



Scipion Tutorial Series

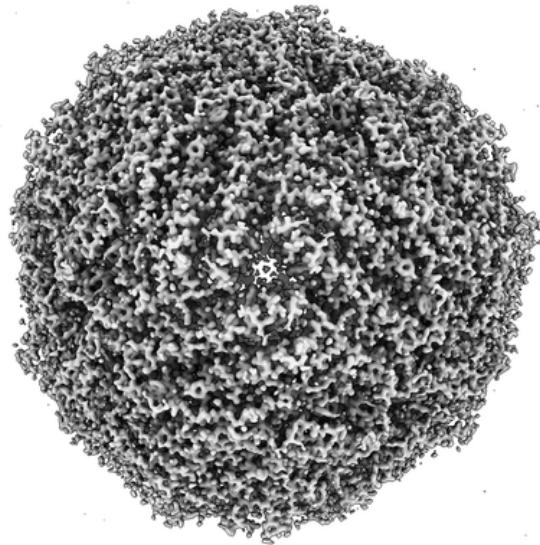
NATIONAL CENTER FOR BIOTECHNOLOGY  
BIOCOMPUTING UNIT

---

## Image Processing

---

December 14, 2020



Cryo-EM map of apoferritin at 1.54 Å resolution (EMD-9865)

CARLOS OSCAR S. SORZANO & DANIEL MARCHÁN

## Revision History

Revision	Date	Author(s)	Description
1.0	11.18.2019	MM	Initial draft created for the S2C2 CryoEM Image Processing Workshop held in Stanford

## Intended audience

The recent rapid development of single-particle electron cryo-microscopy (cryo-EM) allows structures to be solved by this method at almost atomic resolutions. Providing a basic introduction to image processing, this tutorial shows the basic workflow aimed at obtaining high-quality density maps from cryo-EM data by using *Scipion* software framework.

## We'd like to hear from you

We have tested and verified the different steps described in this demo to the best of our knowledge, but since our programs are in continuous development you may find inaccuracies and errors in this text. Please let us know about any errors, as well as your suggestions for future editions, by writing to scipion@cnb.csic.es.

## Requirements

This tutorial requires, in addition to *Scipion*, *cryoSPARC2* (<https://cryosparc.com/>). It is **IMPORTANT** to remark that for executing any *cryoSPARC2* protocol, in the `compute settings` tab, the option `Cache particle images on SSD` has to be **deactivated**.

## Contents

1	Introduction to image processing	4
2	Problem to solve: Apoferritin	6
3	From movies to micrographs	9
4	CTF estimation	14
5	Particle picking	20
6	Extract particles	27
7	2D Classification	29
8	Initial volume	34
9	3D Classification and Refinement	43
10	Volume enhancement and validation	51

# 1 Introduction to image processing

## Definition

Image processing is a structure determination technique that allows to get the 3D density map from a set of cryo-EM images of a particular macromolecule. Although different structural approaches can be followed to analyze the structures of macromolecules, this tutorial focuses on cryo-EM single particle analysis (SPA). Fortunately, cryo-EM SPA is undergoing in this decade a resolution revolution that has allowed the structures of macromolecules to be solved at near-atomic resolution.

## Image processing workflow

The set of successive tasks aimed to get the 3D density map is known as image processing workflow. Main steps of the general workflow are detailed from top (movies) to bottom (refined volume) in the Fig. 1. Some of the tools required to perform the respective tasks are detailed in a non-exhaustive way on the right side of the Figure. All these methods have been integrated in *Scipion* to facilitate interoperability among different software packages, data tracking and reproducibility of results.

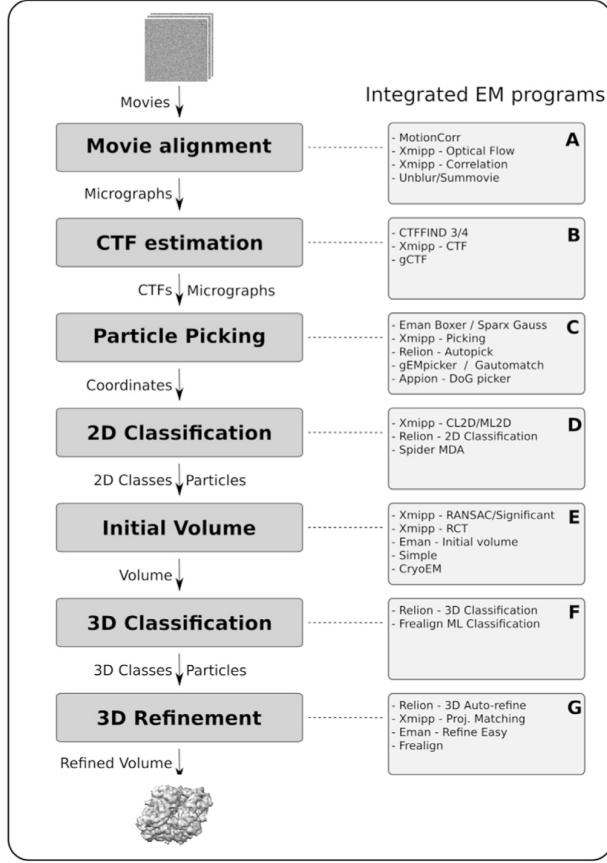


Figure 1: General image processing workflow for SPA (?).

The workflow considers as input the movie frames generated by the microscope. These movies should be global or locally aligned before computing the CTF of individual micrographs. *Scipion* allows to compare different CTF values obtained with distinct algorithms using the CTF consensus protocol. Once the CTF has been corrected, we are ready to extract individual particles of each micrograph by using different protocols of particle picking. As in the case of the CTF, we can retrieve the coordinates of each particle using different protocols of manual and automatic picking, and finally, estimate the agreement between all those methods through a consensus picking protocol. The screened particles are used for further processing. The next step involves the 2D classification of the individual selected particles. 2D classes derived from the last procedure contribute to generate the initial 3D map.

The last part of the workflow includes 3D classification and 3D refinement tasks in order to iterative refine the initial 3D map.

In this tutorial, we show all above mentioned processes of 3DEM processing, as well as the necessary tools to accomplish them, illustrating the combination of different EM software packages in *Scipion*.

## 2 Problem to solve: Apoferritin

Ferritins are iron storage metalloproteins ubiquitously distributed among living organisms. These proteins are involved in iron metabolism in many different types of cells, and play a relevant dual role both in iron detoxification and iron reserve. The ferritin's architecture, similar to a spherical shell, is highly conserved in bacteria, plants and animals, and it allows to accumulate high amounts of Fe(III) atoms (up to 4000 per molecule).

The highly stable iron-free shell is known as apoferritin. Mammalian apoferritins are heteromeric molecules, constituted by 24 monomers structurally equivalent that surround the central cavity. Among these monomers, variable proportions of two types of subunits with different properties, H (heavy) and L (light), can be found. The tissues involved in iron storage contain higher proportion of L chains, whereas the tissues that require higher protection against oxidation, such as heart or brain, have a higher content of H chains. Unlike L chains, H chains display ferroxidase catalytic activity, necessary to oxidize Fe(II) to Fe(III). Concerning the structure of each subunit, it is constituted by 4 long helices, a fifth smaller helix and an additional extended loop. The dinuclear iron site, or ferroxidase site, is located in the center of the four helix bundle.

This tutorial will guide us in the building process of the mouse apoferritin 3D map using the *Scipion* framework (Fig. 2). As starting input data, we are going to use the EMPIAR ID: 10248 data, obtained from mouse heavy chain apoferritin. This cryo-EM data allowed to generate the 3D map EMD-9865 at 1.54 Å resolution

(?). The most recent atomic structure of mouse apoferritin, homo 24-mer of ferritin heavy chain with octahedral symmetry, was also obtained from cryoEM data at 1.84 Å (PDB ID: 6S61). The 24 monomers of this metal binding protein are ligated to 6 Fe(III) and 24 Zn(II) ions.

## Apo ferritin processing workflow in *Scipion*



Figure 2: Apo ferritin processing workflow.

### 3 From movies to micrographs

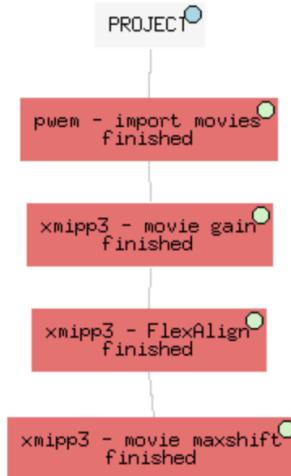


Figure 3: From movies to micrographs workflow.

#### Import movies

The protocol `pwem-import movies` allows to download the mouse apoferritin cryo-EM data in *Scipion*. The protocol form with parameters can be seen in Fig. 4. With this protocol, besides the set of movies, located in the Data folder, acquisition parameters such as accelerating voltage, spherical aberration and sample rate, will be registered in your *Scipion* project.

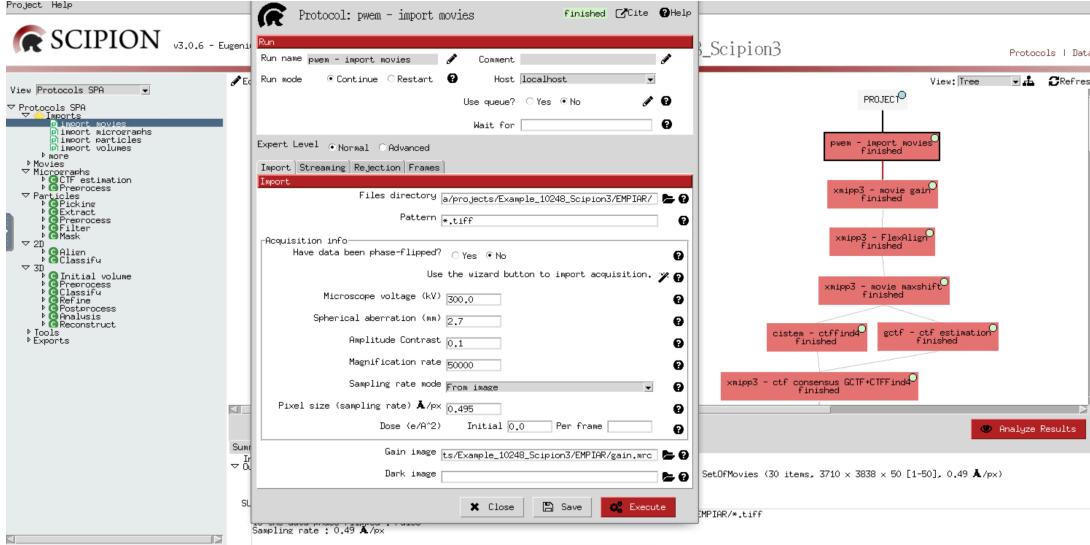


Figure 4: Filling in the protocol to import cryo-EM data.

After executing this protocol, we can visualize the list of 30 movies imported to the project by pressing **Analyze Results**. Each movie contains 50 frames (size 3,710 pixels x 3,838 pixels). Frames contained in each movie can be visualized by right-clicking each entry.

## Computation of movie gain

The protocol **xmipp3-movie gain** is used to compute the movie gain (Fig. 5). Two movie gains will be computed: 1) Without applying the input gain, to orientate the input movie gain; 2) Applying the input gain to estimate the residual movie gain.



Figure 5: Completing the protocol to compute the movie gain.

After executing this protocol, by pressing **Analyze Results** we can visualize the image of the gain computed. None of the movie residual gains computed moves forward the protocol output. The set of movies has now attached the re-oriented input gain.

## Movie alignment

In order to correct BIM-induced image blurring and restore important high resolution information, the stack of individual frames contained in each movie needs to be aligned. Only one image will be generated, and this image is called micrograph. Although in *Scipion* we have integrated several protocols to perform global and local alignment, in this tutorial we are going to use **xmipp3-FlexAlign** that goes on GPUs. We have completed the params of this protocol in Fig. 6.

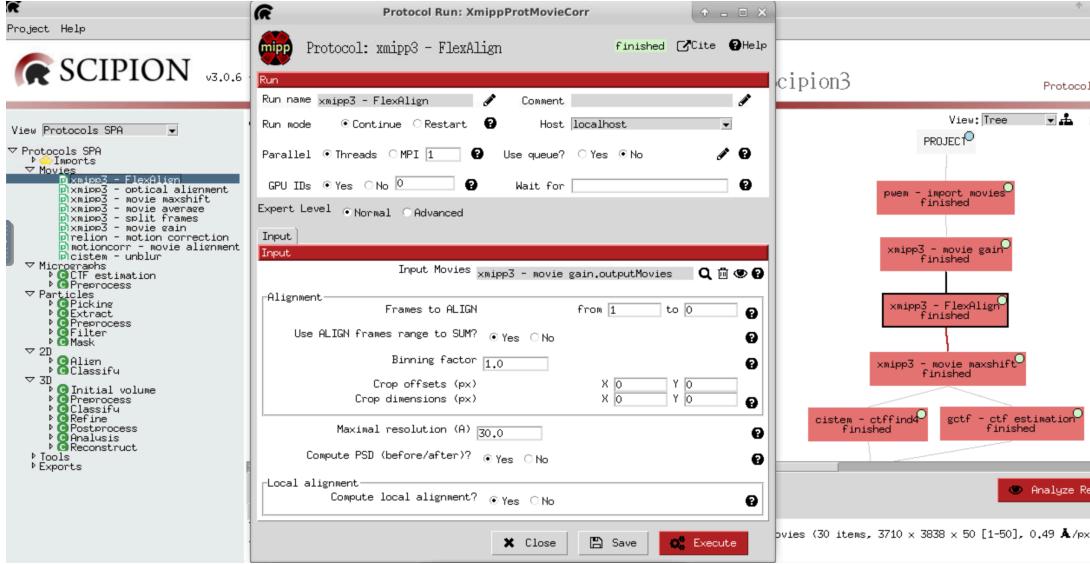


Figure 6: Filling in the protocol to align the frames of each movie.

When the execution of this protocol finishes, we can observe the set of 30 resulting micrographs by pressing **Analyze Results**. The first column contains composite images with half of the PSD of the unaligned micrograph (left side) and half of the PSD of the aligned one (right side). The plots in the second column reflect the estimated shifts of the movie frames. The name of each resulting micrograph appears in the third column. Each micrograph included in the set can be opened for visual inspection by double-clicking or right-clicking its entry.

## Screening of micrographs

Since some of the micrographs generated in the previous step could derive from movies with high drift among frames, we have added in the processing workflow a step to select only the micrographs originated from movies with allowed drifting values among frames. The protocol **xmipp3-movie maxshift**, completed in Fig. 7, was designed to screen micrographs according maximum shift values. Movies will be rejected if they exceed the maximum shift value among frames, the maximum travel value for the whole movie, or both previous conditions.

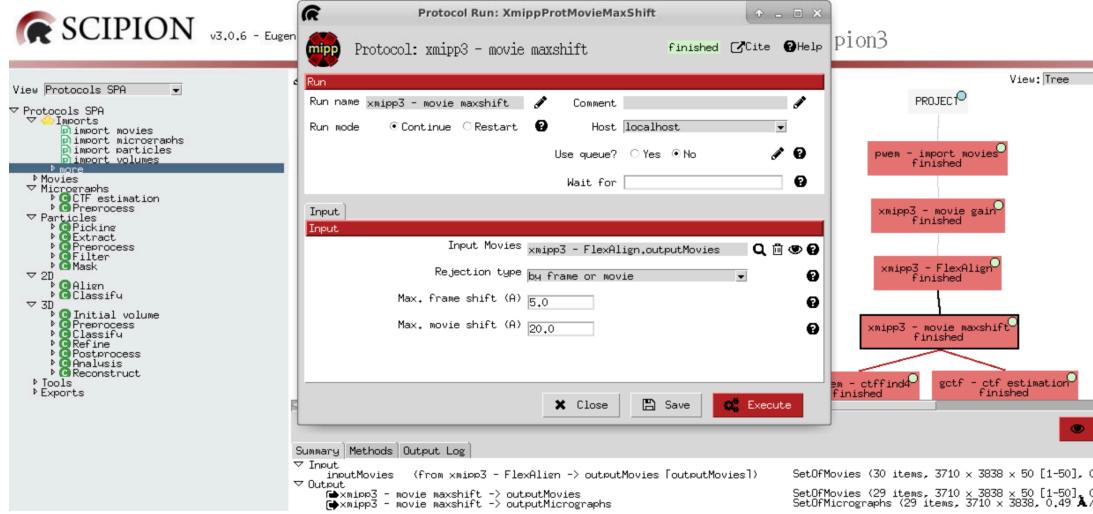


Figure 7: Completing the protocol to screen the micrographs.

Once the protocol is executed, both discarded and accepted lists of micrographs can be visualized by pressing **Analyze Results**. Each micrograph can be opened for visual inspection by double-clicking or right-clicking. In this case, only 1 movie was rejected and the set of 29 micrographs was included in the protocol output. This set will serve as input for further processing steps.

For more information:

- **Video tutorial:** first half of this video [https://www.youtube.com/watch?v=01sBrJbKh7I&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=26](https://www.youtube.com/watch?v=01sBrJbKh7I&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=26).
- **Theoretical lecture:** first half of this video [https://www.youtube.com/watch?v=F3Uslh3v9J8&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=32](https://www.youtube.com/watch?v=F3Uslh3v9J8&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=32).

## 4 CTF estimation

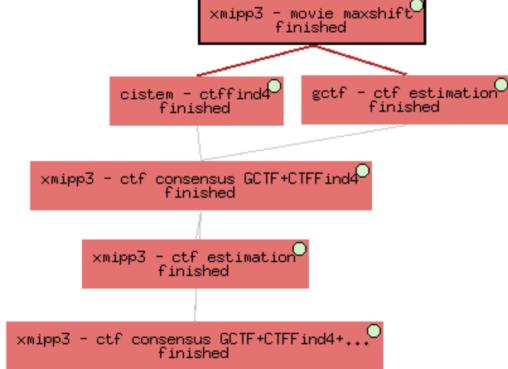


Figure 8: CTF estimation workflow.

