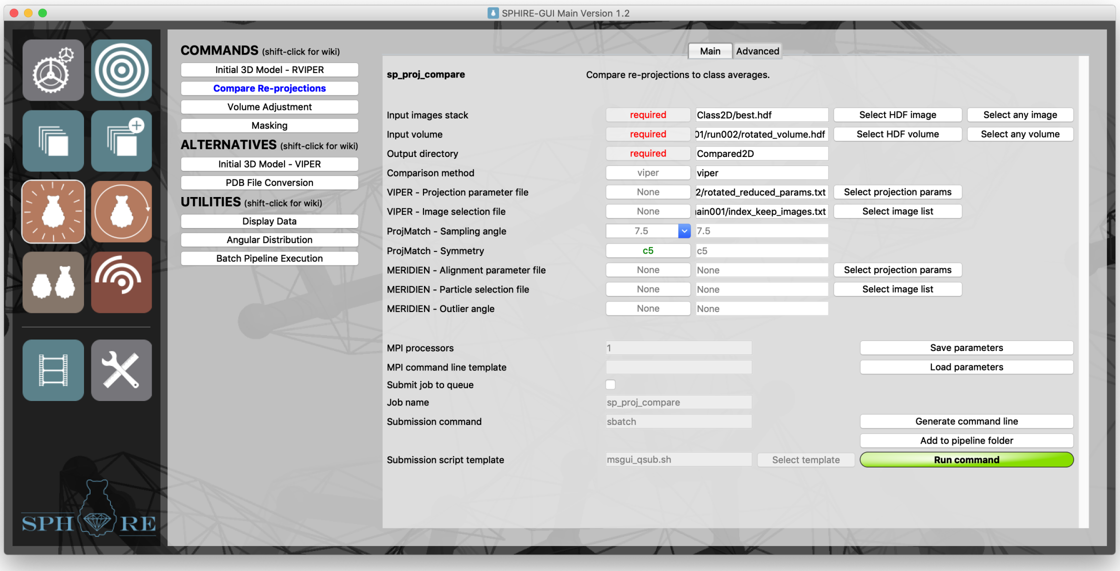


Compare the appearance of the obtained **RVIPER** maps with the images above and check if your map has the correct handedness. If not, it will be necessary to mirror the structure about the x-y plane (that is along the z-axis). This can be performed in the subsequent **Volume adjustment** step.

TIP: *In case neither the handedness of the complex is known nor a homologous structure is available, you should either perform a tilt experiment or proceed with 3D refinement using the obtained **RVIPER** model. Once sufficient resolution is reached, you can establish the correct hand based on the visual appearance of secondary structure elements.*

Compare re-projections

During a RVIPER run, orientation parameters for each 2D class average are determined, which are finally used to compute the initial 3D model. The **Compare Re-projections** tool can now be used to produce side-by-side comparisons of the 2D class averages with the respective projections of the initial 3D model. This provides a helpful tool to validate the quality of the 2D class averages, the assigned orientation parameters and the robustness of the initial 3D model. Click the **Compare Re-projections** button in the middle of the SPHIRE GUI.



sp_proj_compare Compare re-projections to class averages.

Input images stack	required	Class2D/best.hdf
Input volume	required	01/run002/rotated_volume.hdf
Output directory	required	Compared2D
Comparison method	viper	viper
VIPER - Projection parameter file	None	2/rotated_reduced_params.txt
VIPER - Image selection file	None	iain001/index_keep_images.txt
ProjMatch - Sampling angle	7.5	7.5
ProjMatch - Symmetry	c5	c5
MERIDIEN - Alignment parameter file	None	None
MERIDIEN - Particle selection file	None	None
MERIDIEN - Outlier angle	None	None

MPI processors	1
MPI command line template	
Submit job to queue	☐
Job name	sp_proj_compare
Submission command	sbatch
Submission script template	msgui_qsub.sh

Fill out the following fields:

- **Input images stack:** Class2D/best.hdf

  NOTE: *Click the **Select HDF image** button, and use the file browser to select in the **Class2D** folder the stack containing the selected class averages of the prepore state (**best.hdf**), which were also used as input for RVIPER.*

- **Input volume:** Initial3D/main001/run002/rotated_volume.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select an RVIPER volume*

- **Output directory:** Compare2D
- **Comparison method:** viper

  TIP: *If you skipped **RVIPER** and wish to perform the comparison with an available 3D volume, type here the option **projmatch** instead. The program will then perform projection matching to find the best match between the class averages and the 2D projections of the 3D volume, at a user-selected angular interval.*

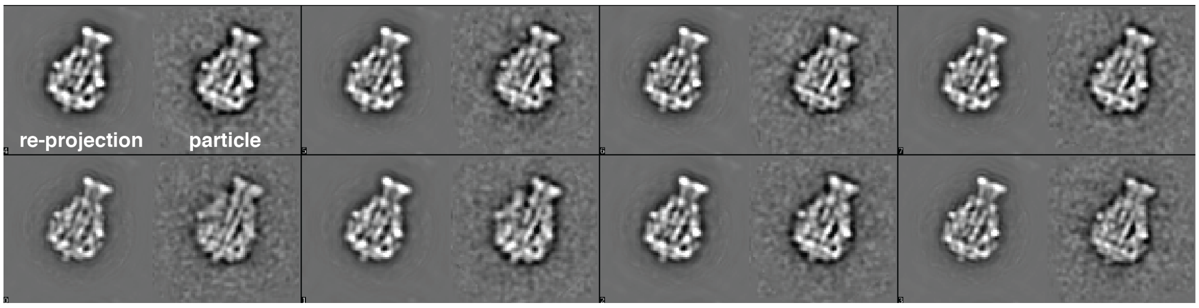
- **VIPER – Projection parameter file:** Initial3D/main001/run002/rotated_reduced_params.txt

  NOTE: *Click the **Select projection params** button, and use the file browser to select the file containing the assigned projection parameters for the respective RVIPER run.*

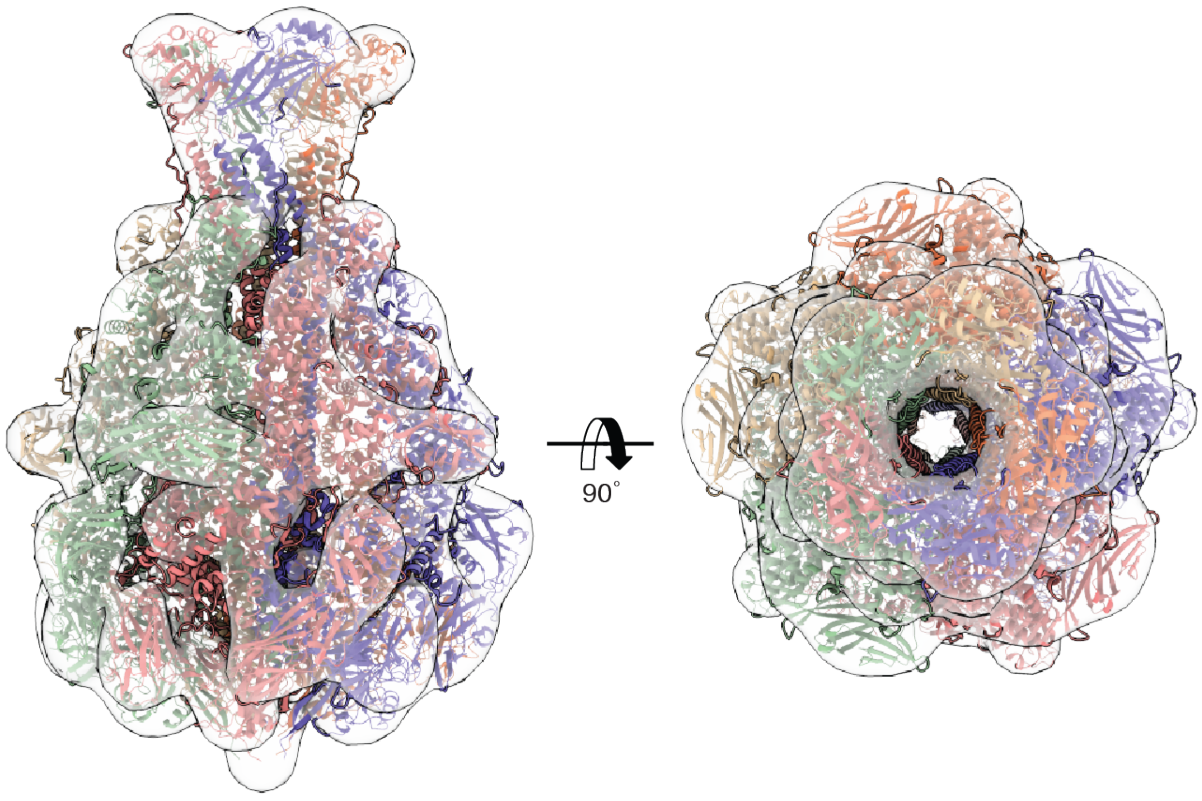
- **VIPER – Image selection file:** Initial3D/main001/index_keep_images.txt

  NOTE: *Click the **Select image list** button, and use the file browser to select the file containing the list of class averages which were considered stable during initial model generation.*

and click the **Run Command** button. This process is usually finished after few seconds. Display the comparison file (**Compare2D/comp-proj-reproj.hdf**) using the ***Display Data*** utility. Each class average will be shown next to the respective re-projection, and both images should show an excellent agreement.



At this point we also strongly recommend to compare at this point the **RVIPER** structure with the available X-ray crystallographic structure of TcdA1 (3.9 Å, PDB-ID: *1VW1* (Meusch et al. 2014)). Load the volume and the PDB file in **UCSF Chimera**, set the voxel size of the volume to 5.7 Å (see Volume Adjustment section below for details) and fit the atomic model into the density map using **Fit in Map**. They should show an excellent agreement, even though at the current resolution only the overall envelope of the EM model is reliable.



You can also create a *.bild* file with the **Angular Distribution** utility to assess the angular distribution of the class averages used during the **VIPER** 3D reconstruction, as described on our SPHIRE wiki page. The file can also be displayed in **UCSF Chimera**.

TIP: *Alternative to the procedure described above, instead of using the downscaled **ISAC2** class averages as input for **RVIPER**, one can also calculate an initial 3D model directly from the beautified class averages with the original pixel size. However, the computational time for **RVIPER** on averages of original image size increases by the square of their size. Depending on the quality of the averages, the advantage of running **RVIPER** on original image size is that you might be able to obtain directly from beautified class averages initial models at 6 Å to 9 Å. For this approach however, you will have to set the **Low-pass filter frequency [1/Pixels]** (**Advanced** option) (filter to be applied to the **RVIPER** structures) to an appropriate value before submitting the job.*

TIP: *If an X-ray structure of a homologous complex is available and there is a high- degree of similarity expected, one can omit the ab initio structure determination with RVIPER, and use the available atomic coordinates instead. To do so, we use the PDB File Conversion utility (Alternatives) to convert the atomic model (typically downloaded from the PDB data base) to a density map.*

If you do not have enough time to perform these steps, you can also copy our precalculated results to the **project directory**.

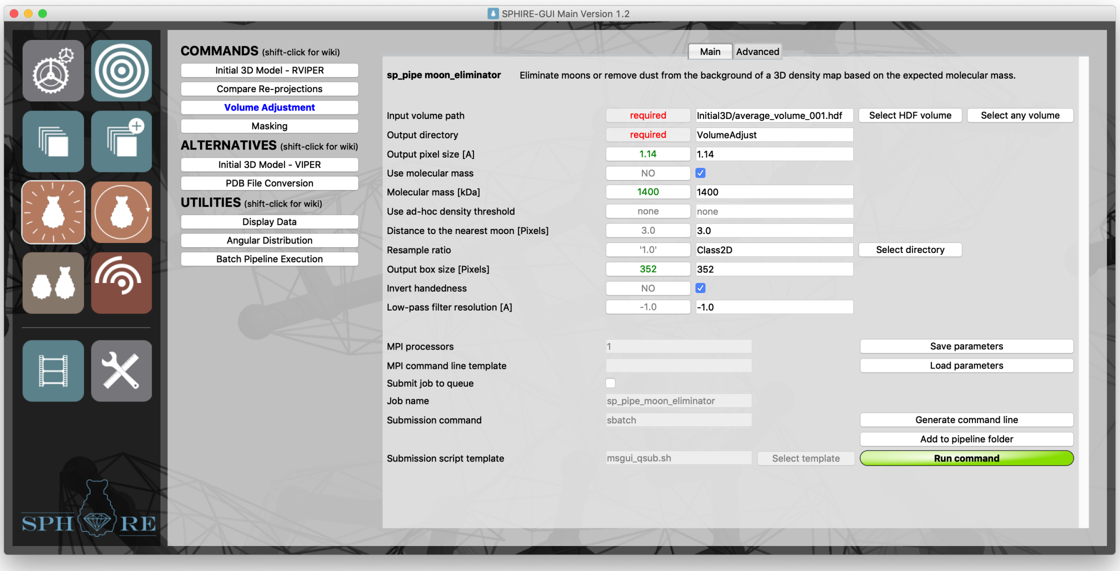
```
cp -Rp SphireDemoResults/Initial3D ./
```

Volume Adjustment

The particles of the dataset were resized during **ISAC2** (in this case from a radius of 145 pixels and box size of 352 pixels to a radius of 29 pixels and box size of 76 pixels). Therefore, the **ISAC2** averages and the **VIPER** 3D model have now an increased pixel size of 5.7 Å. In order to use this model as a reference structure for the 3D refinement of the original dataset, we have to resize and window the **VIPER** model in order to match the dimensions and the pixel size of the original particle stack. In addition, we will mask the reference volume automatically in order to remove background noise and if necessary, invert its handedness. To check the shrink ratio used to downscale the images before **ISAC2** type:

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

Now click the ***Volume Adjustment*** button in the middle of the SPHIRE GUI.



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	Initial3D/average_volume_001.hdf
Output directory	required	VolumeAdjust
Output pixel size [A]	1.14	1.14
Use molecular mass	NO	✓
Molecular mass [kDa]	1400	1400
Use ad-hoc density threshold	none	none
Distance to the nearest moon [Pixels]	3.0	3.0
Resample ratio	'1.0'	Class2D
Output box size [Pixels]	352	352
Invert handedness	NO	✓
Low-pass filter resolution [A]	-1.0	-1.0
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	sp_pipe_moon_eliminator	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following fields:

- **Input Volume Path:** Initial3D/average_volume_001.hdf or Initial3D/average_volume_001_flip.hdf
- **Output Directory:** VolumeAdjust
- **Output pixel size:** 1.14
- **Use molecular mass:** Yes
- **Molecular mass [kDa]:** 1400

  NOTE: *The program will eliminate "moons" (background noise densities) based on the molecular weight of the protein. It will automatically determine a threshold and eliminate densities not connected to the main volume.*

- **Resample ratio:** Class2D

  NOTE: *Click the **Select directory** button, and use the file browser to select the ISAC directory (**Class2D**). The program will automatically extract the resampling ratio from the **README_shrink_ratio.txt** located in the **Class2D** folder and restore the original pixel size and dimensions of the RVIPER model accordingly.*

- **Invert Handedness:** YES or NO?

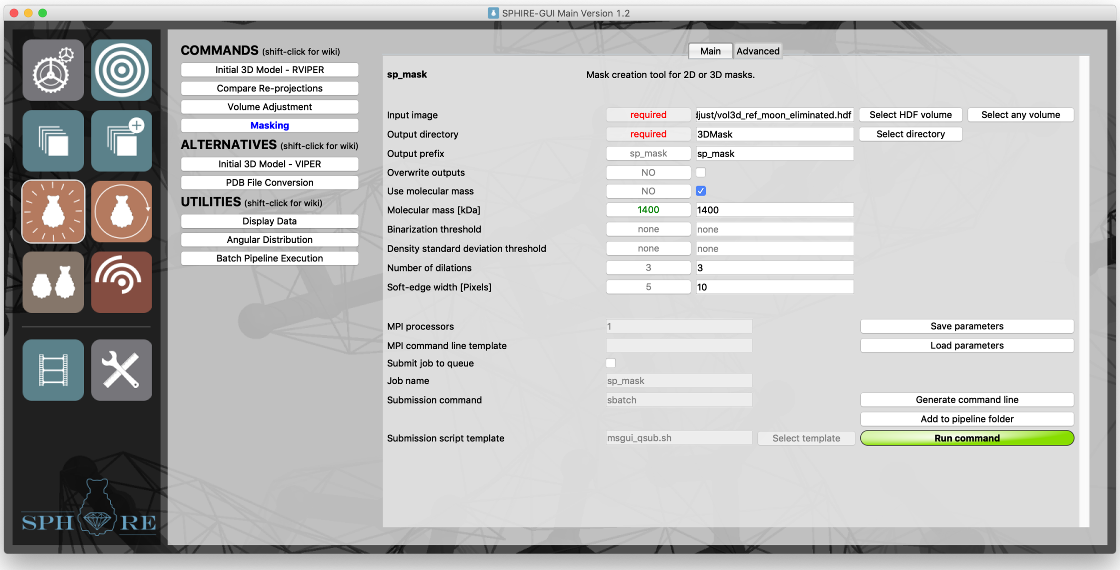
  NOTE: *If your RVIPER volume shows the wrong hand, activate **the Invert Handedness** option.*

Click the **Run Command** button. This step is finished in few seconds. Display the resulting volume (Volume_Adjust/vol3d_ref_moon_eliminated.hdf) using **UCSF Chimera**, and verify it shows all expected structural features.

TIP: *If the molecular weight is underestimated or there is strong flexibility in the complex, the program may eliminate valid regions of the structure (particularly common with negative stain data). In this case, try to increase the molecular weight value and restart the program. Alternatively, instead of the automatic "moon" elimination, you can use an ad-hoc threshold value (**Use ad hoc density threshold**). To do so, first deactivate the **Use molecular mass** option. Determine an appropriate threshold for the volume (using **Volume Viewer in UCSF Chimera**) lower than you would normally use for displaying the structure, and fill the respective field. Noise artifacts can be visible but not connected to the molecule density.*

Adaptive 3D Mask

Now click the ***Masking*** button in the middle of the SPHIRE GUI.



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

Input image	required	just/vol3d_ref_moon_eliminated.hdf
Output directory	required	3DMask
Output prefix	sp_mask	sp_mask
Overwrite outputs	NO	☐
Use molecular mass	NO	☑
Molecular mass [kDa]	1400	1400
Binarization threshold	none	none
Density standard deviation threshold	none	none
Number of dilations	3	3
Soft-edge width [Pixels]	5	10
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	sp_mask	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following fields:

- **Input image:** Volume Adjust/vol3d_ref_moon_eliminated.hdf

  NOTE: *This is the* ***RVIPER*** *result after resizing and windowing to the original box/pixel size.*

- **Output Directory:** 3DMask

- **Use molecular Mass:** Yes

- **Molecular Mass [kDa]:** 1400

  NOTE: *These are the parameters for the moon elimination tool (see the section above for instructions).*

- **Number of dilations:** 3

  NOTE: *Numbers of cycles to extend this initial binarized structure; see the section below for instructions.*

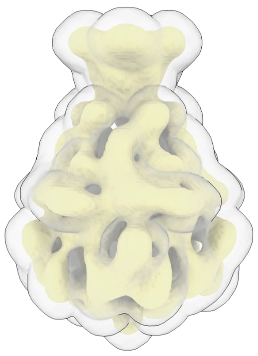
- **Soft-edge width [Pixels]:** 10

  NOTE: *This is the pixel-width for the transition area of the soft-edge mask. The default value is optimal for volumes with pixel size of roughly 1 Å. You will have to adjust this value if you are using a significantly different pixel size.*

Click the ***Run command*** button. This process is usually finished after few seconds. Detailed description regarding the usage of this program is provided on our wiki page.

