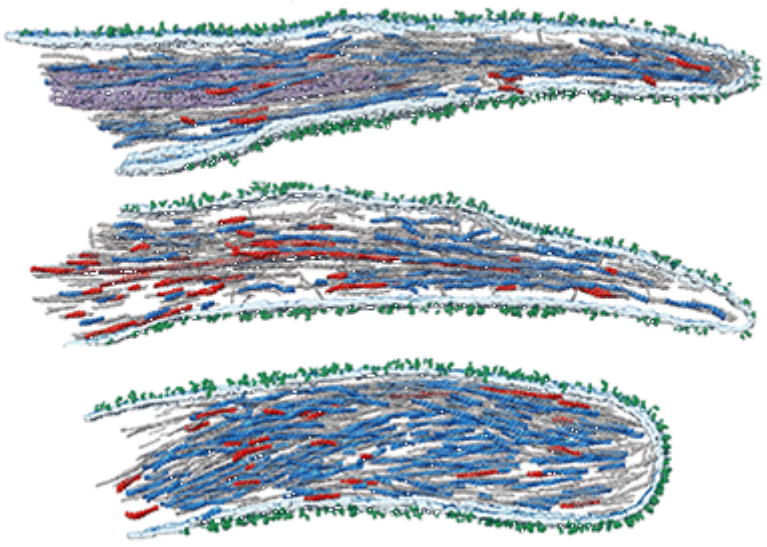
Local Resolution Analysis with MonoTomo



Scipion tutorial series

National Center for Biotechnology
BioComputing Unit

Index

Workflow Summary

The main goal of this tutorial is to illustrate how to analyze the local resolution of a reconstructed tomogram by means of MonoTomo. Therefore it does not pretend to achieve high quality reconstructions or analyze the biology of the sample, it only attempts to illustrate some possibilities that ScipionTomo offers in image processing. Here is a brief summary of the steps that will be done:

1. **Import Tilt series movies.** The Tilt series movies images are loaded, with all acquisition parameters of the microscope session. The result will be a Scipion object called *Set of Tilt Series Movies*.
2. **Movie alignment:** The imported *Set of Tilt Series Movies* is processed and the movie stacks are aligned leading to “tilted-micrographs”. However, the original *Set Of Tilt Series Movies* will be split in two subsets, called odd and even, with the corresponding odd and even frames of the original *Set of Tilt Series Movies*. The result will be three Scipion objects called *Set of Tilt Series*.
3. **Assign alignment**. The Sets of Tilt Series odd and even must present the same alignment in order to estimate the local resolution of the tomogram. To do that, the transformation matrix (alignment parameters) will be assigned to ensure the same alignment parameters in both Sets Of Tilt Series (odd and even) and equal to the Full Set of Tilt Series (with all frames)
4. **Tomogral reconstruction.** The reconstruction of the tomogram with the aligned Tilt Series with the same reconstruction method for the two Set of Tilt Series (odd and even)
5. **Local resolution estimation.** Finally, the local resolution is estimated with MonoTomo making use of odd and even tomograms.

1. Getting Started.

About Scipion

Scipion is an integrative framework for image processing of cryoEM data, in particular: Single Particle analysis, modelling and electron tomography data. It incorporates as plugins the most highlighted software packages as they are: RELION, cryoSPARC, IMOD, XMIPP, EMAN, or DYNAMO among many others. Scipion provides a standard of communication to allow the interoperability between packages, full tracing of the image processing, and analysis/visualization tools.

> **Important**: Scipion does not modify the original binaries of the integrated software, it just provides a wrapper, where those binaries can be executed in an user-friendly environment.

If you use Scipion we really appreciate it if you cite it (J.M. de la Rosa-Trevin et. at. 2016)
For more information visit our website: http://scipion.i2pc.es/

Scipion installation

To use this tutorial Scipion must be installed in the machine. The instructions about how to install Scipion can be found in the next link

https://scipion-em.github.io/docs/index.html

Plugin Requirements

Scipion is an integrative framework composed of a broad variety of software packages called Plugins. The installation of the Plugins is independent on the Scipion, and they can be installed with the Plugin Manager, or alternatively by command line.

This tutorial will make use of the next ones:

- Scipion-em-tomo
- Scipion-em-imod
- Scipion-em-jjsoft
- Scipion-em-xmipp
- Scipion-em-xmipptomo
- Scipion-em-chimera

Data Download

The used data for this tutorial can be found in the next link, which contains a small subset of the EMPIAR entry EMPIAR-10364 (E. colli minicells - Burt et.al. 2020)

Download the data tutorial here, in the **folder (day 1):**

http://scipion.cnb.csic.es/downloads/scipion/data/tutorials/tomography/

Contact with us

We want to hear from you! Any comment, question, or complaints regarding this tutorial, the use of Scipion or xmipp can be sent to these mails: scipion@cnb.csic.es, xmipp@cnb.csic.es.

Also you can follow us in our social media

Twitter: https://twitter.com/instructi2pc

Tutorials about Scipion use, and cryoEM seminars can be found in your YouTube channel

Youtube: https://www.youtube.com/user/BiocompWebs

We also have a **slack channel** where our most active members keep in touch daily. You can request access on scipion@cnb.csic.es

1. Import data

First step is to import data, to have the raw images in the Scipion project. Scipion provides many different kinds of imports like SetOfTomograms, SetOfCTF3D, SetOfCoordinates, or SetOfSubTomograms, among others. In this tutorial, it will be shown how to import a **SetOfTiltSeriesMovies.**

> **Note**: The import is the unique action (leaving out some exceptions) where the user has to interact with files. Once the data is imported in the project, Scipion carries out the files management during the image processing, enhancing the usability.

Import of the Tilt series movies

The import of tilt series movies is carried out by means of the Scipion protocol called **Import tilt-series movies**, located in the left-side panel in the **Tomography tab**. Alternatively, it can be found by searching in the search box of Scipion (**Crtl+F**), and typing “import tilt-series movies”. The Scipion search box also allows searching by keywords like import, or tilt, or tilt series… and therefore identifying the protocol in the deployed list of protocols.
The form of the import looks like Figure 1.1.

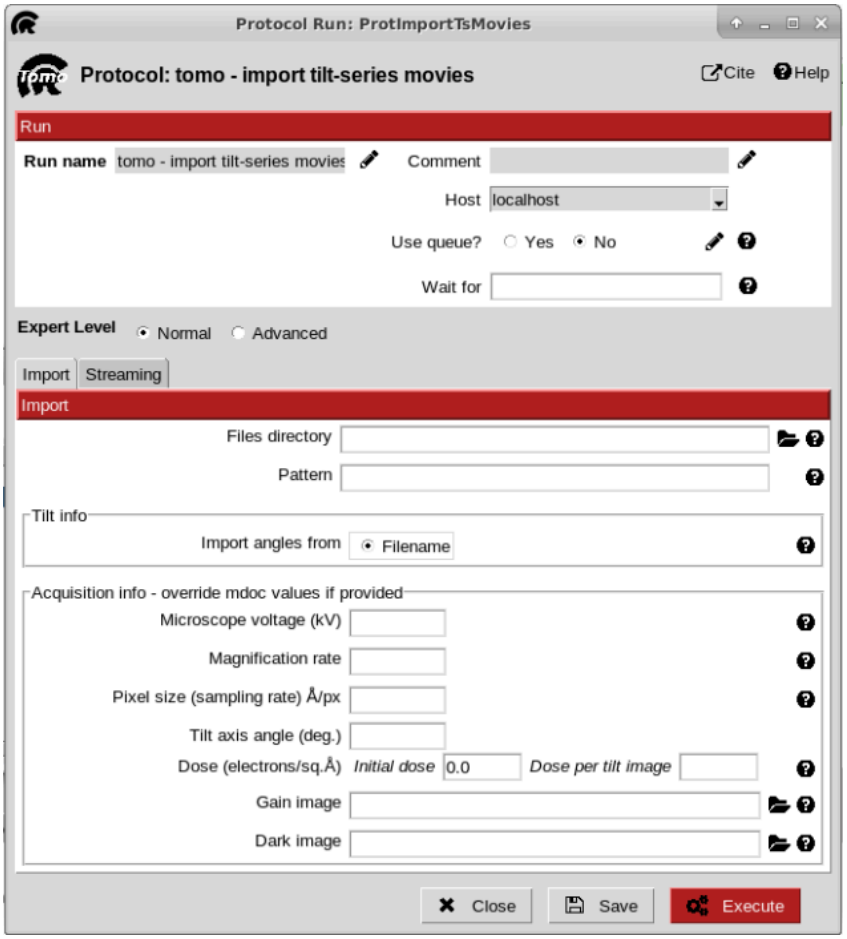


Figure 1.1. Import tilt series movies with the form empty

The **import tilt series movies** has two different possibilities to load the data in the Scipion project. The difference of the two different options depends on the input file selected from which the images binaries and acquisition information is provided. The first and most classical approach is to use a pattern with some keywords ({TS}, {TO}, {TA}), from which scipion will get the image binary location, an identifier ({TS}), a order of acquisition ({TO}), and finally a tilt-angle ({TA}), among some other necessary information coming from the rest of the fields in the form, as sampling rate, tilt axis angle, or voltage. This import relies on the distribution of the different tilt movies stacks in different files.