Since close to focus images of biological specimens embedded in vitreous ice generate very little contrast, we take our movie-frames out of focus, and retrieve systematically distorted images of our specimens. The alteration observed is due to the different transference of contrast for each frequency. In an ideal microscope all the frequencies are transferred with total contrast (+1). In a normal one, some frequencies are transferred with contrast 0 or even -1. The CTF (Contrast transfer function) indicates how much contrast is transferred to the image as a function of the spatial frequency. The estimation of the CTF is the first step to correct it, repair its negative effect and retrieve our specimens undistorted.

How to estimate the CTF?

Since part of the Fourier components are lost, attenuated or inverted, images of the specimens taken in a non-ideal microscope will appear blurry. We define this blurry effect with the PSF (Point spread function). The effect of the PSF makes that a discrete point in the specimen is reproduced in the image as a broad point with a complex shape. The PSF can be directly estimated from the micrographs and, since the PSF and the CTF are related through the Fourier Transform, by computing the Fourier Transform of the PSF, we can directly estimate the CTF.

We count on different protocols to estimate the CTF of the micrographs in *Scipion*. In this tutorial we are going to use three different algorithms: `Gctf(?)`, `CTFFind4(?)` and `Xmipp CTF estimation(?)` executed with protocols `gctf-ctf estimation` (Fig. 9), `cistem-ctffind4` (Fig. 10) and `xmipp3-ctf estimation` (Fig. 11), respectively. Besides of estimating the CTF envelope, this last protocol improves the CTF estimation of CTFFind4. Thus, in this case an additional parameter is the defoci from a `Previous CTF estimation`.

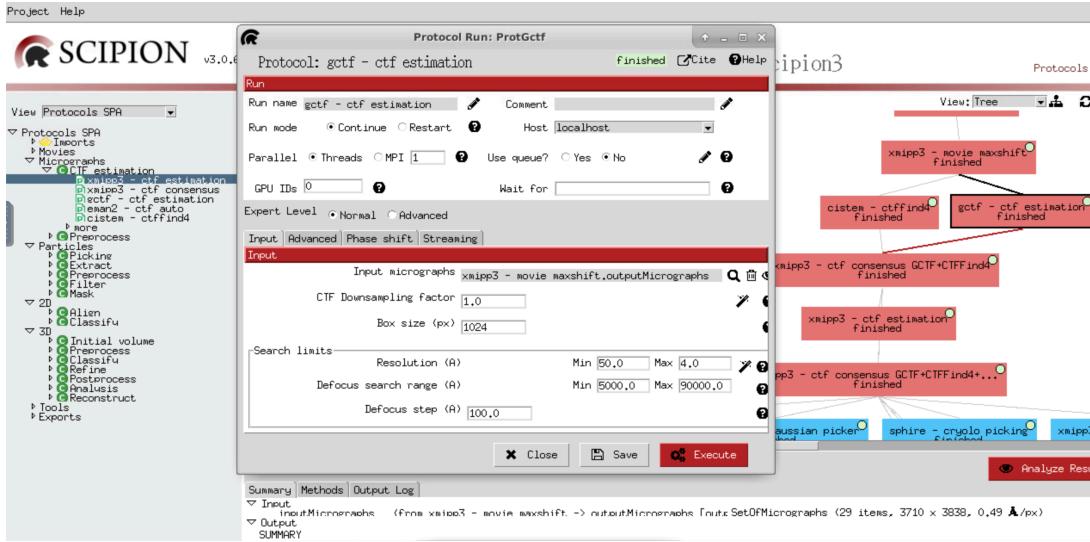
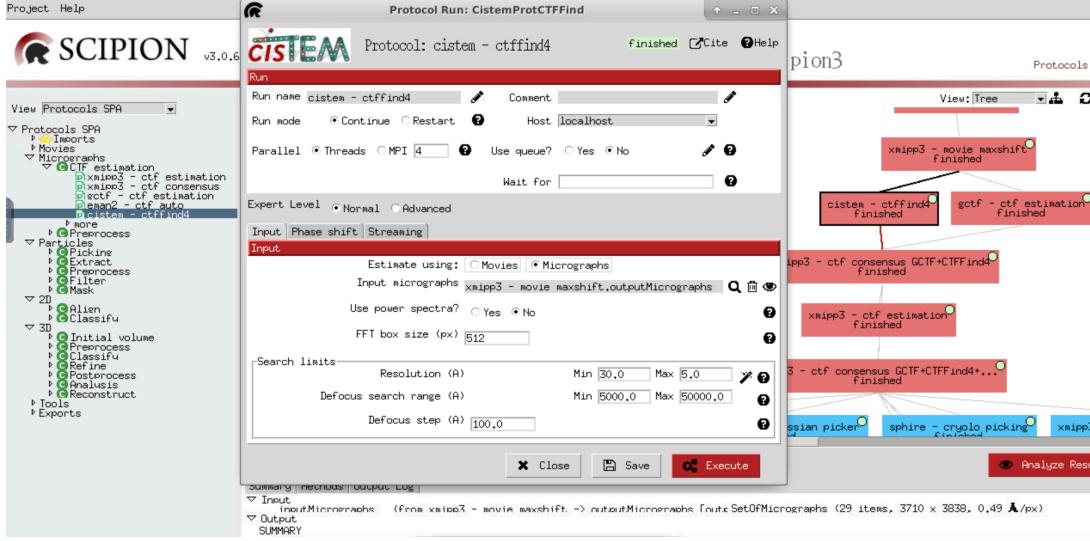
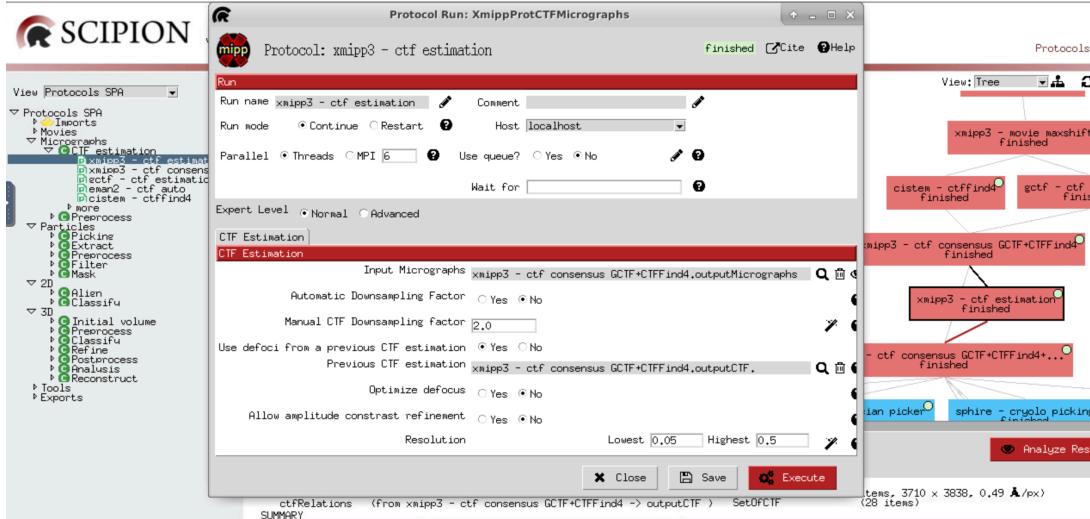


Figure 9: Protocol 1 `gctf-ctf estimation` to compute the CTF.

Figure 10: Protocol 2 `cistem-ctffind4` to compute the CTF.Figure 11: Protocol 3 `xmipp3-ctf estimation` to compute the CTF.

The three algorithms estimate the PSD of the micrographs and the parameters of the CTF (defocus U, defocus V, defocus angle, etc.). They cut micrographs into many smaller images with a desired window size. After that, they compute the

Fourier Transform of each image and calculate an average. The three protocols designed to apply the three respective algorithms contain very similar parameters. To estimate the CTF we need to limit the frequency region to be analyzed between the lowest and highest resolution. The frequency domain selected must include all zeros of the CTF. The wizard displayed on the right helps to choose that frequency region.

After executing each one of these three protocols, results can be observed by pressing **Analyze results**. A table will be opened showing the image of the CTF computed for each micrograph, as well as other CTF parameters. CTFs of good micrographs typically display multiple concentric rings extending from the image center towards its edges. Bad micrographs, however, might lack rings or show very few of them that hardly extend from the image center. Micrographs like these will be discarded, likewise micrographs showing strongly asymmetric rings (astigmatic) or rings that attenuate in a particular direction (drifted). To discard a particular micrograph, select it, click the mouse right button and choose **Disable**. If you want to see the CTF profile, choose the option **Show CTF profile**, and a new window will be opened to show the CTF profile. **Recalculate CTFs** and **Micrographs** are additional options of **Analyze results** menu that can be used after selecting specific micrographs. **Recalculate CTFs** allows to estimate again the CTF when the algorithm has previously failed to find the rings, even if they can be seen by eye. The option **Micrographs** allows to create a new subset of selected micrographs.

Concerning some differences among protocols, we remark that micrographs with CTF estimated with **cistem-ctffind4** display a hole in the center because, in some cases, they have very much power and avoid appreciate what is underneath. In the particular case of **xmipp3-ctf estimation**, four different images of micrographs are shown. Besides the PSD, this last protocol displays the enhanced PSD, the CTF model by quadrants and half planes.

## CTF consensus

The CTF estimation process concludes by applying two different consensus protocols to assess first, the differences among the output of the algorithms **Gctf** and **CTFFind4** and second, the differences among the consensus reached by those two and **Xmipp CTF estimation**. We are going to use **xmipp3-ctf consensus** to perform this task (Fig. 12 and Fig. 13). This protocol allows to screen micrographs according meaningful CTF estimations based on defocus values, astigmatism, resolution and other **Xmipp** criteria (second tap in the protocol form), which will only be used in case that any of the CTFs computed is estimated by **xmipp3-ctf estimation**.

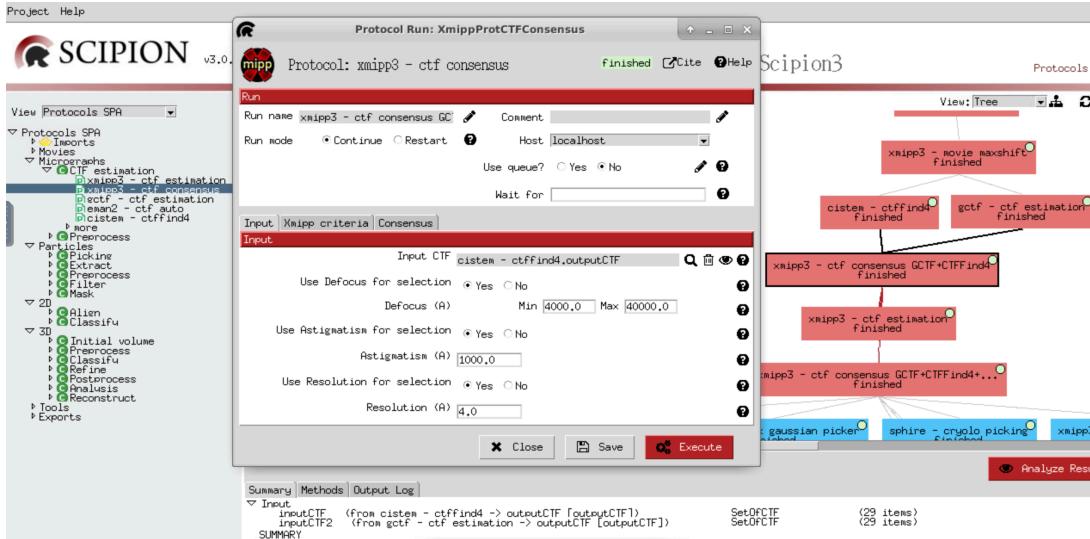


Figure 12: Filling in the consensus protocol 1 **xmipp3-ctf consensus**.

In this first case we have selected, as first input, the estimation of the CTF calculated by **cistem-ctffind4** and, as second input, the estimation obtained by **gctf-ctf estimation**. By pressing **Analyze Results** both accepted (28) and rejected (1) micrographs can be visualized. This consensus was used to obtain a good estimate of the defocus.

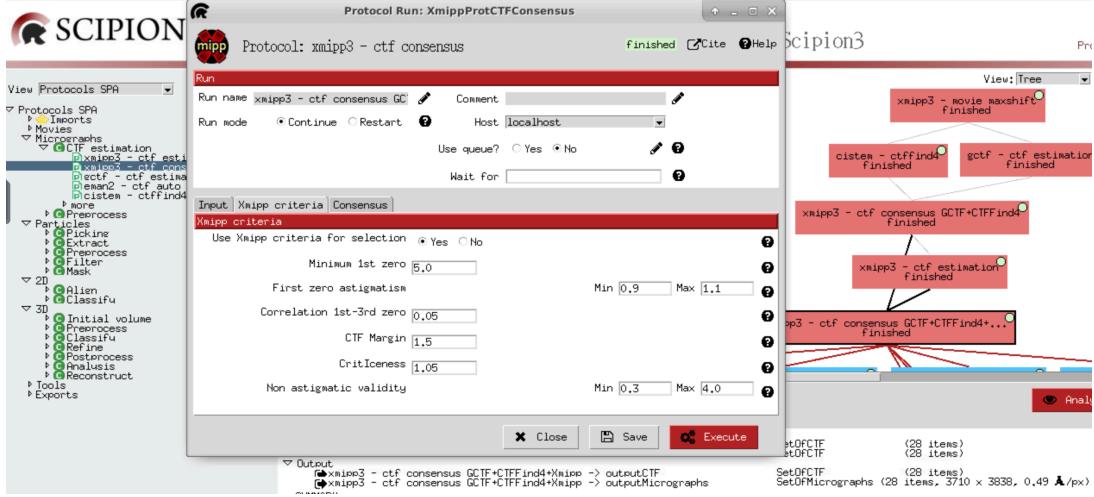


Figure 13: Filling in the consensus protocol 2 `xmipp3-ctf consensus`.

In this second case we have selected, as first input, the estimation of the CTF calculated by `xmipp3-ctf estimation` and, as second input, the estimation obtained by `xmipp3-ctf consensus GCTF and CTFFind4`. By pressing `Analyze Results` both accepted (28) and rejected (0) micrographs can be visualized. On the other hand, once we had a good estimate of the defocus with this consensus we obtained a good estimate of the envelope.

With the proper set of micrographs we can continue the image processing. The negative effects of the CTF will be corrected in the next steps.

For more information:

- **Video tutorial:** second half of this video [https://www.youtube.com/watch?v=01sBrJbKh7I&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=26](https://www.youtube.com/watch?v=01sBrJbKh7I&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=26).
- **Theoretical lecture:** second half of this video [https://www.youtube.com/watch?v=F3Uslh3v9J8&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=32](https://www.youtube.com/watch?v=F3Uslh3v9J8&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=32).

## 5 Particle picking

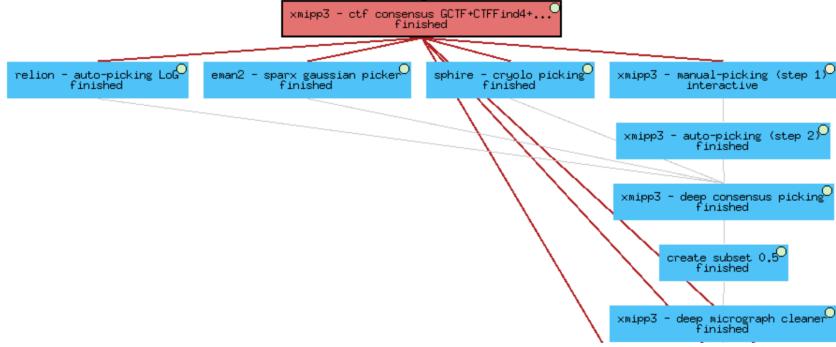


Figure 14: Particle picking workflow.

Since the reconstruction of the 3D density map of the macromolecule is based on images of individual single particles, the next step is essential in the image processing workflow. With the particle picking step we will retrieve the coordinates of each single particle.

Because manual picking can be very tedious, some tools have been designed to help in this task. We have integrated in *Scipion* different picking tools that can be used individually or in combination to obtain the final coordinates. Currently, we have tools available from *Eman2*, *Relion*, *Bsoft*, *Sphire* and *Xmipp*. In this tutorial we are going to use four different protocols that integrate tools from *Relion* (`relion-auto-picking LoG` (?)), *Eman2* (`eman2-sparx gaussian picker`), *Sphire* (`sphire-cryolo picking` (?)) and *Xmipp* (`xmipp3- manual-picking (step1)` and `xmipp3-auto-picking (step 2)` (?)). The reason for using several pickers is that, each of them have different properties and different errors which complement the process to have a better estimate.

The protocol `relion-auto-picking LoG` (Fig. 15) provides particle coordinates in an automatic way. Together with the 28 input micrographs and the size in pixels for each particle, the protocol form allows to set specific parameters to compute the Laplacian of Gaussian (LoG).

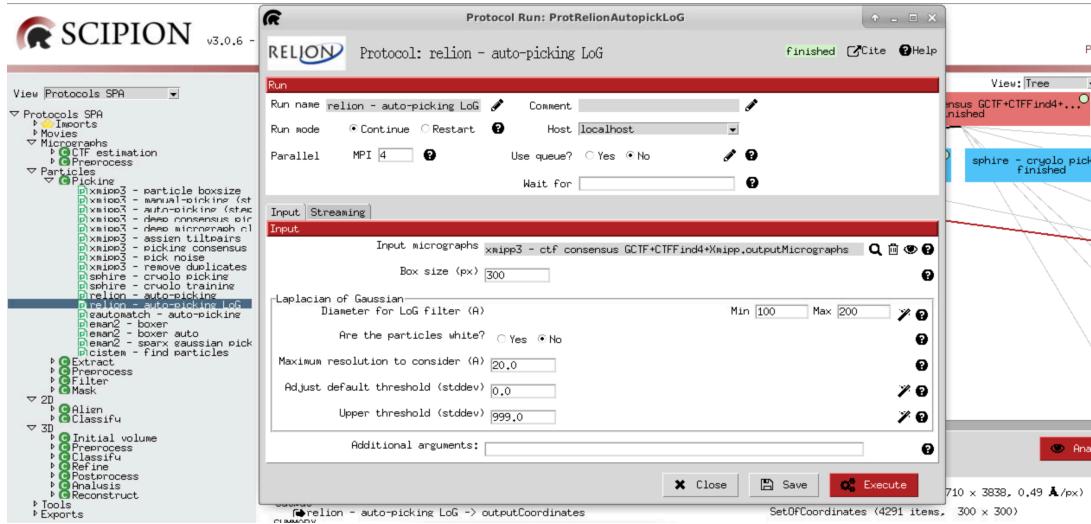


Figure 15: Filling in the protocol 1 [relion-auto-picking LoG].

