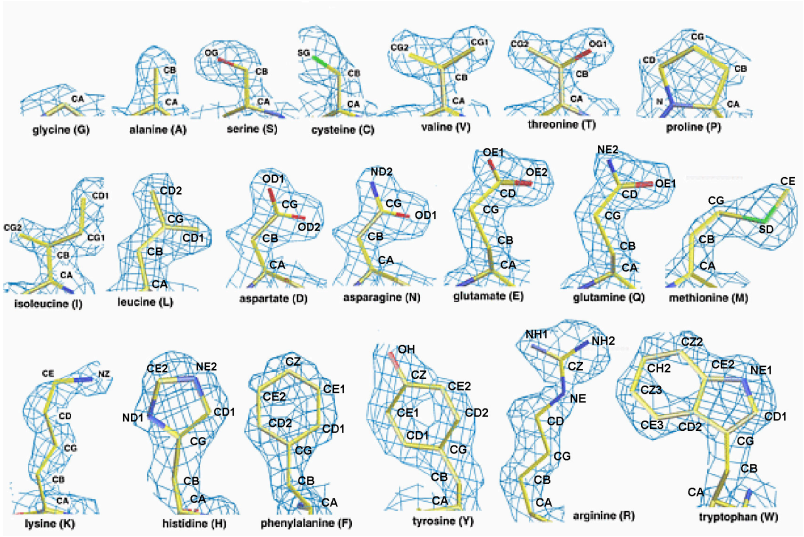
Scipion Tutorial Series

National Center for Biotechnology

Biocomputing Unit

Model Building Basic

December 10, 2020



Density for amino acid side chains from an experimental electron density map at 1.5 Å resolution (http://people.mbi.ucla.edu/sawaya/m230d/Modelbuilding/modelbuilding.html)

Roberto Marabini & Marta Martínez

Revision History

Revision	Date	Author(s)	Description
1.0	11.15.2018	MM, RM	created for first model building workshop
1.1	01.30.2019	MM	added appendices and minor fixes
1.2	04.24.2019	MM	added atomstructutils, contacts and submission protocols
1.3	09.10.2019	MM	added map preprocessing protocols (create mask, and compute local Resolution and Sharpening) and *PHENIX* validation cryoEM
1.4	18.11.2020	MM	migration to python3 and adaptation to *Scipion* version 3.0, replacement of Chimera by ChimeraX (new functionalities in model from template and map subtract), added other preprocessing tools (*DeepEMhancer*), added *PHENIX* dock-in-map protocol, removed singularities for *PHENIX* version 1.13, removed *PowerFit* protocol

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 model building, this tutorial shows the initial workflow aimed at obtaining high-quality atomic models 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*, the *CCP4* suite (`http://www.ccp4.ac.uk/download/#os=linux`) including *Refmac* and *Coot*, and the *PHENIX suite* (`https://www.phenix-online.org/download/`). USCF *ChimeraX* is also required but you only have to follow the *Scipion* instructions to install `ChimeraX v.1.1` (`https://www.rbvi.ucsf.edu/chimerax/download.html`). Basic knowledge of *ChimeraX* and *Scipion* is assumed. Warning: old versions of *Refmac* are not suitable for EM data.

Contents

1 Introduction to Model building

Definition

Model building is the process that allows getting the atomic interpretation of an electron density map. Although a electron density volume can be obtained from different methodologies, in this tutorial we focus in maps obtained by cryo-EM. As an example of these maps, Fig. 1 shows the input electron density map (a), as well as the output haemoglobin tetramer atomic model (b) obtained by the model building process. Since high quality atomic structures are essential to accomplish detailed mechanistic studies and to seek inhibitor drugs of macromolecules, the main aim of model building is obtaining reliable structures of these macromolecules.

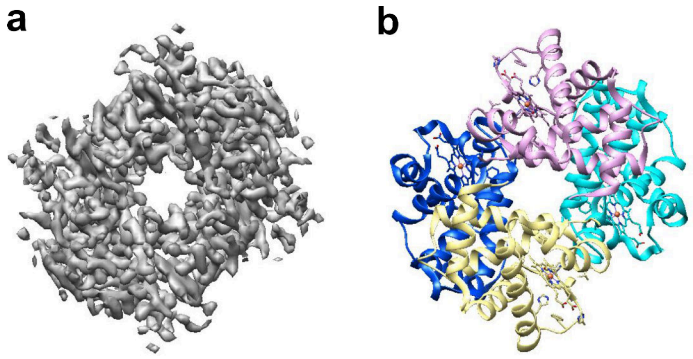


Figure 1: Haemoglobin tetramer (Khoshouei et al., 2017). a) Electron density map at 3.2Å resolution obtained by Cryo-EM single particle analysis with Volta phase plate. b) Atomic structure model inferred from the electron density volume.

Relevance of cryo-EM map resolution

Model building process is limited by the resolution of the starting cryo-EM density map. The higher the resolution, the more detailed and reliable atomic structure will be obtained. Fortunately, single-particle cryo-EM is undergoing in this decade a resolution revolution that has allowed the structures of macromolecules to be solved at near-atomic resolution. The density map is thus sufficiently resolved to build the

atomic model. As a general rule, at resolutions of 4.5Å the molecule backbone can be inferred based on the map alone, and resolutions lower than 4Å allow to trace side chains of some residues.

Model building workflow

The set of successive tasks aimed to get the atomic interpretation of electron density maps is known as model building workflow. Main steps of the general workflow are detailed from top to bottom in Fig. 2. Tasks and tools required are highlighted in green (left side). Before starting those tasks, a detailed study and recruiting of experimental information of the macromolecule itself and similar specimens is recommended. Cryo-EM density map preprocessing is also desirable in order to optimize it by maximizing details and connectivity, as well as extracting the lower asymmetrical element of the starting volume (ASU: asymmetric unit) to save computational resources and facilitate the modeling.

In addition to the map ASU, the workflow considers as input the sequence of each individual structural element (from 1 to n). This sequence is used to get the initial model, *de novo* or by prediction based in structures of homologous sequences. Initial model of each structure element has to be fitted to the volume ASU, and then refined according to the density of this map fraction. Refinement in real and/or reciprocal spaces are included in the workflow. Once refined, the geometry of each individual structure has to be validated regarding the starting volume. The last two steps of refinement and validation will be applied globally to the whole set of structures contained in that map ASU to avoid forbidden steric overlaps among them. Borders between adjacent unit cells will be checked similarly in the reconstruction of the whole atomic structure.

In this tutorial, we show how to obtain an atomic model using a reference homologous structure.