Masking

Open the adjusted RVIPER volume (**Volume Adjust/vol3d_ref_moon_eliminated.hdf** created in the previous step) and the 3D mask (3DMask/sp_mask_mask.hdf) with **UCSF Chimera**. In order to compare both maps in **UCSF Chimera**, set their voxel sizes to $1.14\,\text{Å}$ (**Volume Viewer->Features->Coordinates->Voxel Size**: $1.14\,\text{Å}$). Check the size of the resulting mask relative to the initial map again using **UCSF Chimera** (we usually display the 3D mask transparent, as shown in the figure below).



The generated mask should have the shape of the particle, but it should extend in all directions by a sufficient number of voxels (3 to 5) to prevent masking artifacts during 3D refinement. The amount of extension is controlled by the number of dilations parameter (default set to 3; you may have to adjust this value). The resulting mask should have sufficient clearance around to the map while excluding all satellite densities.

MERIDIEN

In the previous steps we removed all "junk" images and produced a clean subset of particles which was aligned and clustered in a stable and reproducible manner using the powerful **ISAC2** approach. Subsequently, from the validated **ISAC2** class averages we calculated a reproducible model. These results give us confidence that we can now proceed with the structure refinement using the obtained model as a starting reference (3D Refinement; **MERIDIEN**).

MERIDIEN employs a quasi-Maximum Likelihood approach (later referred to as ML). Briefly, the set of 2D input images is randomly split into half-datasets and is refined quasi-independently in order to minimize noise bias and over-refinement. During the projection-matching phase, the program estimates "probabilities" with which each particle image matches reference reprojections of the current approximation of the 3D structure. Next, during the 3D reconstruction phase, the data is backprojected using all significant angular (and translation) directions with probabilities used as weights. These two steps are then iterated. Note both the maps and resolution curves (driver FSCs) generated during each iteration of the program serve only as internal approximations (i.e., the resolution reported after each iteration is usually underestimated). Usable results are obtained after the refinement finishes. We use the ***PostRefiner*** utility to process the so-called unfiltered final half-volumes to produce the ultimate result, including the FSC curve.

MERIDIEN provides the user with the following functionalities:

1. The standard use is to perform full 3D structure that starts from an initial, user-provided reference structure. By default the program will start from exhaustive searches without assuming any prior angular information. Any available alignment parameters (e.g. from **ISAC2**) will automatically be incorporated into the search in order to improve the convergence.

2. Local refinement mode is primarily meant to refine subsets obtained by 3D sorting. This mode is described in detail in the subsequent sections. It can also be used to locally refine datasets obtained by other software packages, as long as orientation parameters (both angles and translations) are stored in the input image headers in the expected format.

3. 3D reconstruction of full-size unfiltered maps using parameters obtained from a user-specified iteration of the **MERIDIEN** refinement.

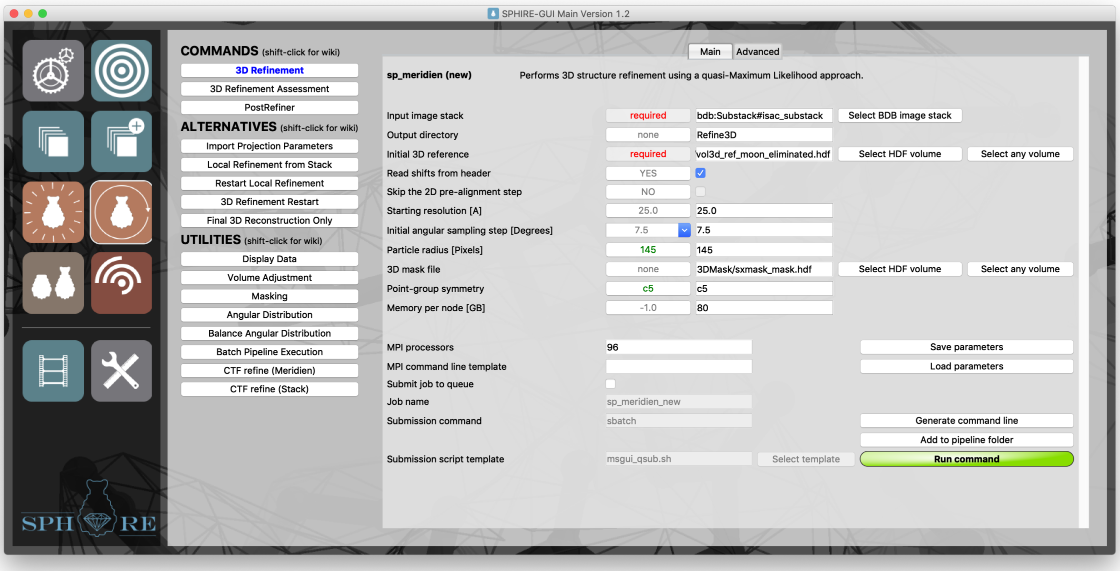
4. Restart mode, which allows user to continue the refinement (either standard or local) should the program crash or run into a time limit (sometimes imposed by a computing cluster).

We will now use mode (1) and perform a high-resolution 3D refinement using the 3D reference volume and 3D mask produced in the previous steps.

TIP: *In most cases* ***RVIPER*** *will deliver a highly reliable and detailed initial model. Therefore, if the overall structure is already established, we recommend beginning the refinement immediately with a 3D mask. Structure refinement that utilizes a reasonably tight mask proceeds faster and yields better (higher resolution) results. In rare cases in which the initial model is of low quality or unreliable you should proceed with caution: first run a 3D refinement without using a 3D mask, validate the outcome and only then create a 3D mask and repeat the refinement.*

3D Refinement

3D refinement of the structure is done using the original individual particle images. The standard default run (mode 1) starts from exhaustive searches using the resized **RVIPER** volume as an initial reference structure. In the main window of the SPHIRE GUI first click the button ***MERIDIEN*** on the left and then the ***3D REFINEMENT*** button in the middle. Fill out the following input fields:



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

Input image stack	required	bdb:Substack#isac_substack
Output directory	none	Refine3D
Initial 3D reference	required	vol3d_ref_moon_eliminated.hdf
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	3DMask/sxmask_mask.hdf
Point-group symmetry	c5	c5
Memory per node [GB]	-1.0	80
MPI processors	96	
MPI command line template		
Submit job to queue		
Job name	sp_meridien_new	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **Input image stack:** bdb:Substack#isac_substack

  NOTE: *Click the* ***Select BDB image stack*** *button, and use your file browser to select the unbinned clean subset of particles, containing only the members of the selected* ***ISAC2*** *class averages.*

- **Output directory:** Refine3D
- **Initial 3D reference:** Volume Adjust/vol3d_ref_moon_eliminated.hdf

NOTE: *Click the **Select HDF volume** button, and use your file browser to select the resized and windowed **RVIPER** structure.*

TIP: *Here you can also directly load any type of 3D structure as an initial reference (i.e. any structure produced by another program, structures derived from X-ray crystallographic model, a simple sphere in case of high point group symmetry, etc.). The file that contains the structure does not have to have any header information. HDF is the primary SPHIRE format for 3D volumes, but .mrc, .spi, bdb:, .img and .dm4 are also supported and can be directly loaded to the GUI by clicking the **Select any volume** button. Keep in mind that the box and voxel size of such an imported volume have to match the input dataset. If this is not the case, the volume has to be resized using the **Volume Adjustment** utilities described above.*

- **Read shifts from header:** YES

  NOTE: *The 2D data should have already translation parameters stored in their headers, as they were computed during 2D alignment in **ISAC2** (and improved further by the **Beautifier** tool). Refinements that use pre-centered particles converge faster and produce better results. Therefore, this flag is set to **YES** by default and the predetermined **ISAC2** shifts are imported from the header automatically. If shifts are not available, set this flag to **NO**. In this case, the program will perform by default a reference-free 2D alignment in an attempt to roughly pre-center the particles. This significantly improves the performance by accelerating the convergence and improving the final resolution. If shifts are not available, but your particles are pre-centered (for example re-extracted particles after a 3D refinement), set this flag to NO and in addition set the flag **Skip the prealignment step** to **YES.** Note that setting flag **Read shifts from header** (see below) to **YES** turns off the initial 2D pre-alignment.*

- **Starting resolution [A]:** 25.0

  NOTE: *The initial **VIPER** model will be low-pass filtered to this resolution to mitigate the initial model bias.*

  TIP: *Be careful if you use an X-ray derived structure as a starting reference. In this case, you should apply a stronger low-pass filter (for example to 40 Å to 50 Å).*

  On the other hand, note that, based on the overall shape of the particle, too restrictive initial filtration may suppress discernible features and the program might fail to converge properly.

- **Initial angular sampling step [Degrees]:** 7.5

  NOTE: *Usually the default value of 7.5° is reasonable to create sufficient number of reprojections for the initial global parameter search for almost every asymmetric and/or low symmetry structures with initial resolution of 15 Å to 25 Å. Refinement of higher resolution initial models and/or structures with high point group symmetry may benefit from a smaller initial angular sampling step, but values lower/equal than 3.75° may result in excessively long time of calculations.*

- **Particle radius [Pixels]:** 145
- **3D mask:** 3DMask/sp_mask_mask.hdf

  NOTE: *This is the 3D mask produced in the previous step.*
- **Point-group symmetry:** C5
- **Memory per node [GB]:** 80

  NOTE: *Depends on cluster specifications. The program has to know the amount of physical memory available on each node as it uses "per node" MPI parallelization. Nodes are basic units of a cluster, and each node has a number of processors. While clusters are often characterized by the amount of memory per processor, here we ask for the total amount of **Memory per node [GB]** as the program will internally adjust the number of processors it is using. For example, a cluster that has 3 GB memory per processor and 16 processors per node has 3 GB × 16 = 48 GB memory per node. It is advisable to use a lower value (about 20 %) as at least some part of the memory of a node is not available to the program.*

IMPORTANT: Listed below are advanced options; please do not change their default values unless you understand their meaning well.

Advanced parameters:

- **Search range [Pixels]:** 5.0
- **Search step size [Pixels]:** 1.0

  NOTE: *Changing the two values above can cause problems. Increasing the initial search range and step might result in an unexpectedly long execution time. Decreasing the range and step may cause the program to underperform. Decreasing the range and step is justified only in cases when previously determined orientation parameters are available.*
- **Correlation peaks to be included [%]:** 99.9

  NOTE: ***MERIDIEN** will computationally match each image with template reprojections of the structure and convert their similarity into probability values. For high quality data and good agreement of the data with the initial model we want to include all computed probabilities as this assures the best performance of the program. However, for lower quality data, the number of non-zero probabilities (and thus the "smear" of the correlation peak) might be excessively large and the running time of the 3D reconstruction step might increase excessively. Should that happen, it is advisable to terminate the program and lower the value of this parameter. 0.0 % value corresponds to usage of one orientation per image, thus "hard matching" refinement. While this setting will result in a very fast execution of the program, the result will be most likely disappointing. We found that for challenging cases, setting of this parameter to 90 % to 95 % provides a good balance between the runtime and the quality of the final result.*

Specify the number of processors (on our cluster that has 24 processors per node, we used 48 processors) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. 3D structure refinement with **MERIDIEN** is computationally expensive and the running time increases significantly with the number of particles and the box size.

The refinement progresses through a sequence of search modes: **INITIAL**, **PRIMARY**, **EXHAUSTIVE**, **RESTRICTED**, and **FINAL**. The main distinction is between **EXHAUSTIVE** searches, in which every data image is matched with all quasi-evenly generated reprojections of the model, and **RESTRICTED**, in which matching is done only within the neighborhood of orientation determined in the previous iteration. The program uses elements of the Maximum Likelihood machinery. **MERIDIEN** is driven by the internal assessment of the resolution between two quasi-independent concurrent refinements. The current resolution is then used to set main parameters of the refinement process (i.e., angular step, shift search range, maximum resolution and thus the window size) using a set of simple heuristics. In order to prevent premature termination at a suboptimal resolution stage, the program also uses heuristic randomization of the refinement process. Thus, repeated runs of the program from the same starting point might yield slightly different results, both in terms of orientation parameters and the ultimate resolution. One can monitor the progress of the job through the standard output, which will provide valuable information about the refinement. For example, during every iteration, the program will print the time it took to process a chunk of 20 % of data. Based on that, one can easily estimate the remaining time for the respective iteration.

```
Number of images : 9 48 18.8% 0.0min
```

In order to quickly check the progress of the run and the resolution from iteration to iteration, type:

```
grep ITERATION my_standard_outputfile
```

NOTE: *The name of the text file containing the standard output depends on your submission script.*

```
2019-05-09 14:38:38 main =>ITERATION # 1. Current state: INITIAL, nxinit: 52, delta: 7.5000,
xr: 5.0000, ts: 1.0000

2019-05-09 14:39:38 main =>Resolution achieved in ITERATION # 1: 16/ 26 pixels, 25.08A/15.43A.

2019-05-09 14:39:38 main =>ITERATION # 2. Current state: PRIMARY, nxinit: 120, delta: 7.5000,
xr: 5.0000, ts: 1.0000

2019-05-09 14:40:56 main =>Resolution achieved in ITERATION # 2: 34/ 46 pixels, 11.80A/ 8.72A.

2019-05-09 14:40:56 main =>ITERATION # 3. Current state: PRIMARY, nxinit: 102, delta: 7.5000,
xr: 5.0000, ts: 1.0000

2019-05-09 14:42:04 main =>Resolution achieved in ITERATION # 3: 37/ 47 pixels, 10.85A/ 8.54A.

2019-05-09 14:42:04 main =>ITERATION # 4. Current state: PRIMARY, nxinit: 108, delta: 7.5000,
xr: 5.0000, ts: 1.0000

2019-05-09 14:43:14 main =>Resolution achieved in ITERATION # 4: 37/ 47 pixels, 10.85A/ 8.54A.

2019-05-09 14:43:14 main =>ITERATION # 5. Current state: PRIMARY, nxinit: 108, delta: 7.5000,
xr: 5.0000, ts: 1.0000
```

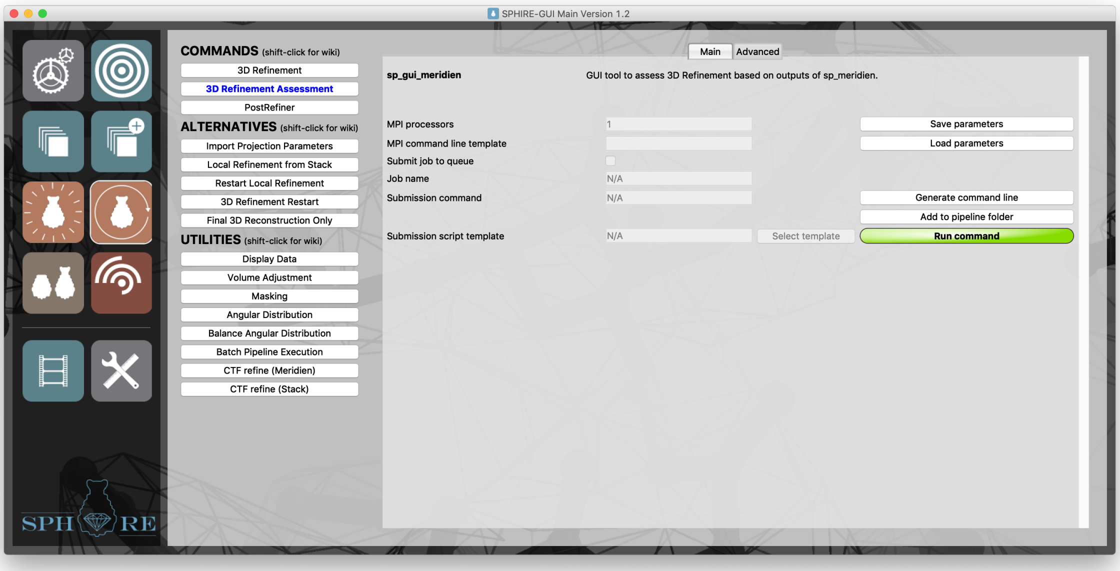
```
2019-05-09 14:44:24 main =>Resolution achieved in ITERATION # 5: 37/ 46 pixels, 10.85A/ 8.72A.

2019-05-09 14:44:24 main =>ITERATION # 6. Current state: EXHAUSTIVE, nxinit: 108, delta: 3.7500,
xr: 4.1276, ts: 0.9663

2019-05-09 14:48:59 main =>Resolution achieved in ITERATION # 6: 46/ 58 pixels, 8.72A/ 6.92A.

2019-05-09 14:48:59 main =>ITERATION # 7. Current state: EXHAUSTIVE, nxinit: 126, delta: 3.7500,
xr: 4.1276, ts: 0.9663
```

The printout contains a sequence of pairs of lines for each iteration. They contain information about the time the iteration started, its number and the resolution achieved, both in Fourier pixels and in Å as read from driver FSC@0.5 and FSC@0.143. Note that the so-called driver FSC is computed during the individual iterations and thus the printed resolutions are not the ultimate results but merely approximations calculated quickly for expediency to guide the program and used for adjustment of its parameters.

```
2019-05-09 14:51:34 main =>ITERATION # 9. Current state: EXHAUSTIVE, nxinit: 128, delta: 3.7500,
xr: 4.1276, ts: 0.9663 2019-05-09 14:52:56 main =>Resolution achieved in ITERATION # 9: 46/ 59
pixels, 8.72A/ 6.80A.
```

For example, the line above informs us that ITERATION number 9 involved exhaustive searches and the parameters used were: processing window size (**nxinit**) 128, angular step (**delta**) 3.75°, translational search range (**xr**) 4.127 6 pixels, and search step (**ts**) 0.966 3 pixels.

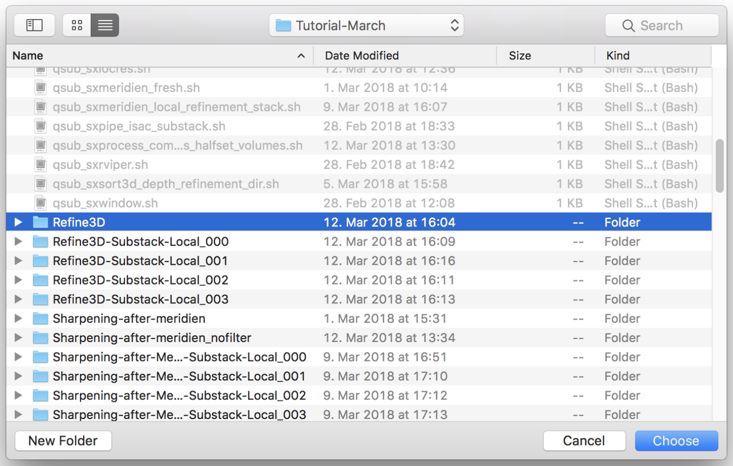
You can also follow the progress of the refinement, and plot the driver FSC for each iteration using the **3D Refinement Assessment** GUI. For this purpose, click the ***Refinement Assessment*** button in the middle:



sp_gui_meridien GUI tool to assess 3D Refinement based on outputs of sp_meridien.