The protocol [eman2-spark gaussian picker] (Fig. 16) provides particle coordinates in an automatic way. Similar to Relion, together with the 28 input micrographs, we have to set the size in pixels for each particle, and the width of the Gaussian kernel used for automated particle picking, the protocol allows us to obtain the particles as a set of coordinates.

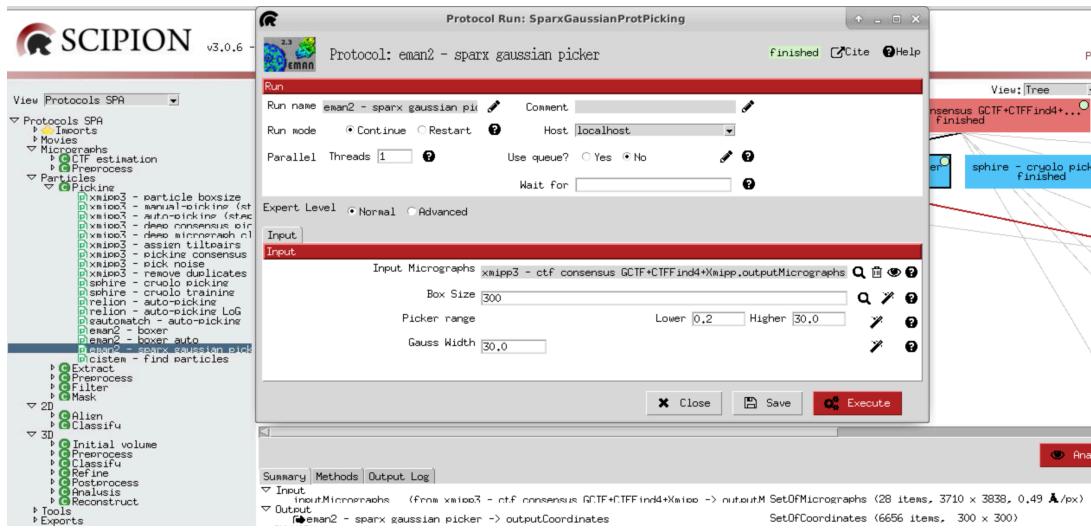


Figure 16: Filling in the protocol 2 [eman2-spark gaussian picker].

The protocol **sphire-cryolo picking** (Fig. 17) integrates a fully automated particle picker based on deep learning. The protocol form also requires the 28 micrographs and the size of particles, and gives you the possibility of using your own network model, obtained in a previous training step, instead of the general one. **Confidence threshold** allows to perform a more or less conservative picking by increasing or decreasing the value of this param.

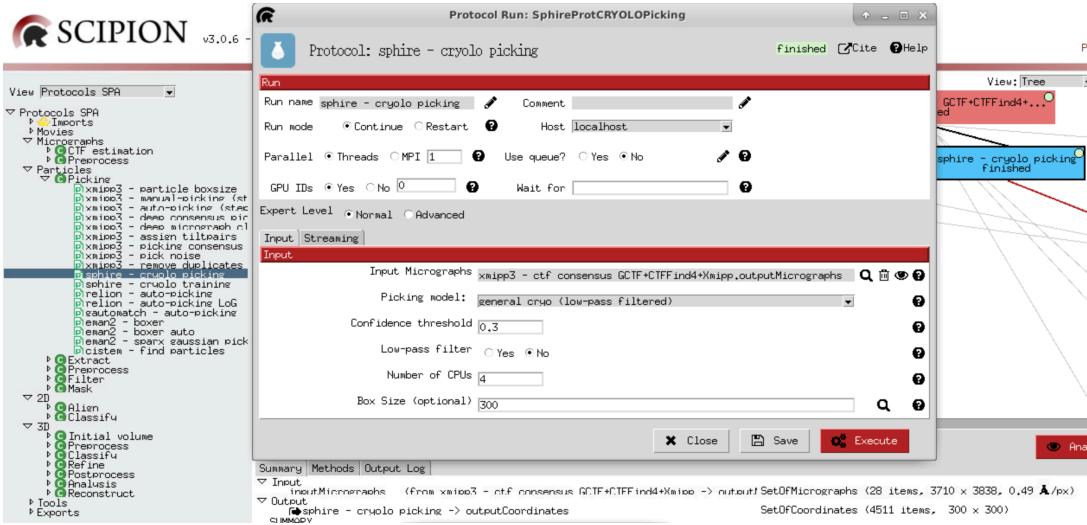


Figure 17: Completing in the protocol 3 **sphire-cryolo picking**.

The protocol **xmipp3-manual-picking (step1)** (Fig. 18) is the first part of the Xmipp picking method, and allows to perform manual picking in a set of micrographs either manually or in a supervised mode. This protocol only requires as input the set of micrographs.

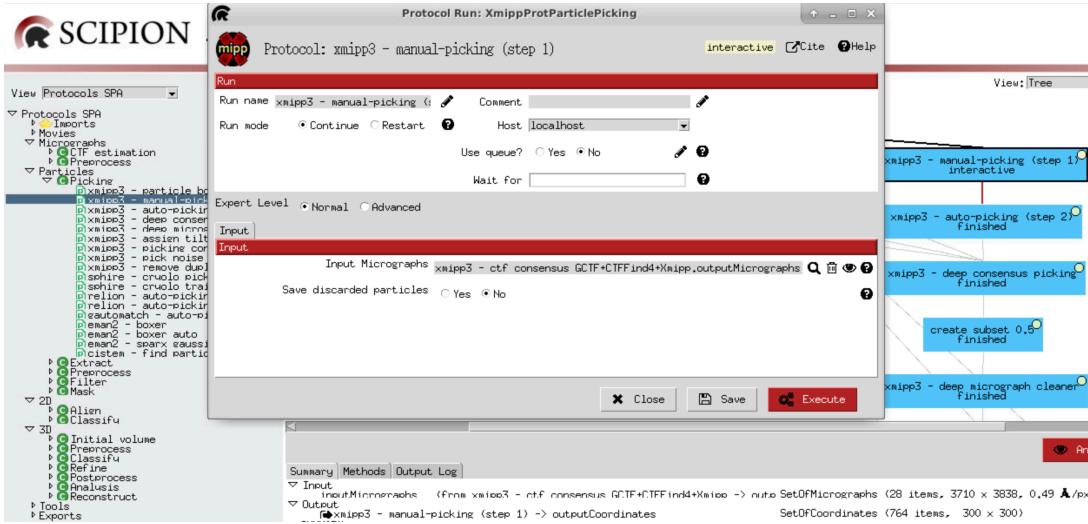


Figure 18: Filling in the protocol 4 `xmipp3- manual-picking (step1)`.

After executing this protocol, the respective box will become light yellow because an interactive job is running and it can be relaunched at any time.

The Xmipp picking GUI contains a control panel with the list of micrographs and some other parameters. The micrograph that we are going to pick is displayed in a separate window and we can apply to it a number of filters/enhancements (like Gaussian blurring, Invert contrast, adjust histogram, etc.) just to improve the visualization of particles. Main control actions are:

- **Shift + Mouse wheel:** Zoom in and out of the overview window.
- **Mouse left button:** Mark particles. You may move its position by clicking the left mouse button on the selected particle and dragging it to a new position.
- **Shift + Mouse left:** Remove a selected particle.
- You can apply filters in the micrographs to see the particles better. Select those filters in the menu **Filters**.

In the manual/supervised step, we start picking manually a few micrographs (5 in this case) and then click the **Active training** button. At this point, the program

will train a classifier based on machine learning and will propose some coordinates automatically. You can “correct” the proposal of the classifier by adding missing particles or removing wrongly picked ones. After training with a few more micrographs, we can register the output coordinates by clicking the **Coordinates** red button.

After manual picking, we can close the GUI and open the protocol **xmipp3- auto-picking (step 2)** (Fig. 19). Select as inputs the previous manual/supervised execution and all micrographs (**Micrographs to pick: Other**). The method will pick the rest of micrographs automatically. At the end, we can review the picking coordinates and we still have the chance to add/remove particles.

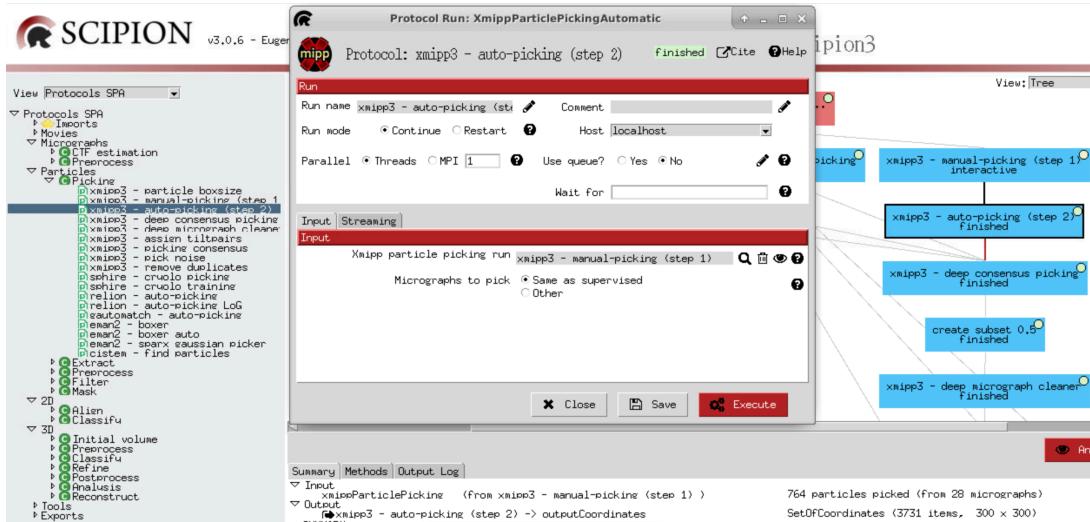


Figure 19: Completing the protocol **xmipp3-auto-picking (step 2)**.

Results of all these protocols can be observed by pressing **Analyze Results**. In all cases a table details the number of particles extracted from each micrograph. Total number of particles appear in the lower part of this table, 4,291, 6,656, 4,511, 764 and 3,731 running **relion-auto-picking LoG**, **eman2-sparx gaussian picker**, **sphire-cryolo picking**, **xmipp3-manual-picking (step1)**, and **xmipp3-auto-picking (step 2)**, respectively. As a conclusion, the three algorithms devoted to particle picking give us a similar result, around 4,500 particles. However, there are some differences among programs and we would like to keep only the coordinates of the good particles selected by the four methods.

## Consensus in particle picking

The protocol **xmipp3-deep consensus picking** (Fig. 20) will try to select consensus particles among different particle picking algorithms. This protocol can also be used to get the consensus of sets of coordinates retrieved after using distinct settings of parameters with the same program. In our case, the whole sets of coordinates retrieved from the four previous methods have to be included as protocol input.

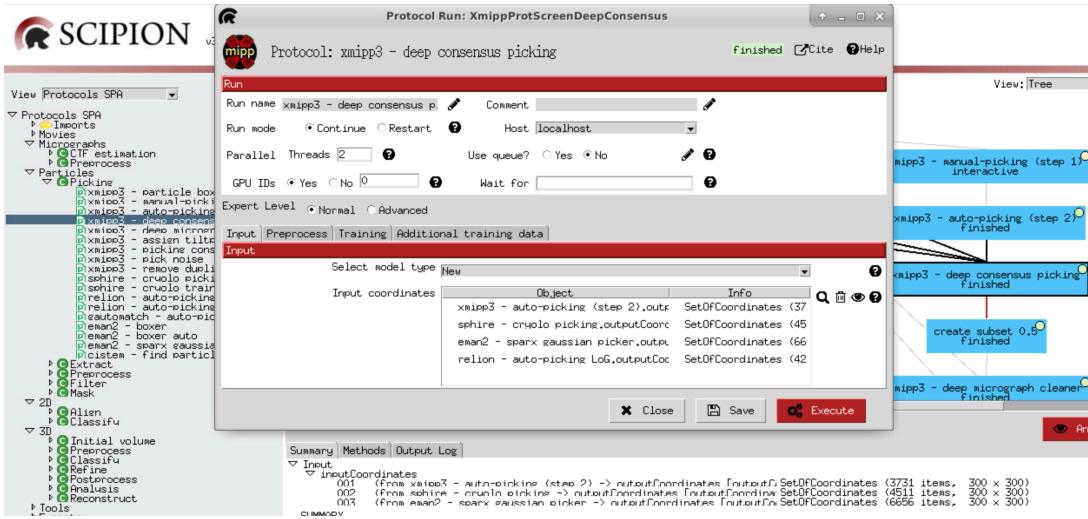


Figure 20: Completing the protocol **xmipp3-deep consensus picking**.

(Note: when executed, if the process takes too long in step 13, stop the protocol and press continue.)

A neural network will be trained with subsets of coordinates from particles picked and not picked. Finally, the method provides a score for each particle according to the neural network predictions. After pressing **Analyze Results**, a menu allows to visualize a table showing an image and the value of the deep learning score of all the particles (Select particles/coordinates with high 'zScoreDeepLearning1' values). Considering that bad particles show scores values close to 0.00 and good particles scores close to 1.00, the threshold, automatically set to 0.50, allows to select good particles. In our example, from the total number of particles (4,006), 0

particles were rejected, and the total number of particles were selected to remain in the processing workflow.

An additional cleaning step, accomplished with the protocol `xmipp3-deep micrograph cleaner`, removes particles located in carbon zones or in large impurities (Fig. 21). Provide as input the previously selected set of coordinates and indicate the set of micrographs from which the computation will be performed. By default, we use the same as coordinates.

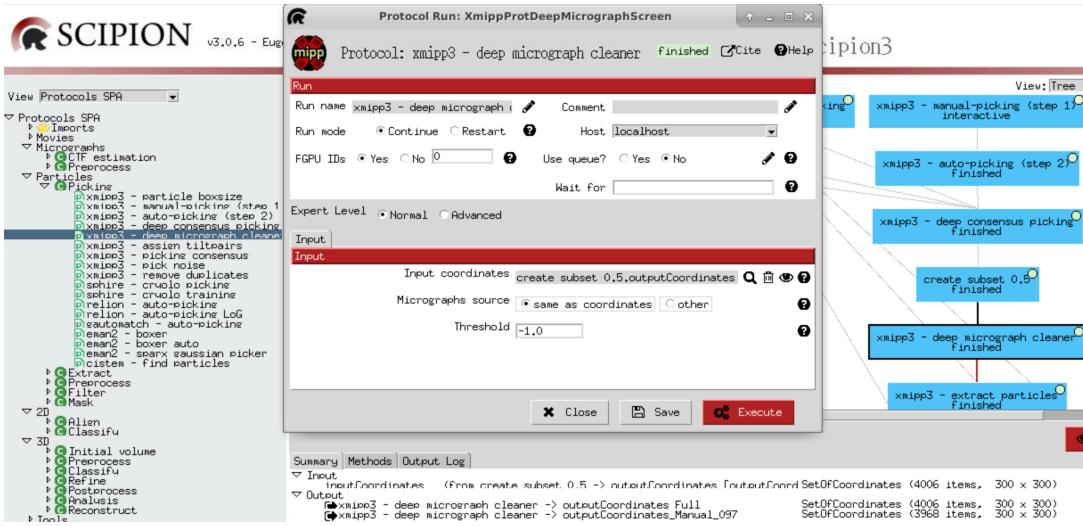


Figure 21: Completing the protocol `xmipp3-deep consensus picking`.

After this additional cleaning step, 38 particles were rejected. The coordinates of the remaining reliable 3,968 particles are selected for further processing.

For more information:

- **Video tutorial:** first half of this video [https://www.youtube.com/watch?v=eVjQoZ8ehw&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=27](https://www.youtube.com/watch?v=eVjQoZ8ehw&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=27).
- **Theoretical lecture:** first half of this video [https://www.youtube.com/watch?v=yVFvN2T\\_soQ&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=33](https://www.youtube.com/watch?v=yVFvN2T_soQ&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=33).

## 6 Extract particles

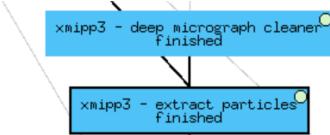


Figure 22: Extract particles workflow.

Once we have the set of coordinates, we can proceed to extract particles with **Xmipp** protocol **xmipp3-extract particles** (Fig. 23). This protocol allows to extract, normalize and correct the CTF phases of the selected particles. As input, this protocol requires the set of coordinates, the consensus CTF values obtained in previous steps, and a downsampling factor. To save computing resources, include in the input the desired reduced size of the particles. In this particular case, 120 pixels.

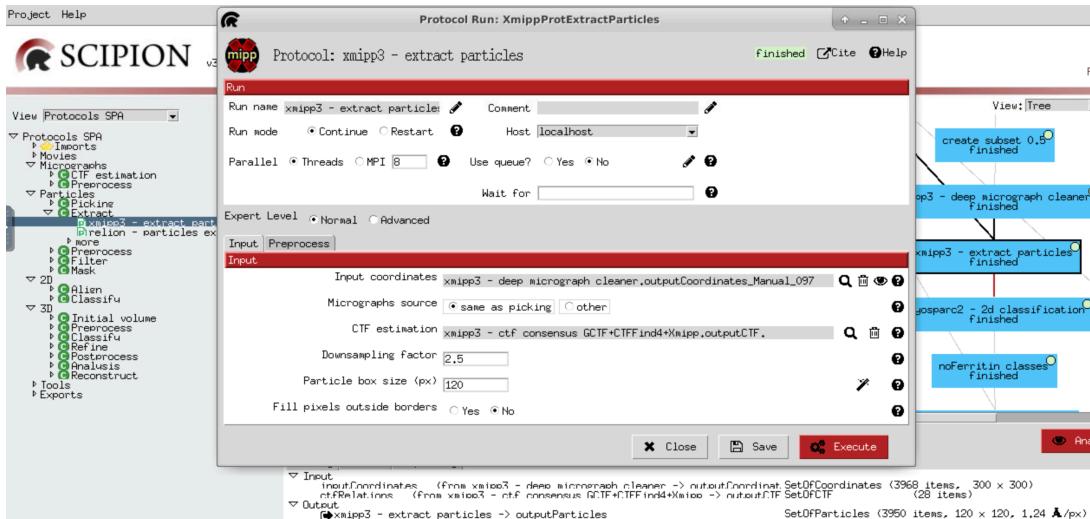


Figure 23: Filling in the protocol **xmipp3-extract particles**.