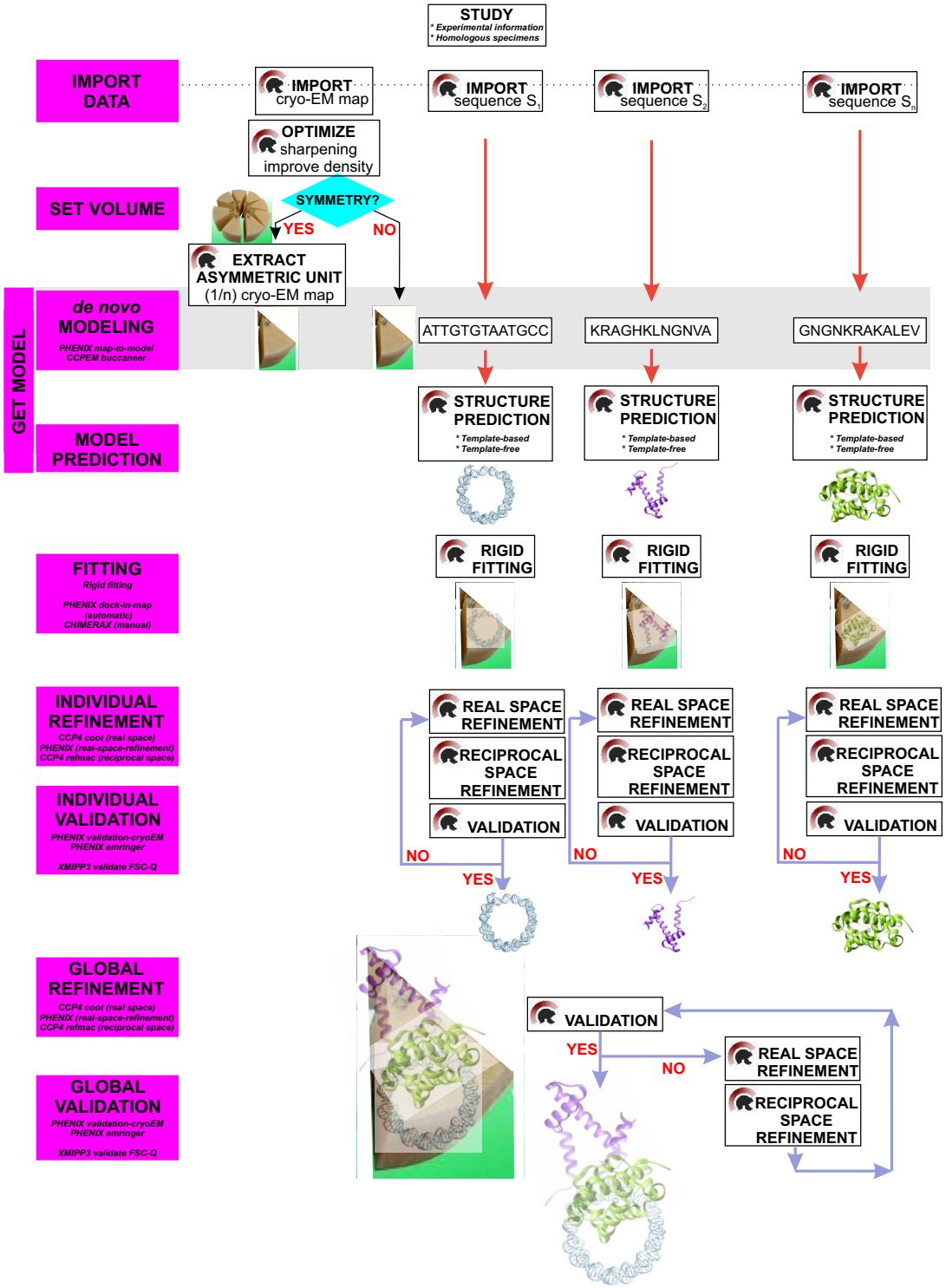


Figure 2: General Model Building Workflow.

2 Problem to solve: Haemoglobin

The metalloprotein Haemoglobin (Hgb) is the iron-containing protein able to transport oxygen, essential to get energy from aerobic metabolic reactions, through red blood cells of almost every vertebrate. The first atomic structure of Hgb was determined in 1960 by X-ray crystallography (Perutz et al., 1960). Hgb was, alongside myoglobin, the first structure solved by this methodology. Due to its emblematic prominence in structural biology History, we have selected Hgb to model its atomic structure.

Hgb is a relatively small macromolecule (molecular weight of 64 KDa) that shows C2 symmetry. This heterotetramer is constituted by four globular polypeptide subunits, two α and two β monomers with 141 and 146 aminoacids in human Hgb, respectively. Each subunit associates to a prosthetic heme group, that consists in an iron (Fe) ion and the heterocyclic ring of porphyrin. Although the molecule is able of binding oxygen only in the reduced ferrous status, human Hgb is commercially distributed in its nonfunctional oxidized ferric status as metHgb. The atomic structure of the human metHgb specimen was inferred by Khoshouei et al. (2017) for the first time from the electron density volume obtained by cryo-EM and using the Volta phase plate. The volume, at 3.2Å resolution, and its atomic interpretation (Fig. 1) are available in the Electron Microscopy Data Bank (EMDB) and Protein Data Bank (PDB) with accession numbers EMD-3488 and PDB-5NI1, respectively.

This tutorial will guide us in the deduction process of the human metHgb atomic structure using the *Scipion* framework, the 3D map and the protein sequences as starting input data, as well as reference atomic structures as homologous models in the way indicated in (Martínez et al., 2020).

3 Input data description

Map

Modeling means atomic interpretation of a map. This map can be the result of our own reconstruction process or can be obtained from a database. In this tutorial we use the haemoglobin map `EMD-3488`, that can be downloaded from `PDBE` (`http://www.ebi.ac.uk/pdbe/entry/emdb/EMD-3488`) (Fig. 3).
WARNING in case you use your own map obtained from cryo-EM images: Take into account that cryo-EM 3D maps benefit significantly of an "optimizing" step, normally referred to as "sharpening" or "density improvement", that tends to increase signal at medium/high resolution. Therefore, we recommend to sharp the map before tracing the atomic model. Either two *Scipion* protocols consecutively applied, xmipp3 - local MonoRes (Vilas et al., 2018) and xmipp3 - localdeblur sharpening (Ramírez-Aportela et al., 2018), or the protocol xmipp3 - deepEMhancer (Sanchez-Garcia et al., 2020), allow map sharpening. Details about the parameters of these protocols are shown in Appendices 18, 17 and 11, respectively.