MPI processors 1
MPI command line template
Submit job to queue ☐
Job name N/A
Submission command N/A
Submission script template N/A

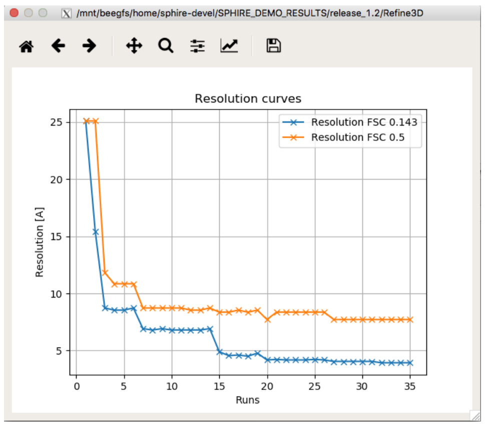
and then click the **Run command** button. In the pop-up window select the current refinement directory (**Refine3D**):



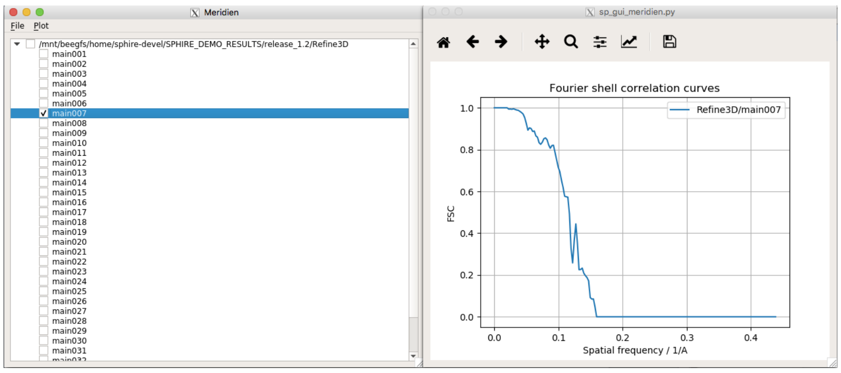
Tutorial-March

Name	Date Modified	Size	Kind
qsub_sxlocres.sh	12. Mar 2018 at 12:36	1 KB	Shell S...t (Bash)
qsub_sxmeridien_fresh.sh	1. Mar 2018 at 10:14	1 KB	Shell S...t (Bash)
qsub_sxmeridien_local_refinement_stack.sh	9. Mar 2018 at 16:07	1 KB	Shell S...t (Bash)
qsub_sxpipe_isac_substack.sh	28. Feb 2018 at 18:33	1 KB	Shell S...t (Bash)
qsub_sxprocess_com...s_halfset_volumes.sh	12. Mar 2018 at 13:30	1 KB	Shell S...t (Bash)
qsub_sxrviper.sh	28. Feb 2018 at 18:42	1 KB	Shell S...t (Bash)
qsub_sxsort3d_depth_refinement_dir.sh	5. Mar 2018 at 15:58	1 KB	Shell S...t (Bash)
qsub_sxwindow.sh	28. Feb 2018 at 12:08	1 KB	Shell S...t (Bash)
Refine3D	12. Mar 2018 at 16:04	--	Folder
Refine3D-Substack-Local_000	12. Mar 2018 at 16:09	--	Folder
Refine3D-Substack-Local_001	12. Mar 2018 at 16:16	--	Folder
Refine3D-Substack-Local_002	12. Mar 2018 at 16:11	--	Folder
Refine3D-Substack-Local_003	12. Mar 2018 at 16:13	--	Folder
Sharpening-after-meridien	1. Mar 2018 at 15:31	--	Folder
Sharpening-after-meridien_nofilter	12. Mar 2018 at 13:34	--	Folder
Sharpening-after-Me...-Substack-Local_000	9. Mar 2018 at 16:51	--	Folder
Sharpening-after-Me...-Substack-Local_001	9. Mar 2018 at 17:10	--	Folder
Sharpening-after-Me...-Substack-Local_002	9. Mar 2018 at 17:12	--	Folder
Sharpening-after-Me...-Substack-Local_003	9. Mar 2018 at 17:13	--	Folder

After clicking the ***Choose*** button, the program will display a plot of two curves showing resolution as a function of iteration number, as determined by the FSC@0.143 and FSC@0.5 criteria, respectively. During runtime the plot will only include information about completed iterations. Once a new iteration is finished, a pop-up window will prompt you to update the plot.

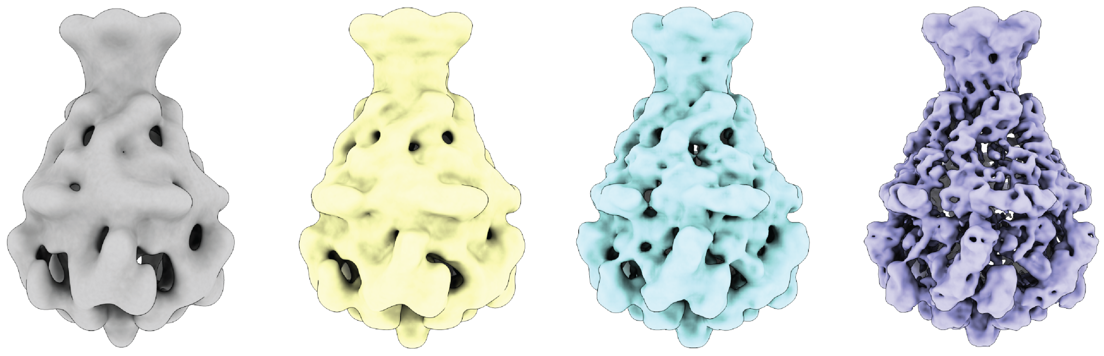


To display the driver FSC of a specific iteration, first check its tick box (i.e., **main003**) at the main window of the **Refinement Assessment** GUI, click the ***Plot*** button and then select the ***New FSC Plot*** option.



NOTE: *Again, please keep in mind the driver FSC@0.143 and driver FSC@0.5 computed during the individual iterations do not reflect the final results.*

TIP: *When we monitor the progress of the refinement, we usually examine the half-maps for selected iterations in **UCSF Chimera**. The nominal improvement in resolution should agree well with the visual appearance of the maps. Note that these maps are not meant for interpretation yet, as they are merely approximations internally used by the program. Thus, you should not attach too much weight to their appearance and, in particular, to the apparent lack of details. Only the so-called "final" maps can be properly interpreted (see below for instructions how to compute them). If one is impatient to examine the details without waiting for the refinement to complete, one can enhance the details using the **PostRefiner** tool in single volume mode that is available in the main **UTILITIES** section of the GUI.*



Iteration	1	4	9	33
Resolution (0.5)(Å)	25	10.8	8.7	7.7

Once the program converged, it will print a notification and then calculate the final unfiltered, full size structures:

```
2018-03-01_11:19:12 =>Convergence criterion A is reached (angular step delta smaller than
3/4 changes in angles))

2018-03-01_11:19:12 =>Resolution achieved in ITERATION #34: 48/ 97 pixels, 8.36A/ 4.14A.
```

The final half-volumes are stored in:

Refine3D/vol_0_unfil_iternr.hdf and **Refine3D/vol_1_unfil_iternr.hdf**.

The final orientation parameters are stored in

Refine3D/final_params_iternr.txt.

Known issue in SPHIRE 1.3

The last iteration is memory intensive and the run may crash if sufficient memory is not available. Check the **MEMORY ESTIMATION** entry for the final iteration in the **MERIDIEN** log file.

```
  2019-05-09 17:43:49 recons3d_final =>MEMORY ESTIMATION. memory per node = 120.0GB,
  volume size = 2.12GB, data size per node = 0.91GB, estimated number of CPUs = 23.96
  2019-05-09 17:43:49 do3d_final =>do_final_rec3d
```

In this case Meridien estimated it required

$$24\,\mathrm{GB} \times 0.91 + 2.12\,\mathrm{GB} = 23.96\,\mathrm{GB}$$

and the available memory per node was set to $80\,\mathrm{GB}$, thus the program assumes there is sufficient memory to calculate the final full-size 3D reconstruction.
If the program crashes due to insufficient memory, try to perform the final reconstruction with the memory per node parameter set to a smaller value (**final 3D reconstruction only**, **ALTERNATIVES**).
If the program does not finalize even if it uses only one processor per node, you have two options:

1. Reduce the size of input particles (and reference volume) by scaling them down. This can be done with the following command:

   ```
   sp_process.py bdb:inputstack bdb:outputstack --changesize --ratio=0.8
   ```

   This will reduce window size to **nx** × 0.8, but has the undesirable associated effect of increasing the pixel size by a factor of 1.0 / 0.8 = 1.25, thus increasing the maximum/Nyquist frequency by the same factor. It is also likely that due to the larger pixel size the alignment accuracy will decrease.

2. Further “clean” your dataset with an additional, more conservative **ISAC2** round, by setting the **Pixel error threshold [Pixels]** parameter of **ISAC2** to a lower value. In general, refinements of “cleaner” stacks have lower average smear, run faster and require less memory.

NOTE: *The actual number of iterations varies from run to run and depending on the number of the iteration with the highest resolution, the parameter file might have a different file name; so, please check the content of Refine3D directory, and use the proper number.*

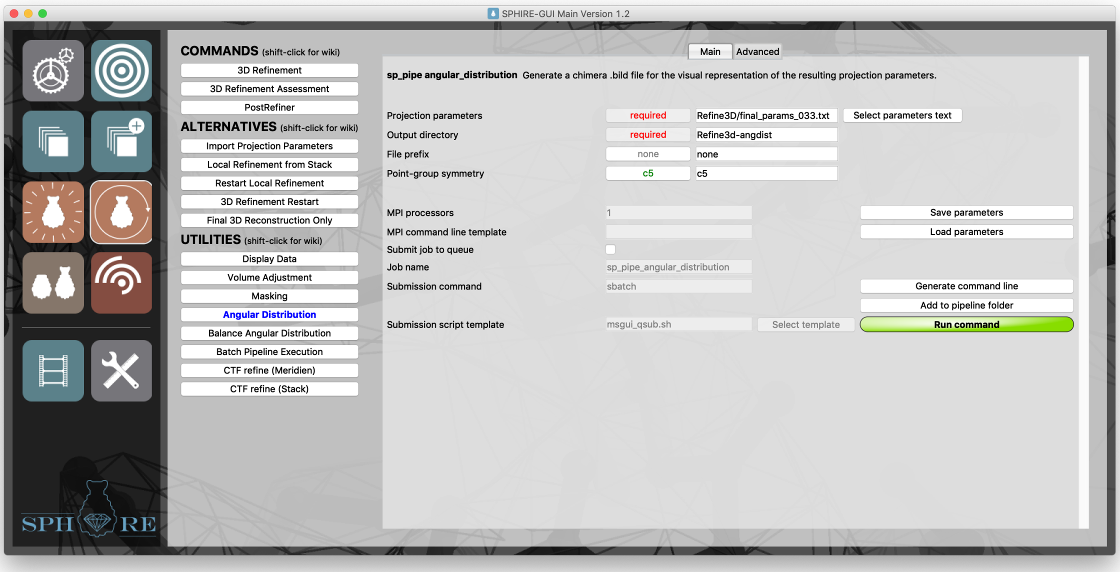
On our cluster this **MERIDIEN** job, running on 48 processors, required about 3 hours to finish. If there is not enough time to perform the 3D refinement, please copy our pre-calculated results to the **project directory**.

```
cp -r SphireDemoResults/Refine3D ./
```

NOTE: *Should a **MERIDIEN** run into a time limit on your cluster or crash, you can simply restart it. For this purpose, first click in the main window of the SPHIRE GUI the button **MERIDIEN** on the left, then the **3D Refinement Restart** button within the **ALTERNATIVES** subsection in the middle and specify the **MERIDIEN** run directory (**Refine3D**). Within the specified directory, the program will automatically identify, the last fully completed iteration automatically and continue from there.*

NOTE: *It is possible to change some of the refinement parameter values upon a restart, but we do not encourage experimenting with the settings. The reason is that ML refinement is an iterative process and by changing the parameters in the middle of the process, we modify the state of the program and this may have unintended consequences.*

Now we can use the ***Angular Distribution*** utility to produce a graphics file (*.bild*) with the angular distribution of your particles. Click the ***Angular Distribution*** button (**UTILITIES**) in the middle of the SPHIRE GUI. Fill out the following fields:



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

Projection parameters	required	Refine3D/final_params_033.txt
Output directory	required	Refine3d-angdist
File prefix	none	none
Point-group symmetry	c5	c5
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	sp_pipe_angular_distribution	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

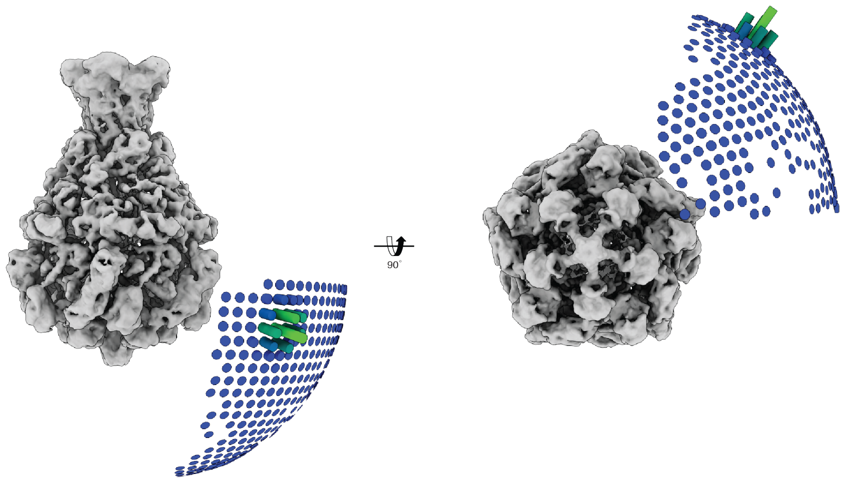
- **Projection parameters file:** Refine3D/final_params_**iternr**.txt
- **Output directory:** Refine3D-angdist

Click the ***Run command*** button and wait for the program to finish.

> **NOTE:** *If you did not set the **Project Settings** described in chapter PROJECT you need to set the values for **Point-group symmetry**, **Pixel size [A] (Advanced)** and **Box size [Pixels] (Advanced)** manually.*

> **NOTE:** *To learn more about angular distribution visualization, visit the wiki page* `http://sparx-em.org/sparxwiki/Euler_angles`.

Wait for the program to finish and then load one of the half volumes (i.e., **Refine3D/vol_0_unfil_iternr.hdf**) together with the resulting *.bild* file into **UCSF Chimera**.



The resulting *VRML model* in **UCSF Chimera** will illustrate the number of particle images assigned to given orientation directions within one asymmetric unit of angular space, as defined by the **Point-group symmetry**. Note this display only contains the distribution of the most probable orientations assigned to the data, as determined by the Maximum Likelihood methodology employed by **MERIDIEN**. Therefore, the picture is somewhat misleading as the actual distribution of angular information may be significantly different. The full information about all orientation parameters is stored in each **Refine3D/mainITERNR/oldparamstructure**. However, at this time we do not have any utilities that would allow the examination of this information in a compact form.

Make sure that the visualized angular distribution is not dominated by just a few directions (especially if you process an asymmetric particle). In such a case, the reconstructed structure will likely have directional artifacts (e.g. elongation in real space).

> **TIP:** *If the data were already subjected to 3D refinement and 3D orientation parameters (shifts and Euler angles) are available, you can use the already determined parameters to start a local refinement (**MERIDIEN/ALTERNATIVES/Local Refinement from Stack**). The local 3D refinement is designed to refine a structure with user-provided orientation parameters taken to initialize the iterative process. This program will do only local searches and it begins with 3D reconstruction of initial reference half-maps.*

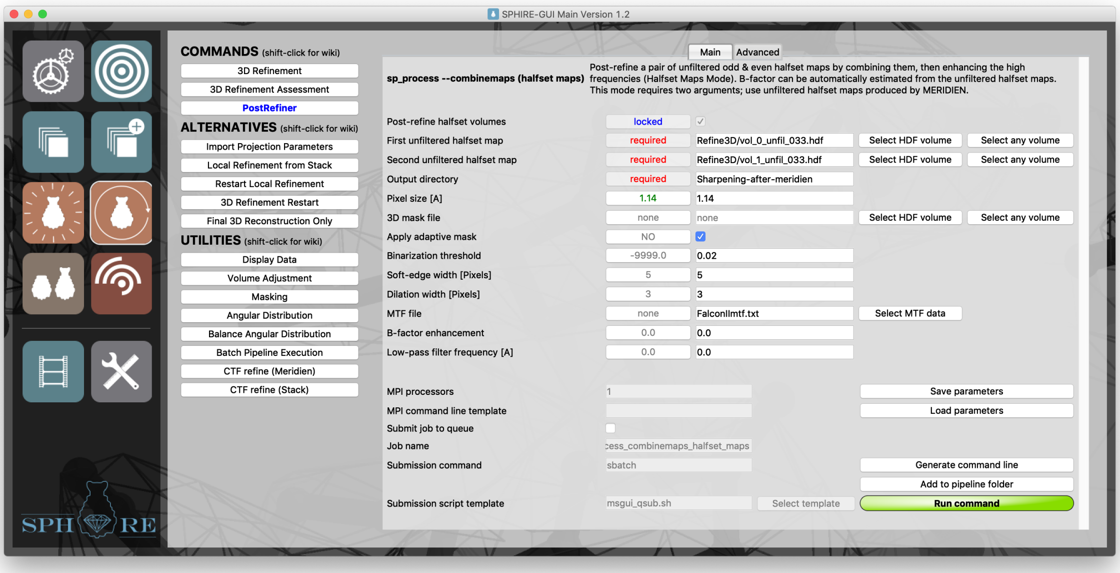
In order to initiate the local refinement (or 3D sorting; see section below), input images should have 3D orientation parameters stored in headers. Alignment parameters from a **MERIDIEN** are imported to the header of the particle using the **Import projection parameters** tool **(Alternatives)**.

> **TIP:** *If you refined your data using **RELION** you can easily convert the files and create a particle stack in BDB file format using the **UTILITIES/RELION to SPHIRE conversion** tool. The particle stack will contain alignment parameters stored in its header and you will be able to directly continue with SPHIRE and to perform a local 3D refinement and/or 3D sorting using the imported data.*

PostRefiner

This utility executes the following sequence of operations: calculation of a soft 3D mask, estimation of the FSC using masked half-maps with FSC rescaling to account for the full-size of the dataset (Penczek 2010), merging of two half-maps, MTF compensation, filtration by the FSC, estimation of the B-factor from the merged map, high-pass filtration of the merged map using a B-factor-parametrized Gaussian function (B-factor is either estimated by the program or defined by the user), and low-pass filtration (either at the estimated resolution or at a cut-off resolution chosen by the user). Some of the operations listed are optional. The output of the utility is a power spectrum-adjusted map both with and without masking being applied. Finally, the program reports the resolution values of FSC@0.5 and FSC@0.143.

In the main window of the SPHIRE GUI first click the button ***MERIDIEN*** on the left, then the ***PostRefiner*** button in the middle and fill out the following input fields:



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	Refine3D/vol_0_unfil_033.hdf
Second unfiltered halfset map	required	Refine3D/vol_1_unfil_033.hdf
Output directory	required	Sharpening-after-meridien
Pixel size [A]	1.14	1.14
3D mask file	none	none
Apply adaptive mask	NO	✓
Binarization threshold	-9999.0	0.02
Soft-edge width [Pixels]	5	5
Dilation width [Pixels]	3	3
MTF file	none	FalconIImtf.txt
B-factor enhancement	0.0	0.0
Low-pass filter frequency [A]	0.0	0.0
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	cess_combinemaps_halfset_maps	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **First unfiltered halfset volume:** Refine3D/vol_0_unfil_**iternr**.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the first final half-volume.*

- **Second unfiltered halfset volume:** Refine3D/vol_1_unfil_**iternr**.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the second final half-volume.*

- **Output directory:** Sharpening-after-meridien
- **Pixel size [A]:** 1.14

- **3D mask file:** none

  NOTE: *Usually we automatically compute a 3D mask, using a 3D adaptive mask procedure (see below). If you want to skip this and use a 3D mask obtained by a different method instead, use the file browser to load your 3D mask file and deactivate the apply adaptive mask option below.*

- **Apply adaptive mask**: YES

  NOTE: *Activation of this option will create a 3D mask directly from the combined half-maps guided by the user-defined parameters listed below.*

  NOTE: *Setting of* ***3D mask file*** *to* ***none*** *and* ***Apply adaptive mask*** *to* ***NO****, will completely deactivate 3D masking. In this case the final output will be power spectrum adjusted but not masked and the FSC will be computed between the two unmasked half-volumes.*

- **Binzarization threshold:** 0.02

  NOTE: *Open one of the half-volumes in* ***UCSF Chimera*** *and visually establish a proper threshold value. The density map should not show any disconnected regions.*

- **Soft edge width [Pixels]:** 5
- **Dilation width [Pixels]:** 3
- **MTF file:** FalconIImtf.txt

  NOTE: *Providing an MTF file name will activate MTF compensation. Click the* ***Select MTF data*** *button, and use the file browser to select the MTF file that comes along with our test data set for the Falcon II detector at 300 kV (stored in the* ***project directory****).*

- **B-Factor enhancement:** 0

  NOTE: *Set to 0, in order to estimate and apply B-factor automatically.*

  Alternative Options: ***-1:*** *Do not apply B-factor.*

  Positive number: *128 (a user specified B-factor of 128 Å^2 will be applied.*

  NOTE: *By default, the program will estimate B-factor in 10 Å to full resolution range. You can adjust the B-factor estimation range in the advanced options.*

- **Low-pass filter frequency [A]:** 0

  NOTE: *Cut-off resolution to filter the final map. Set to 0 to filter the map according to FSC@0.143.*

  Alternative options

  Positive value: *5.8 (low pass filter to 5.8 Å.)*

Click the ***Run command*** button.