However, Scipion also offers a second option that relies on the .doc files for the information input, which is a more convenient and flexible procedure for the user. These files are generated during the acquisition session, and contain all microscope parameters, the most important and relevant to the image processing: pixel size, tilt axis orientation, defocus or dose among others. It also contains many other parameters like the voltage of the electron gun, magnification, exposure time or spot size of the beam…that in general are not relevant for the image processing.

> **Note**: Scipion team highly recommend importing with mdoc files.

In this tutorial, we will use the import tilt series movies using the mdoc files. To do that we select in the **files directory box** of the form the folder where the mdoc files are (see Figure X). The option of importing with mdoc files is carried out by typing *.mdoc in the **pattern box.** In the moment when the user type *.mdoc the options of **Tilt info box** should disappear (see the figure below with the form fulfilled). In the help button of the pattern box, tips about how to introduce the pattern are explained. In this tutorial the user will type *.mdoc, meaning that all files with extension mdoc of the **Files directory** folder will be imported. Finally, just click on execute and Scipion will import the tilt series.

> **Note**: Importing with mdoc the parameters of the **Acquisition info box** are not required. However, they are enabled to be fulfilled just in case the information of the mdoc file could be not fully correct, or even if the user wants to refine some acquisition parameters.

This is a very illustrative dataset (according to the previous note), where not all the needed information is provided in the mdoc file, specifically, the information related to the dose. Thus, it will be necessary to **input manually the dose value** as it is shown in the following Figure 1.2.

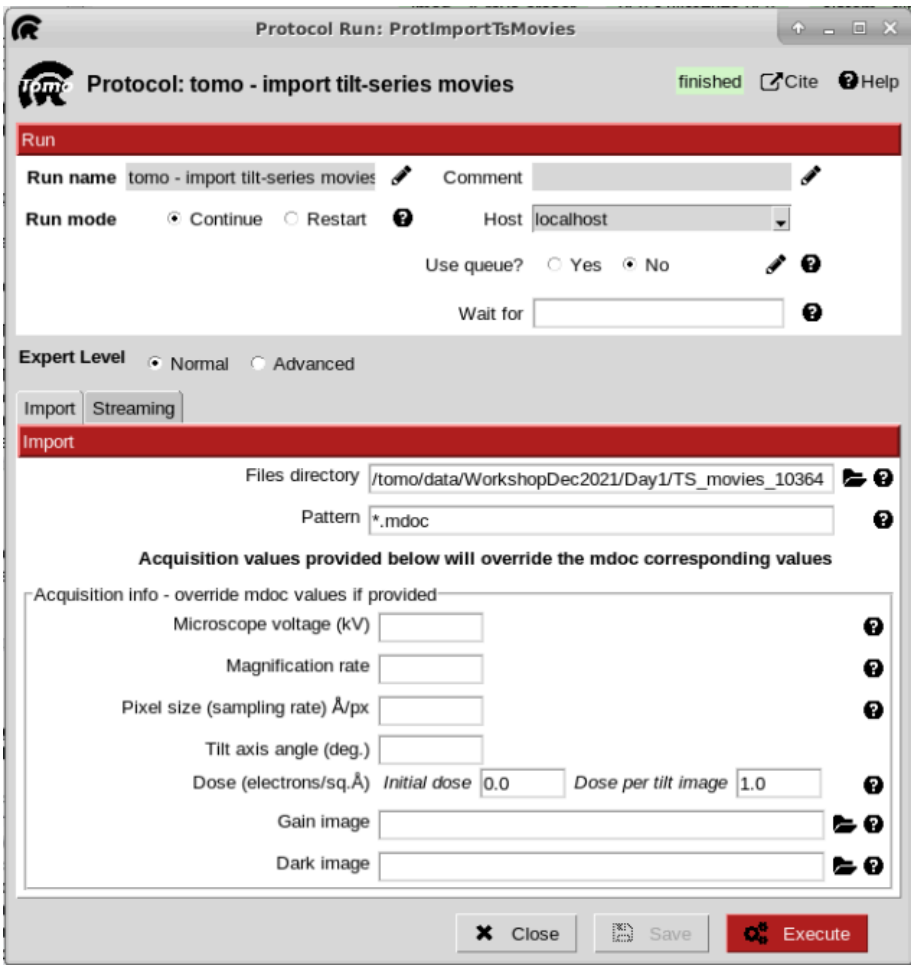


Figure 1.2. Tilt series movies import from .mdoc file. The empty boxes are not mandatory, they can be empty.

The result of the import will be a SetOfTiltSeriesMovies in the lower part of the Scipion main window (see Figure). In this case, 3 tilt series composed of 61 images with dimensions 3838x3710 have been imported with pixel size 2.24A and dimensions. Moreover, Scipion points out that they were imported from mdoc files as well as the path of the mdoc folder.

2. Movie alignment

In this section, some processing tools will be applied to the previous set of tilt series movies in order to align the stack of movies into “tilted micrographs”. Thus, we will convert the **input set of tilt series movies** to an **output set of tilt series**. For this purpose **Flexalign** from **Xmipp** will be used.

xmipptomo - FlexAlign

FlexAlign Reference: Strelak 2020
Plugin: scipion-em-xmipptomo

The movie alignment protocol used is the **tilt series Flexalign** from **xmipptomo** plugin. In order to carry out this process inside the Scipion framework, we can search in the left panel in the section Tilt-series movies or directly search between all the protocols using the **Ctrl+F** command, or **in the left side panel Tilt Series movies-> FlexAlign**.

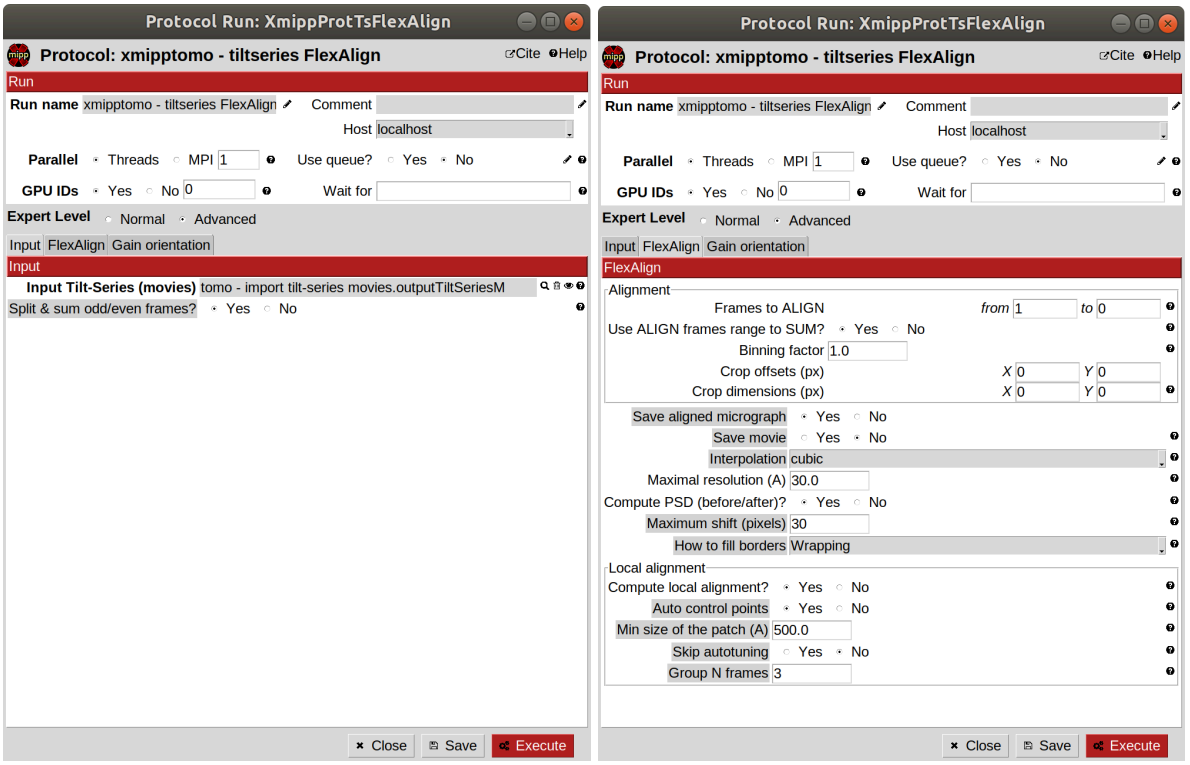


Figure 2.1: **xmipptomo - tilt-series Flexalign** form, with the execution values (default values in this case)

For the execution of this protocol no modifications are made to the **default parameters**, aligning all the frames that conform each “tilt stack” of movies and without binning the output images. Only it will be necessary to input the set of tilt series movies previously imported.

Figure 2.2: State of the workflow. Result in the Scipion three of the xmipptomo - tilt-series Flexalign.

Note: As it can be seen in the summary section the input of the protocol is a SetOfTiltSeriesMovies and the output SetOfTiltSeries. **This is the only step in the pipeline where a protocol inputs a SetOfTiltSeriesMovies**.