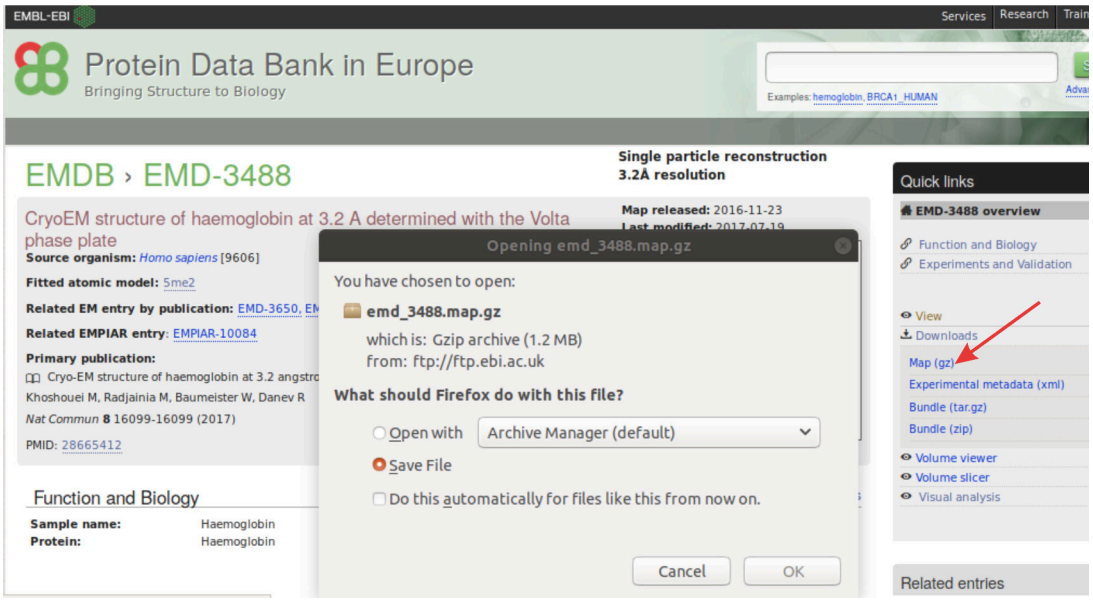


Figure 3: Downloading the volume from `PDBe`.

Once downloaded the volume, unpack it (command line: `gunzip emd-3488.map.gz`) and save it in your tutorial folder.

Sequences

The sequences of Hgb α and β subunits are included in UniProtKB. Accession numbers are P69905 and P68871, respectively. Next, we show both sequences in fasta format:

```
>sp|P69905|HBA_HUMAN Haemoglobin subunit alpha
MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHFDLSHGSAQVKGHG
KKVADALTNAVAHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTP
AVHASLDKFLASVSTVLTSKYR

>sp|P68871|HBB_HUMAN Haemoglobin subunit beta
MVHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPK
VKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFG
KEFTPPVQAAYQKVVAGVANALAHKYH
```

These protein sequences were determined by direct translation from the experimental sequence obtained from complementary DNA (cDNA), i.e., DNA synthesized or retro-transcribed from messenger RNA (mRNA). In this way, it is quite unlikely that these sequences include post-translational modifications. Although methionine is added with the translation Met-tRNA initiation factor, the removal of methionine aminoacid from the N-terminus of a polypeptide is a common post-translational modification. Since Met appears at the N-terminal end of both proteins, we can predict that these are not the polypeptide mature forms and Met will be removed in the mature ones that are present in the atomic structures.

Those two sequences can be retrieved from UniProtKB using *Scipion* import sequence protocol, which allows direct downloading from the database.

4 Import Input data

Taking advantage of *Scipion* software framework, we are going to import the above indicated input data using protocols import volumes and import sequence. Details about

the parameters of these two protocols are shown in Appendices 16 and 14, respectively.

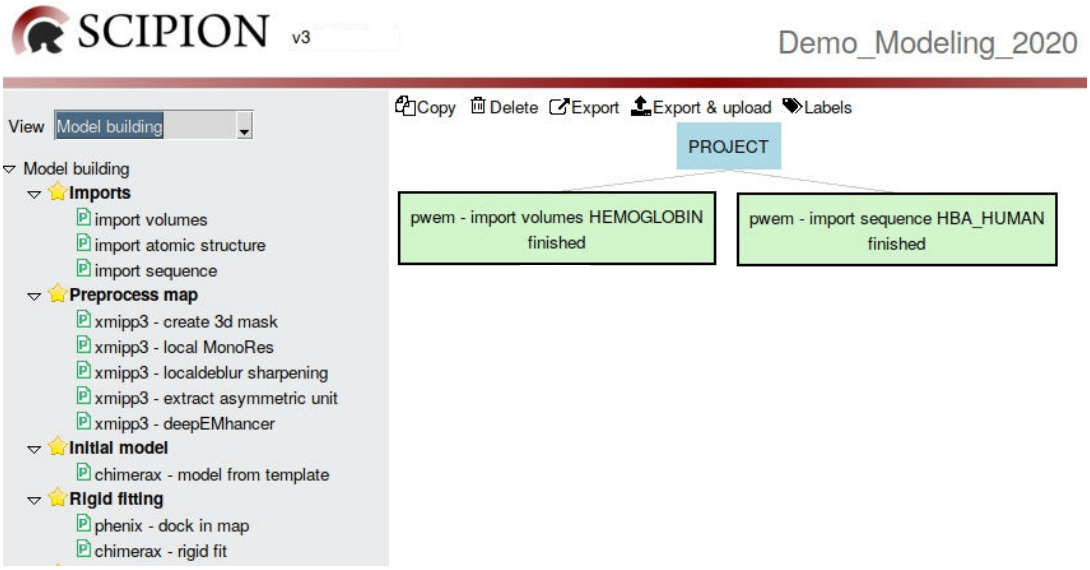


Figure 4: *Scipion* framework with import workflow.

(Note: The notation `Fig. X (a)` means that the step is shown in figure number X and there will be an arrow labeled with "a" marking the region of interest.)

Volume

First open the import volumes protocol (Fig. 5 (1)), fill in the form and execute it (2), and finally you may visualize the volume (3).

As you can see, when we import a map we directly assign its sampling rate and its origin of coordinates. If for any reason we have to work with other maps previously generated during the reconstruction process that do not have the desired sampling and origin, we can use the auxiliar protocol assign Orig & Sampling, detailed in Appendix 26, to assign them.