Monitor the progress of the job at the terminal through the logfile **log.txt**:

```
tail -f Sharpening-after-meridien/log.txt
```

The output looks like this:

```
2019-05-13 16:14:48 logLine =>---------- >>>Summary <<<------------

2019-05-13 16:14:48 logLine =>Final resolution 0.5/0.143 is 4.14/ 3.43[A]

2019-05-13 16:14:48 logLine =>B-factor is -56.03[A^2]

2019-05-13 16:14:48 logLine =>FSC curves are saved in halves.txt, full.txt, masked_halves.txt,
masked_full.txt

2019-05-13 16:14:48 logLine =>The final volume is Sharpening-after-meridien/vol_combined.hdf

2019-05-13 16:14:48 logLine =>Guinierlines in logscale are saved in Sharpening-after-meridien/guinier-
lines.txt

2019-05-13 16:14:48 logLine =>Tanl low-pass filter is applied using cutoff frequency 1/ 3.43[1/A]

2019-05-13 16:14:48 logLine =>---->>>Analysis of enhancement <<<-----

2019-05-13 16:14:48 logLine =>B_factor : 56.026815

2019-05-13 16:14:48 logLine =>Low-pass filter cutoff : 3.429743[A] (0.332386[absolute])

2019-05-13 16:14:48 logLine =>Low-pass filter falloff : 0.010000[absolute]

2019-05-13 16:14:48 logLine =>Max enhancement point : 115[pixels]

2019-05-13 16:14:48 logLine =>Max enhancement ratio : 33.665805

2019-05-13 16:14:48 logLine =>First zero pw spectrum point : 132[pixels]

2019-05-13 16:14:48 logLine =>Falloff width : 17[pixels]

2019-05-13 16:14:48 logLine =>-----------------------------------------------------

2019-05-13 16:14:48 logLine =>---------- >>>DONE <<<------------
```

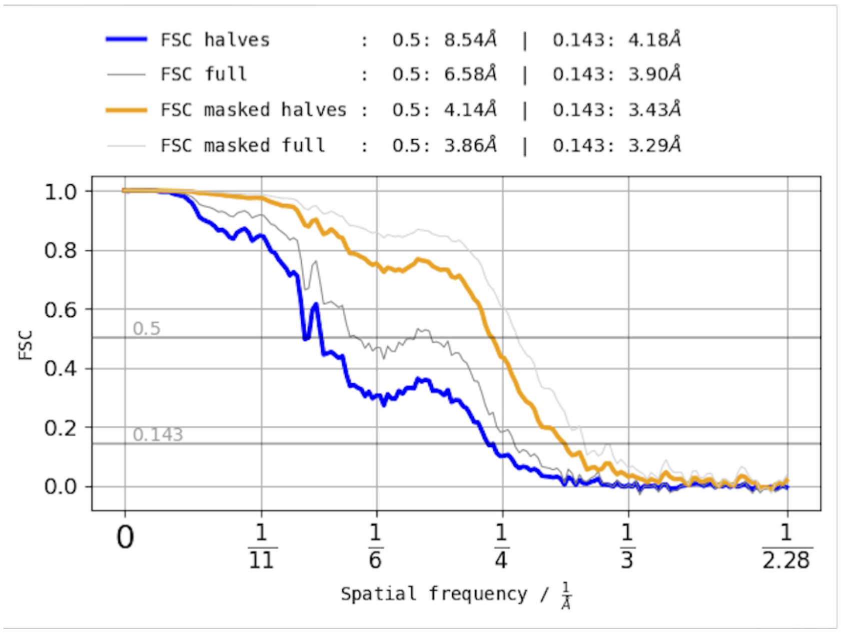
On our Linux cluster, this program job finished after few seconds.

However, if you do not have enough time to wait for the results, you can find the precalculated results for this step in the folder **SphireDemoResults/Sharpening-after-meridien**.

In our case, according to the masked FSC@0.143, the final map has a resolution of 3.43 Å. A B-factor of -56 Å^2 and a low-pass filter at 3.5 Å were thus applied to the map. In the output directory you can now find the following files:

- **vol_combined.hdf:** The merged, sharpened and masked final structure, filtered to the achieved resolution of 3.5 Å.
- **vol_combined_nomask.hdf:** The final structure, with no mask applied.
- **vol_adaptive_mask.hdf:** The soft 3D mask used for the FSC calculations.
- **fsc.png:** A plot of all FSC curves.

The program will also output text files for all FSC curves and the Guinier plot. Display the final structure using **UCSF Chimera** and check the FSC plots carefully (see note below). Compare the EM map with the available X-ray structure.

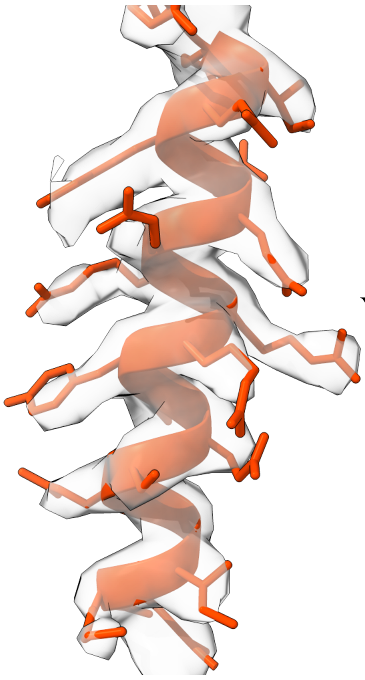


NOTE: *As is well known, the application of a very tight mask may result in an overestimation of resolution. While in general this effect is impossible to avoid (short of not using any mask at all), one has to take precautions to avoid an excessive impact of the mask. Examine the FSC curve carefully. If the FSC curve rises in high frequencies or it never drops below zero, repeat the FSC estimation using a more generous mask. Another telltale sign is emergence of strange B-factor values estimated by the program and/or high-resolution artifacts in the map. Ideally, right after it drops to 0.143, the FSC should oscillate around the zero line. For an in-depth discussion and more examples see (Penczek 2010).*

TIP: *You should always visually inspect the resulting map and confirm that the features of the density agree with the nominal resolution (i.e., a high-resolution map should show clearly discernible side chains). You can also simply download atomic coordinates of an X-ray crystallographic structure, convert them to a pseudo-electron density map using box and pixel size of the map in question (**PDB File Conversion** in **UTILITIES**) and apply a low-pass filter using the same cut-off resolution as the one determined in the post-refinement step. The appearance of surfaces, as visualized by **UCSF Chimera**, should be similar between the EM map and the converted X-ray model. In particular, spurious surface features, strange sharpness of the surface, disconnected pieces, and/or lack of secondary features that at the same time are discernible in X-ray map, set a red flag. Conversely, an exceedingly smooth appearance of the EM map means that either its resolution and/or the B-factor were underestimated.*

IMPORTANT: Scaling of the FSC (**FSC masked full**) is still an experimental feature under development and might report higher resolutions. Check the output structure carefully – report only the resolution according to the FSC@0.143 of the **masked halves** curve.

Results of the 3D refinement highly depend on the quality of the initial model. Therefore, we usually recommend starting a second round of 3D refinement using the newly produced, sharpened volume as the initial reference, as well as the 3D mask obtained by the **PostRefiner** utility. This may improve resolution of the newly refined structure.



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

SORT3D

3D Variability

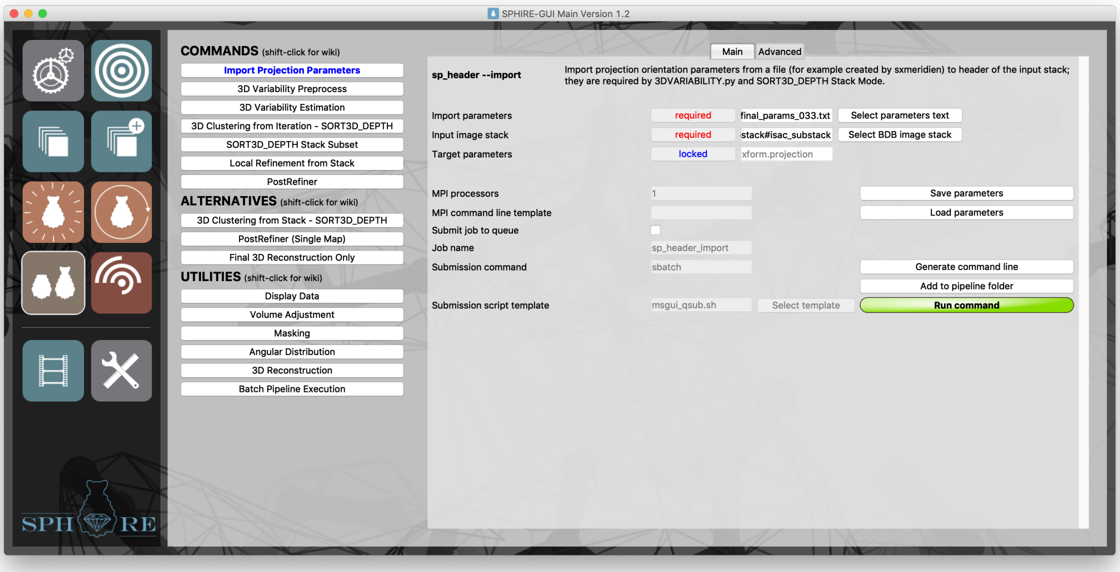
After the 3D structure refinement is completed, we assess possible structural variability of the complex. 3D variability might be caused by conformational changes, substoichiometric ligand binding and/or compositional heterogeneity. In general, regions with higher variability are less reliable and the data may require further analysis. In order to detect such regions, we calculate a 3D variability map using the EM image data and their orientation parameters.

> **TIP:** *Although regions with higher variability typically show fewer details, this step is not done to estimate the local resolution, but it is rather used to guide the subsequent step of 3D sorting. SPHIRE includes a dedicated tool for the local resolution estimation (**LOCALRES**), which is usually performed at the end of the workflow, as described below.*

> **NOTE:** *Ideally, we would want to compute a 3D variance map using the projection data. However, it is not immediately apparent that this is mathematically possible as the data is given in form of projections. Moreover, methods suggested in the literature call for massive calculations. Therefore, in SPHIRE we settle for a good approximation of the variance map, that we termed "variability map".*

First, we will import the orientation parameters of the final refinement iteration (with the highest resolution) to the header of the clean particle stack (the particle stack created after **ISAC2**, that we used as input for the 3D refinement).

In the main window of the SPHIRE GUI click the button ***SORT3D*** on the left and then the ***Import Projection Parameters*** button in the middle:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
Main
Advanced
Select parameters text
Select BDB image stack
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

sp_header --import Import projection orientation parameters from a file (for example created by sxmeridien) to header of the input stack; they are required by 3DVARIABILITY.py and SORT3D_DEPTH Stack Mode.

Import parameters	required	final_params_033.txt
Input image stack	required	stack#isac_substack
Target parameters	locked	xform.projection
MPI processors		1
MPI command line template		
Submit job to queue		☐
Job name		sp_header_import
Submission command		sbatch
Submission script template		msgui_qsub.sh

Fill out the following input fields:

- **Import parameters:** Refine3D/final_params_**iternr**.txt

  NOTE: *Click the **Select parameters text** button, and use the file browser to select the text file with the orientation parameters of the refinement iteration with the highest resolution.*

- **Input image stack:** bdb:Substack#isac_substack

  NOTE: *Click the **Select BDB image stack** button, and use the file browser to select the stack used as input for the high-resolution refinement.*

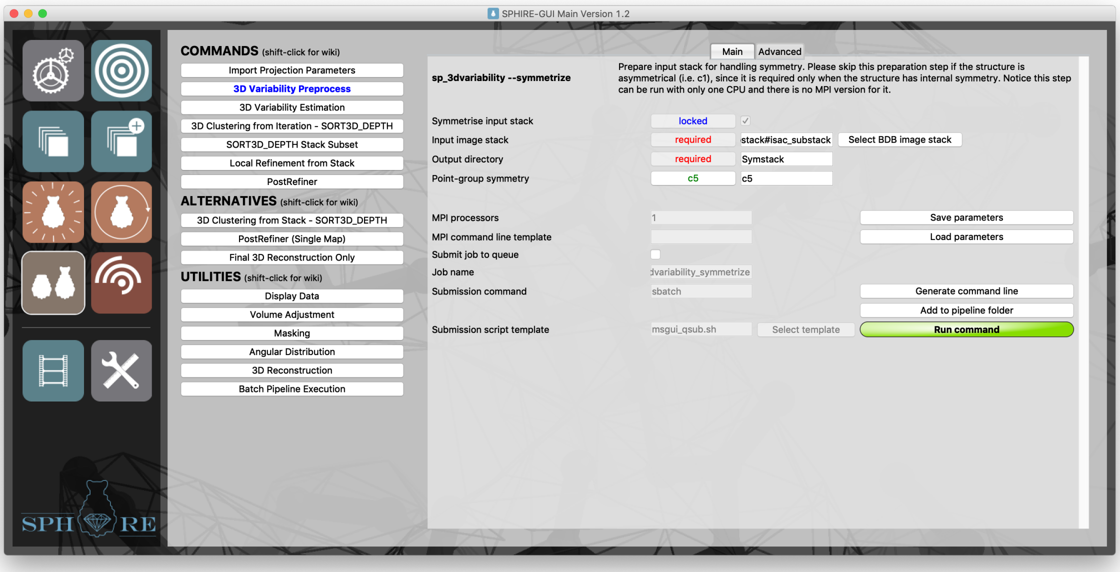
NOTE: *During the 3D refinement, **MERIDIEN** outputs, and uses multiple projection directions (probability-weighted) per particle. However, for the purpose of variability analysis, we use only the highest-probability orientations for each particle.*

Click the ***Run command*** button.

In the next step, we will perform a "symmetrization" of this dataset in order to cover the entire 3D angular space for the subsequent calculation of the 3D variability field.

TIP: *If you are processing an asymmetric particle, please skip this step and proceed directly to the 3D Variability estimation.*

In the main window of the SPHIRE GUI first click the button ***SORT3D*** on the left and then the ***3D Variability Preprocess*** button in the middle:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
Main
Advanced
Select BDB image stack
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

sp_3dvariability --symmetrize

Prepare input stack for handling symmetry. Please skip this preparation step if the structure is asymmetrical (i.e. c1), since it is required only when the structure has internal symmetry. Notice this step can be run with only one CPU and there is no MPI version for it.

Symmetrise input stack	locked	✓
Input image stack	required	stack#isac_substack
Output directory	required	Symstack
Point-group symmetry	c5	c5
MPI processors		1
MPI command line template		
Submit job to queue		☐
Job name		dvariability_symmetrize
Submission command		sbatch
Submission script template		msgui_qsub.sh

Fill out the following input fields:

- **Input image stack:** bdb:Substack#isac_substack

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select the stack used as input for the high-resolution refinement.*

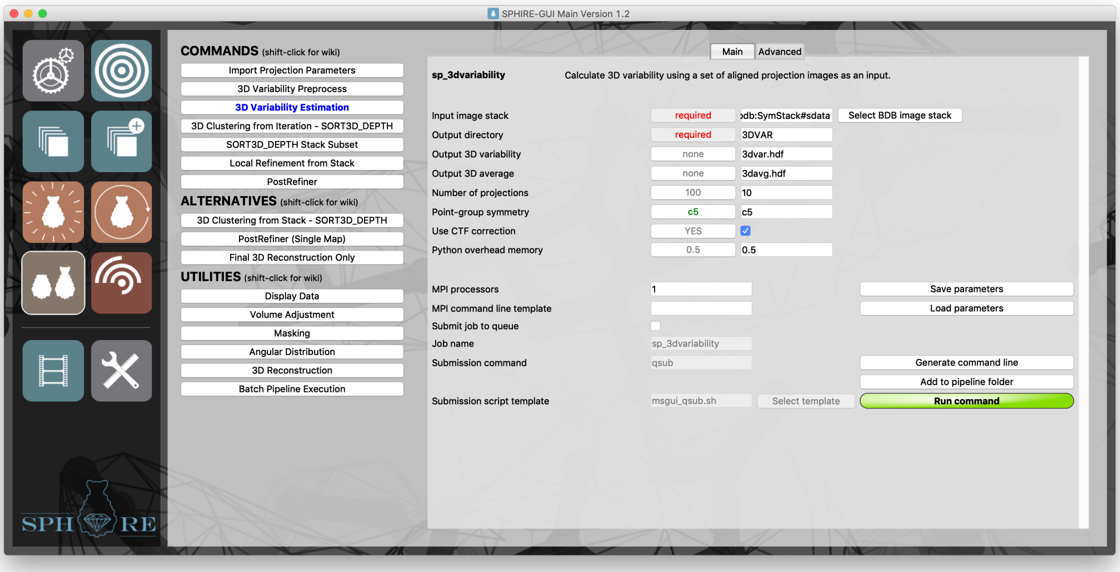
- **Output directory:** Symstack
- **Point-group symmetry:** C5

A symmetrized dataset named **bdb:Symstack#sdata** will then be stored directly in the output directory. (This is a virtual stack, so no excessive amount of disk space is consumed). Use **EMAN2**'s **e2iminfo.py** to check the number of particles in this dataset.

```
e2iminfo.py bdb:Symstack#sdata
```

After "symmetrization" (C5 symmetry is used in our example) this stack should contain five times more projections than your input dataset ($5 \times 10949 = 54745$ particles).

In the main window of the SPHIRE GUI click the button ***3D Variability*** Estimation in the middle:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
Main
Advanced
Select BDB image stack
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

sp_3dvariability Calculate 3D variability using a set of aligned projection images as an input.