The form tap **Preprocess** gives you additional options:

- **Dust removal**: Option **Yes** (recommended) sets pixels with unusually large values to random values from a Gaussian with zero-mean and unity-standard deviation.

- **Invert contrast:** Option **Yes** means that bright regions become dark and the other way around.
- **Phase flipping:** Option **Yes** means that the protocol corrects CTF phases of the particles.
- **Normalize:** Option **Yes** (recommended) means that the particles are normalized with zero mean and one as standard deviation for background pixels.

As output, the protocol generates a new set of 3,950 particles after discarding other 18 particles. The extracted particles have a smaller selected size and almost 3 times the initial sampling rate. The images of the normalized extracted particles can be seen pressing **Analyze Results**. By default, particles displayed in gallery mode can be sorted by **Zscore**. To visualize the score associated to each particle, switch the table view by pressing the top left button. If you want to remove any of the particles showing lower score values, select them, press the mouse right button and choose **Disable**. A new subset of particles can be created by clicking on **Particles** red button. However, this new subset of selected particles is considered reliable for further image processing.

## 7 2D Classification

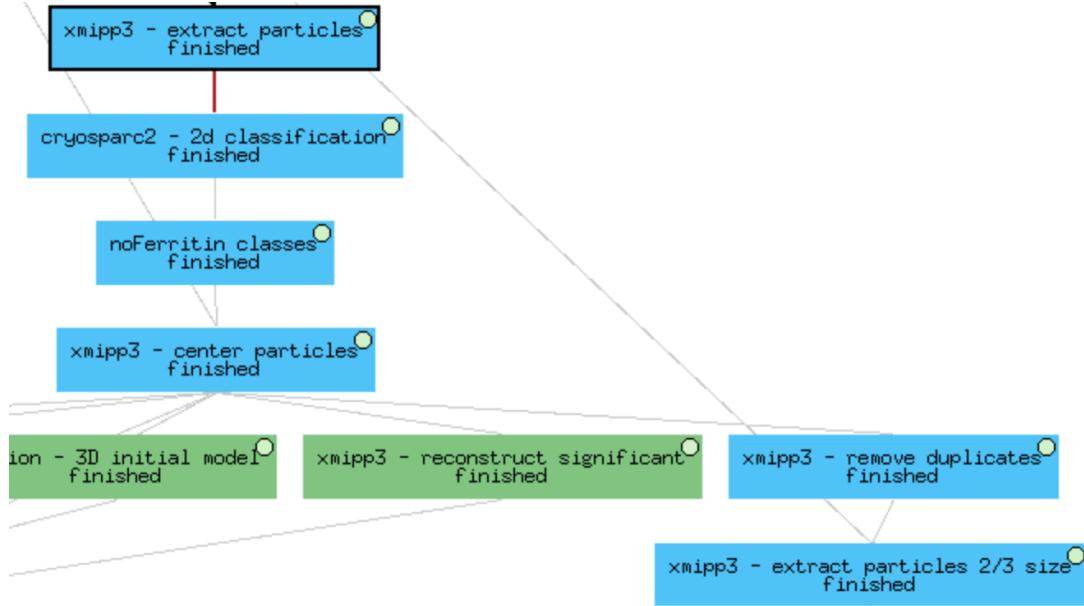


Figure 24: 2D classification workflow (Blue color).

The next step in image processing involves the 2D classification of particle images to group similar ones. This process can serve as an exploratory tool of your data and might also be used to throw away bad particles. In addition, by overlapping similar images we can obtain the average images or 2D classes. Since these classes are the projections of the 3D object that we try to reconstruct, they can also be used in the reconstruction of the 3D object.

Although there exist several 2D classification algorithms, in this tutorial the 2D classes will be created with the *cryoSPARC* (?) 2D classification method (?), integrated in the protocol [cryosparc2-2D classification](#). We used as input the subset of particles previously selected. The 2D classification parameters can be observed in the central tap of the protocol form in the Fig. 25. Remark that we choose in advance the number of classes, 50 in this case.

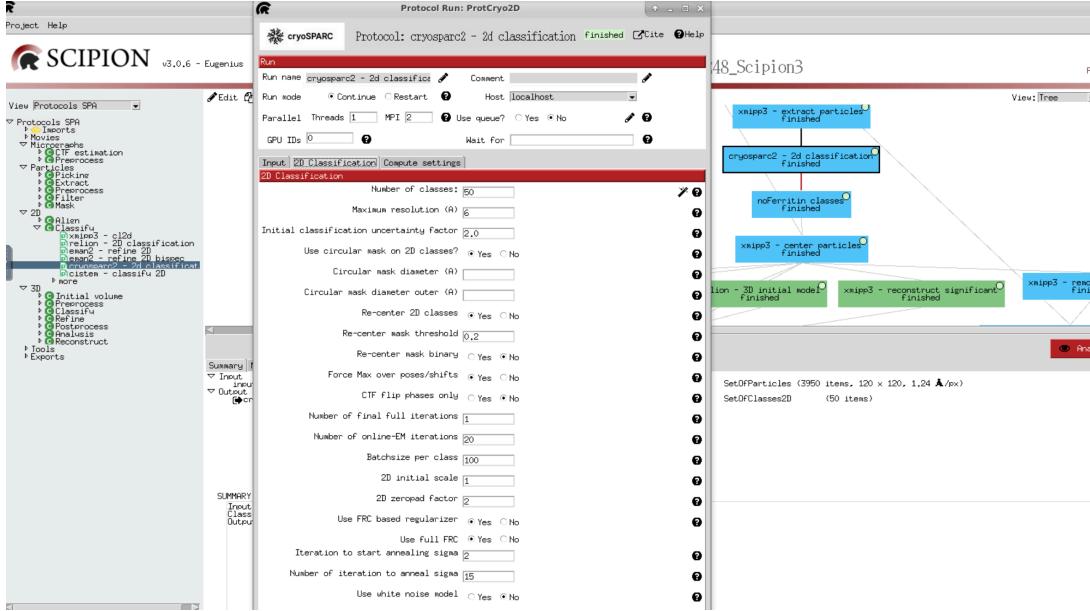


Figure 25: Filling in the second tap of the protocol **cryosparc2-2D classification**.

After running the protocol, particle classes can be visualized selecting any of the two options of the menu opened with **Analyze Results**, the common *Scipion* viewer or the *cryoSPARC* GUI. By double-clicking or right-clicking the classes we can see the particles behind those classes. The final number of classes also appears in the Summary output, which in this case is 50 classes.

Once inspected the different classes, in *Scipion* we can discard manually the ones that still have the iron attached (Ferritin molecules), which will appear as a white dot in the core of the particle. In our case we found that 378 particles had the iron attached. Then by selecting the classes that we are interested in (38 from 50) and pressing **Classes**, a new set of 38 Classes, which contained both the average (class representative particles) and the particles behind them (3,491), will be created included in the box **noFerritin classes**.

The next protocol **xmipp3-center particles** will recenter the particles. For centering

the particles, we needed as input both the classes and the set of micrographs, as we can see in the protocol form Fig. 26.

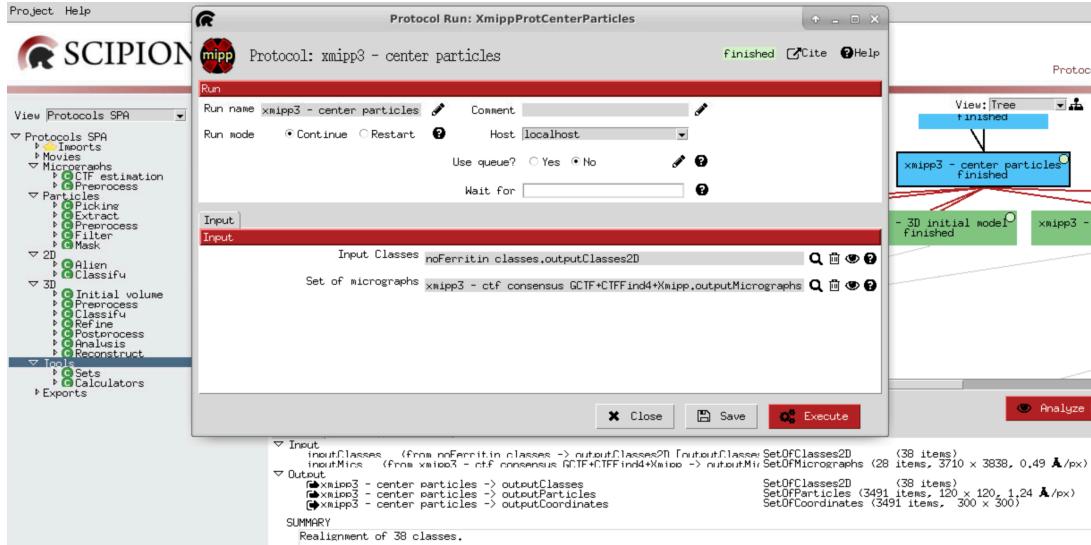


Figure 26: Centering of particles selected in `noFerritin classes` box.

This operation will allow us to compute better the initial volumes and also to eliminate duplicates with `xmipp3-remove duplicates` protocol, as with the recentered particles we can have a better estimate of which are the coordinates that are pointing to the same particle within a radius x (defined to 100).

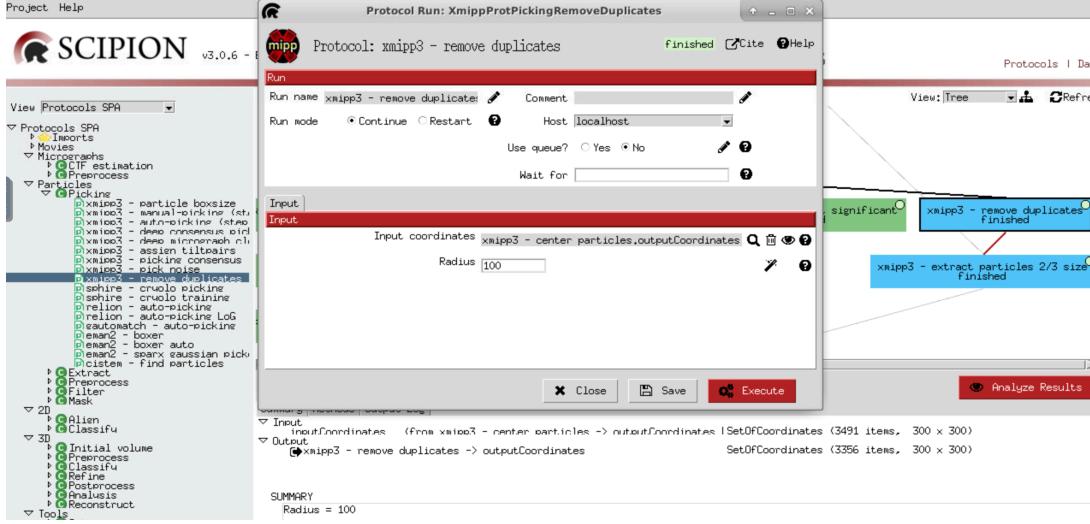


Figure 27: Remove duplicates from the recentered set of particles.

After running the protocol, particles can be visualized selecting the **Analyze Results**. In principle, we do not see many overlapping particles in the micrographs. In this case, for 3491 particles 141 were duplicated obtaining a clean set of coordinates of 3,356 particles.

Once we have a set of non-duplicated coordinates, we can proceed to extract particles with Xmipp protocol **xmipp3-extract particles** (Fig. 28). This protocol allows to extract, normalize and correct the CTF phases of the selected particles. As input, this protocol requires the set of coordinates and the consensus CTF values obtained in previous steps, and a downsampling factor 1.5 to obtain more or less 1 Å per pixel resolution and 250 box size to correct the CTF effect.

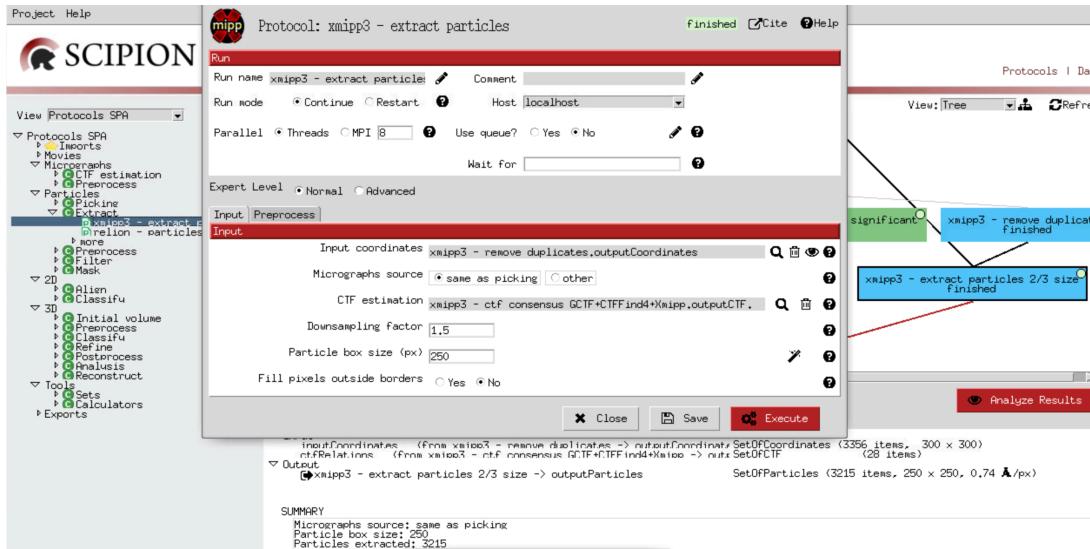


Figure 28: Filling protocol `xmipp3-extract particles`.

The recentered set of particles will be used in the next step in the processing workflow to generate the initial volume and the extracted non-duplicated set of particles will be used in the step in the processing workflow as input of the 3D Refinement and Classification.

For more information:

- **Video tutorial:** second half of this video [https://www.youtube.com/watch?v=eVjQoZ8ehw&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=27](https://www.youtube.com/watch?v=eVjQoZ8ehw&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=27).
- **Theoretical lecture:** second half of this video [https://www.youtube.com/watch?v=yVFvN2T\\_soQ&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=33](https://www.youtube.com/watch?v=yVFvN2T_soQ&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=33).

## 8 Initial volume

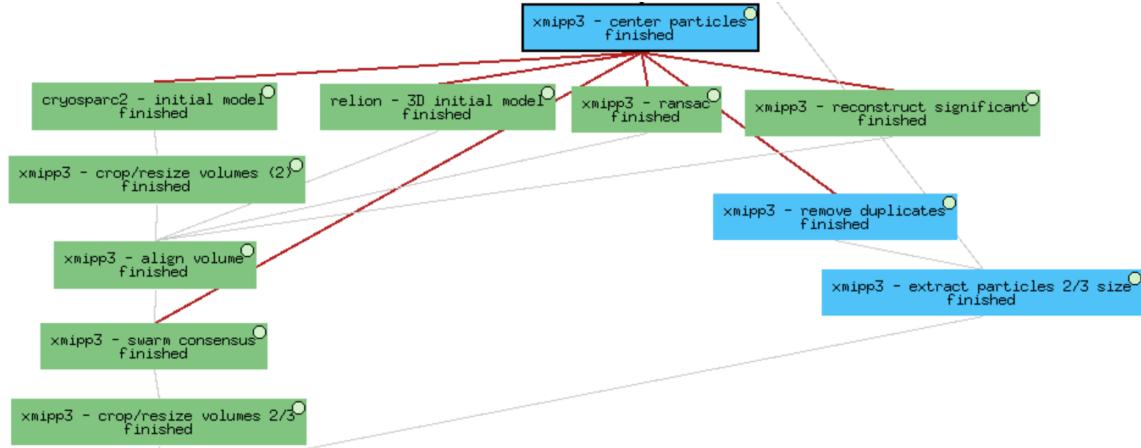


Figure 29: Initial volume (Purple color).

Right angles of each projection image are needed to reconstruct the 3D map. In 3D EM, however, these angles are unknown and we have to estimate them. The most popular way of estimating them is comparing the projections of a volume similar to ours, known as initial volume, with the images obtained from the microscope. Since the 3D map reconstruction process requires an approximate low resolution map to be refined in further steps according to the projection images of particles, in this tutorial we are going to generate a *de novo* initial map model combining the results obtained by different algorithms: First, to compute the initial volume using the set of aligned particles as input, we have used *cryoSPARC Stochastic Gradient Descent (SGD)* and *Relion Stochastic Gradient Descent (SGD)* (Fig. 29, left). Second, to generate the initial volume from the class representative particles, we have run *Xmipp reconstruct significant* and *Xmipp RANSAC* (Fig. 29, right). *Xmipp reconstruct significant* sets the map in a **Weighted Least Square** framework and calculates weights through a statistical approach based on the cumulative density function of different image similarity measures. *Xmipp RANSAC* is based on an initial non-lineal dimension reduction approach with which selecting sets of class representative images that capture the most of the structural information of each particle. These reduced sets will be used to build maps starting from random ori-

tation assignments. The best map will be selected from these previous assumptions using a random sample consensus (RANSAC) approach.

### *cryoSPARC* Stochastic Gradient Descent (SGD)

The algorithm *cryoSPARC* Stochastic Gradient Descent (SGD) has been implemented in the protocol `cryosparc2-initial model` (Fig. 30). The set of particles selected in the previous step is used as input (see the Input tap of the protocol form).