3. Tilt series alignment

Once the tilt-series are properly imported and preprocessed into Scipion it is time to perform the last step before the reconstruction of the tomogram, the tilt-series alignment. This task is a 3 process step that is compose of a previous prealignment of the tilt-series based on cross-correlation, followed by the construction of the landmark (or fiducial) models, that will be finally used to obtain the properly aligned tilt-series.

Tilt-series prealignment

Imod References: Kremer 1996, Mastronarde 2017

The first step in the alignment process is the **prealignment of the tilt-series**. In this step only the translational alignment (shifts) is solved (no angle correction yet). This calculation is performed by cross correlating the successive images from the tilt-series and stretching the images with the larger tilt angle perpendicular to the tilt axis (cosine stretching).

In order to carry out this process inside Scipion environment, the **binned SetOfTiltSeries** generated in the previous step will input the protocol, **imod - Xcorr prealignment**, from the imod plugin, that can be found in the left menu under **Tilt-series > alignment> Xcorr prealignment**, or searching with the **Ctrl.+F** hotkey, along with the following parameters.

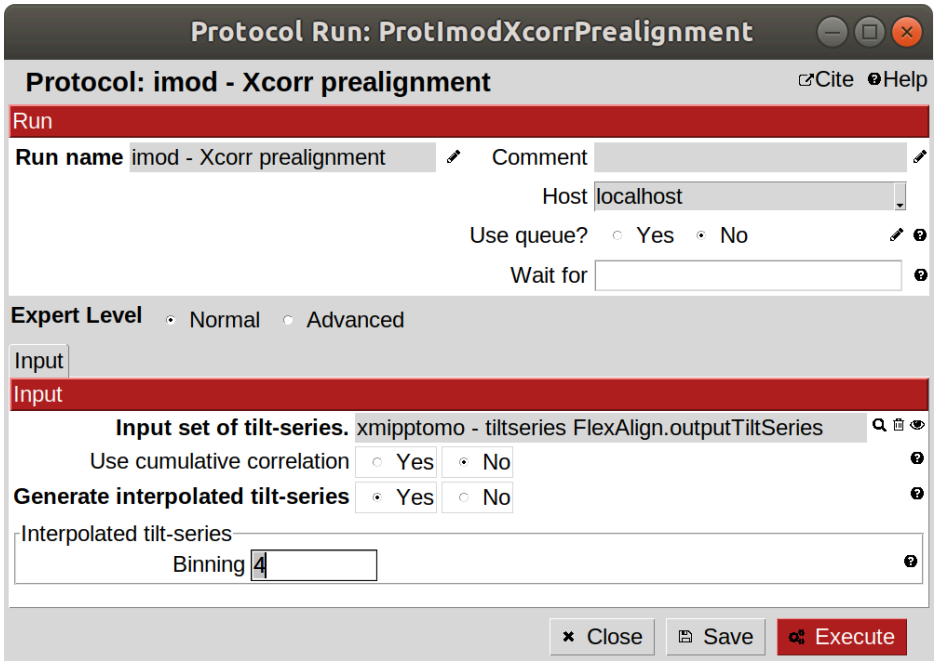


Figure 3.1: Imod - Xcorr prealignment form with the corresponding parameters for this tutorial

Note: It is also possible to modify the alignment algorithm using cumulative correlation (another cross-correlation algorithm that follows a different strategy) or modify the filter parameters. Since a cosine stretching is applied to the tilt-series, it is important to **properly determine the tilt-axis angle** in order to ensure the quality of the obtained results.

Since we selected the option “Generate interpolated tilt-series”, as it can be seen in the following figure, **a double output will be generated**. First, a **non-interpolated tilt-series** whose alignment transformation information will be safe but not applied (so, if we visualize this tilt-series we would see no changes respect to the input one), and secondly, an **interpolated** (and binned) **output** from which we can judge the goodness of the prealignment.

This is a **common behaviour for all the alignment protocols inside Scipion** pursuing the **reduccion in the number of interpolations** applied to the same tilt series and **reducing the disk space** used.

Fiducial model generation

Imod References: Kremer 1996, Mastronarde 2017

Once the prealignment process is finished, it is possible to generate the landmark (fiducial) models associated with each tilt series. Before diving into the theory behind these models it is important to be aware that they are designed to work with tilt-series presenting gold beads as fiducial markers, in order to perform the posterior alignment. These protocols will underperform (or simply do not work) with fiducial-less data, so this approach is not recommended if this is the case.
The landmark (fiducial) model generation consists of a model that provides information of the position of each gold bead in the image. Specifically, it defines the 2D coordinates of one fiducial along all the images in which it has been detected (not necessarily the whole tilt-series). Thus, it is possible to track the position of several landmarks along the whole tilt series, making it possible to posteriorly align them, correcting its tilt axis position.

We will input to this algorithm with the previous **non-interpolated SetOfTiltSeries**. The reason to do so, as it has been said previously, is to avoid excessive interpolation of our data. The protocol will apply the alignment matrices to input data and calculate the landmark (fiducial) models to posteriorly offer this information.

In order to carry out this process inside Scipion environment, the **prealigned SetOfTiltSeries** generated in the previous step will input the protocol, **imod - generate fiducial models**, from the imod plugin, that can be found in the left menu under **Tilt-series > alignment**, or searching with the Ctrl.+F hotkey, along with the following parameters.

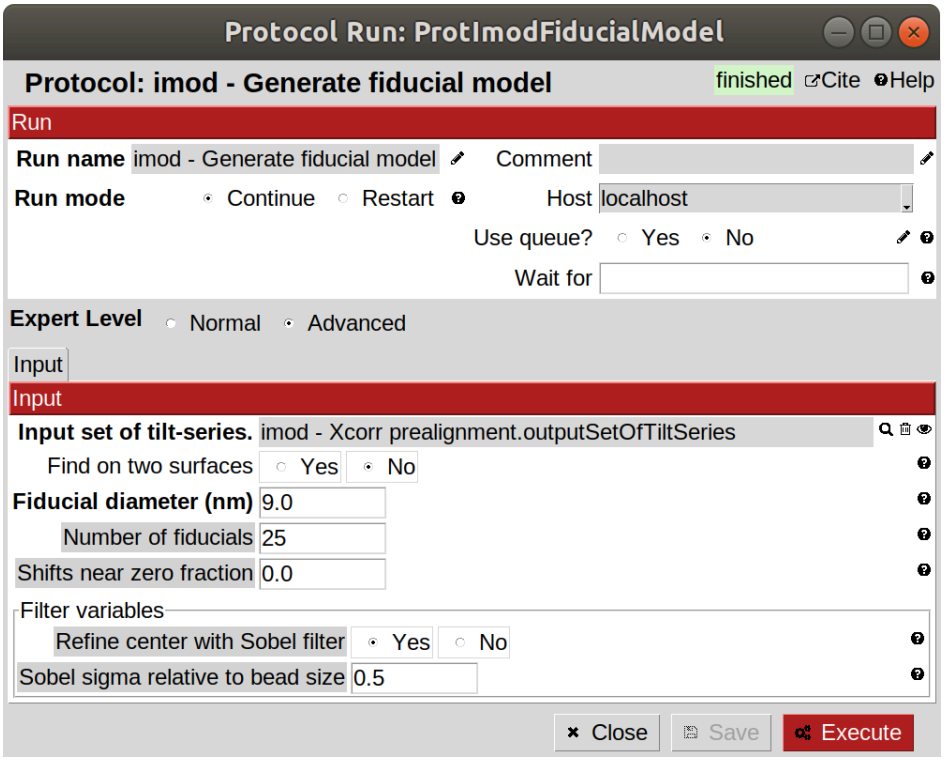


Figure 3.2: Imod - generate fiducial models form with the corresponding parameters for this tutorial

> **Note**: In this protocol, it is important to **properly set the fiducial radius** (in nanometers) since, if the indicated size is significantly different from the real one, the algorithm will fail in the fiducial location and posterior tracking. Also, it is possible to set the algorithm to differentiate between those gold beads that are in front (or over) the sample and the ones that are in the rear part (or under it), using the "**Find on two surfaces option**".

The output of this protocol is a new Scipion abstraction that has not appeared before, the **SetOfLandmarkModels**. This object is able to store the position information of each gold bead through the tilt-series for every tilt series belonging to the set.