Input image stack	required	db:SymStack#sdata
Output directory	required	3DVAR
Output 3D variability	none	3dvar.hdf
Output 3D average	none	3davg.hdf
Number of projections	100	10
Point-group symmetry	c5	c5
Use CTF correction	YES	✓
Python overhead memory	0.5	0.5
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	sp_3dvariability	
Submission command	qsub	
Submission script template	msgui_qsub.sh	

Fill out the following input fields:

- **Input image stack:** bdb:SymStack#sdata

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select the symmetrized stack located in the* ***SymStack*** *project directory.*

  NOTE: *After the successful 3D refinement, projection parameters are now available for each particle. The program will find and borrow its neighbors (the size of the angular neighborhood is user-defined) and will compute average and variance using this grouping. During this tutorial, we will not output these 2D averages and variances, as they consume a large amount of disk space and are not particularly interesting per se. The program will, however, use them to produce a 3D average and variability map (see below). If you do want to output the intermediate 2D results, activate the relevant option in the advanced parameters. Both the averages and variances will have 3D orientation parameters stored in headers, so you can compute 3D maps independently. Note that both averages and variances are CTF-corrected, so they do not have CTF information stored in headers.*

- **Output directory:** 3DVAR
- **Output 3D variability:** 3dvar.hdf

  NOTE: *3D variability map.*

- **Output 3D average:** 3davg.hdf

  NOTE: *3D map computed from neighbor-based 2D averages.*

- **Number of projections:** 10

TIP: *The larger the number, the less noisy the variability map but the lower the resolution. In general, you have to consider here the size of your dataset and the type of heterogeneity you expect and want to visualize and adjust this value accordingly. For larger datasets (more than 100.000 particles), increase the number of projections to 50-100.*

- **Point-group symmetry:** C5
- **Use CTF:** YES

TIP: *If YES, the program will use CTF correction during calculation of both 2D averages and variances. Deactivate this flag for negative-stain data. For negative stain data, variability analysis is more challenging and should be performed only if one expects compositional heterogeneity of large-molecular-weight components or very large and well-defined conformational changes. To further enhance the results, consider restricting analysis of the variability to data extracted from micrograph areas of similar stain thickness.*

Advanced parameters:

- **Image decimate factor:** 0.25

NOTE: *Very large datasets with large box sizes might fail due to insufficient memory. In addition, most conformational changes can be easily detected using a larger pixel size. Therefore, by default, the 2D averages and variances are binned 4x during the calculation of the 3D variability map (**decimate factor** of 0.25). If you wish, however, to detect the variability of high-resolution features (e.g. binding of small ligands) it might be necessary to increase the factor and adjust the pixel size accordingly.*

- **Target image size [pixels]:** 290

NOTE: *To further reduce memory consumption, you can in addition clip the images and use a smaller box size, without changing the pixel size. Make sure, however, that the new smaller box size is still slightly larger than the longest axis of the particle. By default, images are not clipped.*

- **Low-pass filter frequency [1/Pixel]:** 0.0
- **Low-pass filter fall-off [1/Pixel]:** 0.0

NOTE: *In order to suppress noise, before the variability calculation, it is possible to apply a low-pass filter. By default, images are not filtered. Use the resolution pop-up converter (**Use resolution [A]**) to calculate resolution in Å. If you binned your images (image decimate factor option), make sure to enter the new pixel size (original pixel size/decimate factor).*

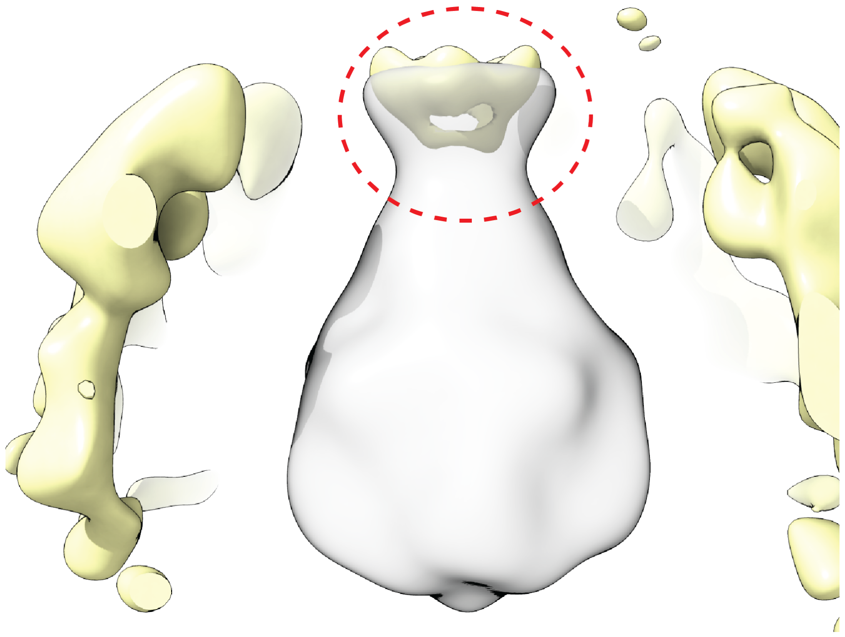
Specify the number of processors (for this job we used 48) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. This process should finish after few minutes (or hours for larger datasets).

The 3D average and variability volumes (**3davg.hdf**, **3dvar.hdf**) are located in the output directory **3DVAR**. Display both maps in **UCSF Chimera**.

Otherwise, to save time, you can copy the precalculated variance and average maps to the **project directory** and display them:

```
cp -Rp SphireDemoResults/3DVAR ./
```

The average map and the simplified variability map are shown in gray and red, respectively. The first observation is the high amount of noise in the variability map.



While there is some variability detected at the channel opening (binding domain of component A, see dashed line), the bulk of our tutorial particle can be considered rather rigid, so the 3D variability analysis yields rather limited information.

Nevertheless, in order to demonstrate how to identify homogeneous subpopulations of particles, we will continue with a 3D heterogeneity analysis in the next section.

3D Clustering

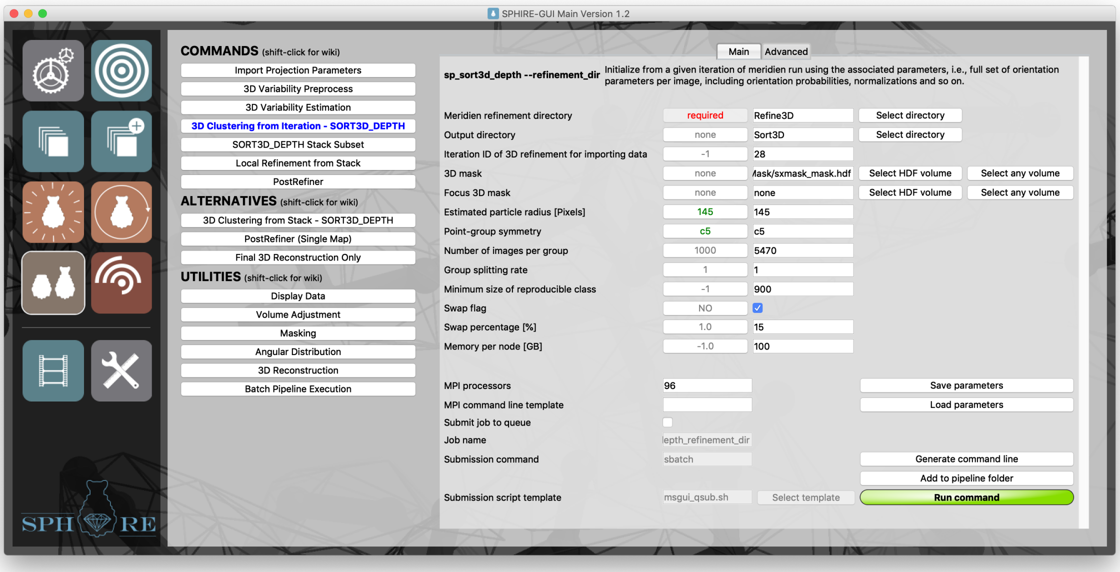
We will now perform a 3D cluster analysis of the 3D refined dataset using the **SORT3D** program. In the protocol adopted in SPHIRE, we separate 3D refinement from 3D sorting of the data. Thus, in a first step we refine the entire dataset using **MERIDIEN** (see previous section) assuming that it is relatively homogeneous, i.e., it does not contain different categories of macromolecular shapes (for example a mixture of ribosomes and proteasomes). In particular, any heterogeneity or flexibility of the analyzed complex should be "secondary" with respect to the overall shape and thus it should make sense to initially treat all projections as belonging to one, quasi-homogeneous dataset. The overall structure derived from **MERIDIEN** refinement should, to a degree, "make sense" and its reprojections should match the reference-free 2D class averages produced by **ISAC2**.

Given these assumptions, we can now proceed with the next step, the clustering, which treats the orientation parameters as constant (i.e., there is no alignment during sorting). In **SORT3D** we employ a variant of the K-means algorithm in which we maintain an approximately equal sizes for all sorted groups. To provide validation of the outcome, the **SORT3D** program employs a strategy of multiple independent random restarts. Thus, the program will actually perform clustering of the data multiple times using different random initializations (different initial structures), compare resulting group assignments and accept results only if they are above the level expected from chance sorting. It follows that the outcome is validated in a sense of assuring the user that the results can be reproduced with reasonable certainty and are not random artifacts. Finally, the program will output the results of statistical ANOVA tests designed to verify that the outcome of sorting is not influenced by parameters that are unrelated to the structure itself and, for example, avoid results that were sorted "by defocus", which regrettably is a relatively common artifact.

> **TIP:** *This step is computationally expensive, and the running time increases significantly with the number of particles and groups.*

> **TIP:** *In other software packages, 3D sorting is sometimes performed to remove "junk" particles. In SPHIRE, most of these particles have already been eliminated by **ISAC2** during 2D clustering.*

In the main window of the SPHIRE GUI first click the button ***SORT3D***, then the button ***3D Clustering from Iteration - SORT3D_DEPTH*** in the middle and fill out the following fields:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
SPHIRE
Main
Advanced
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Run command

sp_sort3d_depth --refinement_dir Initialize from a given iteration of meridien run using the associated parameters, i.e., full set of orientation parameters per image, including orientation probabilities, normalizations and so on.

Meridien refinement directory	required	Refine3D	Select directory	
Output directory	none	Sort3D	Select directory	
Iteration ID of 3D refinement for importing data	-1	28		
3D mask	none	/ask/sxmask_mask.hdf	Select HDF volume	Select any volume
Focus 3D mask	none	none	Select HDF volume	Select any volume
Estimated particle radius [Pixels]	145	145		
Point-group symmetry	c5	c5		
Number of images per group	1000	5470		
Group splitting rate	1	1		
Minimum size of reproducible class	-1	900		
Swap flag	NO	✓		
Swap percentage [%]	1.0	15		
Memory per node [GB]	-1.0	100		
MPI processors	96			
MPI command line template				
Submit job to queue	☐			
Job name	lepth_refinement_dir			
Submission command	sbatch			
Submission script template	msgui_qsub.sh	Select template		

- **Meridien run directory:** Refine3D

  NOTE: *Click the* ***Select directory*** *button, and use the file browser to select the 3D refinement directory.*

  TIP: *The program takes advantage of all of the ML information produced in the 3D reconstruction step* ***(MERIDIEN)*** *and will make use of the probability distributions associated with the individual particle images.*

- **Output directory:** Sort3D
- **3D refinement iteration ID:** 28

  NOTE: *The* ***MERIDIEN*** *iteration number whose alignment parameters will be used by the program.*

  TIP: *The choice of iteration number depends on the goals set by the user. It does not have to be the final iteration and in fact to select it is usually counterproductive as the last few iterations use a larger window size and, which results in longer runtimes of* ***SORT3D****. For the same reason one should avoid iterations with large smear values. Also, early iterations should not be used, including early* ***RESTRICTED*** *ones as the structure is not yet sufficiently formed to permit sensible sorting.*

- **3D mask:** 3DMask/sp_mask_mask.hdf

NOTE: *This is the global mask used for the sorting procedure. It should be always soft-edged. For the tutorial dataset, the mask previously used in* ***MERIDIEN*** *3D refinement is a good choice. Click the* ***Select HDF volume*** *button, and use the file browser to select the soft 3D mask used during refinement. However, we prefer to use a more generous mask here (not the 3D mask obtained during sharpening), because the heterogeneity revealed by the 3D variability analysis was not resolved in the refined map. In general, tight masks should be avoided as one may inadvertently mask out an as yet undetected region of interest.*

TIP: *When processing your own data display the 3D variability map, the refined map and the 3D mask together jointly in* ***UCSF Chimera****, to confirm that the soft 3D mask encloses the protein map completely (particularly any dynamic components that are not yet resolved in the refined map, but clearly present in the variability map).*

- **Focus 3D mask:** none

  NOTE: *A binary focus mask. It is used to perform focused clustering. It allows the user to focus sorting on a particular region of a 3D structure e.g. a flexible region, as discussed above. In the case of the tutorial dataset, since the 3D variability map did not reveal any local heterogeneity (e.g. substoichiometric ligand binding, a flexible domain, etc.), we will not conduct any focused classification and thus set this input to "none".*

  TIP: *If the variability analysis indicates a clear local flexibility in the map, use the* ***Masking*** *utility and the* ***3D Variability map*** *as input****,*** *to create a focus mask for* ***SORT3D.*** *The focus mask has to be a* ***binary*** *mask; therefore set the* ***Soft edge width*** *option (****Masking****) to 0.*

- **Number of images per group:** 5470

  NOTE: *The entered value will be used to determine the initial number of groups, as given by the following relation:*

$$\textit{Total Number of Particles} = \frac{10949}{\textit{Particles per group}} = 5470 \approx 2\,.$$

  However, the ultimate number of groups determined by the program depends on the heterogeneity of the dataset and may be either larger or smaller than the initial number. In particular, if there is no reproducible clustering of the dataset, given user-settings, the program will terminate and only return a single group.

  The program will try to maintain groups of the size given by this parameter. However, the ultimate group sizes will not necessarily be equal to the value used and can differ significantly. In general, larger values will result in larger group sizes.

  IMPORTANT: The desired number of images per group has to be larger than the minimum group size set (the next parameter).

> **TIP:** *Adjust this parameter group size based on the size, resolution, and quality of your dataset, the available computational resources, and the goal of* ***SORT3D****. The runtime will increase linearly with the number of groups.*

- **Minimum size of reproducible class:** 900

  > **NOTE:** *The clusters determined by the program will have at least the number of images determined by the value of this parameter. Smaller candidate clusters will be eliminated. A minimum group size cannot be too small (for example 10), as it is impossible to compute a valid 3D structure from such a small number of images. We recommend setting it to a large value without exceeding the desired group size.*

- **Swap flag:** YES
- **Swap percentage [%]:** 15
- **Memory per node [GB]:** 100

Specify the number of processors (for this job we used 96) and submit the job to the queuing system of your cluster using an appropriate submission script by clicking the ***Run command*** button.

You can monitor the progress of the program by following the information that appears in the logfile **log.txt**.

```
tail -f Sort3D/log.txt
```

On our cluster, this job finished in about 75 minutes. Otherwise, to save time, copy the precalculated results to the **project directory** and continue with those. In this case, type:

```
cp -Rp SphireDemoResults/Sort3D ./
```

The **SORT3D** results are reported at the end of the logfile (**log.txt**) saved in the output directory. For each determined group, the printout states group ID, number of images assigned to the group, group reproducibility, group random reproducibility (based on Monte Carlo sampling), the standard deviation of the group random reproducibility, the name and the location of the group selection file, and finally the location and name of the map file of this group. The logfile also contains information about the total number of images, the number of images in all groups ("accounted for" images) and the number of the images not assigned to any groups ("unaccounted for" images; the numbers of which is saved in the text file **Unaccounted.txt** in the output directory). What follows are results of statistical ANOVA analysis computed for group defocus values, image norms per group, and the smearing (spread of probabilities per image) per group. These results make it possible to determine whether the determined grouping of the data reflects factors other than structural heterogeneity. Of particular interest here are the defocus values. Should their average values differ significantly between groups, it is quite likely that the apparent structural differences are due to differences in group defocus values, and not due to actual differences in the respective structures.

At the terminal type:

```
cat Sort3D/log.txt
```

```
2019-05-17 16:36:53 logLine =>Final results saved in Sort3D

2018-03-05_18:13:39 =>-----------------------------

2019-05-17 16:36:53 logLine =>Group ID size determined in generation reproducibility random
reproducibility std selection file map file

2019-05-17 16:36:54 logLine =>0 6363 0 90.6 42.7 5.2 Cluster_000.txt vol_cluster000.hdf

2019-05-17 16:36:54 logLine =>1 3929 0 85.7 28.9 5.7 Cluster_001.txt vol_cluster001.hdf

2019-05-17 16:36:54 logLine =>Images 10949 accounted for images: 10292 unaccounted for images:
657

2018-03-05_18:13:53 =>Unaccounted images saved in Unaccounted.txt
```

From the above table we learn that out of 10949 images, the program assigned 10292 images (accounted for) to two groups (groups 0 and 1), containing 6363 and 3929 images respectively. 657 images were not assigned to any groups (unaccounted for images). Due to the intrinsic randomness of the sorting algorithm, the numbers of images per group may slightly differ from one run to another.

The final results table contains the results of internal reproducibility tests. As the program performs independent clustering it is possible to compute the reproducibility of the individual groups obtained in these independent runs, in other words, computing the percentage of images that were assigned to the same group across different runs. Here they are 90.6 % and 85.7 %, respectively. In order to assess whether these reproducibility levels are above what one would expect if the assignments were entirely random, we perform **Monte Carlo** simulation using the group sizes as input. The random reproducibility levels are 42.7 % and 28.9 % while their standard deviations are 5.2 and 5.7, respectively. We set the group rejection threshold at random reproducibility plus its two standard deviations (which approximately corresponds to 5 % significance level) and remove those groups whose reproducibility is below this level. Please note that at this level (and even at three σ level) all obtained groups are significantly more reproducible than what we would obtain if groups were due to random chance assignments, i.e., if in reality there were no groups in the data.

The output logfile also contains the results of the statistical analysis of the possible influence of external parameters that are a source of data heterogeneity on the output of the sorting. We will discuss analysis of variance of average defocus values per group, i.e., we want to determine whether the obtained groups differ by defocus. If that was the case, the sorting result would be in question. For this, the relevant part of the logfile is:

```
2019-05-17 16:37:05 logLine =>ANOVA of defocus

2019-05-17 16:37:05 logLine =>ANOVA F-value Significance

2019-05-17 16:37:05 logLine =>ANOVA 0.00 100.00

2019-05-17 16:37:05 logLine =>

2019-05-17 16:37:05 logLine =>ANOVA: defocus mean of all clusters: 1.571500

2019-05-17 16:37:05 logLine =>ANOVA: Group averages

2019-05-17 16:37:05 logLine =>ANOVA GID N mean std 2019-05-17 16:37:05 logLine =>ANOVA 0 106
1.5715 0.3673

2019-05-17 16:37:05 logLine =>ANOVA 1 106 1.5715 0.3673
```

```
2019-05-17 16:37:05 logLine =>2019-05-17 16:37:05 logLine =>ANOVA Pair-wise tests
2019-05-17 16:37:05 logLine =>ANOVA A B avgA avgB F_value Significance
2019-05-17 16:37:05 logLine =>ANOVA 0 1 1.5715 1.5715 0.000 100.0000
```

According of the results of the overall ANOVA of defocus, the differences are not significant. This is further confirmed by pair-wise tests and therefore we can reject the hypothesis that defocus influenced results of 3D sorting. (Significance should be <= 0.05 in order to reject null hypothesis of equality of defocus means).

In the output directory, you can find following files:

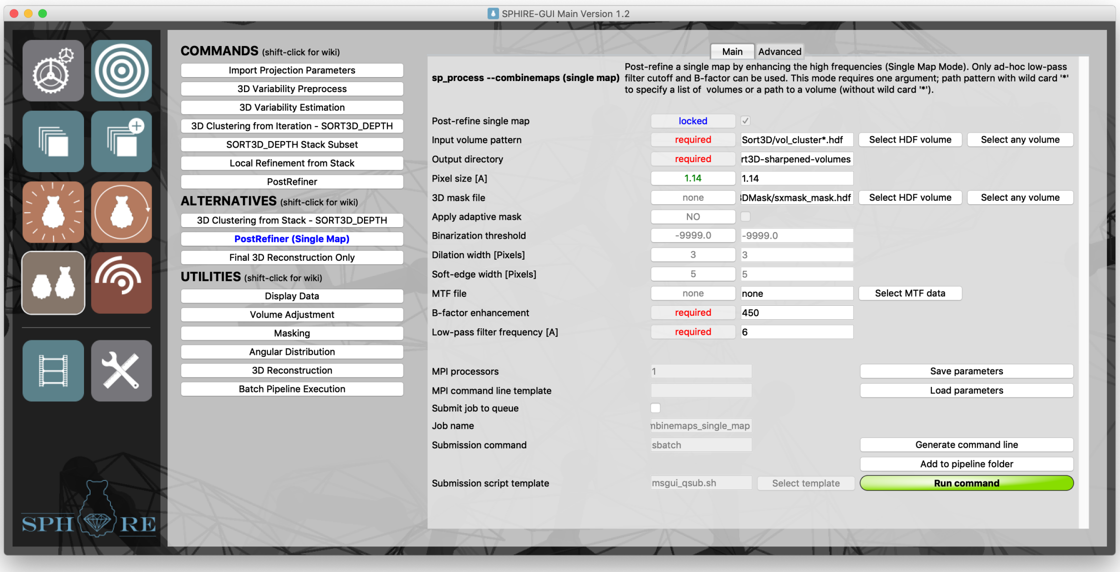
– **Cluster_ITERNR.txt**

  NOTE: *The numbered ClusterITERNR.txt files contain the IDs of the particles assigned to the respective cluster.*

– **vol_clusterITERNR.hdf**

These are the volumes calculated from each cluster of images. Note, however, that these are internal **SORT3D** maps (similar to half-maps stored for every iteration during **MERIDIEN** run in mainITERNR directories), i.e., during 3D reconstruction they were truncated at the maximum resolution of the refined input data. Consequently, they will be featureless if displayed in **UCSF Chimera**. If you wish to bring up some details, you can use the **PostRefiner** utility and apply a large B-factor to these volumes. However, do keep in mind that these volumes are truncated and the final resolution can be obtained only after running a local refinement on the respective substack of particles (see below).