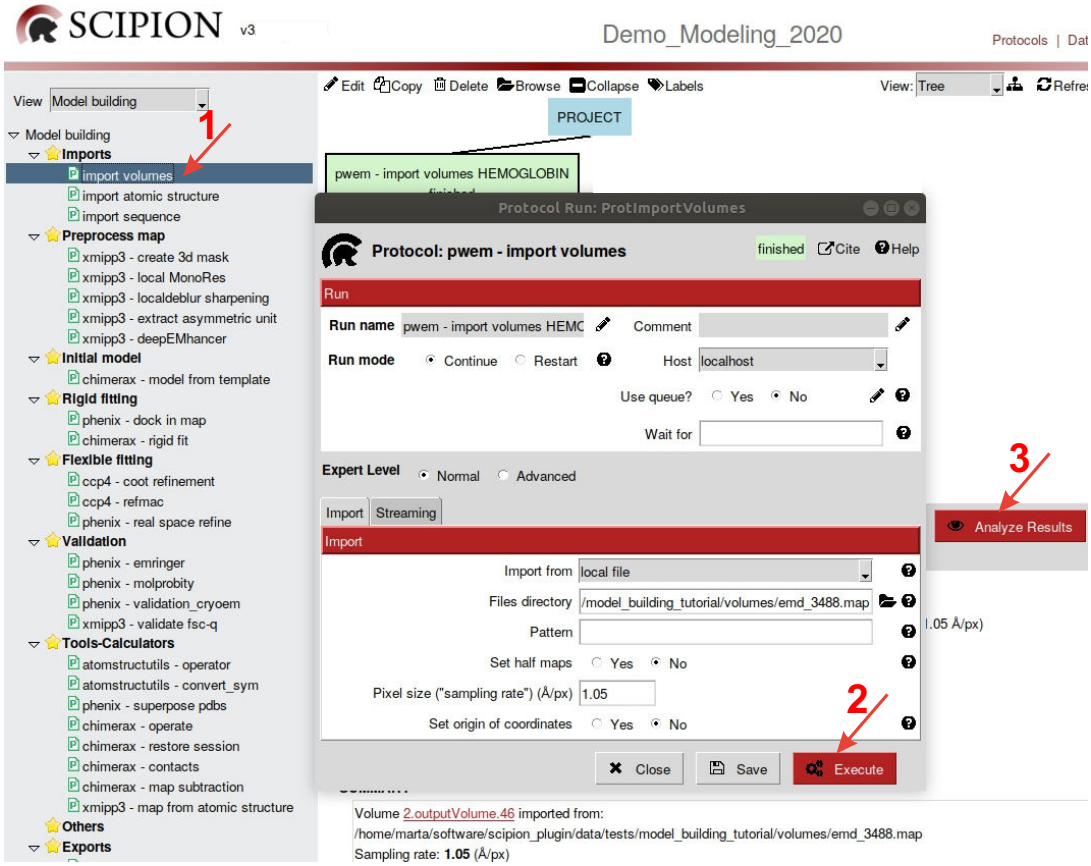


Figure 5: Importing the volume in *Scipion*.

By default *ChimeraX* (Goddard et al., 2018) is used for visualization. Clicking in the viewer menu (Fig. 6 (1)), *ChimeraX* shows the 3D map and the x (red), y (yellow) and z (blue) axes.

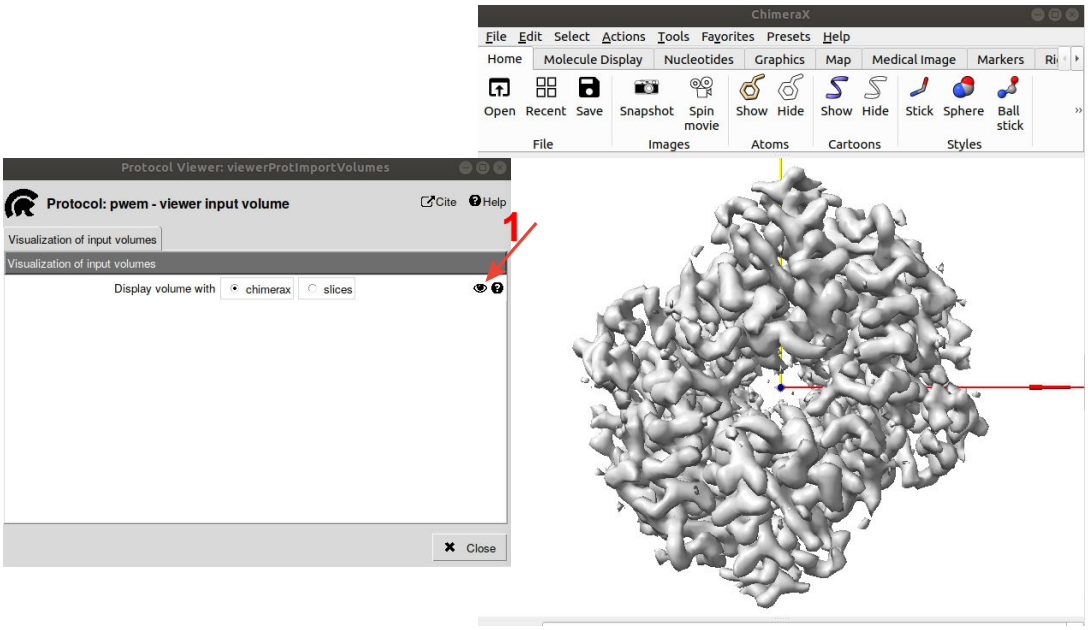


Figure 6: Volume visualized with *ChimeraX*.

Sequences

The sequences of `Hgb` α and β subunits will be independently downloaded from `UniprotKB`. First of all, open the form of import sequence protocol (Fig. 7 (1)), then complete the form to download `HBA_HUMAN` protein with `UniProtKB` accession code `P69905`, execute the process (2), and finally visualize the sequence (3) in a text editor. The sequence will appear in `fasta` format as it has been written above. Follow the same protocol to download `HBB_HUMAN` with accession code `P68871`.

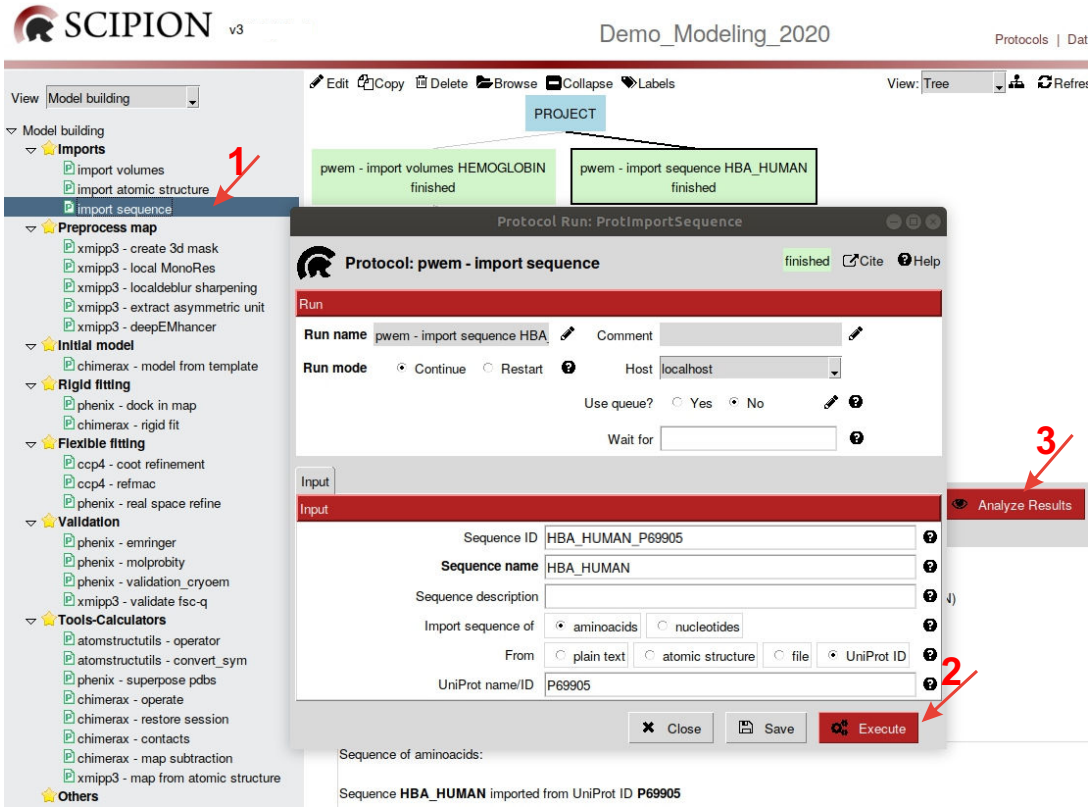


Figure 7: Importing a `UniProtKB` sequence in *Scipion*.