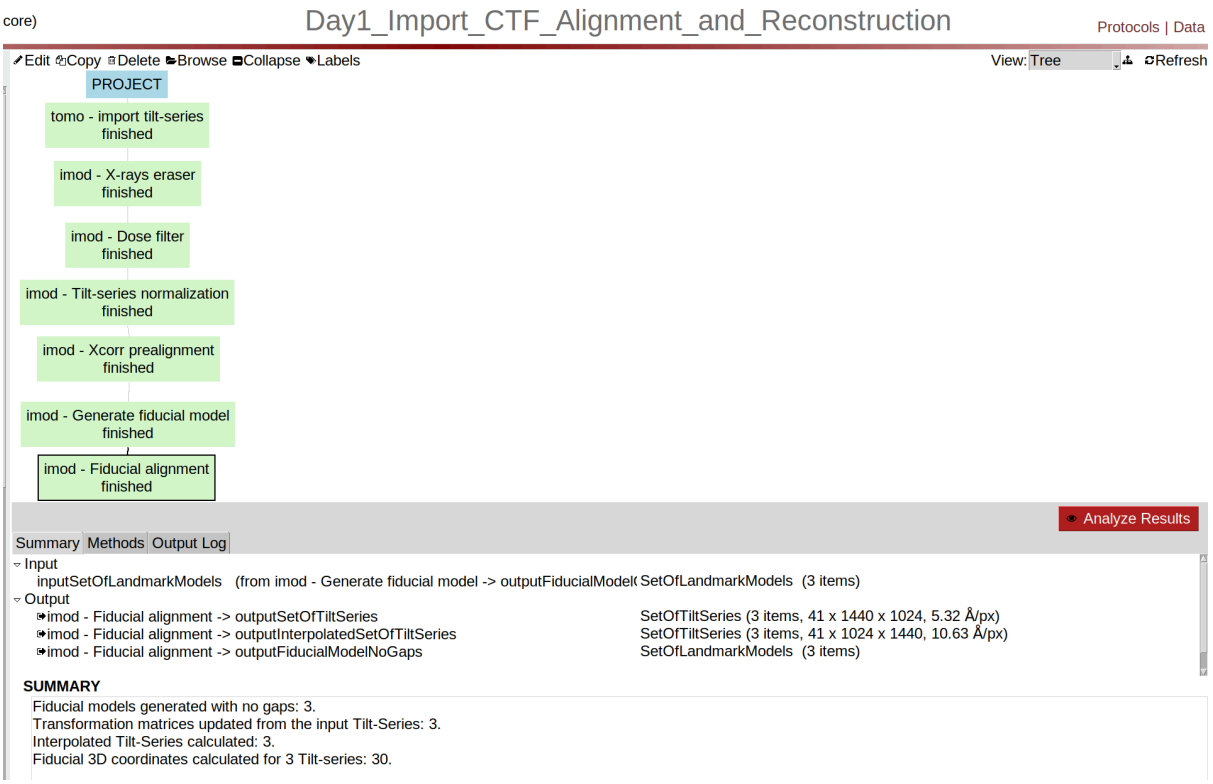


Figure 4.4: Imod - generate fiducial models form with the corresponding parameters for this tutorial

Fiducial alignment

Imod References: Kremer 1996, Mastronarde 2017

Once the landmark (fiducial) models are generated it is possible to calculate the final **alignment of the tilt-series**. In this final alignment not only the translational movements (shifts) are corrected, basically inherited from the prealignment correction, but also the **rotation of the images** (angle), aligning the tilt axis with the vertical (Y) axis of the image.

Note: Scipion uses the imod convention putting the tilt axis in the vertical (Y-axis)

This is done since it is the expected disposition of the input data for reconstruction algorithms. The reason for this is a **more efficient reconstruction**, since the algorithm will need less resources by reconstructing the tomogram slice by slice in XZ planes (perpendicular to the aligned tilt axis in the Y axis).

The input for this algorithm is just the **SetOfLandmarkModels** object, obtained in the previous step. This abstraction holds all the information needed, both the Landmark chains locations and the tilt-series associated with them.

In order to carry out this process inside Scipion environment, the **binned SetOfTiltSeries** generated in the previous step will input the protocol, **imod - Fiducial alignment**, from the

imod plugin, that can be found in the left menu under **Tilt-series > alignment**, or searching with the **Ctrl.+F** hotkey, along with the following parameters.

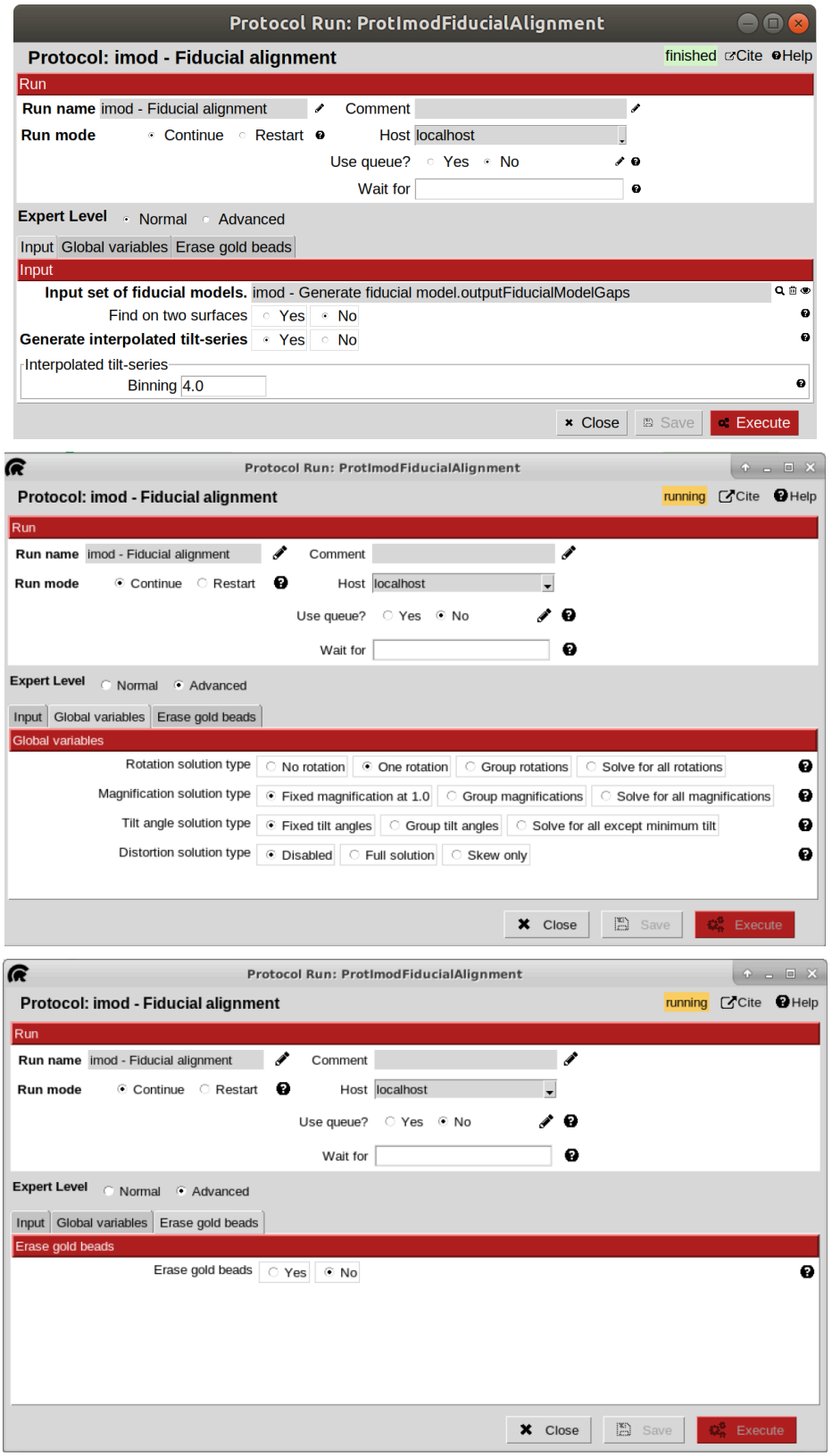


Figure 3.3: Imod - fiducial alignment form with the corresponding parameters for this tutorial

Note: This is a very option intensive protocol. In the first case, and as in the prealignment we will **generate the interpolated tilt-series**, at a higher binning to save some space in disk. According to the alignment options, it will be solved **only for one rotation** and with a **fixed magnification** 1 (no magnification of the images calculated). Also, the **tilt angles will be fixed** and **no distortion** of the images will be calculated.

In the first place, this protocol will generate the transformation matrices according to the landmark models to posteriorly generate a finally aligned tilt series. As it has been said in the previous protocol the input tilt series will have it prealignment matrix associated to the (a non-interpolated said) and in the output **the protol will combine this information**. And, in second and since we have selected again the “Generate interpolated tilt-series”, this information will be applied in order to generate the **final interpolated tilt-series**.

Also, this protocol generated a refined **SetOfLandmarkModels** with **no gaps**, meaning that the position of the landmark lost for some images in the previous steps are now interpolated from the transformation matrices calculated. And finally it also generates a **setOfCoordinates3D**. These are the coordinates of the fiducials (gold beads) in the third-dimensional space, being possible to **calculate their positions** because we already know the final alignment.