To apply a B-factor on these volumes, click the button ***PostRefiner (Single Map)*** in the middle and fill out the following fields:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
SPHIRE
Main
Advanced
Select HDF volume
Select any volume
Select MTF data
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command

sp_process --combinemaps (single map) Post-refine a single map by enhancing the high frequencies (Single Map Mode). Only ad-hoc low-pass filter cutoff and B-factor can be used. This mode requires one argument; path pattern with wild card '*' to specify a list of volumes or a path to a volume (without wild card '*').

Post-refine single map	locked	✓
Input volume pattern	required	Sort3D/vol_cluster*.hdf
Output directory	required	rt3D-sharpened-volumes
Pixel size [A]	1.14	1.14
3D mask file	none	DMask/sxmask_mask.hdf
Apply adaptive mask	NO	
Binarization threshold	-9999.0	-9999.0
Dilation width [Pixels]	3	3
Soft-edge width [Pixels]	5	5
MTF file	none	none
B-factor enhancement	required	450
Low-pass filter frequency [A]	required	6
MPI processors	1	
MPI command line template		
Submit job to queue		
Job name	nbinemaps_single_map	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **Input volume pattern:** Sort3D/vol_cluster*.hdf

  NOTE: *Click the* ***Select HDF Volume*** *button, and use the file browser to select a cluster volume. Then, replace the variable part of the file name with the wildcard character "*". The program will batch process all cluster volumes using the settings below.*

- **Output directory:** Sort3D-sharpened-volumes
- **Pixel size [A]:** 1.14
- **3D mask file:** 3DMask/sp_mask_mask.hdf

  NOTE: *This is the global mask used for the sorting procedure.*

- **Apply adaptive mask:** NO
- **MTF file:** none
- **B-factor enhancement:** 450

  TIP: *The choice of B-factor is somewhat arbitrary, since enhanced maps are only used for a visual assessment of the sorting outcomes to allow users to decide which, if any, groups should be subjected to subsequent local refinement. The value of B-factor should be adjusted using visual inspection of the outcome. The enhanced map should retain "structural integrity" and the histogram of densities should have an "ice peak" (routinely set to zero by the reconstruction program) at about 0.25 of the histogram range.*

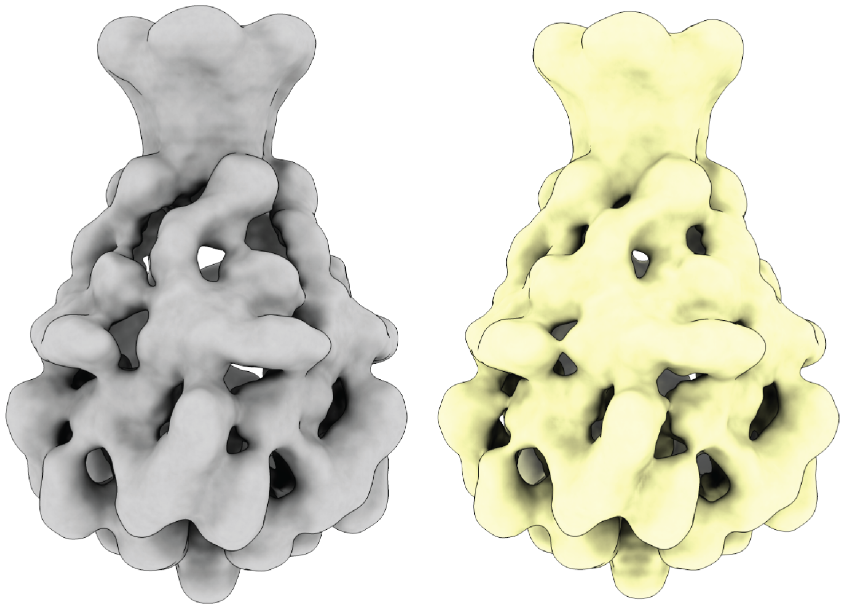
- **Low-pass filter frequency [A]:** 6

  NOTE: *This should be resolution used by the* ***SORT3D*** *program.*

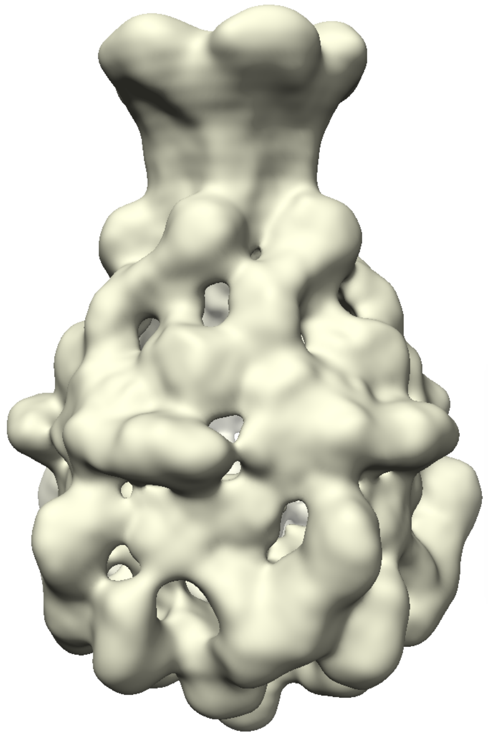
Start the **PostRefiner (Single Map)** job by clicking the ***Run command*** button.

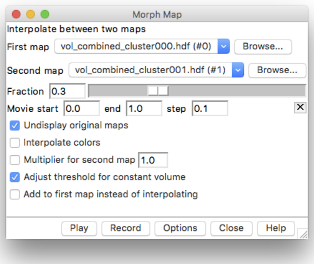
NOTE: *This step (and* ***SORT3D*** *in general) does not output any resolution values or FSC curves.*

Display the resulting maps with **UCSF Chimera**.



At first glance, the maps appear similar. Now use **UCSF Chimera**’s **Morph Map** utility to compare them.





Interpolate between two maps

First map: vol_combined_cluster000.hdf (#0)

Second map: vol_combined_cluster001.hdf (#1)

Fraction: 0.3

Movie start: 0.0 end: 1.0 step: 0.1

[✓] Undisplay original maps

[ ] Interpolate colors

[ ] Multiplier for second map: 1.0

[✓] Adjust threshold for constant volume

[ ] Add to first map instead of interpolating

In the morphed structures, we observe a rather quasi-continuous but clear flexibility of several domains of the protein. The flexibility is localized in the upper-region of the complex: the TcB-binding site and adjacent domains. Further studies revealed that the binding of the component TcB to TcA and the formation of the holotoxin complex stabilizes this structural region. However, such an in-depth analysis of such heterogeneity requires however a much larger dataset and would therefore be beyond the scope of this tutorial.

Keep in mind that the orientation parameters used during sorting relate the individual particle images to the average structure, and consequently, due to superposition of different conformers, the fine details might not be fully resolved at this point. Therefore, the full potential and resolution limit of each determined group can only be revealed after within-group 3D local refinement (see next section Local Subset Refinement and Final Reconstruction).

However, we usually perform a local refinement only for groups whose visual appearance promises delivery of interesting (and biologically interpretable) results. This is the basic outline and, needless to say, the procedure can be more complex in practice. For example, sorting may be applied again within selected groups after their local refinement. In addition, we expect users to compute and analyze 3D variability maps at all stages as well as continuously examine the results of statistical tests done by **SORT3D** in order to avoid accepting any chance groups.

To complete the protocol, it is necessary to use **MERIDIEN** in the local mode to perform a local refinement of all groups of interest independently with the possible additional gain of improved resolution, as now the data subsets have improved homogeneity.

> **TIP:** *If **SORT3D** reveals large conformational changes, every group should be refined independently, and in this case you might additionally want to consider starting the **MERIDIEN** runs of each group with exhaustive searches (mode 1) (instead using the local search mode).*

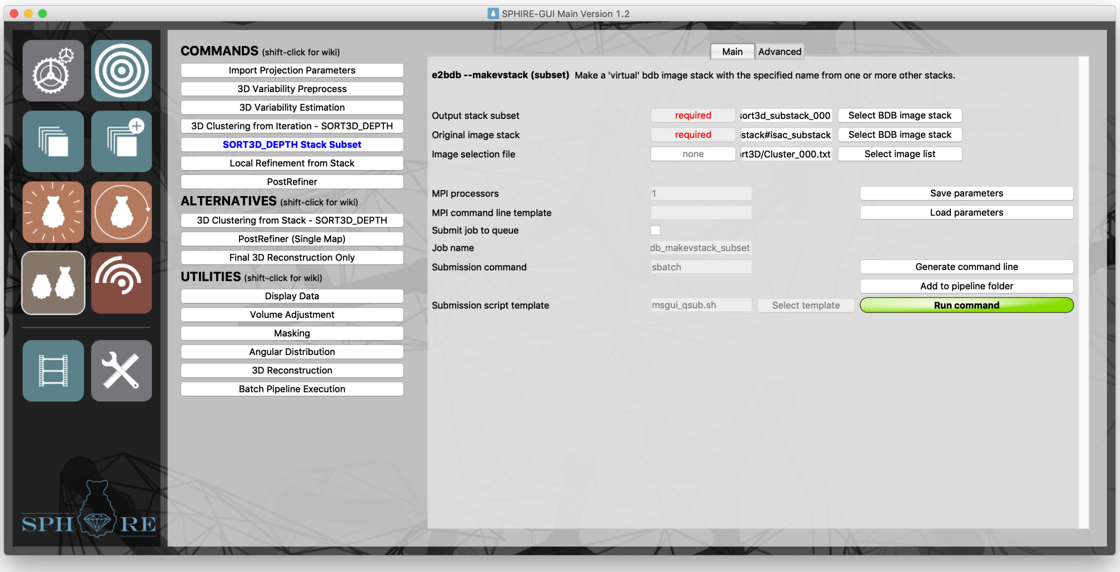
Now, we will create virtual "clean" particle stacks for each sorted cluster that will contain only the particles with IDs included in **Cluster_ITERNR.txt** files produced by **SORT3D**.

> **TIP:** *It is advisable to verify at this point that the stack header is properly populated with the orientation parameters of the **MERIDIEN** run. At the terminal, type:*

```
sp_header.py bdb:Substack#isac_substack --params=xform.projection
--export=try_orientation_parameters.txt
```

> *Should the command crash or produce an empty text file, there are no parameters in the input stack and one should carefully check all the steps of the analysis to find out what went wrong. The parameters should have been imported after **MERIDIEN** run was completed and **SORT3D** was initialized using the **Import Projection Parameters** tool.*

In the main window of the SPHIRE GUI click the button ***SORT3D*** and then the ***SORT3D_DEPTH Stack Subset*** in the middle:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
Main
Advanced
Select BDB image stack
Select BDB image stack
Select image list
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

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

Output stack subset	required	sort3d_substack_000
Original image stack	required	stack#isac_substack
Image selection file	none	rt3D/Cluster_000.txt
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	db_makevstack_subset	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Next, fill out the following input fields:

- **Output stack subset:** bdb:Substack#sort3d_substack_000
- **Original image stack:** bdb:Substack#isac_substack

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select in the* ***Substack*** *folder the virtual stack used as input for the previous* ***SORT3D*** *run.*

- **Image selection file:** Sort3D/Cluster_000.txt

  NOTE: *Click the* ***Select image list*** *button, and use the file browser to select the file with the ID values of all particles accounted for by* ***SORT3D****.*

  Click the ***Run command*** button.

  This will create a virtual stack containing all particles of **Cluster_000.txt**.

  TIP: *The above command has to be* ***repeated for all groups we intend to do an independent local refinement on****. One has to input the selected* ***Cluster_ITERNR.txt*** *files and set the number* ***bdb:Substack#sort3d_substack_ITERNR*** *accordingly. For the results described here, one should create two different substacks.*

  - *bdb:Substack#sort3d_substack_000*
  - *bdb:Substack#sort3d_substack_001*

  After the job has finished, you can verify the number of particles in the resulting stack using ***EMAN2****'s* ***e2iminfo.py*** *command. At the terminal, type:*

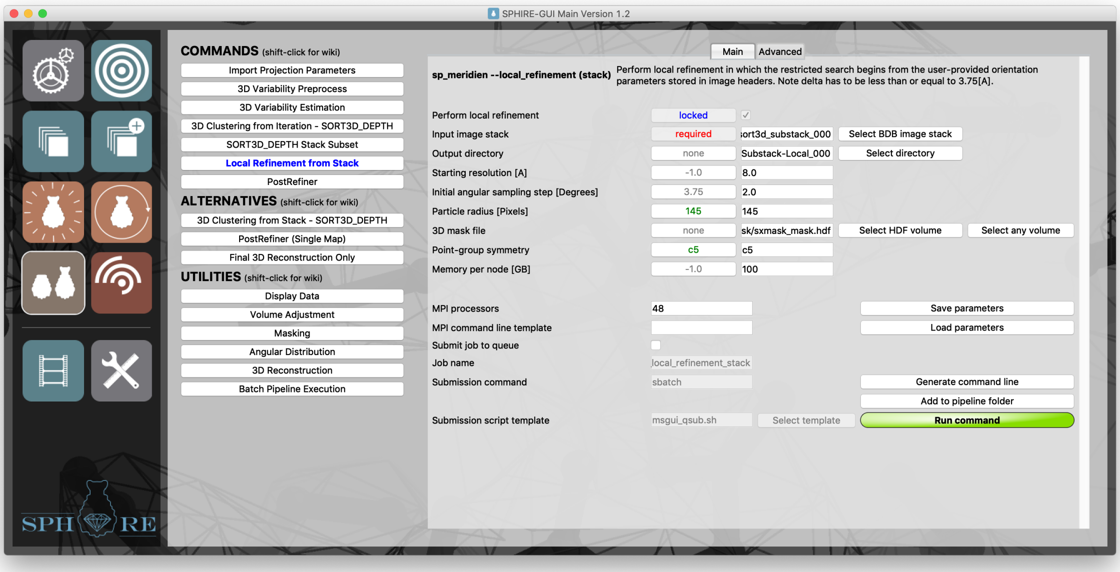
  ```
  e2iminfo.py bdb:Substack#sort3d_substack_000
  e2iminfo.py bdb:Substack#sort3d_substack_001
  ```

The substacks in the precalculated results contain 6363 and 3929 particles, respectively.

Local Subset Refinement and Final Reconstruction

The group substacks created in the previous section contain 3D orientation parameters (recall: these were imported to the full stack at the beginning of the sorting protocol, and thus will be inherited by any virtual substack derived from it), and consequently have already the correct format to serve as inputs for a local 3D refinement by **MERIDIEN**.

In the main window of the SPHIRE GUI click the button ***SORT3D*** on the left and then the ***Local Refinement from Stack*** in the middle and fill out the following input fields:


SPHIRE-GUI Main Version 1.2
COMMANDS (shift-click for wiki)
Import Projection Parameters
3D Variability Preprocess
3D Variability Estimation
3D Clustering from Iteration - SORT3D_DEPTH
SORT3D_DEPTH Stack Subset
Local Refinement from Stack
PostRefiner
ALTERNATIVES (shift-click for wiki)
3D Clustering from Stack - SORT3D_DEPTH
PostRefiner (Single Map)
Final 3D Reconstruction Only
UTILITIES (shift-click for wiki)
Display Data
Volume Adjustment
Masking
Angular Distribution
3D Reconstruction
Batch Pipeline Execution
Main
Advanced
Select BDB image stack
Select directory
Select HDF volume
Select any volume
Save parameters
Load parameters
Generate command line
Add to pipeline folder
Select template
Run command
SPHIRE

sp_meridien --local_refinement (stack) Perform local refinement in which the restricted search begins from the user-provided orientation parameters stored in image headers. Note delta has to be less than or equal to 3.75[A].

Perform local refinement	locked	✓
Input image stack	required	ort3d_substack_000
Output directory	none	Substack-Local_000
Starting resolution [A]	-1.0	8.0
Initial angular sampling step [Degrees]	3.75	2.0
Particle radius [Pixels]	145	145
3D mask file	none	sk/sxmask_mask.hdf
Point-group symmetry	c5	c5
Memory per node [GB]	-1.0	100
MPI processors	48	
MPI command line template		
Submit job to queue	☐	
Job name	local_refinement_stack	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **Input image stack:** bdb:Substack#sort3d_substack_000

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select the substack created after* ***SORT3D****.*

- **Output directory:** Refine3D-Substack-Local_000
- **Starting resolution [A]:** 8.0

> **NOTE:** *Enter the resolution that the program is likely to estimate using the provided initial orientation parameters. The starting resolution here has only a limited impact on the refinement process and it is only used to accelerate the initial 3D reconstruction, the size of which will be limited in order to correspond to the resolution given. If the parameter is omitted, the program will compute full size half-maps to estimate initial resolution. This is time consuming and is in general unnecessary.*

- **Initial angular sampling step [Degrees]:** 2.0

  NOTE: *The initial angular step has to be smaller than 3.75°.*

- **Particle radius [Pixels]:** 145
- **3D mask file:** 3DMask/sp_mask_mask.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the soft-edge 3D mask used during the refinement step. However, if the 3D sorting reveals large conformational changes, you should create a different mask from each **SORT3D** volume to be used as a reference for the local refinement. For this purpose, you can use the **Adaptive 3D Mask** utility.*

- **Point-group symmetry:** C5

  NOTE: *The initial angular step has to be smaller than 3.75°.*

- **Memory per node [GB]:** 100

Advanced parameters:

- **Search range [Pixels]:** 2.25
- **Search step size [Pixels]:** 1.1

  NOTE: *The values of the sampling parameters above depend on many factors, such as the resolution of the initial structure, the kind of refinement that was already performed, where the orientation parameters originated from and, importantly, the goal of the local refinement to be performed. Therefore, we cannot provide default values and it is more than likely that you will have to experiment with different initial settings and adjust them based on the performance of the program and the quality of the result. As a rule of thumb, the initial angular step and search range should be 1.5 × of the values of the **MERIDIEN** iteration used as an input to **SORT3D**.*

Specify the number of processors (we used 48) and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button.

Monitor the progress of the job as discussed in the standard 3D refinement section. On our cluster, this job with 48 processors finished in about 3 hours.

However, if there is no time to wait for the results, copy the precalculated results to the **project directory**.

```
cp -Rp SphireDemoResults/Refine3D-Substack-Local_* ./
```

NOTE: *The above local refinement has to be done for all selected groups. In this case we will refine all 2 groups. Please change the **Input file stack** name and **do not forget** to change the **Output directory** name accordingly. If you have sufficient computational resources, you can run all local refinements simultaneously.*

Once the two local refinements are completed, we apply the **PostRefiner** tool to obtain the power-spectrum adjusted map and report the final resolution of the local refinement. Again, this step has to be done independently for each refined group.

In the main window of the SPHIRE GUI click the **SORT3D** button and then **PostRefiner** button in the middle.

1. Specify the two final half maps (
   Refine3D-Substack-Local_000/vol_0_unfil_ITERNR.hdf
   and
   Refine3D-Substack-Local_000/vol_1_unfil_ITERNR.hdf).
2. Define **Sharpening-after-Meridien-Substack-Local_000** as output directory for **PostRefiner** results of local refinement of **Cluster_000.txt**.
3. Activate the **Apply adaptive mask** Flag.
4. Set the appropriate **Adaptive mask threshold** for the adaptive mask.
5. Use the exact same parameters as in the previous run of **PostRefiner**.
6. Click the ***Run command*** button.