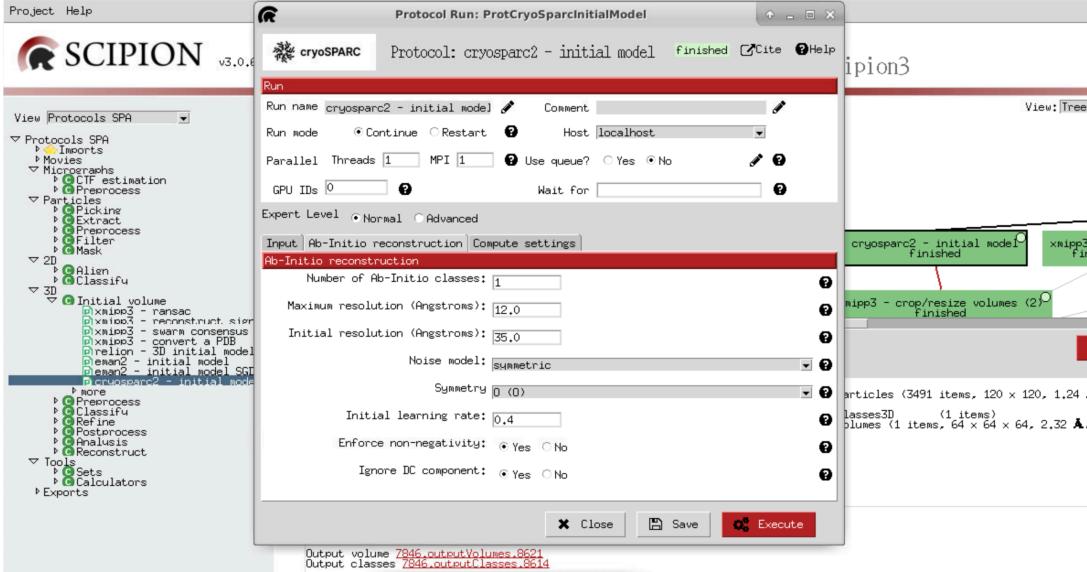


Figure 30: Completing the second tap of the protocol `cryosparc2-initial model` with one **Ab-Initio** class and octahedral **Symmetry**.

In this case, only one **Ab-Initio** class has been selected and octahedral **Symmetry** has been considered (Fig. 30). The **Ab-Initio** class will be randomly initialized, unless an initial map is provided. In that case, the class will be a random variant of the initial map. Regarding symmetries, enforcing symmetry above C1 is not recommendable for *ab-initio* reconstructions. The volume or the 3D class generated can be appreciated by pressing `Analyze Results`.

Since the size and sampling rate of maps generated with [cryosparc2-initial model](#) differ from the size and sampling rate of the input particles, a resizing intermediate method has to be applied to recover these dimensions. Protocol [xmipp3-crop/resize volumes](#) will help us with this task (Fig. 31). As input, select the output volumes of the previous protocols, Sampling Rate for Resize option, and 120 pixels as New image size.

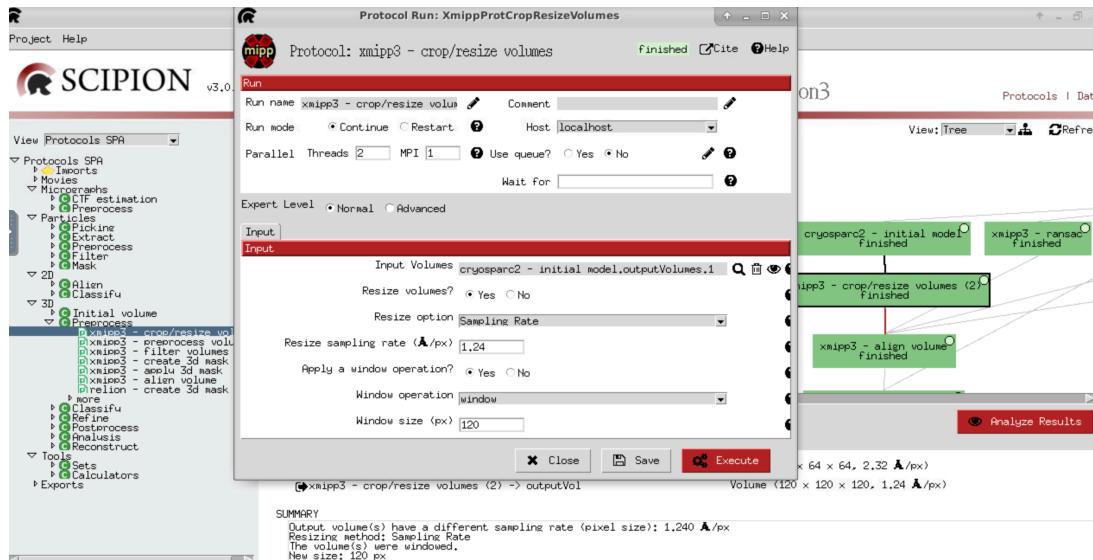


Figure 31: Completing the protocol [xmipp3-crop/resize volumes](#).

### Relion Stochastic Gradient Descent (SGD)

*Relion Stochastic Gradient Descent (SGD)* has been implemented in the protocol [relion-3D initial model](#). As input we are using the same set of particles as *cryoSPARC Stochastic Gradient Descent (SGD)*. And to fill in the param values we used one class and octahedral Symmetry and we executed it (Fig. 32).

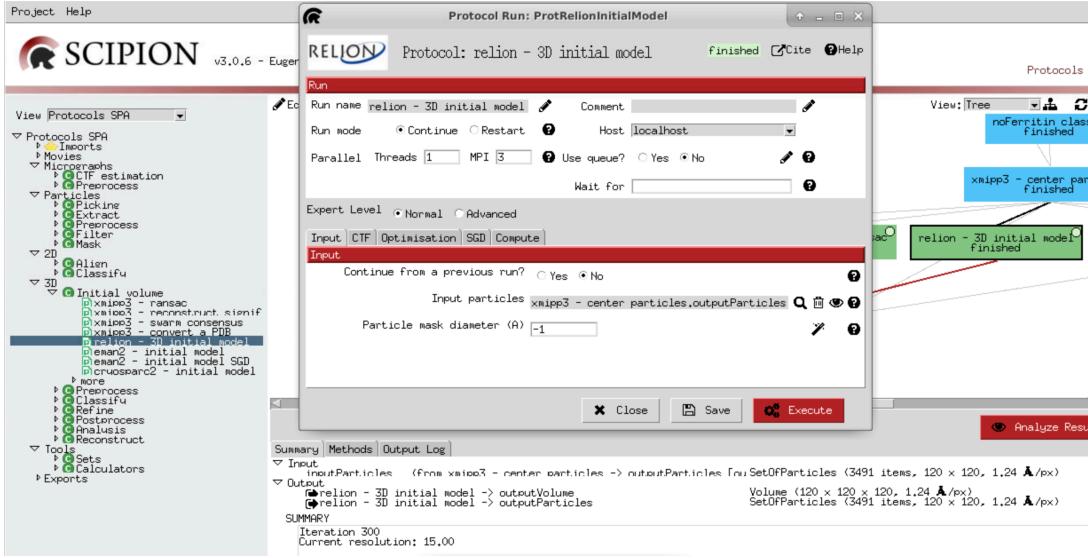


Figure 32: Filling in the Input tap of the protocol `relion-3D initial model`.

Only one volume has been generated with this protocol that keeps size and sampling rate of the input particles. You can visualize it with *Chimera* in 3D by pressing `Analyze Results` and selecting in the Volumes box `Display volume with chimera`.

### *Xmipp*

Using the 38 class representative particles as input, as well as the type of symmetry (octahedral), the protocol `xmipp3-reconstruct significant` (Fig. 33) also generates one initial volume and preserves the size and sampling rate of the input 2D classes.

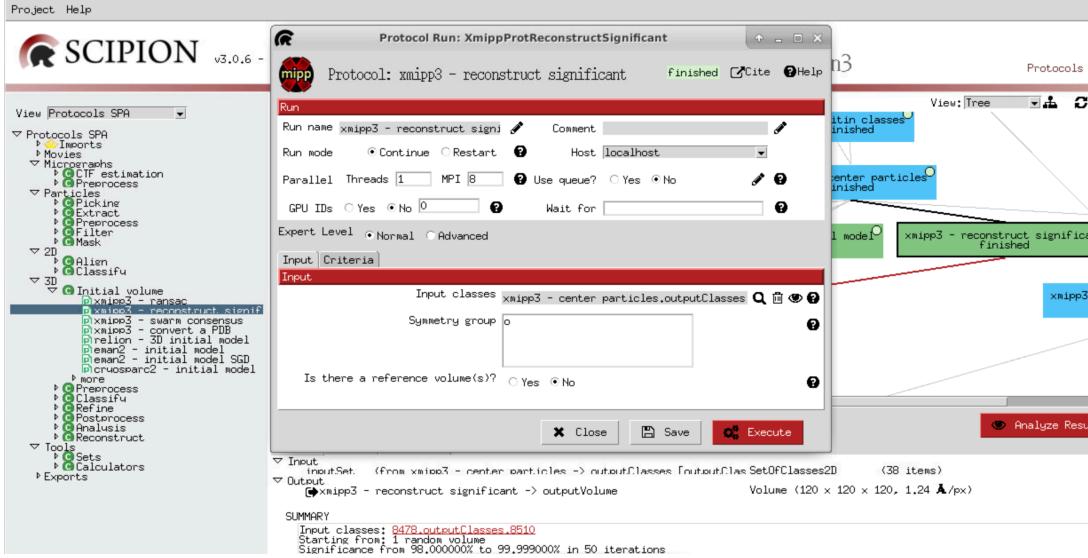


Figure 33: Completing the Input tap of the protocol **xmipp3-reconstruct significant**.

*Xmipp* RANSAC algorithm, implemented in the protocol **xmipp3-ransac** (Fig. 34), although starts from the same input than *Xmipp* reconstruct significant, generates 10 different maps and preserves the size and sampling rate from the input 2D classes. You can choose a different number of output maps in the advanced param Number of best volumes to refine.

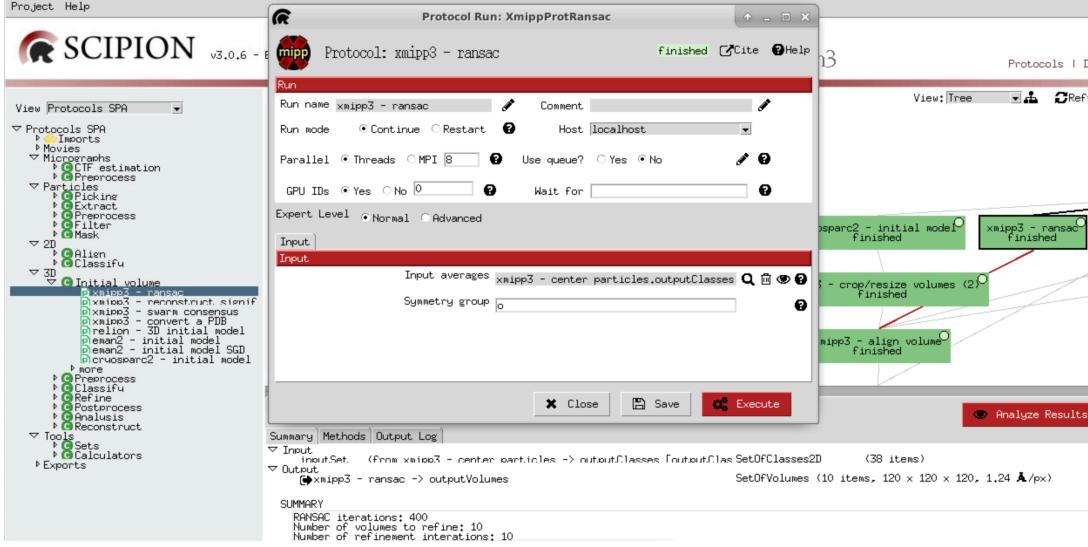


Figure 34: Filling in the params of the protocol `xmipp3-ransac`.

## Map alignment and swarm consensus

Next, we perform a fast fourier alignment of the 13 maps (volumes) generated, starting both from particles and 2D classes, using the protocol `xmipp3-align volume` (Fig. 35). As **Reference volume** we select the initial volume obtained by the *Xmipp* reconstruct significant algorithm.

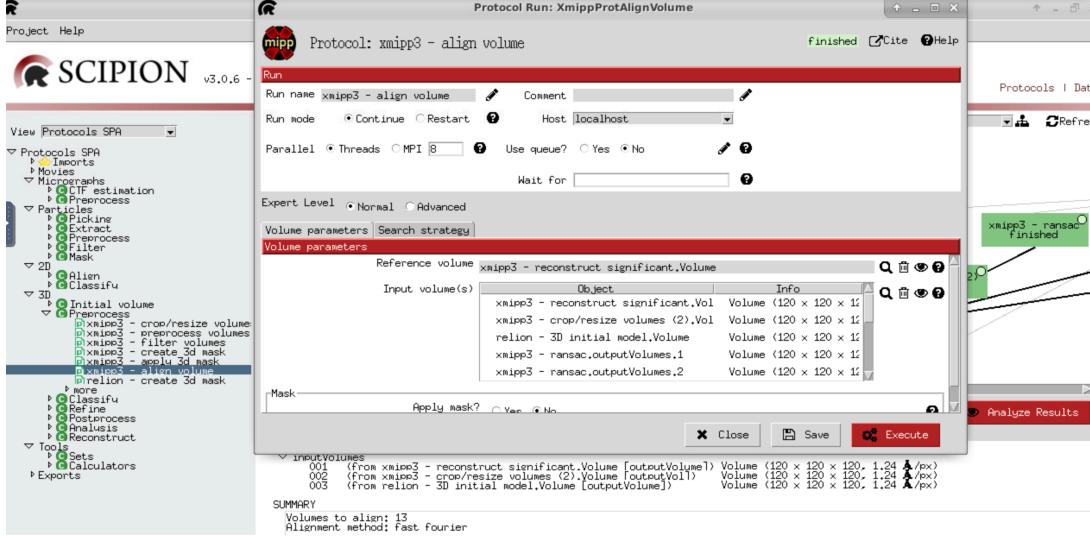


Figure 35: Completing the params for protocol `xmipp3-align volume`.

A new set of 13 volumes has been created keeping the same size and sampling rate shown by the initial particles. These volumes can be visualized by pressing **Analyze Results**.

Next, in order to have only one initial volume partially refined against the selected set of particles, we use the protocol `xmipp3-swarm consensus` (Fig. 36). The inputs of this protocol are the set of 13 maps and the set of 3,491 *Relion* extracted particles, previously generated. In this case, maps and particles have the same size and sampling rate. The program try to optimize the correlation between the swarm of volumes and the set of particles. Only a fraction of the particles are used to update this stochastic maximization.

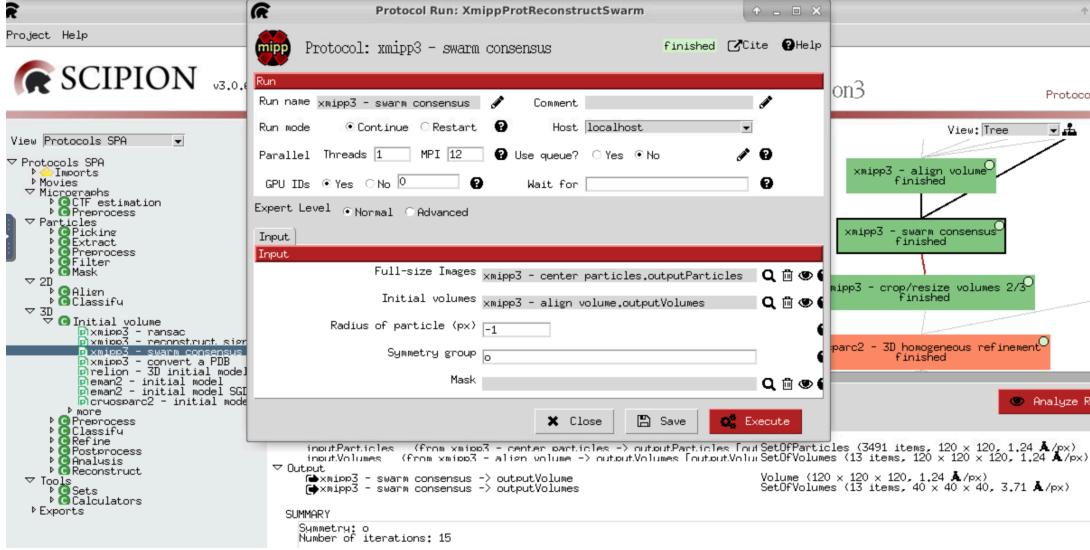


Figure 36: Completing in the params of the protocol **xmipp3-swarm consensus**.

After 15 iterations, this protocol generates two outputs, a downsampled set of volumes and a volume with the size and sampling rate of the inputs. However the **xmipp3-swarm consensus** produced a volume that is of a different pixel size than the extracted particles from the step before, therefore we need to apply **xmipp3-crop/resize volumes** (Fig. 37) to obtain the same dimensions. As input, select the output volumes of the previous protocols, **Sampling Rate for Resize** option, and 250 px as **Window size**.

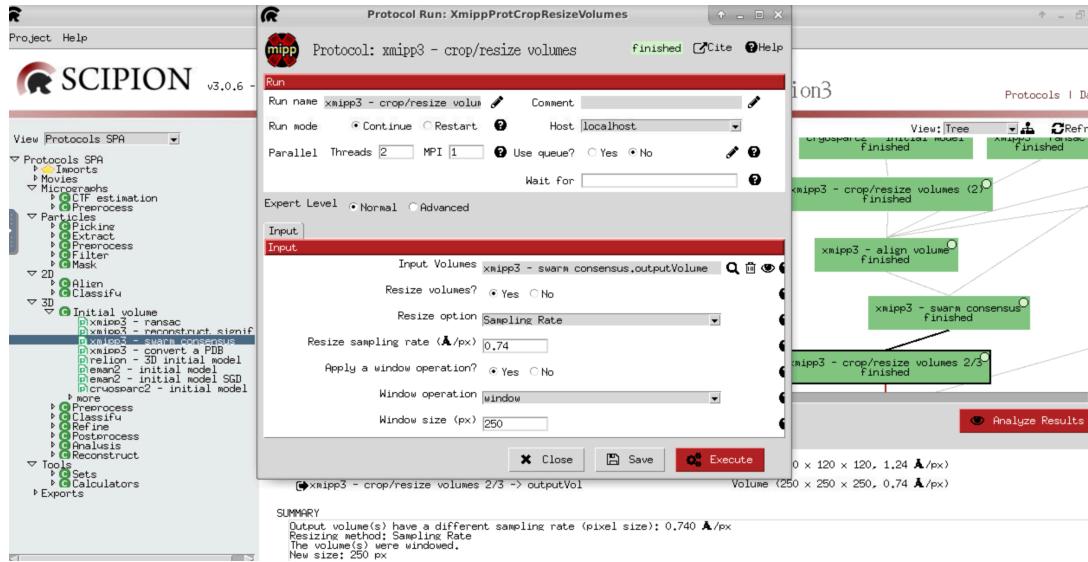


Figure 37: Params of protocol `xmipp3-crop/resize volumes`.