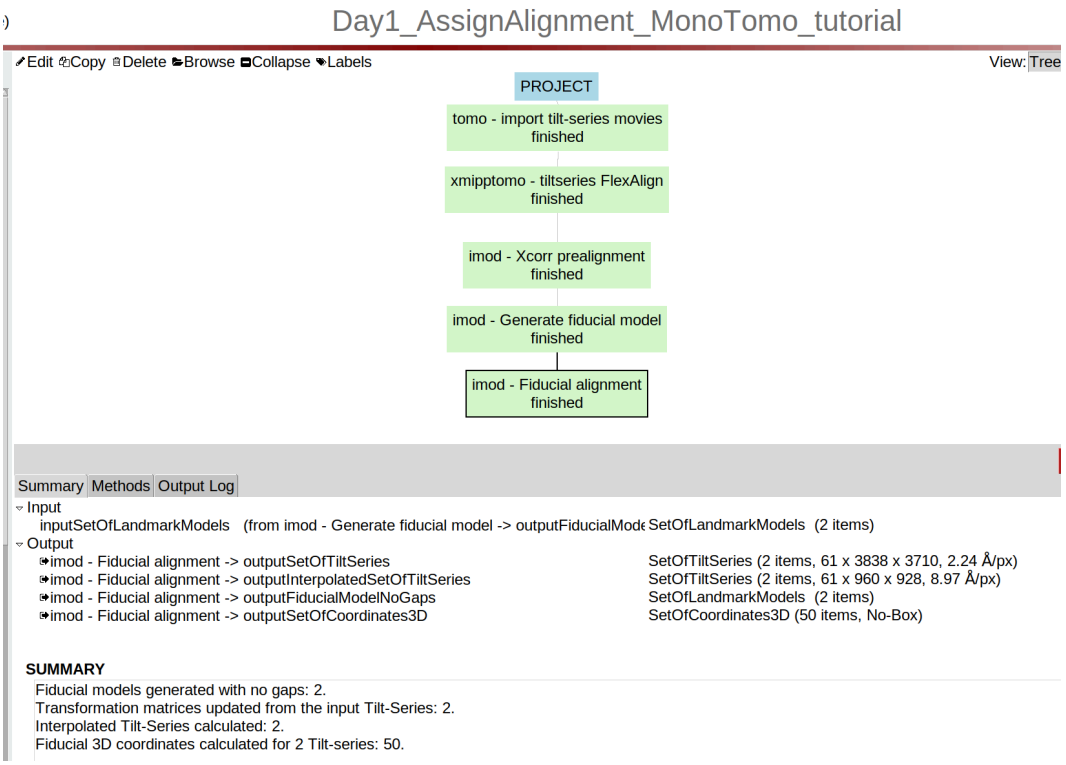


Figure 3.4: Current state of the workflow after the alignment step

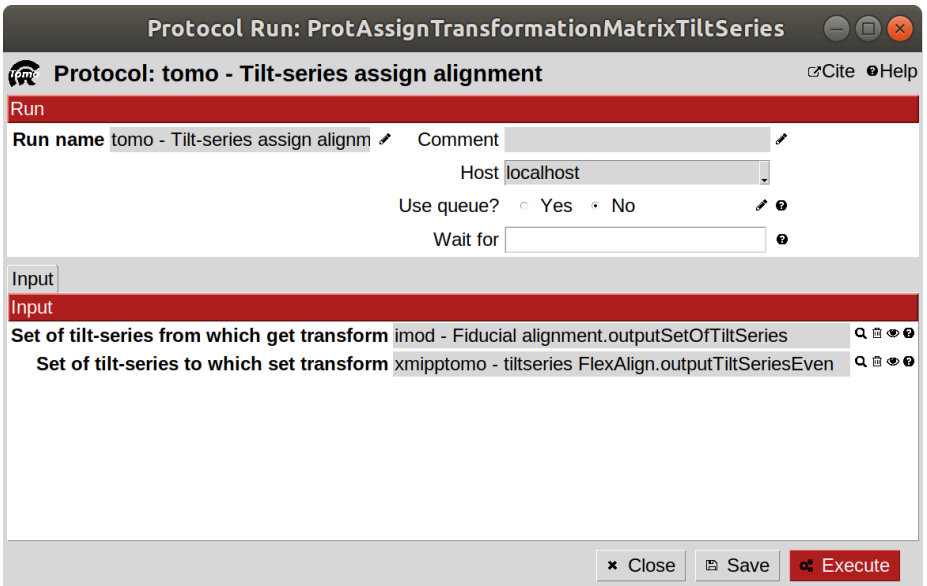
4. Assign Alignment

Imod References: Kremer 1996, Mastronarde 2017

The estimation of local resolution with MonoTomo requires the use of two half tomograms (odd and even) reconstructed with the same alignment parameters. Up to this section, we have carried out the alignment of the Set of Tilt Series involving all frames. The use of all frames ensures the reliability of the alignment. Now that the transformation matrix (alignment) is known, it will be applied to the odd and even Set Of Tilt series to achieve two independent Tilt Series with the same alignment parameters and then reconstruction of the odd and even tomograms.

The transformation matrix (alignment) of the whole SetOfTiltSeries will be taken and assigned to the odd and even SetOfTiltSeries. To do that, the protocol **tomo - Tilt-series assign alignment,** located in **Tilt Series>Alignment>tomo - Tilt-series assign alignment** can be used. Alternatively, searching with **Ctrl+F** with the keyword *assign*. This protocol just takes a transformation matrix from an aligned **SetOfTiltSeries** and attaches this transformation matrix to another SetOfTiltSeries. In particular, we will take as input the aligned and non interpolated **SetOfTiltSeries** from output of the fiducial alignment, and the transformation matrix will be assigned to the odd/even **SetOfTiltSeries** (output of FlexAlign). In Figure 4.1 the two forms with the corresponding parameters for assigning the alignment to the odd and even **SetOfTiltSeries** are shown.

> **Note**: Scipion also offers the possibility of importing transformation matrices, and then to assign such transformation matrices to a tilt series and align.



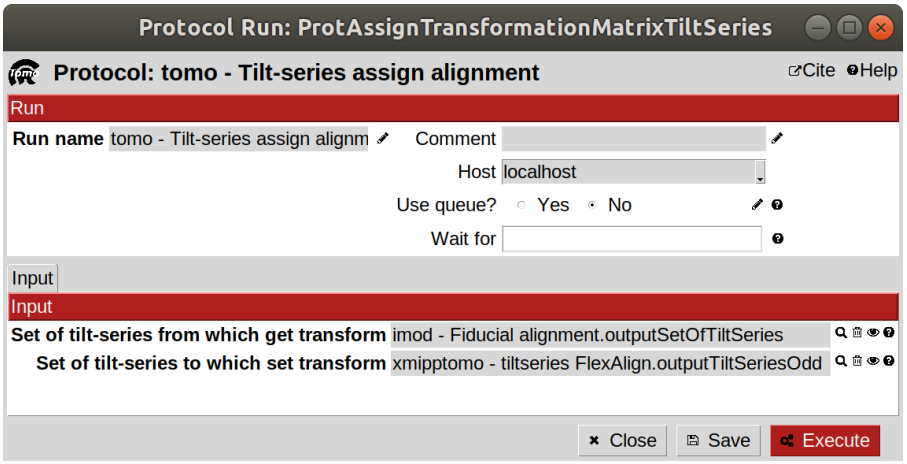


Figure 4.1: **Tomo-tilt series alignment** with the used parameters to assign the alignment. The two forms correspond to the even and odd sets of tilt series

> **Note**: The protocol Tomo-tilt series alignment only associates the transformation matrix to the tilt series, but it does not apply the transformation matrix.

Now that the tilt series has an associated alignment, it remains to apply such alignment. The protocol **imod - apply alignment** performs that task. The protocol can be found in **Tilt Series> alignment> imod - apply alignment** or alternatively with the hotkeys **Ctrl+F** searching by *apply*. The protocol takes an input a SetOfTiltSeries with an associated transformation matrix (that is the reason why it was assigned previously), and applies such matrix to the tilt series. It also allows to bin the aligned tilt series in order to facilitate its visualization. In figure 4.2, the protocol **imod - apply alignment** with the used parameters is shown. A **binning factor 4** has been set to speed up the visualization with 3dmod (just click on analyze results to visualize)

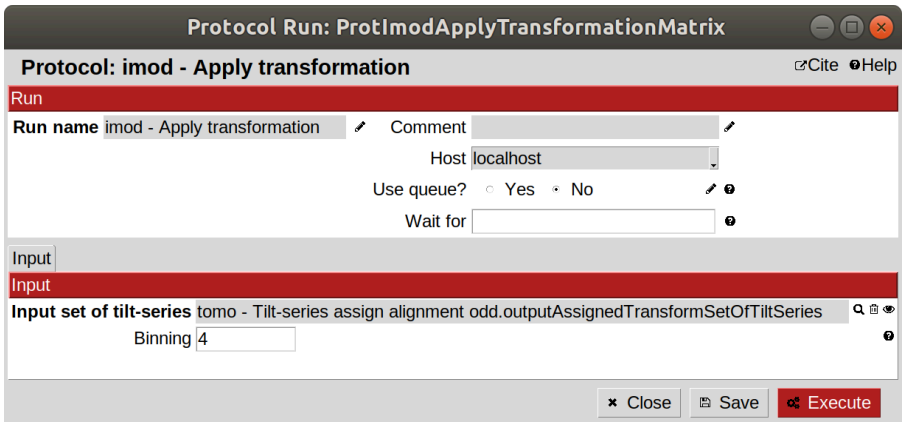


Figure 4.2: I **imod - apply alignment** protocol with the used parameters. This form is used twice, one for the **even SetOfTiltseries** and again for the **odd SetOfTiltseries**