5 3D Map preprocessing

Fig. 8 shows the *Scipion* workflow that we are going to detail in this section.

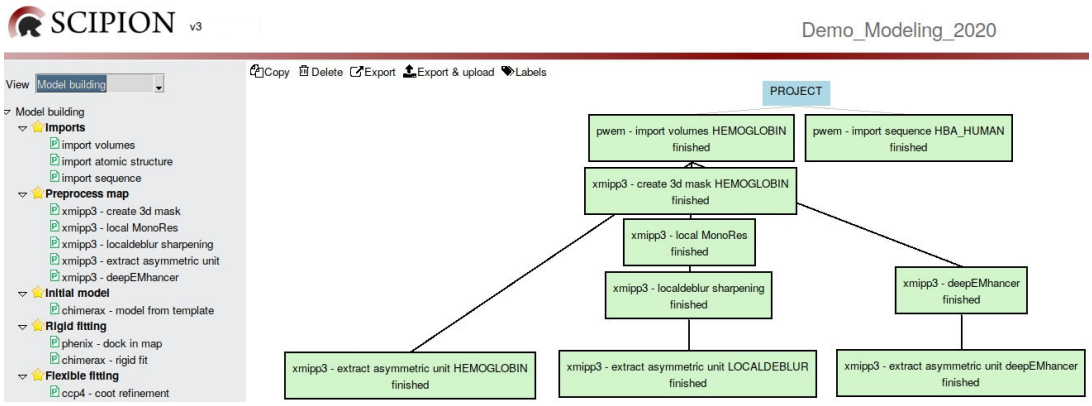


Figure 8: *Scipion* framework detailing the workflow generated after 3D map preprocessing.

Map sharpening

As we have indicated before, since 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 (Ramírez-Aportela et al., 2018). 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) (Appendix 17)), b) deep learning-based sharpening approach (xmipp3 - deepEMhancer (Sanchez-Garcia et al., 2020) (Appendix 11)). 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.

Although the common first rule in both sharpening strategies is counting on half maps to get the best performance of the methods, or the average raw map otherwise, exceptionally in this case, to illustrate the procedure we are going to use the final postprocessed map deposited in the database, where no half maps have been submitted together with the final map.

a) Sharpening with *LocalDeblur*

Since *LocalDeblur* takes advantage of map local resolution to increase the signal, we have to compute this local resolution as first step to apply the *LocalDeblur* sharpening method. Although different algorithms could be used to compute local resolution, we have selected *MonoRes* (Vilas et al., 2018), implemented in *Scipion* in the protocol xmipp3 - local MonoRes (Appendix 18).
Since a map binary mask has optionally to be included as a parameter in this protocol, we will build a mask by using the *Scipion* protocol xmipp3 - create 3d mask (Appendix 10) as starting step in the local resolution estimation process. Open the protocol form (Fig. 9 (1)) and fill in the tap `Mask generation` (2) with the input volume (3) and the density threshold (4). By default, the level value observed in *ChimeraX* main graphics window (Fig. 6) `Tools -> Volume Data -> Volume Viewer -> Level` can be selected as threshold. In the `Postprocessing` tap (Fig. 9 (5)), select `Yes` in `Apply morphological operation` (6) and maintain the rest of options by default. After executing this protocol (Fig. 9 (7)), the morphology of the mask generated can be checked in slices by clicking `Analyze Results` (8).

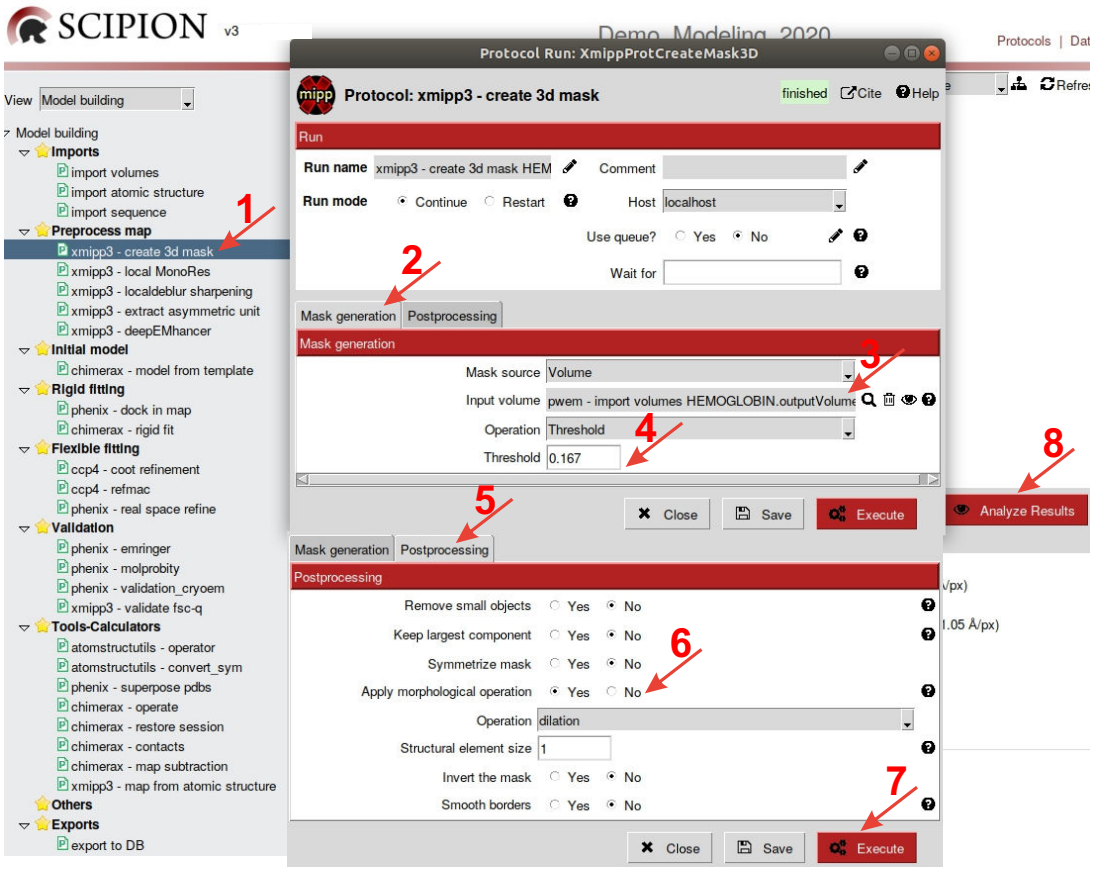


Figure 9: Filling in the protocol to create a mask of the initial volume.

ShowJ, the default *Scipion* viewer, allows visualize the mask with shape similar to the starting volume (Fig. 10).