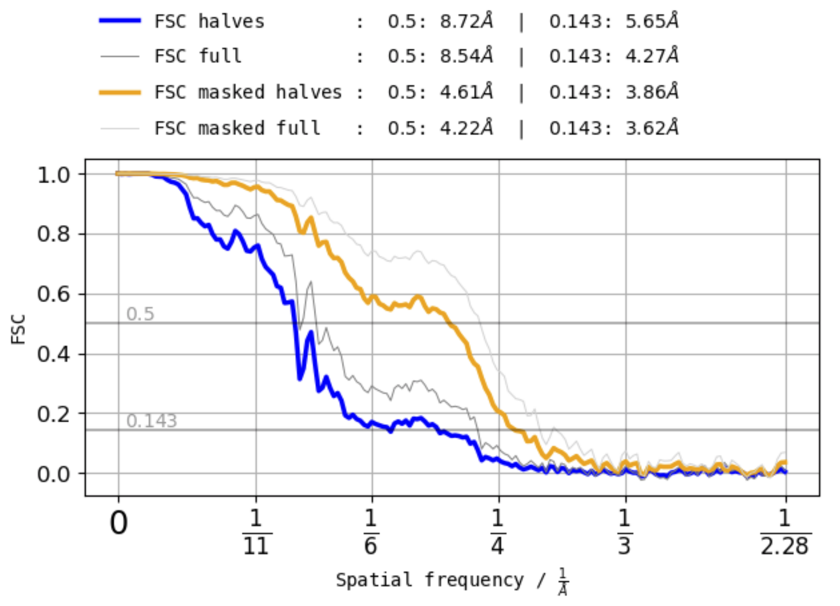
Monitor the progress of the job at the terminal through the logfile **log.txt**:

```
tail -f Sharpening-after-Meridien-Substack-Local_000/log.txt
```

```
2019-05-23 17:16:10 logLine =>---------- >>>Summary <<<------------

2019-05-23 17:16:10 logLine =>Final resolution 0.5/0.143 is 4.61/ 3.86[A]

2019-05-23 17:16:10 logLine =>B-factor is -58.65[A^2]

2019-05-23 17:16:10 logLine =>FSC curves are saved in halves.txt, full.txt, masked_halves.txt,
masked_full.txt

2019-05-23 17:16:10 logLine =>The final volume is Sharpening-after-Meridien-Substack-Local_000
/vol_combined.hdf

2019-05-23 17:16:10 logLine =>Guinierlines in logscale are saved in Sharpening-after-Meridien-
Substack-Local_000 /guinierlines.txt

2019-05-23 17:16:10 logLine =>Tanl low-pass filter is applied using cutoff frequency 1/ 3.86[1/A]

2019-05-23 17:16:10 logLine =>---->>>Analysis of enhancement <<<-----

2019-05-23 17:16:10 logLine =>B_factor : 58.650948

2019-05-23 17:16:10 logLine =>Low-pass filter cutoff : 3.858461[A] (0.295455[absolute])

2019-05-23 17:16:10 logLine =>Low-pass filter falloff : 0.010000[absolute]

2019-05-23 17:16:10 logLine =>Max enhancement point : 102[pixels]

2019-05-23 17:16:10 logLine =>Max enhancement ratio : 26.232149
```

```
2019-05-23 17:16:10 logLine =>First zero pw spectrum point : 117[pixels]
2019-05-23 17:16:10 logLine =>Falloff width : 15[pixels]
2019-05-23 17:16:10 logLine =>------------------------------------------------------
2019-05-23 17:16:10 logLine =>---------- >>>DONE <<<------------
```

Now check the final FSC plot.



The resolution of locally refined group 000 is 3.85 Å according to FSC@0.143 between the two masked half-maps.

Now run the **PostRefiner** tool for the remaining group and check the respective output file. For the precalculated results, the reported resolution of the two groups 6363 and 3929 particles is 3.85 Å and 4.4 Å, respectively.

In our example, although we improved the homogeneity of each substack by 3D clustering, the achieved resolution after local refinement is rather limited by the low number of particles.

You can now display the maps with **UCSF Chimera**, morph them and compare them with the available X-ray structure. However, due to the localized continuous heterogeneity, detailed analysis of this flexibility at near-atomic resolution level of detail would require a much larger dataset.

Advanced Refinement

CTF Refinement

Accurate estimation of the CTF is very critical for high-resolution 3D refinement. Using the highest resolution reconstruction achieved so far as reference (3D refinement with the "clean" particle stack after 2D clustering), we will refine the defocus of each particle in the respective stack and repeat the refinement.

TIP: *If the resolution of your map is worse than 4.5 Å, CTF refinement will most probably not make a significant difference.*

In the main window of the SPHIRE GUI first click the button ***CTF*** on the left and then the **CTF refine (Stack)** button in the middle (Utilities). Fill out the following fields:

- **Input stack path:** bdb:Substack#isac_substack

  NOTE: *Click the **Select BDB Image stack** button, and use the file browser to select the particle stack.*

- **Output directory:** Substack_CTF
- **Path to volume:** Sharpening-after-meridien/vol_combined.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the sharpened, masked 3D volume obtained after sharpening.*

- **Params file:** Refine3D/final_params_035.txt

  NOTE: *Click the **Select parameters text** button, and use the file browser to select the projection parameters file.*

IMPORTANT: Do not combine the results from different refinements.

Submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button.

Monitor the progress of the job through the standard output. At the terminal, type:

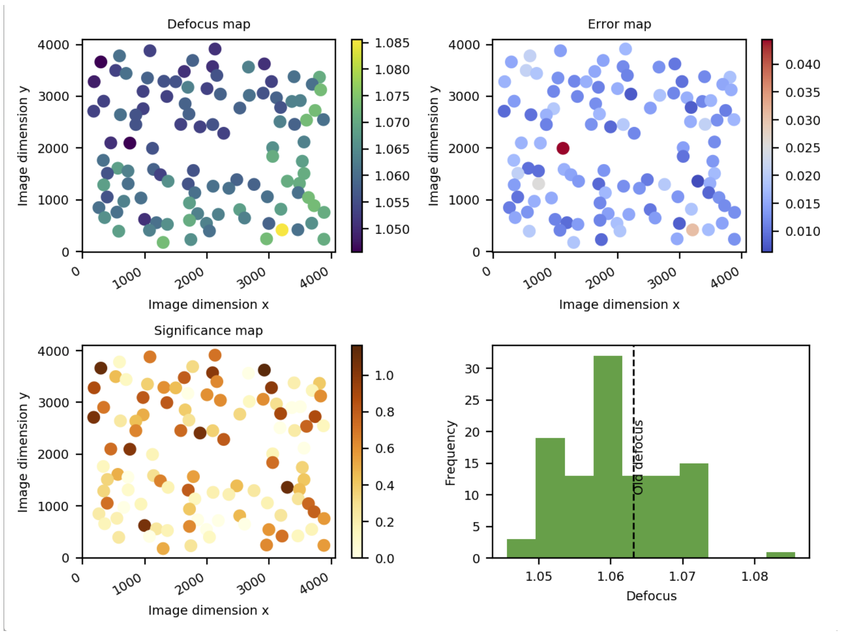
```
tail -f ctf_refine_manual_logfile

####Start refinement####

41%|####1 | 45/109 [09:41<13:35, 12.74s/it]%
```

On our cluster, this job (single process) finished in about 30 minutes. In the output directory Substack_CTF , you can find following files:

– **bdb:ctf_refined:** this is a new stack, containing CTF-refined particles

– **Statistics:** a folder containing additional information about the CTF Refinement.

Within the Statistics/img subdirectory, you can find figures summarizing the statistics of the CTF refinement for each micrograph.



Each image contains four plots:

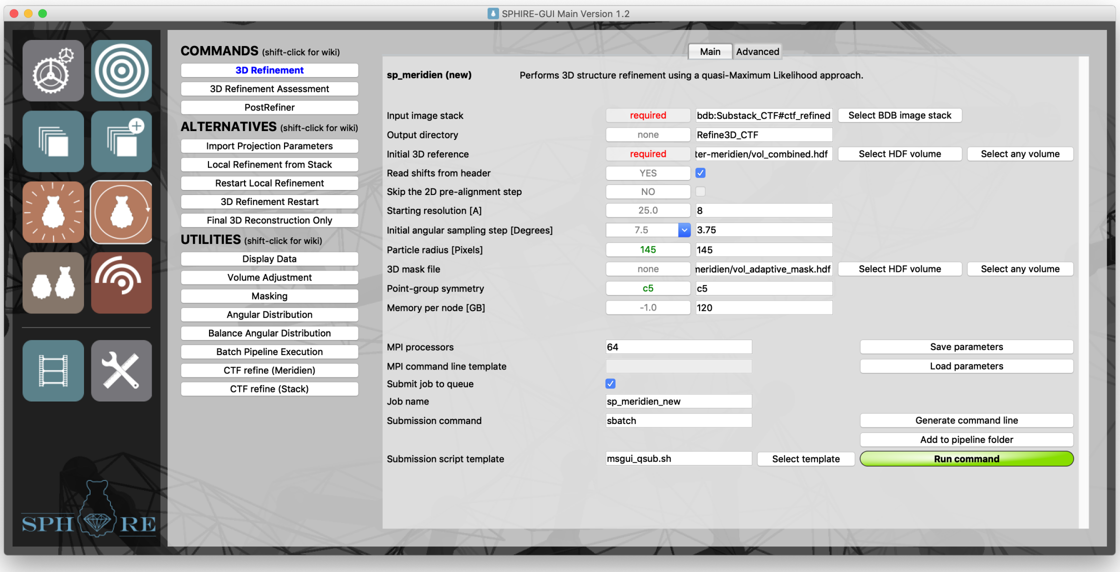
– **Defocus map**: Each dot is a particle while the color represents the estimated defocus value. If your come from a tilted micrograph, you should see a color gradient.

– **Error map:** The color represents the estimated error in terms of standard deviation of the estimated defocus values. We consider an error below 0.05 micrometer as low.

– **Significance map:** Here the color represents the ratio of the difference between the old and the new defocus value divided by the estimated error.

– **Histogram:** The distribution of the refined defocus value for the selected micrographs. The dashed vertical line highlights the previous average defocus value.

We will now use the CTF refined particle stack as the input to start a new round of 3D refinement. In the main window of the SPHIRE GUI click the button ***MERIDIEN*** on the left and then the ***3D REFINEMENT*** button in the middle and fill out the following input fields:



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

Input image stack	required	bdb:Substack_CTF#ctf_refined
Output directory	none	Refine3D_CTF
Initial 3D reference	required	ter-meridien/vol_combined.hdf
Read shifts from header	YES	✓
Skip the 2D pre-alignment step	NO	
Starting resolution [A]	25.0	8
Initial angular sampling step [Degrees]	7.5	3.75
Particle radius [Pixels]	145	145
3D mask file	none	neridien/vol_adaptive_mask.hdf
Point-group symmetry	c5	c5
Memory per node [GB]	-1.0	120
MPI processors	64	
MPI command line template		
Submit job to queue	✓	
Job name	sp_meridien_new	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **Input image stack:** bdb:Substack_CTF#ctf_refined

  NOTE: *Click the* ***Select BDB image stack*** *button, and use the file browser to select the ctf refined subset of particles*

- **Output directory:** Refine3D_CTF
- **Initial 3D reference:** Sharpening-after-meridien/vol_combined.hdf

  NOTE: *Click the* ***Select HDF volume*** *button, and use the file browser to select the 3.4 Å masked, sharpened density map, obtained using the PostRefiner tool.*

- **Read shifts from header:** YES
- **Starting resolution [A]:** 8

  NOTE: *The initial reference is a near-atomic resolution structure obtained by processing these data using a reference-free approach. There is no significant risk for model bias in this case and therefore, we will not apply a very strong low-pass filter to the reference volume.*

- **Initial angular sampling step [Degrees]:** 3.75

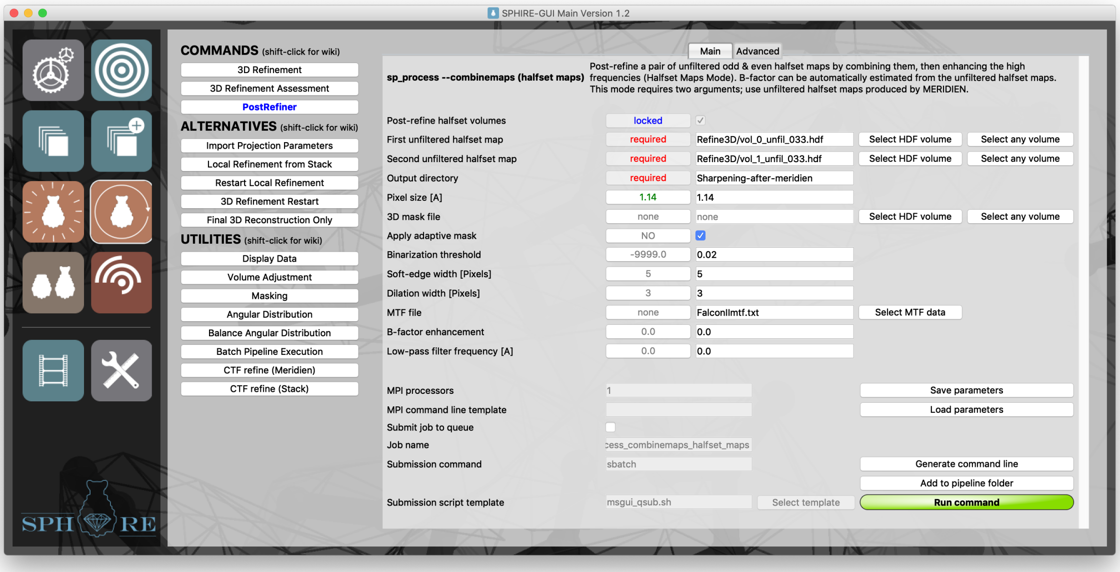
  NOTE: *We will reduce the sampling step from 7.5° to 3.75°. The refinement of symmetric initial models with high resolution symmetry may benefit from a smaller initial angular sampling step.*

- **Particle radius [Pixels]:** 145
- **3D mask:** Sharpening-after-meridien/vol_adaptive_mask.hdf

  NOTE: *This is the 3D mask automatically produced by the PostRefiner tool.*

- **Point-group symmetry:** C5
- **Memory per node [GB]:** 120

Specify the number of processors and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button. On our cluster, this job with 64 processes finished in about 3 hours and 30 minutes Now run the PostRefiner and check if CTF Refinement improved the resulting density map. In the main window of the SPHIRE GUI click the button ***MERIDIEN*** on the left and then the ***PostRefiner*** button in the middle and fill out the following input fields:



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	Refine3D/vol_0_unfil_033.hdf
Second unfiltered halfset map	required	Refine3D/vol_1_unfil_033.hdf
Output directory	required	Sharpening-after-meridien
Pixel size [A]	1.14	1.14
3D mask file	none	none
Apply adaptive mask	NO	✓
Binarization threshold	-9999.0	0.02
Soft-edge width [Pixels]	5	5
Dilation width [Pixels]	3	3
MTF file	none	FalconIImtf.txt
B-factor enhancement	0.0	0.0
Low-pass filter frequency [A]	0.0	0.0
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	cess_combinemaps_halfset_maps	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

- **First unfiltered halfset volume:** Refine3D_CTF/vol_0_unfil_031.hdf
- **Second unfiltered halfset volume:** Refine3D_CTF/vol_1_unfil_031.hdf
- **Output directory:** Sharpening-after-meridien_CTF
- **Pixel size [A]:** 1.14
- **3D mask file:** none
- **Apply adaptive mask**: YES
- **Binarization threshold:** 0.04
- **Soft edge width [Pixels]:** 5
- **Dilation width [Pixels]:** 3
- **MTF file:** FalconIImtf.txt
- **B-Factor enhancement:** 0

- **Low-pass filter frequency [A]:** 0

Click the ***Run command*** button.

and monitor the progress of the job at the terminal through the logfile **log.txt**:

```
2019-05-20 10:16:32 logLine =>---->>>Analysis of enhancement <<<-----

2019-05-20 10:16:32 logLine =>B_factor : 56.616364

2019-05-20 10:16:32 logLine =>Low-pass filter cutoff : 3.459310[A]

2019-05-20 10:16:32 logLine =>Low-pass filter falloff : 0.010000

2019-05-20 10:16:32 logLine =>Max enhancement point : 114[pixels]

2019-05-20 10:16:32 logLine =>Max enhancement ratio : 33.879259

2019-05-20 10:16:32 logLine =>First zero pw spectrum point : 131[pixels]

2019-05-20 10:16:32 logLine =>Falloff width : 17[pixels]
```

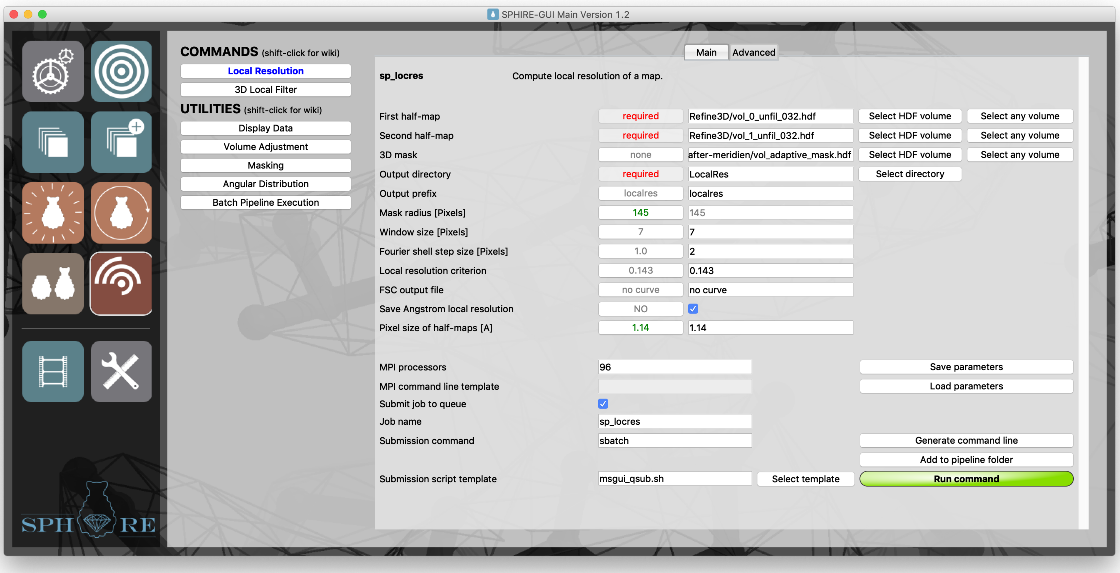
In this case, the CTF Refinement did not have a significant effect on the final resolution in this particular case, most likely due to the very limited number of particles.

LOCALRES

Local Resolution

To interpret a cryo-EM density map properly, it is necessary to know the resolution to which the reliable structural information extends. The highest resolution we achieved so far was 3.4 Å FSC@0.143, and was obtained after 3D refinement, using a dataset cleaned in 2D by **ISAC2**. This value, however, represents an average resolution for the entire map. In addition, due to the intrinsic flexibility, as revealed by 3D clustering using **SORT3D**, and image processing artifacts, the resolution is likely to vary locally. Using the **LOCALRES** tool, we will now compute the local resolution of our final map.

In the main window of the SPHIRE GUI click the button ***LOCALRES*** on the left and then the ***Local Resolution*** button in the middle.



sp_locres   Compute local resolution of a map.