The current state of the workflow with the **odd/even SetOfTiltseries** well aligned can be observed in Figure 4.3

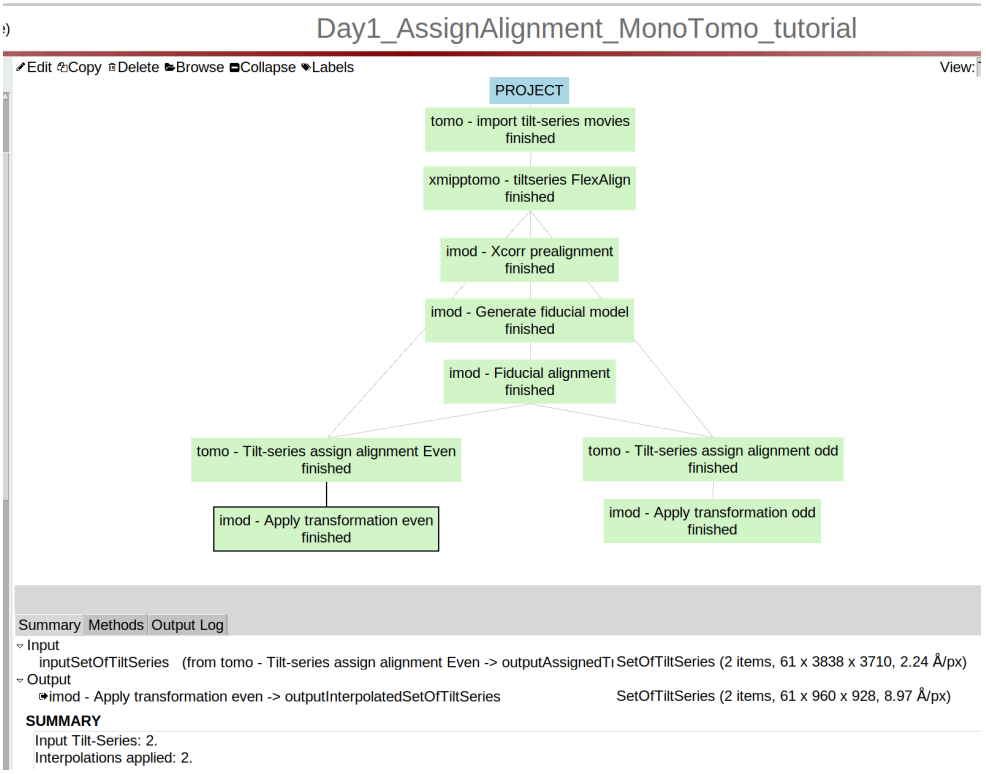


Figure 4.3: Current state of the workflow after assigning and applying the alignment to the **odd and even SetOfTiltSeries**

5. Tomogram Reconstruction (imod and jjsoft)

jjsoft References: JI Agulleiro 2011, JI Agulleiro 2015

The last step of this workflow is the reconstruction of the corresponding tomograms. Scipion provides several methods to undertake the reconstruction task as they are: **imod-reconstruction**, **jjsoft-tomo3d**, **aretomo-reconstruction**, or **nova-ctfreconstruction**. In this tutorial, we will use tomo3d.

The protocol tomo3 can be found in the left side panel **Tomogram->Reconstruction-> jjsoft-tomo3d** respectively (alternatively searching with **Ctrl+F**). In Figure 5.1 the form of the protocol is fulfilled with the used parameters. as importante parameters we selected the **SIRT method** (WBP - weighted back projection can also be used) and in the advanced parameters a **thickness of 300 px**.

> **Note**: WBP is faster than the SIRT method, but SIRT provides higher contrast. The use of SIRT filter attempts to enhance the contrast of the WBP methods.

The input of the reconstruction will be the aligned **SetOfTiltSeries**. In particular the **odd and even SetOfTiltSeries** that we have already aligned. In Figure 5.1 the form of the protocol is shown in the used parameters

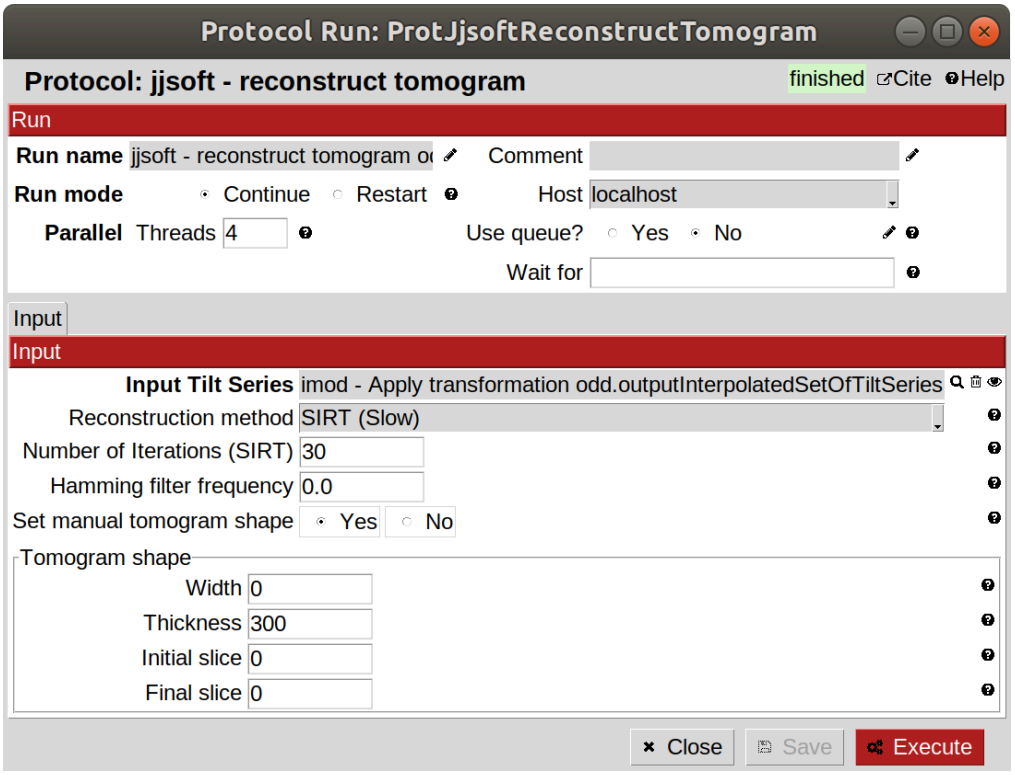


Figure 5.1: jjsoft-tomo3d forms with the used parameters of this tutorial

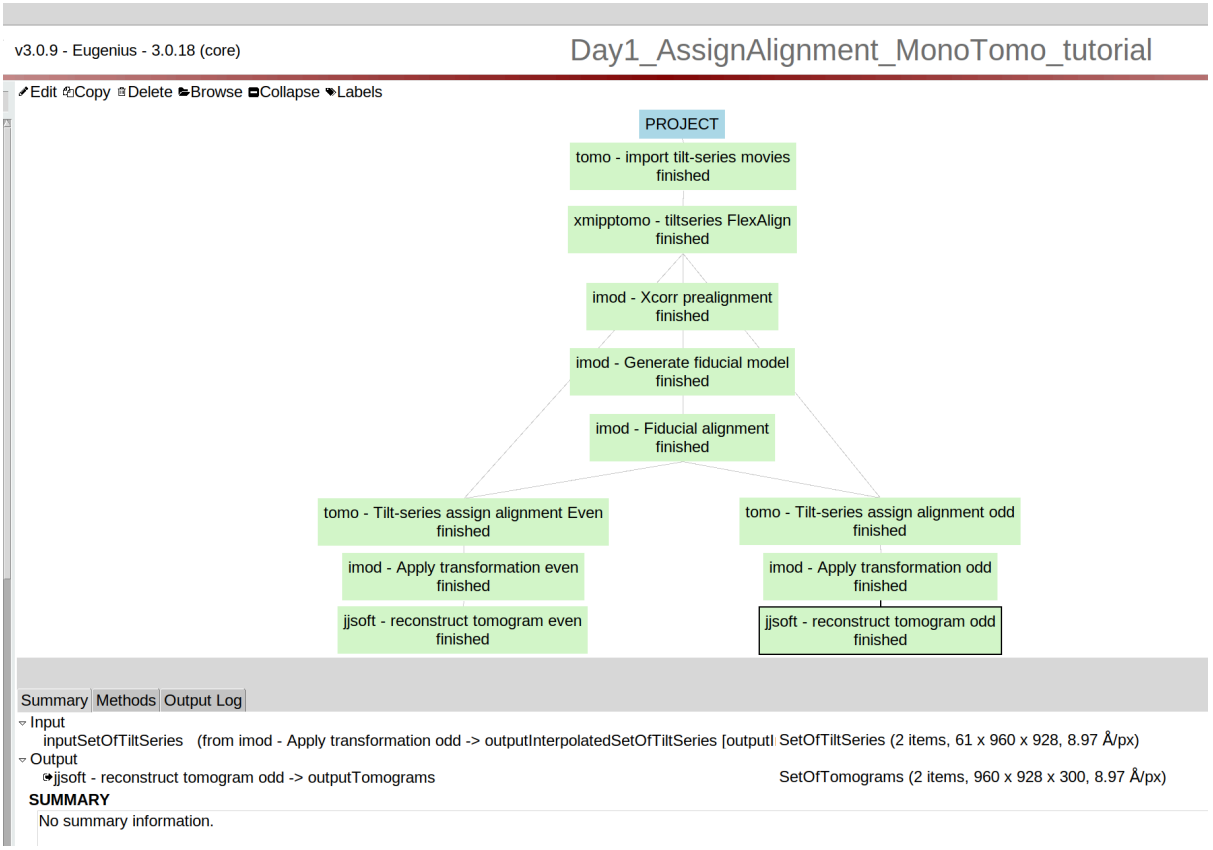


Figure 5.2: State of the workflow after carrying out the two reconstructions with imod and jjsoft.