At this point, we have an initial volume that has been chosen from many proposed algorithms and from the previous step we have extracted the set of particles that are not duplicated and had pass through all the picking and 2D filtering process and both of the same dimensions. These will be used for the next step in the workflow of 3D Classification and Refinement.

For more information:

- **Video tutorial:** [https://www.youtube.com/watch?v=ppZXg4B7Q7U&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=28](https://www.youtube.com/watch?v=ppZXg4B7Q7U&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=28).
- **Theoretical lecture:** [https://www.youtube.com/watch?v=jzEXzW3VB1w&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=34](https://www.youtube.com/watch?v=jzEXzW3VB1w&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=34).

## 9 3D Classification and Refinement

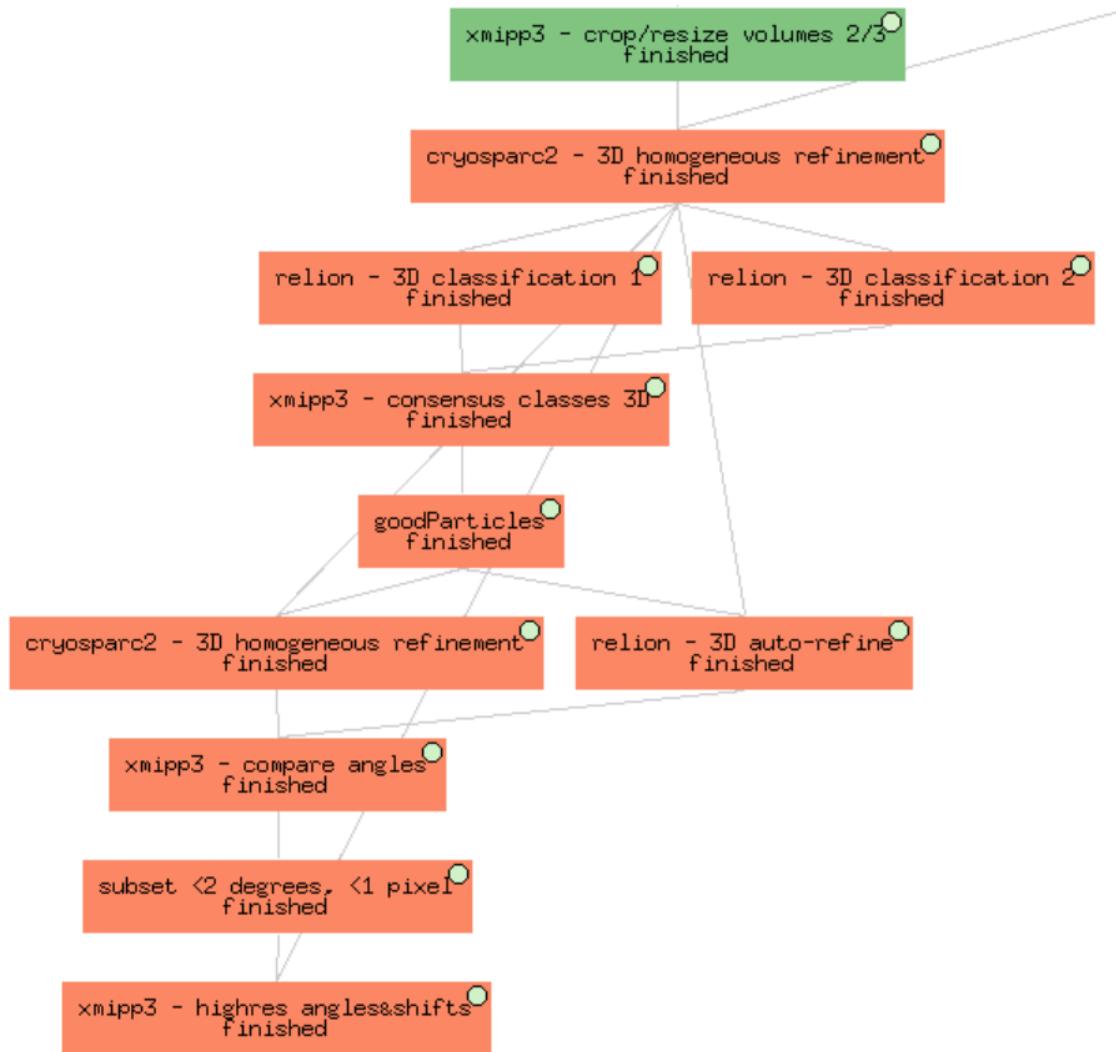


Figure 38: Refinement and 3D Classification.

3D Classification and Refinement are the two last overlapping steps in image processing. They consume the most time and resources with the aim of obtaining a 3D map at the highest possible resolution. This is only feasible if data is homogeneous enough, *i.e.*, if data represent a unique conformation of the specimen.

## Refinement of the initial map

Before starting with the 3D classification properly, a refinement step will be performed with our initial map. The first approach to get a high resolution map in a fully automated manner was performed with the algorithm *Cryosparc homogenous refinement*. This procedure rapidly refines a single homogeneous structure to high-resolution and validate using the gold-standard Fourier Shell Correlation (FSC). We have implemented it in the protocol `[cryosparc2-homogeneous refinement]` (Fig. 39). In the **Input** tap of this protocol form we include the subset of homogeneous particles extracted previously and the initial volume computed, as well in the **Refinement** tap we will choose the symmetry (octahedral), we will select all the options and we will choose in **Noise model** symmetric.

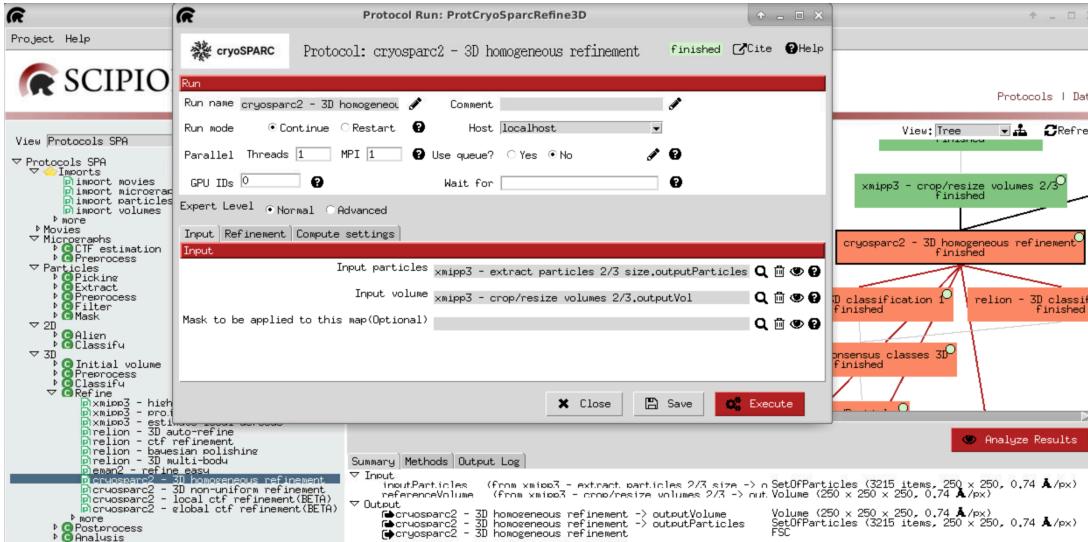


Figure 39: Completing the params of the protocol

`[cryosparc2-homogeneous refinement]`.

After executing, a refined map of 3.3 Å of final resolution was obtained as output, with the same size and sampling rate that we had in the inputs. Press `Analyze Results` to visualize the FSC, the volume or the set of particles. With our initial volume and set of particles we can see that it nicely converge in a good 3D structure, however, we do not know if all the particles that we have used are consistent with that structure

and we also do not know if the parameters of those have been correctly identified. These would be tried to be solved in the next section.

### 3D classification

To continue with the refinement process to obtain a better resolution and to answer the first question, if all the particles belong to that structure, we start executing two independent times the same algorithm of *Relion 3D classification* that we have implemented in the protocol `relion-3D classification` (Fig. 40). In the tap **Input** we include the particles derived from executing the previous protocol `cryosparc2-homogeneous refinement`. The volume derived from this protocol will be the **Input volume(s)** in the tap **Reference 3D map** and will be low-pass filtered by 15Å. The optimization params appear in the tap **Optimization**: 2 **Number of classes** and 25 **Number of iterations**. As **Regularisation parameter T** values as 3-4 are common for 3D classification.

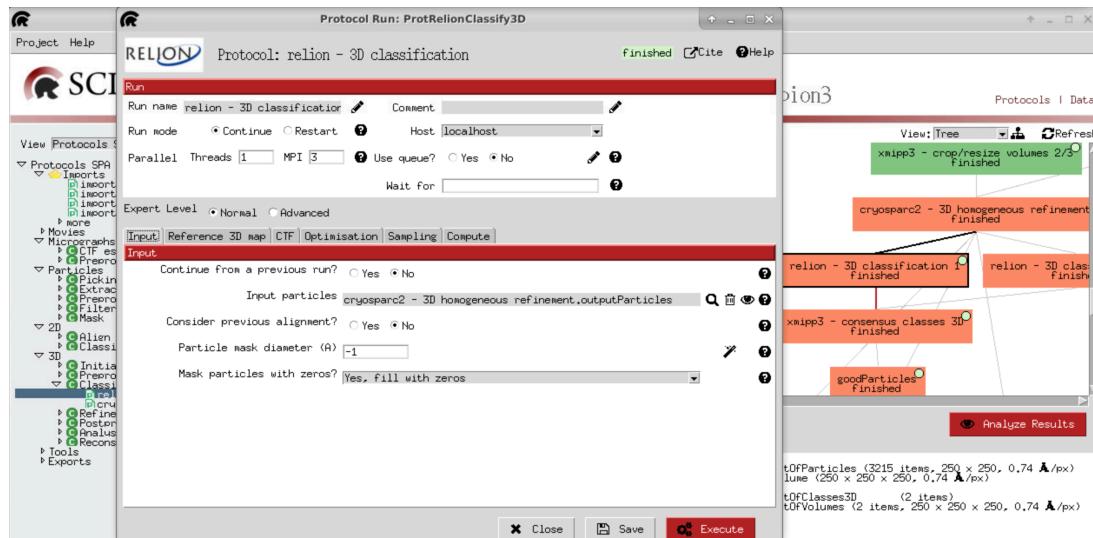


Figure 40: Completing the params of the protocol `relion-3D classification`.

The output of each one of these two `relion-3D classification` protocols are 2 maps with the initial size and sampling rate, reconstructed from different groups of reclas-

sified particles. By pressing **Analyze Results** and **Particles/ Show classification** in Scipion, a table will be opened showing the projection representative of each map and the number of particles contributing to its reconstruction:

- 2,428 and 787 in the first classification.
- 2,460 and 755 in the second one.

The results of the first and second 3D classifications are similar, in both cases the first class contains most of the particles. In order to have a consensus of these results, we execute the protocol **xmipp3-consensus classes 3D** that compares several sets of 3D classes and return the intersection of the input classes (Fig. 41).

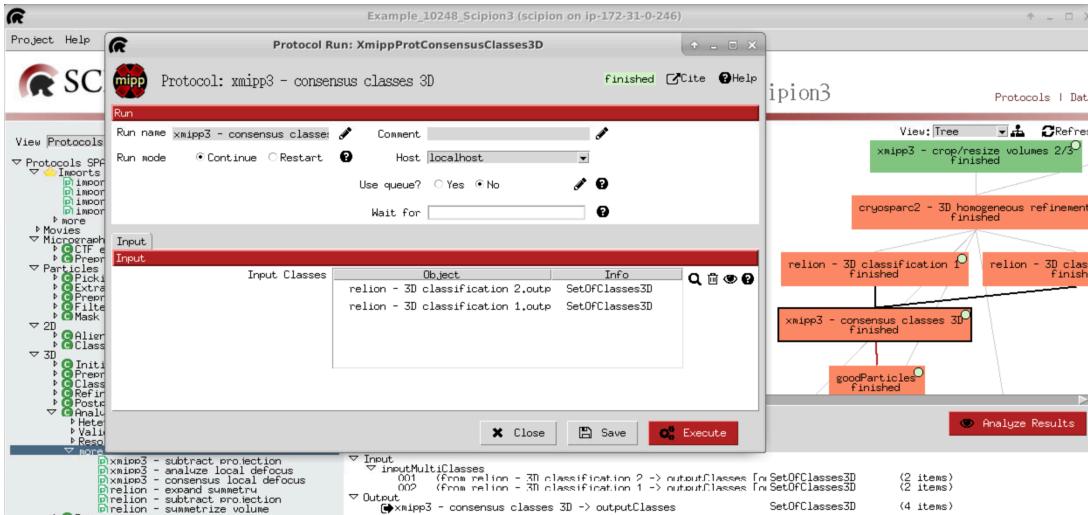


Figure 41: Filling in the params of the protocol **xmipp3-consensus classes 3D**.

By pressing **Analysis Results** you can visualize the 4 intersection 3D classes with the number of particles assigned to each one. The first of these classes derive from about 1,944 particles (aprox. 60% of total particles). Once inspected the different classes, by selecting the classes that we are interested in (only the first one) and pressing **Particles**, a new subset of 1,944 particles will be created included in the box **goodParticles**.

## Refinement

For a further refinement step, we will use again the protocol **cryosparc2-homogeneous refinement** (Fig. 42) and *Relion* algorithm **3D auto-refine** that we have implemented in the protocol **relion-3D auto-refine** (Fig. 43). As input, we include the set of good particles that were consistently assigned to the same class and the initial volume computed in the step before.

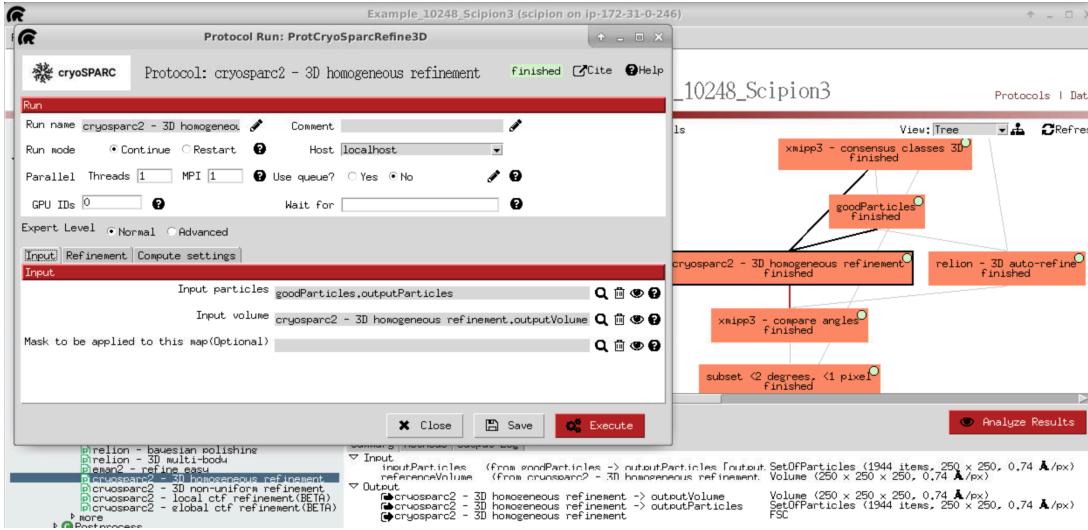


Figure 42: Completing params of protocol **cryosparc2-homogeneous refinement**.

The algorithm of *Relion auto\_refine*, is based on an empirical Bayesian approach. This procedure employs the so-called gold-standard Fourier Shell Correlation (FSC) to estimate the resolution. Combined with a novel procedure to estimate the accuracy of the angular assignments, the algorithm converges. In the **Input** tap of this protocol form we include the subset of good particles selected previously. The initial volume in the **Reference 3D map** tap and as we trust our previous map we do not need to filter it much so we apply a **Initial low pass-filter (A)** of 10.0 Å. This tap gives you the possibility of using **Reference mask (optional)**.

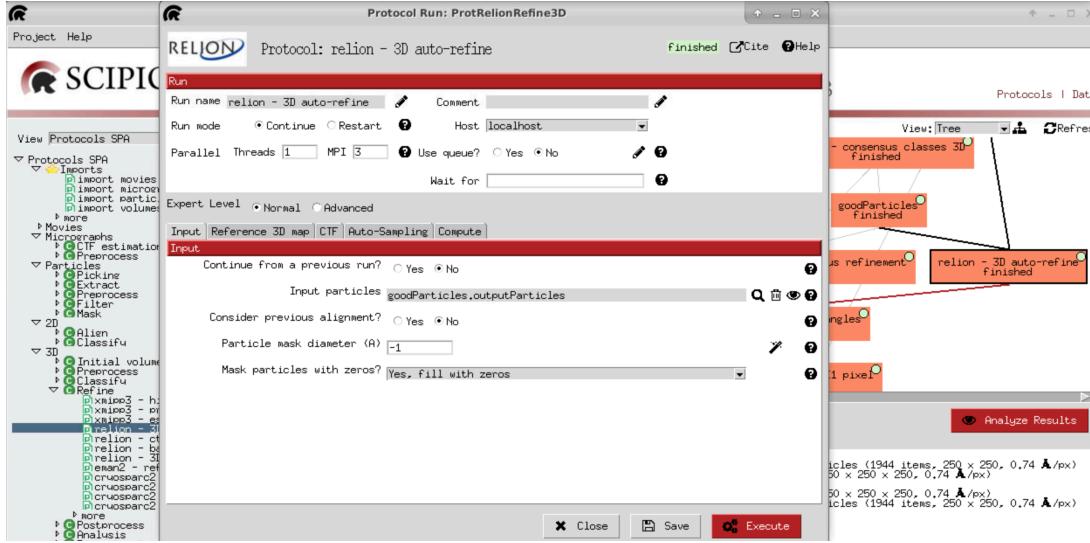


Figure 43: Completing params of protocol `relion-3D auto-refine`.