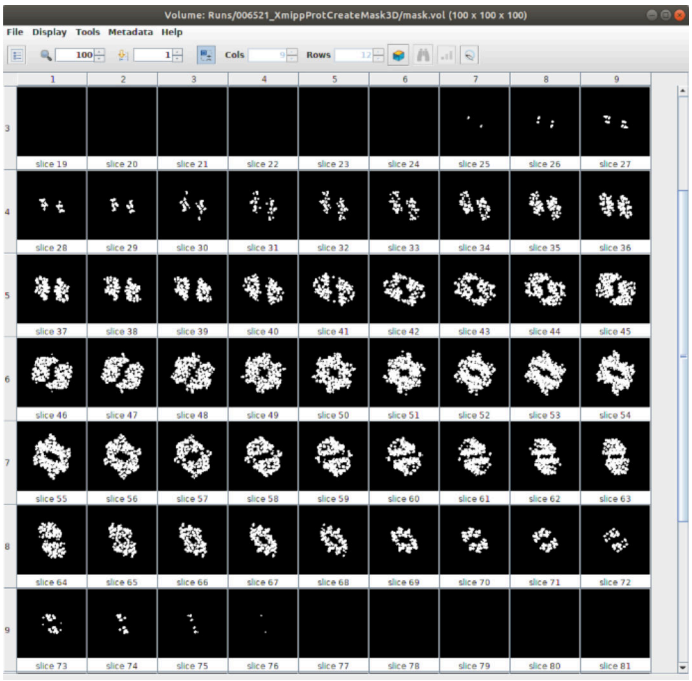


Figure 10: Visualizing the mask of the initial volume.

NOTE: In case you would like to use a previous computed mask, you can do it simply by importing it using the protocol import mask (Appendix 15).

Once the mask of the starting map has been created, the protocol of xmipp3 - local MonoRes can be completed to get the estimation of local resolution. Open the protocol (Fig. 11 (1)) and include the starting map (2), as well as the binary mask (3). Finally, based on the map resolution (3.2 Å), select the default resolution range between 0.0 and 6.0 Å(4).

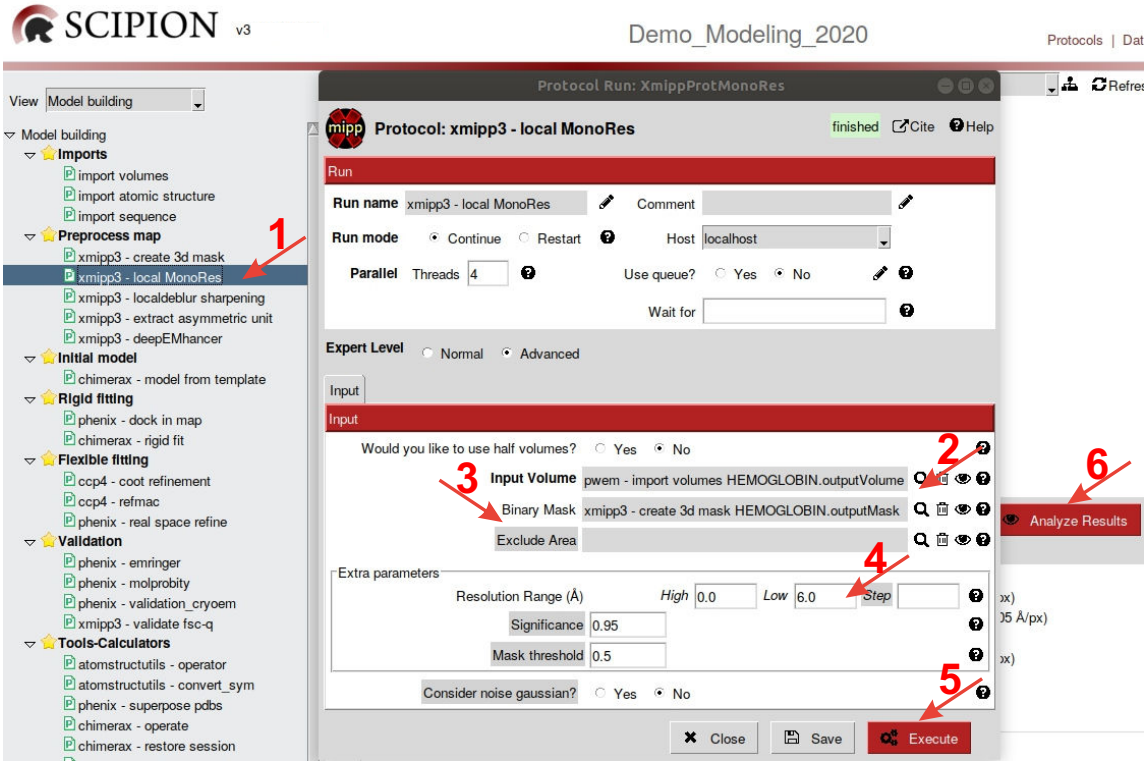


Figure 11: Completing the protocol to estimate the local resolution of the `metHgb` map.

Execute this protocol (Fig. 11 (5)) and analyze the results (6). The menu of results (Fig. 12 (A)), among other views, shows the histogram of local resolutions (1) and the resolution map in *ChimeraX* (2). The histogram of resolutions, which displays the number of map voxels showing a certain resolution, allows to conclude that the majority of voxels evidence a resolution between 3.2 and 3.5 Å, quite close to the published map resolution (3.2 Å).The resolution map shown by *ChimeraX* details the resolution of each voxel (Fig. 13). The bar on the left indicates the color code for resolution values.

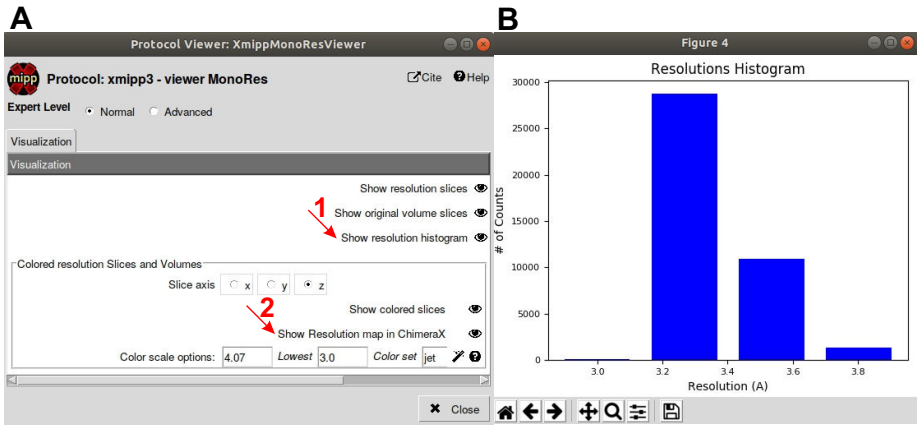


Figure 12: xmipp3 - local MonoRes menu of results (A) and histogram of resolutions (B).