The output will be a **SetOfTomograms** with dimensions 960x928x300 px, see Figure 5.2. They can be visualized by clicking on **Analyze results** or alternatively by choosing the visualization tool by right-clicking on the output in the Summary box. The most suitable viewer to analyze a tomogram is the **imod viewer (**see Figure 5.3)**.**

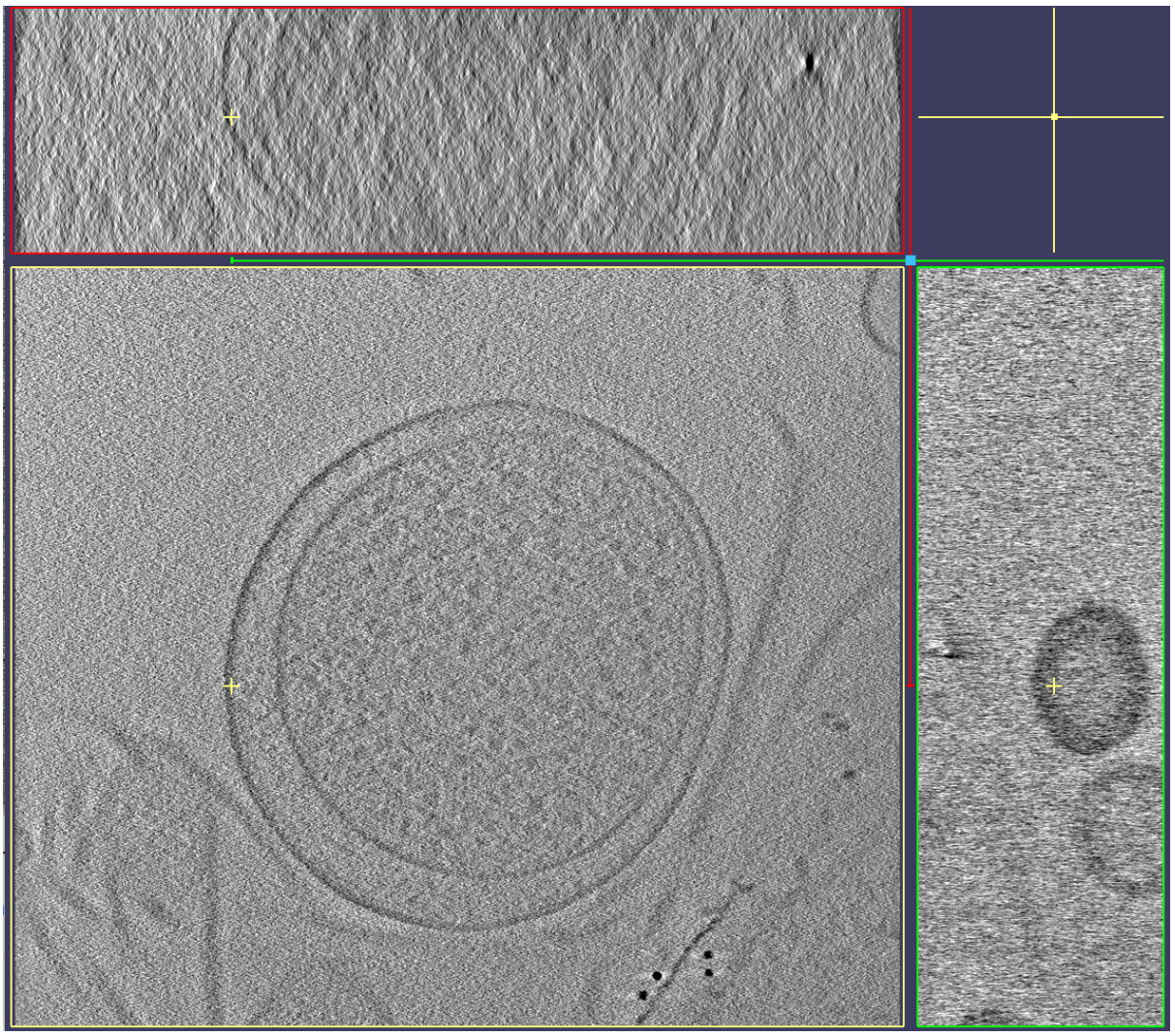


Figure 5.3: Results of the **jjsoft - tomo3d** visualized with 3dmod.

6. Local Resolution Analysis MonoTomo

MonoTomo Reference: JL Vilas 2020
Plugin: scipion-em-xmipptomo

The last step of this tutorial is the local resolution analysis of the reconstructed tomograms. **xmipptomo - local resolution MonoTomo** allows the local resolution estimation of the whole tomograms by means of two half tomograms reconstructed with the half of the data. The tomograms are called odd and even are are reconstructed with the odd and even frames respectively.

> **Note**: They can also reconstructed with odd and even images of the tilt series (not recommended). We highly recommend to split in odd even frames, instead of splitting in odd and even tilt images. Splitting the movies in frames enhances the reliability of the MonoTomo result.

MonoTomo works by establishing comparisons (in a statistical sense - hypothesis tests) at different frequencies between noise and signal. Understanding the difference odd - even (tomograms) as noise, and signal the addition, odd+even (tomograms).

MonoTomo can be found in the left side panel **Tomogram->Quality analysis> local resolution MonoTomo,** or alternatively with the hotkeys **Ctrl+F** and searching **monotomo. The form of xmipptomo - local resolution monotomo,** can be observed in Figure 6.1 and is quite simple, it only requires the two **SetOfTomograms** (odd and even), and eventually a mask if the users wants to analyse a specific are (not mandatory, and not used in this tutorial). Moreover, the local resolution range and the step size of the search. Due to the low resolution of the electron tomograms, a broad range is recommended (between Nyquist to around 140A), this is the introduced range of this tutorial. And as input Tomograms we selected the reconstructed ones odd and even of the tomo3d. The form with the used parameters can be seen in Figure 6.1.

> **Note**: Resolution steps lesser than 0.5A, can fall in local minima, we recommend 1A

MonoTomo result will be a local resolution tomogram that can be visualized with the MonoTomo viewer, or alternatively in Chimera, see Figures 6.4 and 6.5. However, MonoTomo viewer, in Figure 6.3, provides many other features in other to visualize the results, like the possibility of estimating the histogram of local resolution or representation of local resolution colored slices.

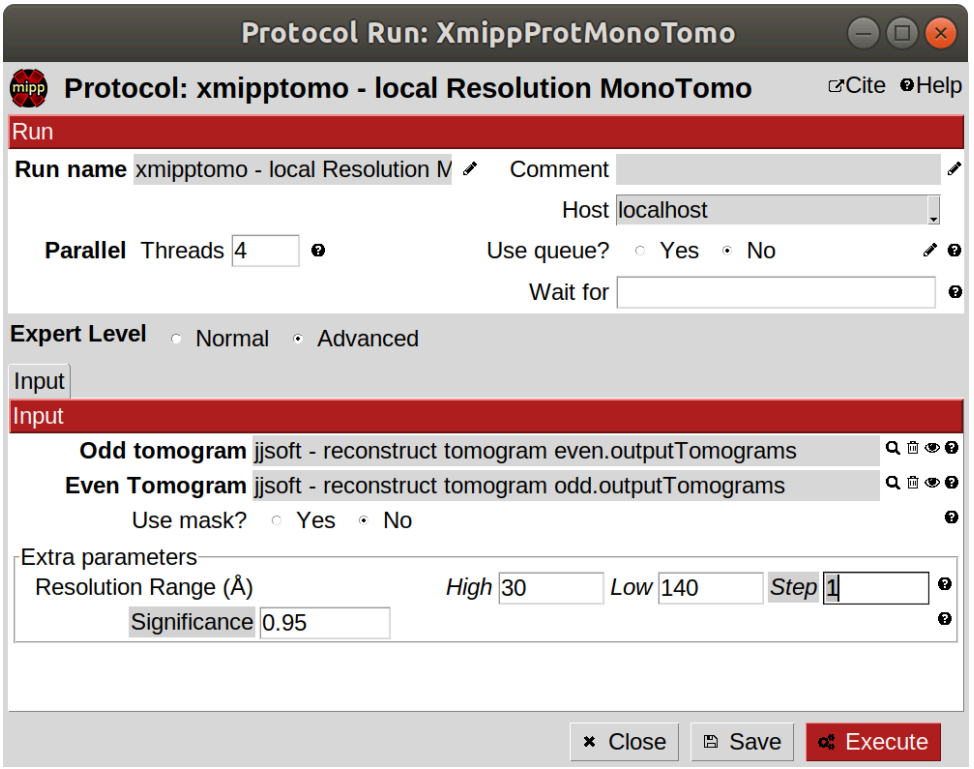


Figure 6.1: xmipptomo - local resolution MonoTomo form with the used parameters in this tutorial