There are three questions in the tab CTF:

- Do `CTF correction?`, set to `Yes` to perform full phase + amplitude CTF correction.
- Has `reference been CTF-corrected?`, set to `No` because the Fourier transforms of the reference projections are not multiplied by the CTF in the first iteration.
- Do `manual grouping ctf?`, set to `No` because we have enough number of particles that we do not need to group them.

The `Angular sampling interval (deg)` option in the tab Auto-Sampling will be used only in the first few iterations. Later, the algorithm will automatically increase its value until convergence. For symmetries lower than octahedral or icosahedral we use the default values of `Angular sampling interval (deg)` and `Local search from auto-sampling (deg)`.

After executing both protocols, by looking to the resolution FSC plots, a refined map of 2.8-3Å of final resolution was obtained as output, with the same size and

sampling rate that we had in the inputs. Press **Analyze Results** and by comparing the output volumes same slices, we can visually see the difference between the two maps obtained by the two protocols. In this particular case, we observe haze on top of relion output volume and this typically is coming from badly align particles. To identify these badly align particles, we can compare the angular assignment of relion and cryosparc protocols using **xmipp3-compare angles** protocol (Fig. 44). As we have an octahedral symmetry we do not have to align the volumes. If needed, you can use **xmipp3-align volume and particles** protocol.

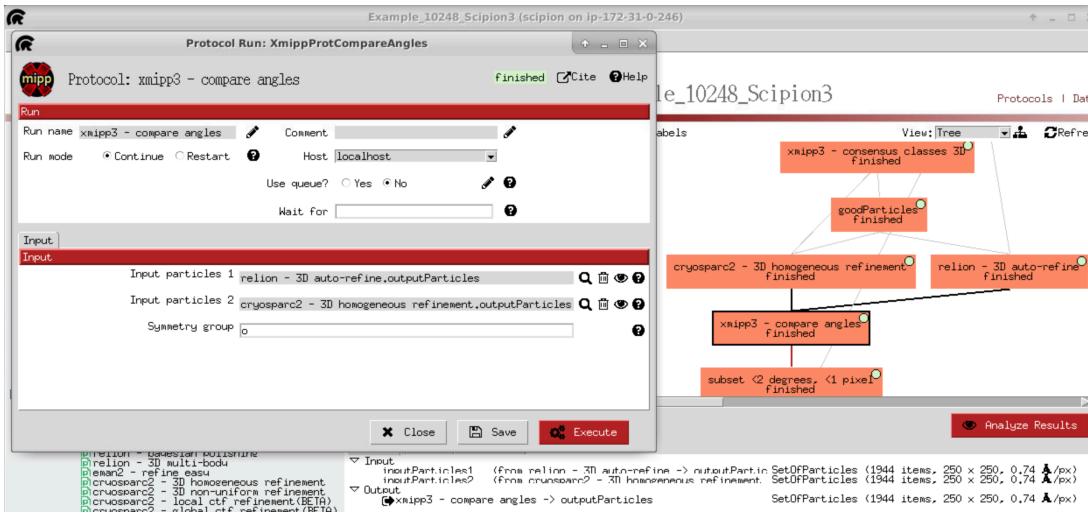


Figure 44: Completing params of protocol **xmipp3-compare angles**.

The results given is a set of particles sorted by the angular difference, and for every particle we have its angular difference (average of the Euler angles) and the shift difference. As they are sorted, if we plot the column of angular difference we can observe that most are around 1 degree difference. However, there are others that the angular error is much bigger and those are the particles that their angular assignment is not so convincing. If we select the ones below 2 degree differences, sort by angular shift, repeat the process and select those below 1 pixel shift, we will create a new subset (1,903) whose particles have an angular assignment consistent between two algorithms (relion and cryosparc) with different implementations. Then, we proceed to make the angular reassignment of this new subset of particles to our refined

map of protocol `cryosparc2-homogeneous refinement` (Fig. 39) using the *Xmipp* algorithm `highres` (?) that we have implemented in the protocol `xmipp3-highres` (Fig. 45).

This method computes a weight for each particle and performs both global or local alignment. Iterations can be performed one by one, removing particles that worse fit the map from one iteration to the next one. This 3D refinement protocol uses as input the refined map obtained from *Cryosparc homogenous refinement* and the subset of particles created before. The **Symmetry group** has been also included in the Input tap. In the **Angular assignment** tap we choose Local as **Image alignment** as we have a previous estimated assignment computed before, and 1 **Number of iterations** with 2 Å as **Max.** **Target Resolution.** In this particular case, as we have few data it is not recommended to optimize the scale, the grey values or the defocus.

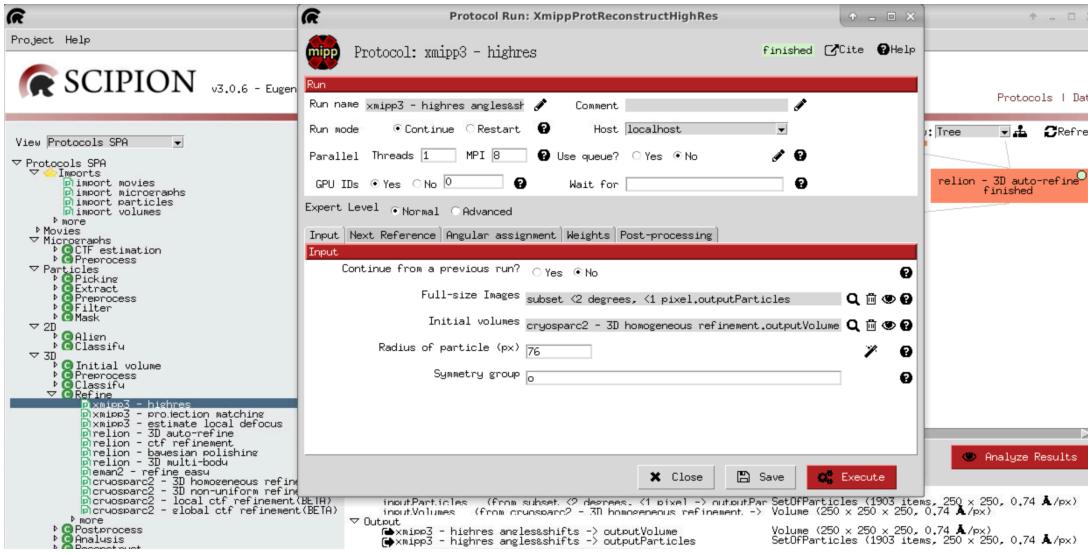


Figure 45: *Xmipp* `highres` map local refinement.

This protocol generates one map as output with the initial size and sampling rate. Press **Analyze Results** to check the results. Particles and map can be visualized. We can observe that no particles have been rejected. The output resolution by displaying the FSC, is around 3.2Å. We can select particles according to the value of

the `_xmipp_cost` param and we can observe the angular distribution. The output volume obtained looks very good, we get rid off the haze and if we can appreciate more details in the 3D structure. In the next section, we will see how to validate our processing and to enhance the volume obtained with this workflow.

For more information:

- **Video tutorial:** [https://www.youtube.com/watch?v=ial950ZXU0&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=29](https://www.youtube.com/watch?v=ial950ZXU0&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=29).
- **Theoretical lecture:** [https://www.youtube.com/watch?v=taCREkFAPoE&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=36](https://www.youtube.com/watch?v=taCREkFAPoE&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=36).

## 10 Volume enhancement and validation

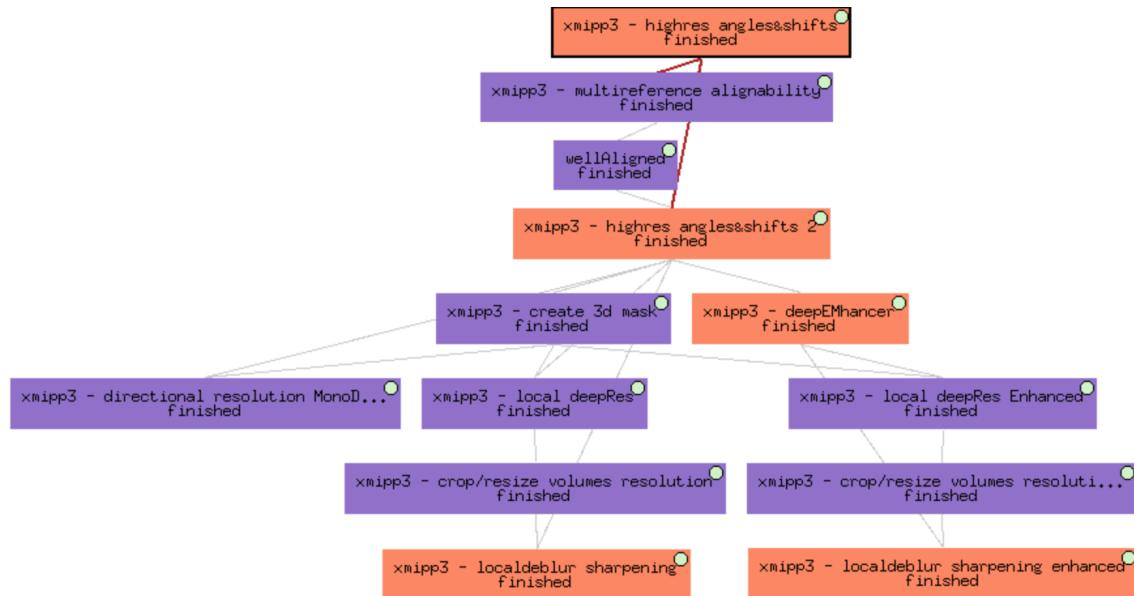


Figure 46: Enhancement and Validation.

In this last part of the tutorial, we will verify if what we constructed is reasonable and for this, we have some tests that we can do for this number of particles.

The first one, *Xmipp* algorithm `multireference alignability` used in protocol `xmipp3-multireference alignability` (Fig. 47), is going to assess the output map of previous step regarding soft alignability and overfitting of particles and 3D map. The input of the protocol requires the map and particles generated in the last refinement step.

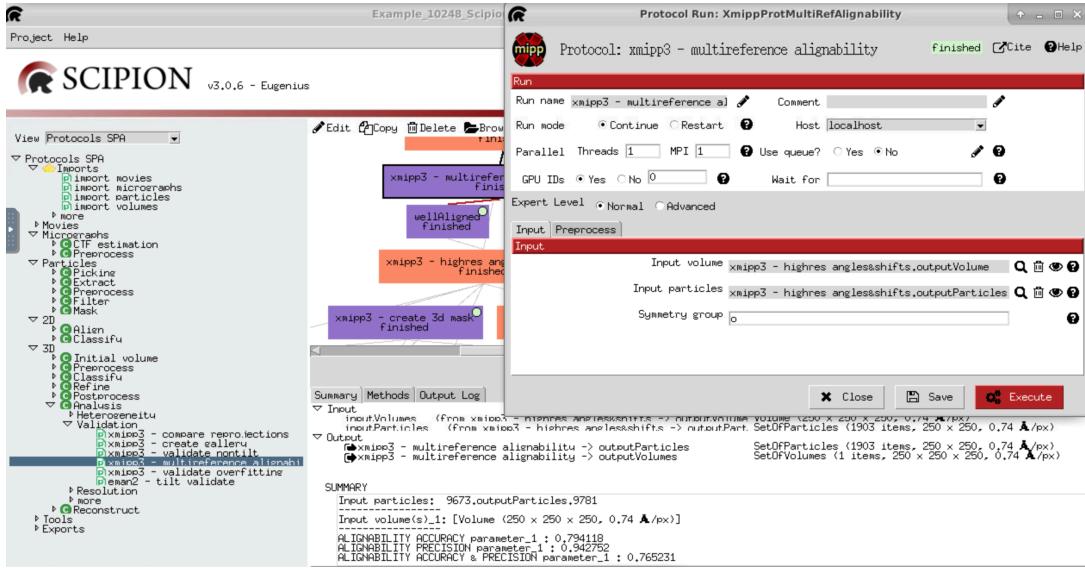
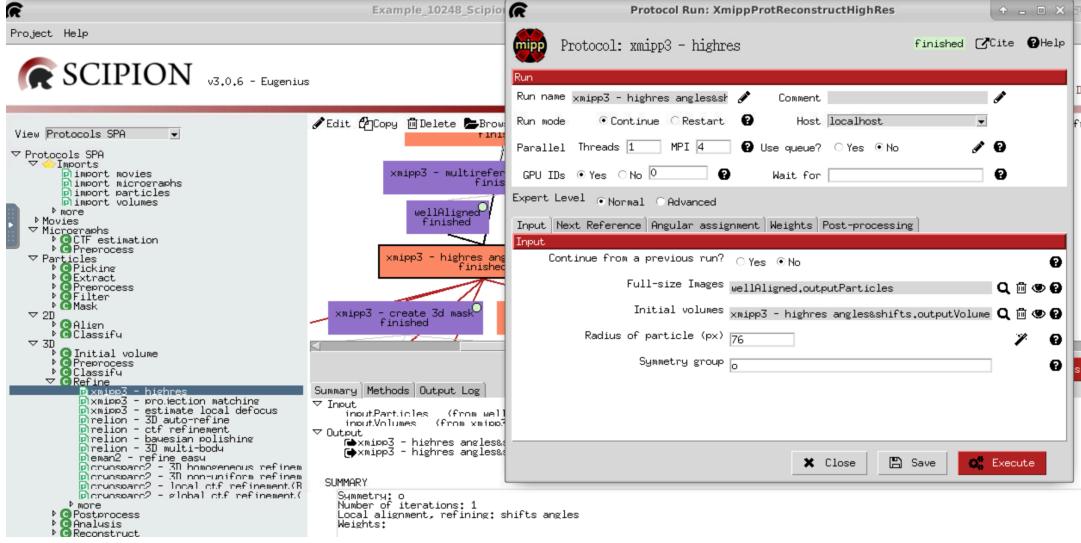


Figure 47: Completing the form of protocol `xmipp3-multireference alignability`.

The output values of particle alignment, precision and accuracy, generated by `xmipp3-multireference alignability` (press `Analyze Results`) allow us to discard particles with worse alignment. In this case, 446 particles (23.4% of the total input) are rejected. A new subset of 1,457 particles will be created included in the box `wellAligned`. In order to improve the refined map resolution, the remaining 1,457 particles will be used to perform the local refinement iteration with *Xmipp* `highres` algorithm. The protocol params will remain unchanged compared with the first implementation.

Figure 48: *Xmipp highres*.

The sampling of the output map is the same than in the previous implementation, despite the selection of best aligned particles. We have thus achieved convergence and we can compute the global or local resolution for sharpening the map. Map sharpening contributes to increase signal at medium/high resolution, we recommend to perform this map preprocessing step before tracing the atomic model of cryo-EM 3D maps. To accomplish this task a couple of automatic alternatives are available in Scipion: a) local sharpening method independent of initial model, based on local resolution estimation `xmipp3 - localdeblur sharpening` (Ramírez-Aportela et al., 2018), b) deep learning-based sharpening approach `xmipp3 - deepEMhancer` (Sanchez-Garcia et al., 2020). Although both sharpening methods display good results, these are not identical but complementary since LocalDeblur maximizes specially details like the secondary structure, whereas DeepEMhancer maximizes connectivity, favoring the fair tracing of the molecule skeleton.

## Map sharpening: a) Local sharpening method independent of initial model, based on local resolution estimation

At this point, we may define a mask, and this mask will help directional resolution and local resolution algorithms to concentrate on which area they should analyze. Basically it is used to see the distribution of the noise and know where is the noise to get rid of it. For this we will use the protocol `xmipp3-create 3D mask`, the mask can be created with a given geometrical shape (Sphere, Box, Cylinder...) or it can be obtained from operating on a 3D volume or a previous mask. In this case it comes from the previous 3D volume as you can see in the next figure Fig. 49.

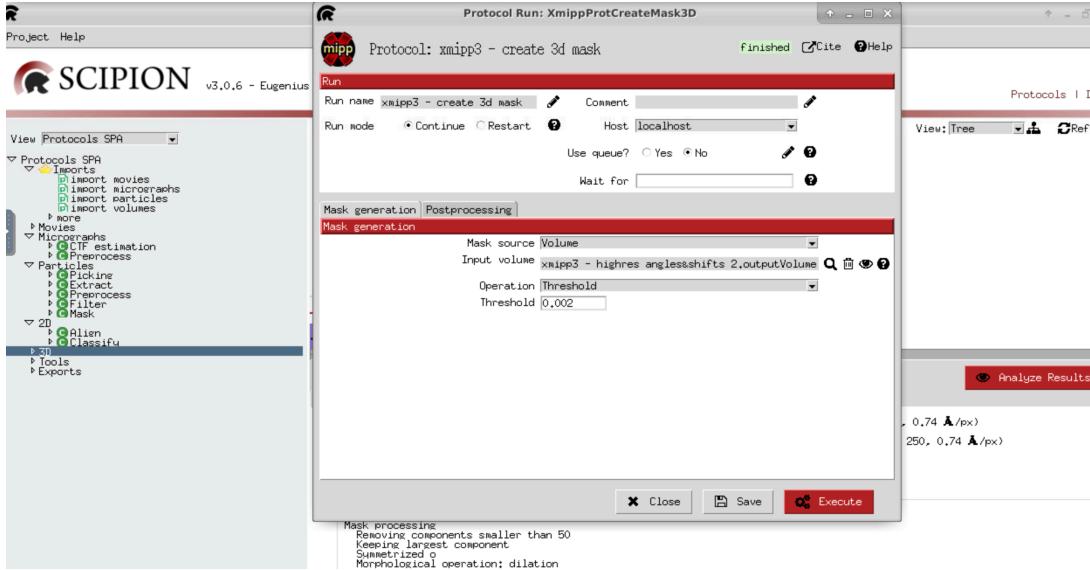


Figure 49: *Xmipp create 3Dmask*.

Once the map mask is created, the protocol `xmipp3-local deepRes` (Fig. 50) can be completed to get the estimation of local resolution. We have to include the input volume, as well as the binary mask and finally, based on the map resolution ( $3.2 \text{ \AA}$ ), select the default resolution range between  $2.5 \text{ \AA}$  and  $13 \text{ \AA}$ . Given a map the protocol will assign local resolutions to each voxel of the map.