First half-map	required	Refine3D/vol_0_unfil_032.hdf
Second half-map	required	Refine3D/vol_1_unfil_032.hdf
3D mask	none	after-meridien/vol_adaptive_mask.hdf
Output directory	required	LocalRes
Output prefix	localres	localres
Mask radius [Pixels]	145	145
Window size [Pixels]	7	7
Fourier shell step size [Pixels]	1.0	2
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
MPI processors	96	
MPI command line template		
Submit job to queue	✓	
Job name	sp_locres	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following fields:

- **First half-volume:** Refine3D/vol_0_unfil_032.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the first final half-volumes.*

- **Second half-volume:** Refine3D/vol_1_unfil_032.hdf

NOTE: *Click the **Select HDF volume** button, and use the file browser to select the second final half-volumes.*

- **3D mask:** Sharpening-after-meridien/vol_adaptive_mask.hdf

  NOTE: *Click the **Select HDF volume** button, and use the file browser to select the soft-edge 3D mask obtained in the **PostRefiner** step.*

  NOTE: *Voxels of the output local resolution volume are assigned a resolution value (in absolute frequency units). The analysis will be restricted to the voxels within the region outlines by the mask. This will speed up the process.*

- **Output Directory:** LocalRes
- **Window size [Pixels]:** 7

  NOTE: *Real space FSC will be performed within small window areas. This parameter defines their size in pixels. The correct size depends on the resolution of the map. Setting the window size too small will result in inaccurate estimations that will vary widely from location to location, whereas too large windows will produce a coarsely sampled resolution map, that most likely will not reflect local resolution variations sufficiently well. We usually use a window size of 7 Å to 10 Å. As an example, if the resolution of a map is 7 Å and the pixel size is 1 Å, the window size should be at least 7 pixels. For a map with nominal resolution of 12 Å and pixel size 2 Å, the window size should be at least 5 pixels.*

- **Fourier shell step size [Pixels]:** 2

  NOTE: *Thickness of the Fourier shell used to focus the resolution estimation in reciprocal space. Setting a larger step size will speed up the process, but also produce a less precise resolution map. Using a small window size might result in inaccurate and usually underestimated FSC values.*

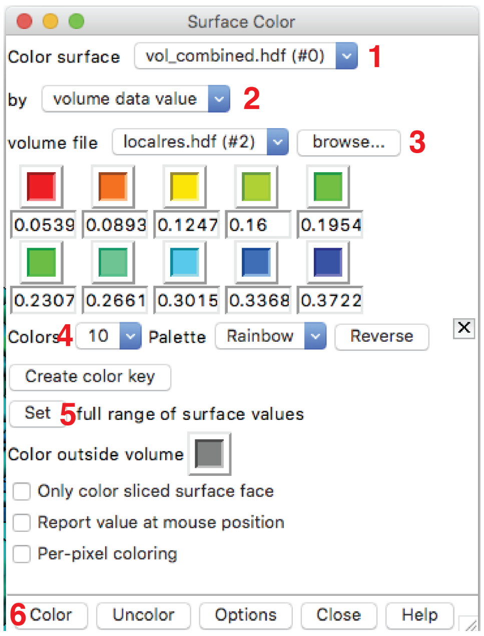
- **Save Angstrom local resolution:** YES

  NOTE: *By default, the program produces a local resolution map with resolution values given in absolute frequency units. Activate this option to obtain a second local resolution map with resolution values given in Å. This map is usually used only for visualization in Chimera (see section below).*

Specify the number of processors and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button.

On our cluster, this job with 96 processors finished in about 2 minutes.

Load the resulting Ångstrom resolution volume (**localres_ang.hdf**) and the sharpened map (**Sharpening-after-meridien/vol_combined.hdf**) into **UCSF Chimera**. Open the **Surface Color** tool available in **UCSF Chimera**.


Surface Color

Color surface: vol_combined.hdf (#0) 1

by: volume data value 2

volume file: localres.hdf (#2) browse... 3

0.0539 0.0893 0.1247 0.16 0.1954

0.2307 0.2661 0.3015 0.3368 0.3722

Colors 4 10 Palette Rainbow Reverse

Create color key

Set 5 full range of surface values

Color outside volume

Only color sliced surface face

Report value at mouse position

Per-pixel coloring

6 Color Uncolor Options Close Help

– (1) Click the **Options Button**. Color surface of volume: **vol_combined.hdf**.

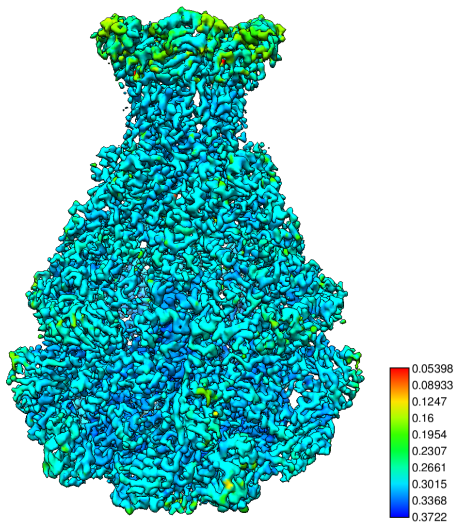
– (2) By volume data value.

– (3) of volume file **localres.hdf**.

– (4) Set number of colors to 10. Select the **Palette Rainbow**.

– (5) Click the **Set** button.

– (6) and then the **Color** button.



TIP: *We strongly recommend comparing the local resolution results with the 3D variability map and the 3D clustering results, obtained in the previous sections.*

TIP: *The values are given in **absolute frequency** units. You can also use the **localres_ang.hdf** map to color the volume ((3)) with values in Å. The distribution of the individual resolution values should match the level of detail of the cryo-EM density map.*

However, if you do not have enough time, you can copy the precalculated local resolution map to the **project directory**.

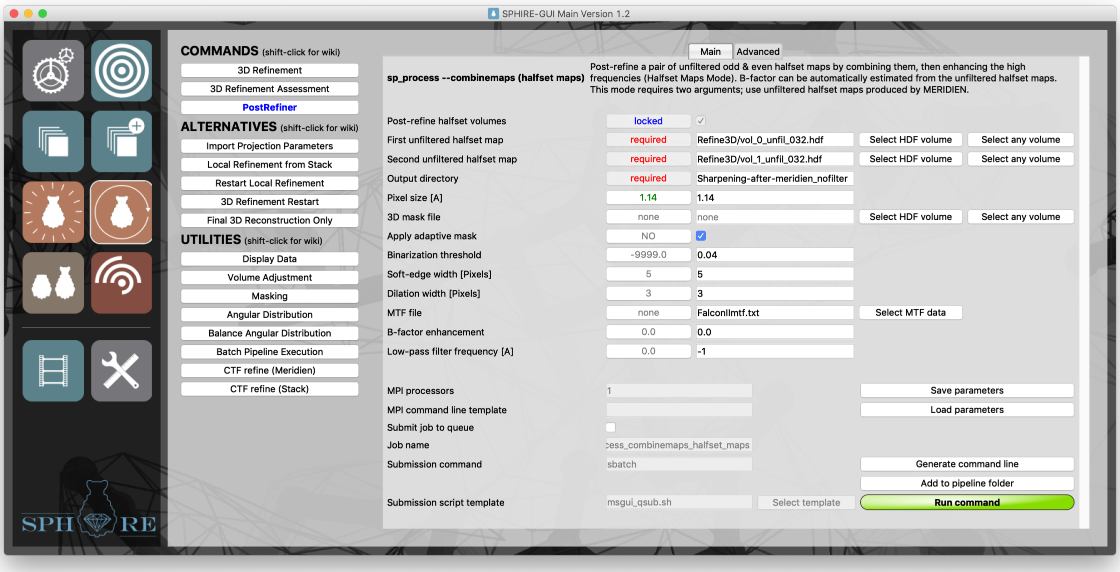
```
cp -Rp SphireDemoResults/LOCALRES ./
```

Local Filtering

After calculating the local resolution map of our final structure, we will now filter the map accordingly. A locally filtered map is more suitable for interpretation and can be used for further analysis (i.e., model building) without the risk of over-interpreting individual features. Poorly resolved regions (corresponding to flexible domains) will be filtered to their respective local resolutions and not to the average (higher) resolution. Similarly, regions better resolved than the average will not be suppressed and local filtering might allow improved molecular modeling in these regions.

However, before applying the local filter, we need to re-run the **PostRefiner** step in order to obtain a masked, sharpened, but **unfiltered** map (the previous map was filtered to the average resolution according to the FSC@0.143).

In the main window of the SPHIRE GUI click the button ***MERIDIEN*** and then the ***PostRefiner*** button in the middle.



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	Refine3D/vol_0_unfil_032.hdf
Second unfiltered halfset map	required	Refine3D/vol_1_unfil_032.hdf
Output directory	required	Sharpening-after-meridien_nofilter
Pixel size [A]	1.14	1.14
3D mask file	none	none
Apply adaptive mask	NO	✓
Binarization threshold	-9999.0	0.04
Soft-edge width [Pixels]	5	5
Dilation width [Pixels]	3	3
MTF file	none	FalconIImtf.txt
B-factor enhancement	0.0	0.0
Low-pass filter frequency [A]	0.0	-1
MPI processors	1	
MPI command line template		
Submit job to queue	☐	
Job name	cess_combinemaps_halfset_maps	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Fill out the following fields:

- **First unfiltered halfset volume:** Refine3D/vol_0_unfil_032.hdf
- **Second unfiltered halfset volume:** Refine3D/vol_1_unfil_032.hdf
- **Output directory:** Sharpening-after-meridien_nofilter
- **3D mask file:** none
- **Apply adaptive mask:** YES
- **MTF file:** FalconIImtf.txt
- **B-factor enhancement:** 0
- **Low-pass filter frequency [A]:** -1

  NOTE: *-1 means no low-pass filtration.*

Monitor the progress of the **PostRefiner** job at the terminal through the logfile **log.txt**:

```
tail -f Sharpening-after-meridien_nofilter/log.txt
```

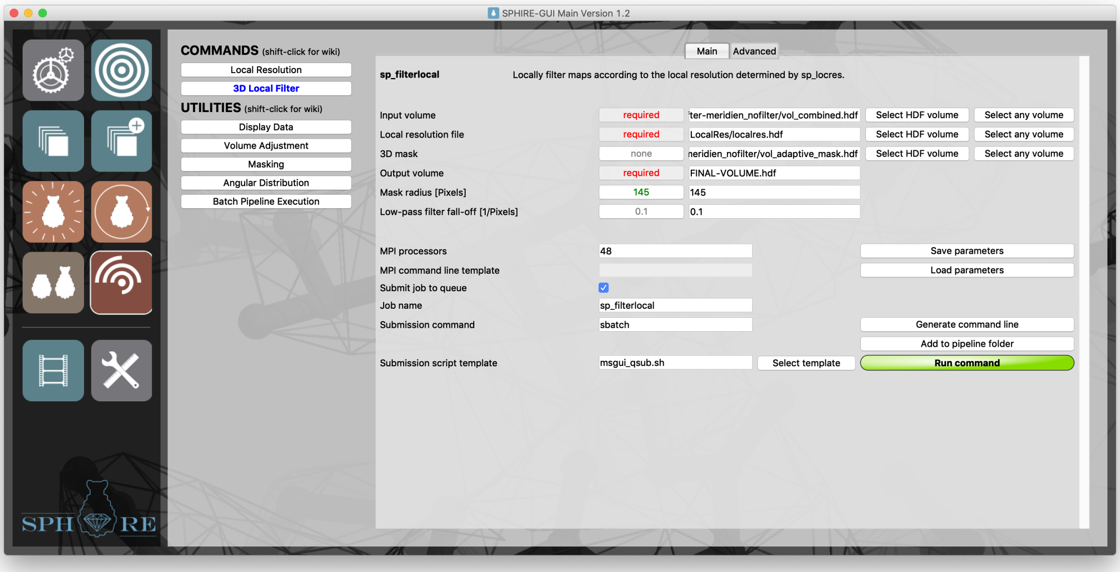
The output will look like this:

```
2019-05-20 14:49:04 logLine =>---------- >>>Summary <<<------------

2019-05-20 14:49:04 logLine =>Final resolution 0.5/0.143 is 4.14/ 3.43[A]

2019-05-20 14:49:04 logLine =>B-factor is -56.03[A^2]

2019-05-20 14:49:04 logLine =>FSC curves are saved in halves.txt, full.txt, masked_halves.txt,
masked_full.txt

2019-05-20 14:49:04 logLine =>The final volume is Sharpening-after-meridien_nofilter/vol_com-
bined.hdf
```

```
2019-05-20 14:49:04 logLine =>Guinierlines in logscale are saved in Sharpening-after-meridien_nofil-
ter/guinierlines.txt

2019-05-20 14:49:04 logLine =>The final volume is not low-pass filtered.

2019-05-20 14:49:04 logLine =>---------------------------------------------------- 2019-05-20 14:49:04
logLine =>---------- >>>DONE <<<------------ 2019-05-20 14:49:04 logLine =>-------------------------------------
```

Now we will apply the local filter to the output volume (**Sharpening-after-meridien_nofilter/vol_combined.hdf**).

In the main window of the SPHIRE GUI click the button ***LOCALRES*** and then the ***3D Local Filter*** button in the middle.



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

Input volume	required	ter-meridien_nofilter/vol_combined.hdf
Local resolution file	required	LocalRes/localres.hdf
3D mask	none	neridien_nofilter/vol_adaptive_mask.hdf
Output volume	required	FINAL-VOLUME.hdf
Mask radius [Pixels]	145	145
Low-pass filter fall-off [1/Pixels]	0.1	0.1
MPI processors	48	
MPI command line template		
Submit job to queue	✓	
Job name	sp_filterlocal	
Submission command	sbatch	
Submission script template	msgui_qsub.sh	

Set the following input fields:

- **Input volume:** Sharpening-after-meridien_nofilter/vol_combined.hdf

  NOTE: *Click the* ***Select HDF volume*** *button, and use the file browser to select the sharpened but unfiltered map created in the previous step.*

- **Local resolution file:** LocalRes/localres.hdf

  NOTE: *Click the* ***Select HDF volume*** *button, and use the file browser to select the local resolution volume with resolution values given in* ***absolute frequency*** *units. Do* ***not*** *use here the localres_ang.hdf file here (local resolution volume with resolution values in Å.*

- **3D mask:** Sharpening-after-meridien_nofilter/vol_adaptive_mask.hdf

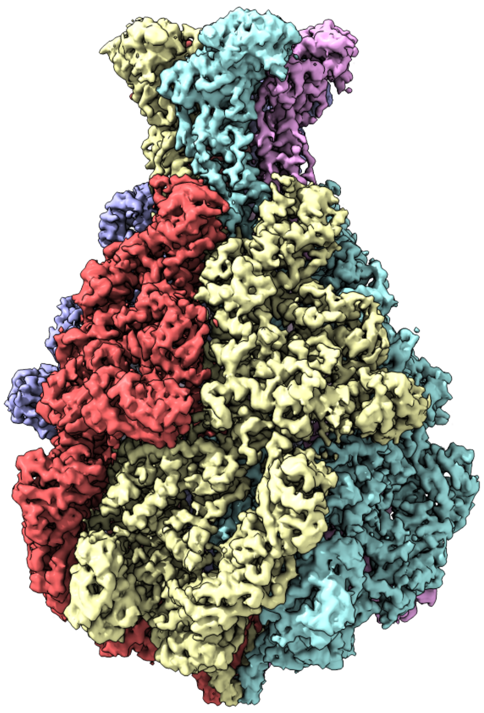
  NOTE: *Click the* ***Select HDF volume*** *button, and use the file browser to select the soft-edge 3D mask obtained in the previous step.*

- **Output volume:** FINAL-VOLUME.hdf

Specify the number of processors (for this job we used 48), and submit the job to the queuing system of the cluster using an appropriate submission script by clicking the ***Run command*** button.

On our cluster this job finished in about 3 minutes.

Display the final map with **UCSF Chimera**. Compare the density with the available X-ray structure.



Congratulations

You have finished the tutorial successfully.

Now it is time to process your own data.

Acronyms

ANOVA	analysis of variance
BDB	Berkley Data Bank
cryo-EM	cryo-electron microscopy
Cs	Spherical Aberration
CTF	Contrast Transfer Function
EM	Electron Microscopy
FSC	Fourier shell correlation
GUI	graphical user interface
HDF	Hierarchical Data Format
ML	Maximum Likelihood
MPI	Message Passing Interface
MTF	modulation transfer function
PDB	protein data bank
SD	standard deviation
SPA	single particle analysis
SPHIRE	SParx for HIgh-REsolution electron microscopy
XFEG	X-Field Emission Gun
SGE	Sun Grid Engine

Citing SPHIRE

The reference for SPHIRE is in preparation.. Until it becomes available, please cite our recent SPHIRE JoVE video protocol:

Moriya, T., Saur, M., Stabrin, M., Merino, F., Voicu, H., Huang, Z., et al. High-resolution Single Particle Analysis from Electron Cryo-microscopy Images Using SPHIRE. J. Vis. Exp. (123), e55448, doi:10.3791/55448 (2017).

Available from `http://sphire.mpg.de`

and (Penczek et al. 2014), (Yang et al. 2012)

If you have any suggestions to improve this practical guide, do not hesitate to contact the author: christos.gatsogiannis@mpi-dortmund.mpg.de

Bibliography

Gatsogiannis, C., A.E. Lang, D. Meusch, V. Pfaumann, O. Hofnagel, R. Benz, K. Aktories, and S. Raunser (2013). “A syringe-like injection mechanism in Photorhabdus luminescens toxins”. In: *Nature* 495, pp. 520–523. DOI: 10.1038/nature11987.

Gatsogiannis, Christos, Felipe Merino, Daniel Prumbaum, Daniel Roderer, Franziska Leidreiter, Dominic Meusch, and Stefan Raunser (2016). “Membrane insertion of a Tc toxin in near-atomic detail”. In: *Nature Structural and Molecular Biology* 23, pp. 884–890. DOI: 10.1038/nsmb.3281.

Grant, T. and N. Grigorieff (2015). “Measuring the optimal exposure for single particle cryo-EM using a 2.6 Å reconstruction of rotavirus VP6”. In: *eLife* 4, e06980. DOI: 10.7554/eLife.06980.

Meusch, D., C. Gatsogiannis, R.G. Efremov, A.E. Lang, O. Hofnagel, I.R. Vetter, K. Aktories, and S. Raunser (2014). “Mechanism of Tc toxin action revealed in molecular detail”. In: *Nature* 508, pp. 61–65. DOI: 10.1038/nature13015.

Penczek, P.A. (2010). “Resolution measures in molecular electron microscopy”. In: *Methods in enzymology* 482, pp. 73–100. DOI: 10.1016/S0076-6879(10)82003-8.

Penczek, P.A., J. Fang, X. Li, Y. Cheng, J. Loerke, and C.M.T. Spahn (2014). “CTER-rapid estimation of CTF parameters with error assessment”. In: *Ultramicroscopy* 140, pp. 9–19. DOI: 10.1016/j.ultramic.2014.01.009.

Pettersen, E.F., T.D. Goddard, C.C. Huang, G.S. Couch, D.M. Greenblatt, E.C. Meng, and T.E. Ferrin (2004a). “UCSF Chimera?A visualization system for exploratory research and analysis”. In: *J. Comput. Chem* 25, pp. 1605–1612. DOI: 10.1002/jcc.20084.

Pettersen, Eric F., Thomas D. Goddard, Conrad C. Huang, Gregory S. Couch, Daniel M. Greenblatt, Elaine C. Meng, and Thomas E. Ferrin (2004b). “UCSF Chimera—A visualization system for exploratory research and analysis”. In: *Journal of Computational Chemistry* 25, pp. 1605–1612. DOI: 10.1002/jcc.20084.

Tang, Guang, Liwei Peng, Philip R. Baldwin, Deepinder S. Mann, Wen Jiang, Ian Rees, and Steven J. Ludtke (2007). “EMAN2: An extensible image processing suite for electron microscopy”. In: *Journal of Structural Biology* 157, pp. 38–46. DOI: http://dx.doi.org/10.1016/j.jsb.2006.05.009.

Yang, Z., J. Fang, J. Chittuluru, F.J. Asturias, and P.A. Penczek (2012). “Iterative Stable Alignment and Clustering of 2D Transmission Electron Microscope Images”. In: *Structure/Folding and Design* 20, pp. 237–247. DOI: 10.1016/j.str.2011.12.007.