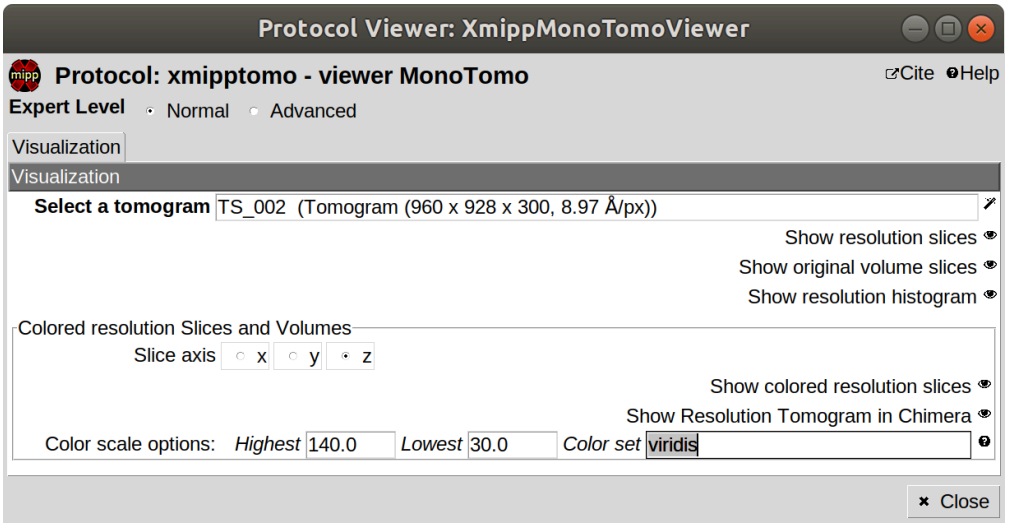


Figure 6.3: xmipptomo - local resolution MonoTomo viewer with all visualization options

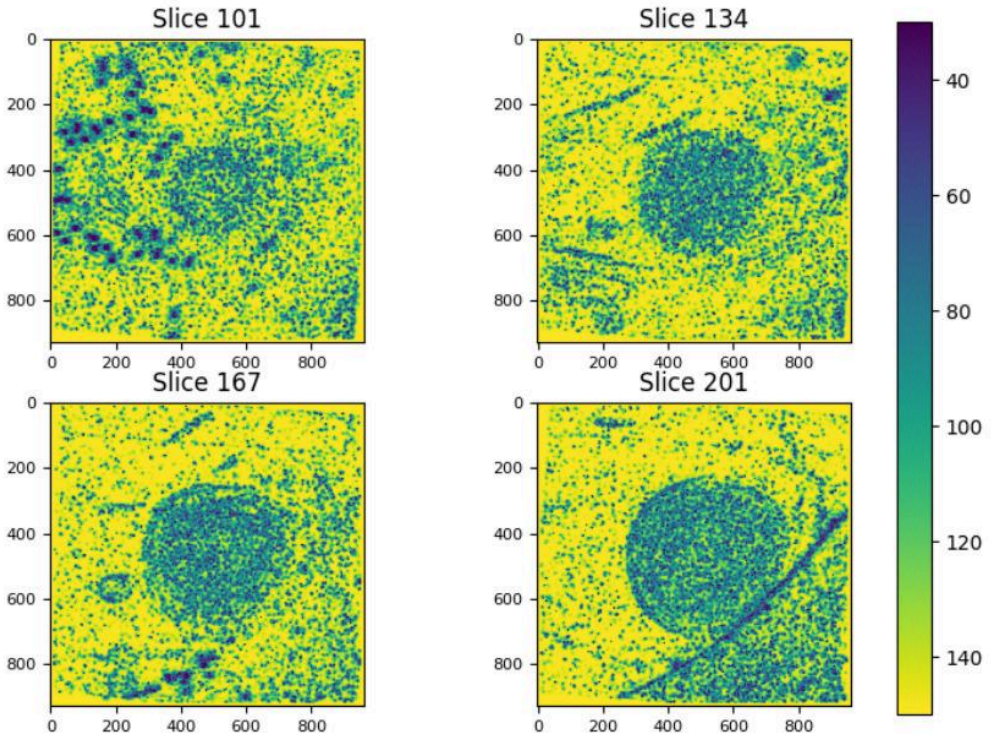


Figure 6.4: Local Resolution slices estimated with **xmipptomo - local resolution monotomo**

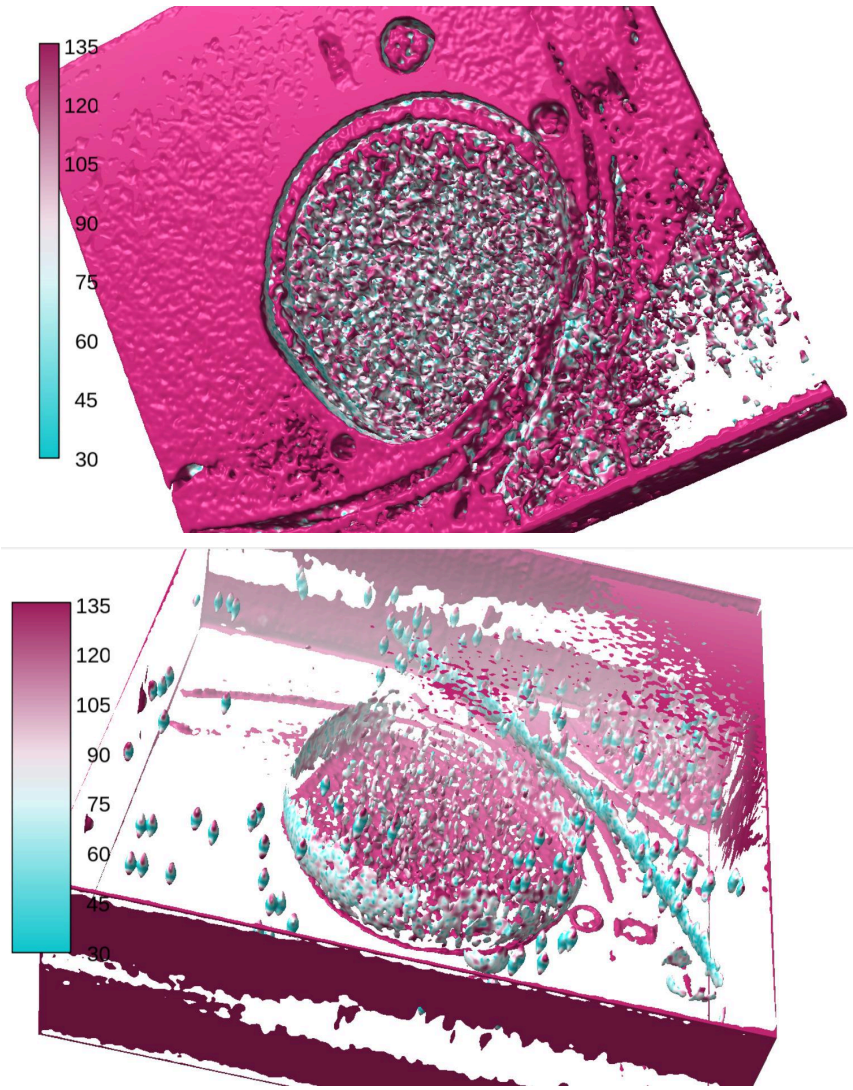


Figure 6.5: Colored tomogram with chimeraX

References

1. JM De la Rosa-Trevín, A Quintana, L Del Cano, *et al.* Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. Journal of Structural Biology, 195,1, 93-99 (2016).
2. D. Střelák, J. Filipovič, A. Jiménez-Moreno *et al* FlexAlign: An Accurate and Fast Algorithm for Movie Alignment in Cryo-Electron Microscopy. *Electronics*. 9(6):1040, (2020).
3. J.R. Kremer, D.N. Mastronarde, J.R McIntosh, Computer Visualization of Three-Dimensional Image Data Using IMOD, Journal of Structural Biology, 116, 1, 71-76 (1996)
4. D.N. Mastronarde, S.R. Held, Automated tilt series alignment and tomographic reconstruction in IMOD, Journal of Structural Biology, 197, 2, 102-113 (2017)
5. JI Agulleiro, JJ Fernandez. Fast tomographic reconstruction on multicore computers. *Bioinformatics* 27:582-583, (2011).
6. JI Agulleiro, JJ Fernandez. Tomo3D 2.0--exploitation of advanced vector extensions (AVX) for 3D reconstruction. *Journal of Structural Biology 189:147-152, (2015).*
7. J.L. Vilas, J. Oton, C. Messaoudi, et. al. Measurement of local resolution in electron tomography, Journal of Structural Biology: X, 4, 100016 (2020)