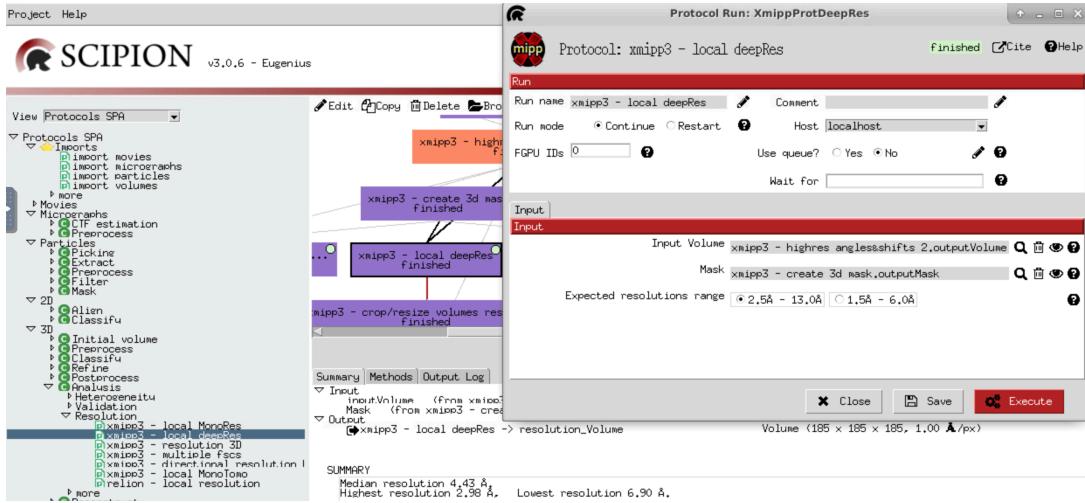


Figure 50: *Xmipp local deepRes*.

Press **Analyze Results** to see the menu of results, among other views, it shows the histogram of local resolutions and the resolution map in ChimeraX. The histogram of resolutions, which displays the number of map voxels showing a certain resolution, allows us to conclude that the majority of voxels evidence a resolution between 4.2 Å. The resolution map shown by ChimeraX details the resolution of each voxel by colors.

Resolution is not only local but also directional, in different directions you can have different resolution. To compute this difference, we used **xmipp3-directional resolution MonoDir** protocol (Fig. 51).

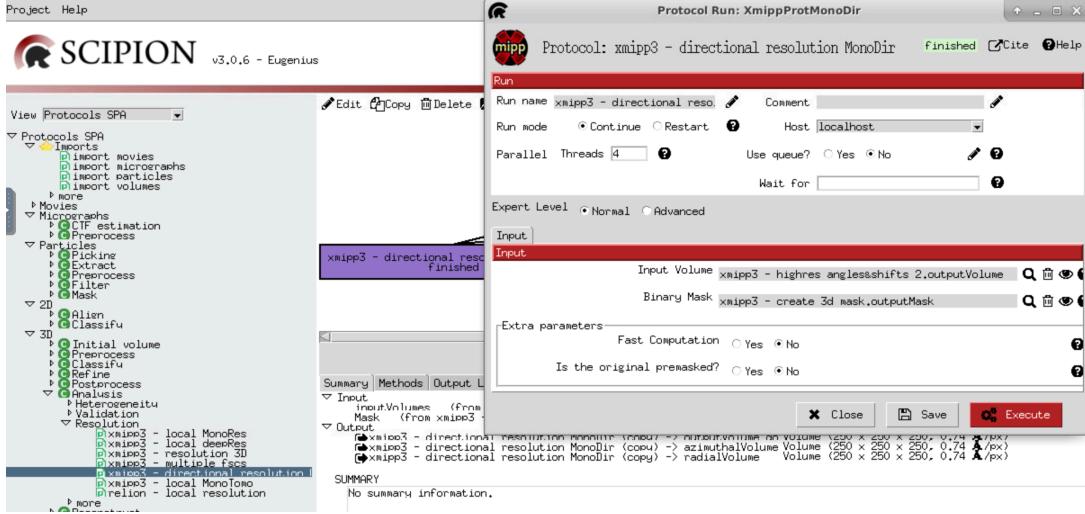
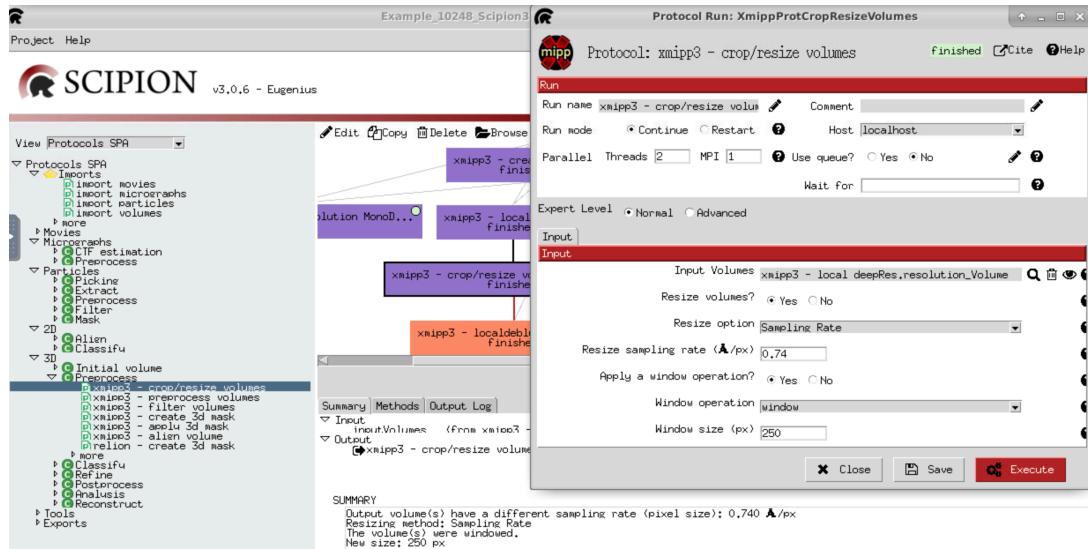


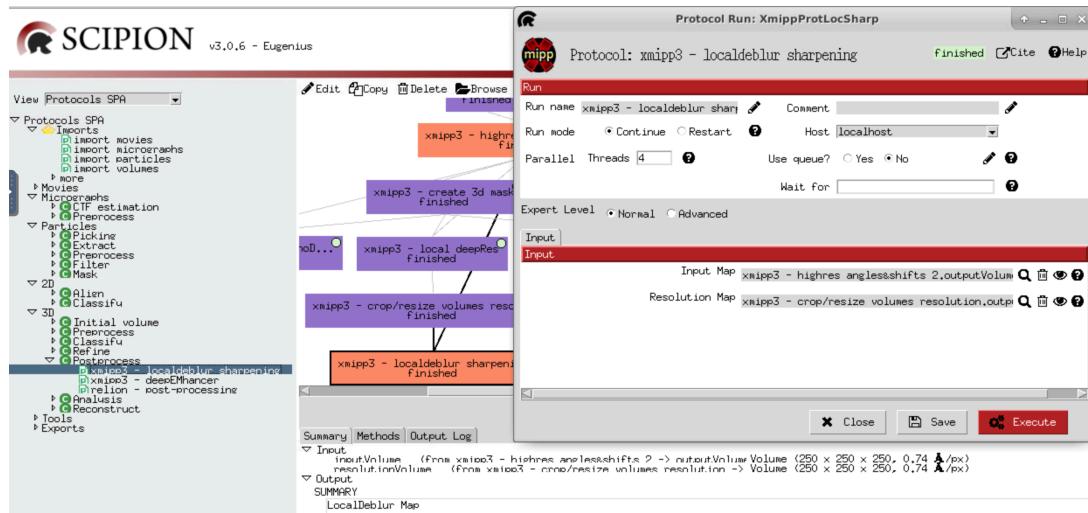
Figure 51: *Xmipp* directional resolution MonoDir.

The interesting views of **Analyze Results** are, the Half Interquartile Range between different resolutions. In this case we can appreciate that the region of the structure looks very consistent and all the inconsistency comes from the surrounding. Another interesting plot is the **Directions histogram**, basically how many voxels have their maximum resolution along that direction. In this case it looks uniform and this can be confirmed by looking to the **radial averages** (radial averages of the resolutions in different directions).

Back to the sharpening of the map, local resolution values of the input map allow to compute the sharpened map by **xmipp3-localdeblur sharpening** protocol, which implements an iterative steep descending method that not requires initial model. To accomplish this step, first we have to resize our resolution volume since **xmipp3-local deepRes** computed a resolution volume with different dimensions and sampling frequency than our actual volume. To do so, we use **xmipp-crop/resize** protocol on the volume resolution to obtain 250x250x250 dimensions and 0.74Å /px sampling rate (Fig. 52).

Figure 52: *Xmipp* crop/resize.

LocalDeblur takes advantage of the map local resolution to increase the signal, so as input of **xmipp3-localdeblur sharpening** protocol we should include the starting map and the map of resolution values, maintaining the default values for the rest of parameters (Fig. 53).

Figure 53: *Xmipp* localdeblur sharpening.

After three iterations, the sharpening algorithm reached the convergence criterion, i.e. a difference between two successive iterations lower than 1% and stopped. The three maps obtained in the respective iterations can be observed with double-clicking the black output arrow shown in the summary. Additionally, by clicking **Analyze Results** the sharpened map obtained after the third iteration, i.e. the last map, can be also visualized and compared with the initial one in ChimeraX.

## Map sharpening: b) deep learning-based approach

DeepEMhancer is an alternative automatic sharpening method based on deep learning (Sanchez-Garcia et al., 2020), implemented in Scipion in the protocol `xmipp3-deepEMhancer`. Given a map the protocol performs automatic deep post-processing to enhance visualization. We have to include the map, the type of normalization desired and the deep learning mode to use, in this particular case tight target (Fig. 54).

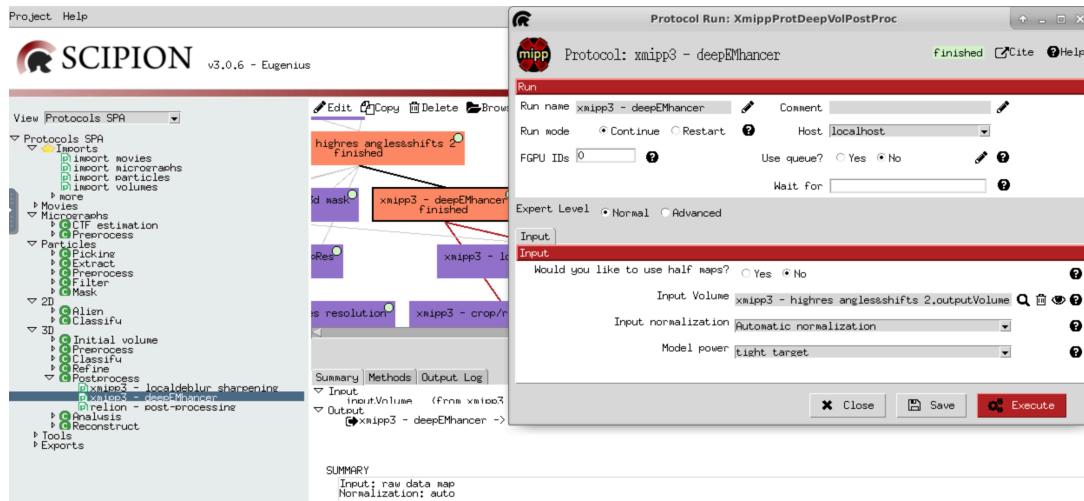


Figure 54: *Xmipp* deep EMhancer.

After executing the protocol with **Analyze Results**, we can check the results. ChimeraX viewer will open and show the sharpened map compared with the initial one.

For further refinement, we can compute the local estimation as it was done before with `xmipp3-local deepRes` (Fig. 55), then the resolution volume output is resized again with `xmipp-crop/resize` protocol (Fig. 56) and we can restore the map running `xmipp3-localdeblur sharpening` protocol (Fig. 57).

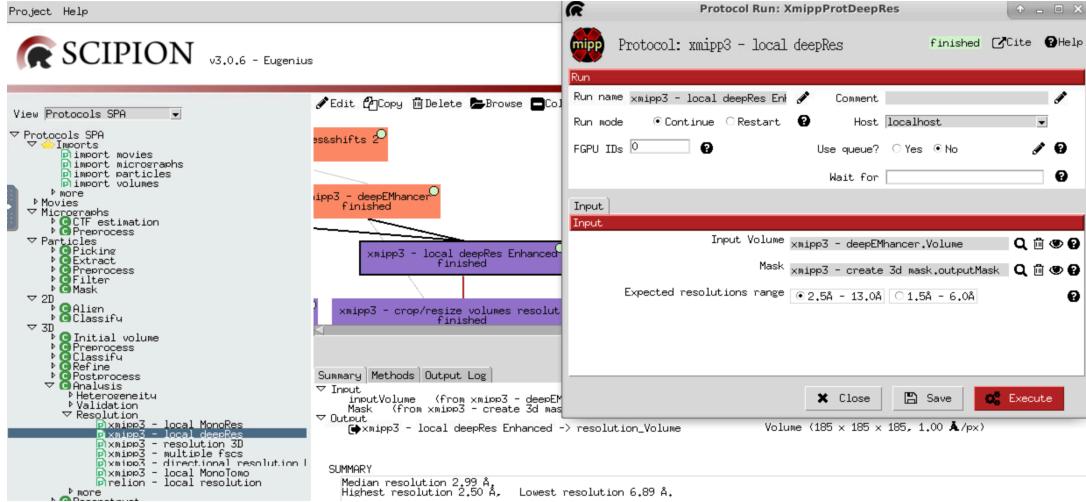


Figure 55: *Xmipp local deepRes Enhanced*.

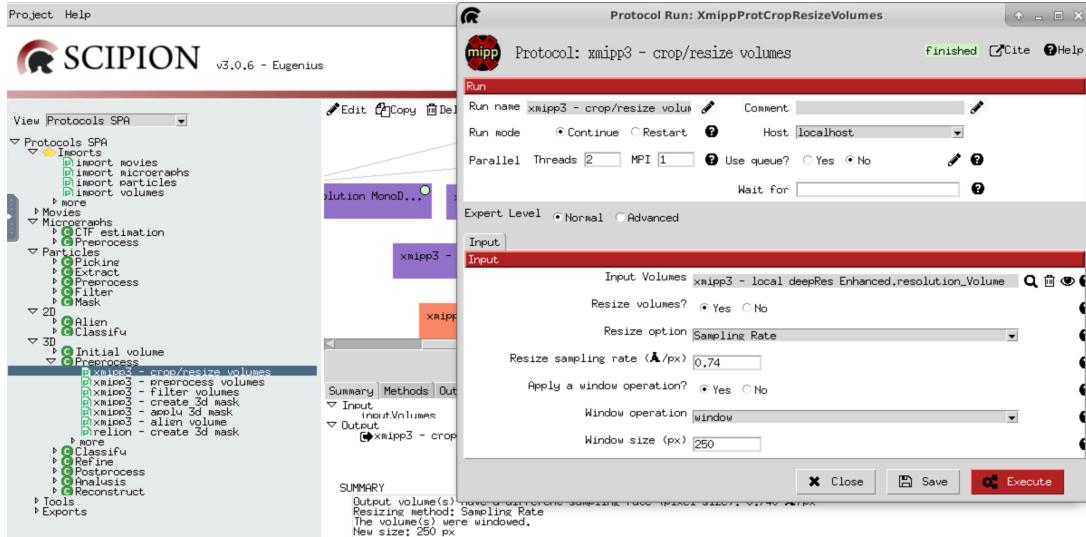


Figure 56: *Xmipp crop/resize volume resolution*.

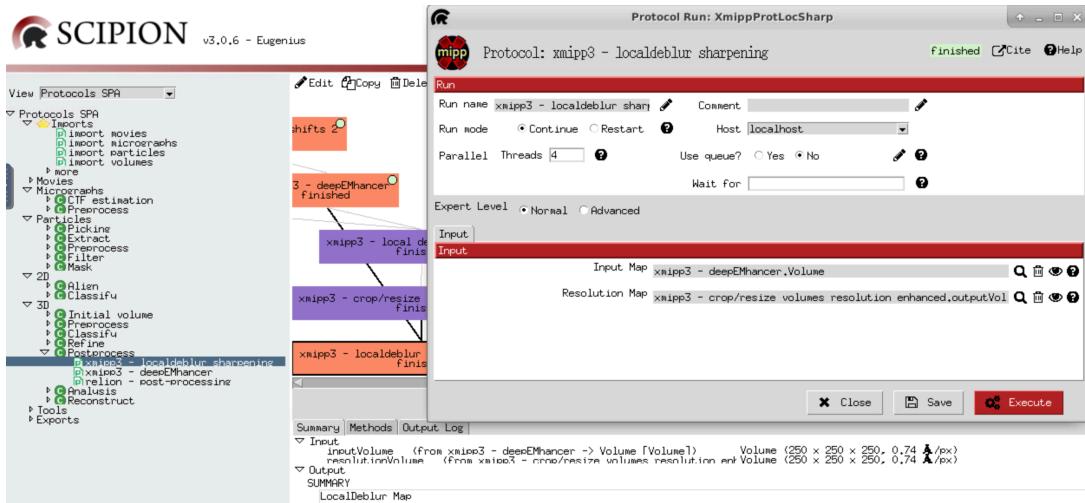


Figure 57: *Xmipp* localdeblur sharpening.

After executing these protocols, we can observe that most of the results were improved, we increased the resolution and we obtained a further refined 3D structure. This tutorial is a good example that with good data we can reach high resolution in a short time.

For more information:

- **Video tutorial:** [https://www.youtube.com/watch?v=d2Lm2H0T7cc&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=30](https://www.youtube.com/watch?v=d2Lm2H0T7cc&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=30).
- **Theoretical lecture:** [https://www.youtube.com/watch?v=gglZdNEskJs&list=PLQjWIcrmtc4JjyC-\\_BM99\\_XW-VsDa4\\_i3&index=37](https://www.youtube.com/watch?v=gglZdNEskJs&list=PLQjWIcrmtc4JjyC-_BM99_XW-VsDa4_i3&index=37).