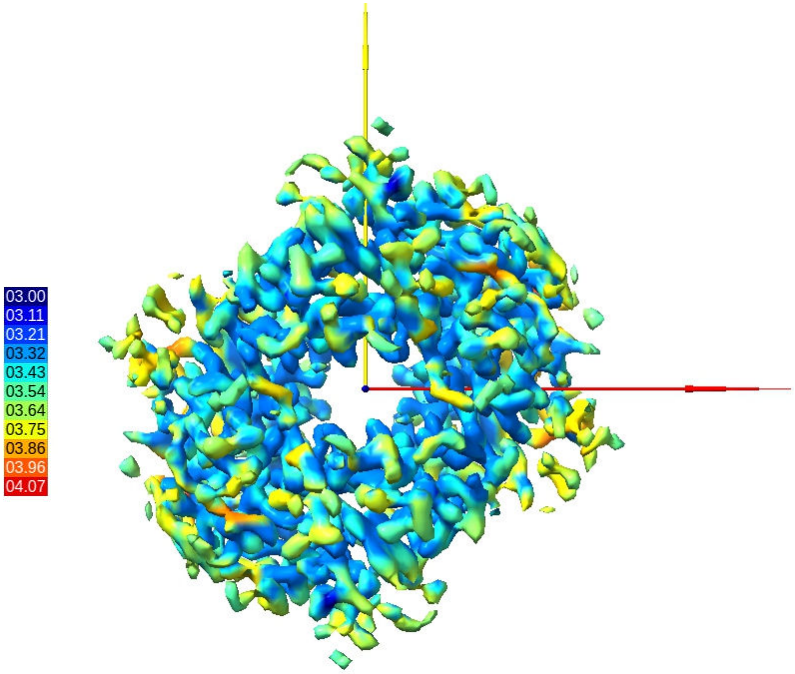


Figure 13: Resolution map in *ChimeraX*.

Local resolution values of the input map allow to compute the sharpened map

by the xmipp3 - localdeblur sharpening protocol, which implements an iterative steep descending method that not requires initial model. To accomplish this step, open the protocol (Fig. 14 (1)) and include the starting map (2) and the map of resolution values (3), maintaining the default values for the rest of parameters (4, 5).

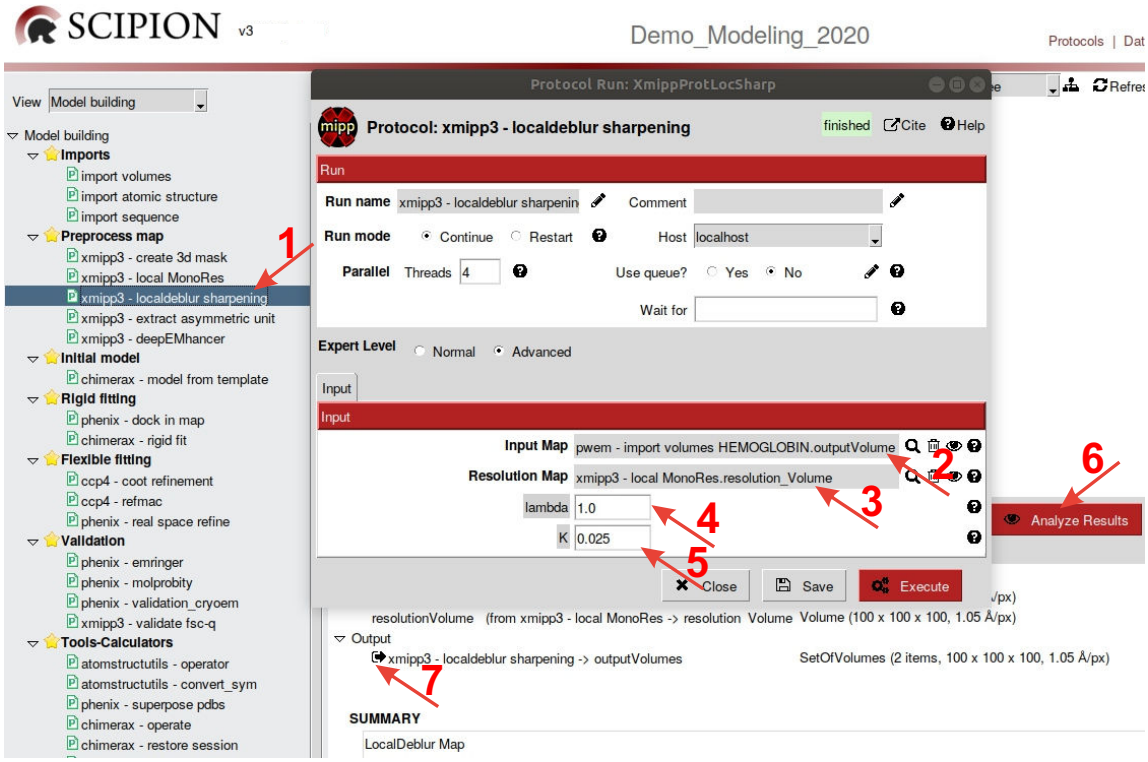


Figure 14: Filling in the protocol to compute the sharpened map.

After two iterations, the sharpening algorithm reached the convergence criterion, *i.e.* a difference between two successive iterations lower than 1 %, and stopped. The two maps obtained in the respective iterations can be observed with *ShowJ* by clicking the black arrow shown in Fig. 14 (7) with the right mouse botton and selecting `Open with DataViewer`. Resulting map for each iteration will be shown, as indicated in Fig. 15. Visualization in *ChimeraX* is also possible selecting `Open with ChimeraX` in the menu option `File` (Fig. 15 (1)).

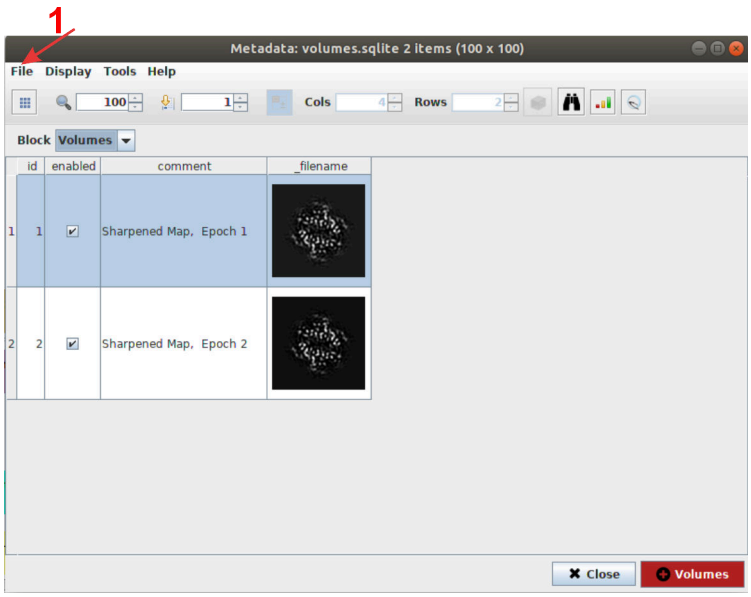


Figure 15: Sharpened maps generated after two iterations.

Additionally, by clicking `Analyze Results` (Fig. 14 (6)) the sharpened map obtained after the second iteration, *i.e.* the `last` map, can be also visualized and compared with the initial one in *ChimeraX* (Fig. 16).

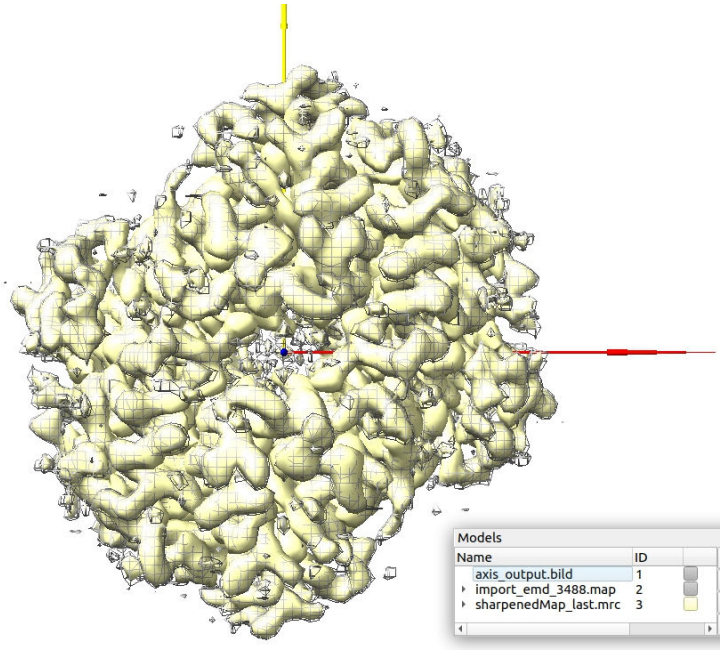


Figure 16: *LocalDeblur* `last` iteration sharpened map (yellow surface) and input map (grey mesh) in *ChimeraX*.

b) Sharpening with *DeepEMhancer*

DeepEMhancer is an alternative automatic sharpening method based on deep learning ((Sanchez-Garcia et al., 2020)), implemented in *Scipion* in the protocol xmipp3 - deepEMhancer (Appendix 11). Open this protocol (Fig. 17 (1)) and complete it as indicated. Since only the refined map is available, we are not going to use half maps (2). Include your map (3), the type of normalization desired (4) and the deep learning mode to use (5), in this particular case `highRes` due to the map high resolution.

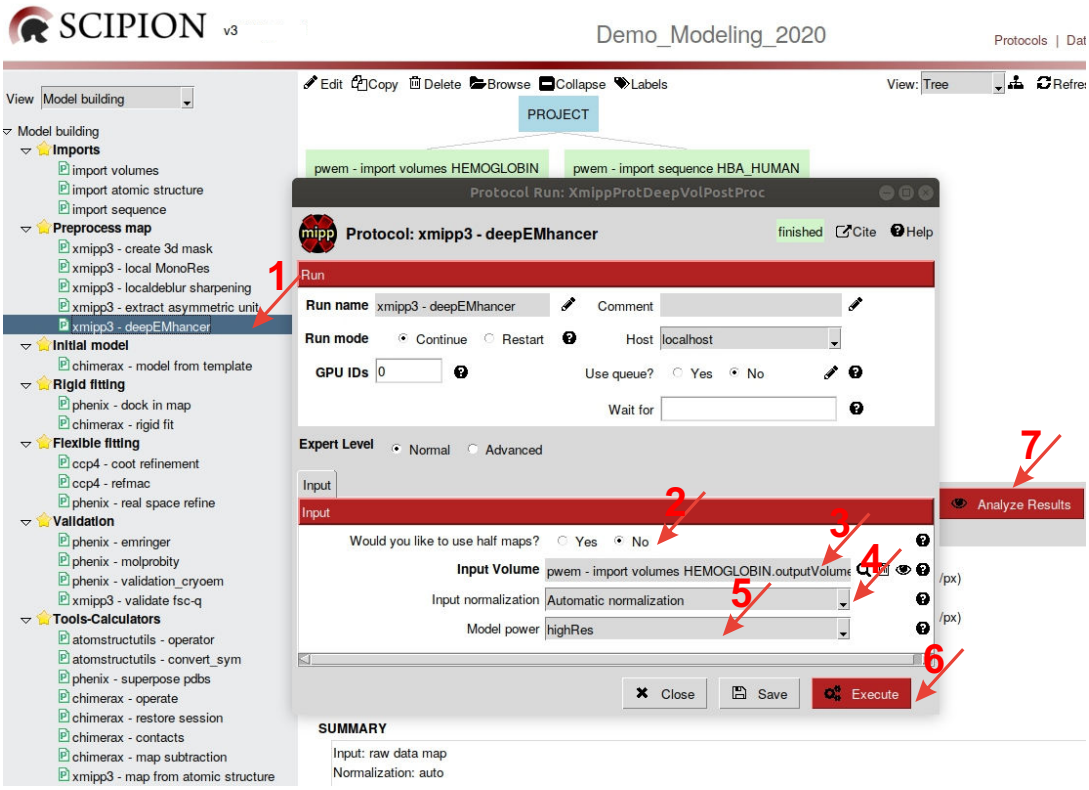


Figure 17: Filling in the protocol to generate a sharpened map with *DeepEMhancer*.

After executing the protocol (Fig. 17 (6)), we can check the results (7). *ChimeraX* viewer will open and show the sharpened map compared with the initial one (Fig. 18).

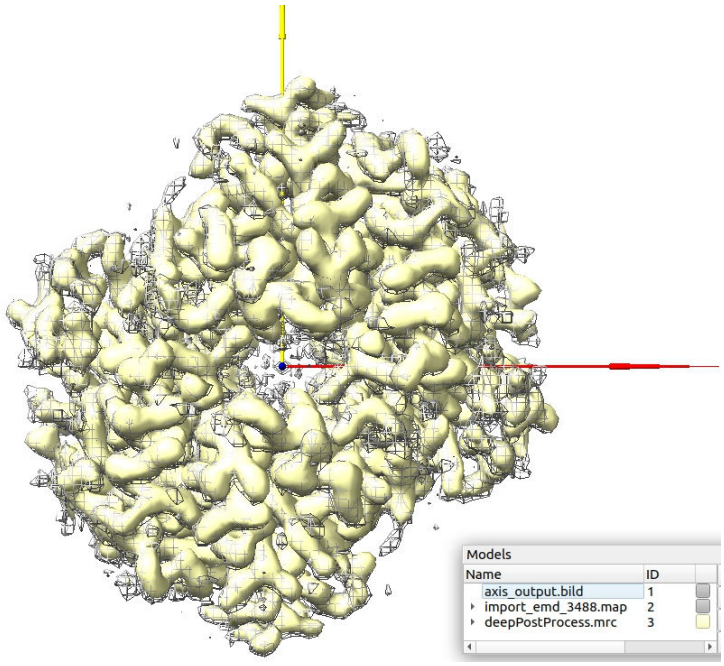


Figure 18: *DeepEMhancer* sharpened map (yellow surface) and input map (grey mesh) in *ChimeraX*.

Comparison of maps

Realize that at this point we have generated two optimized maps derived from the initial one. Additionally, some other maps could have been obtained using other map optimization methods. A comparison among them would be interesting to consider which one(s) of them should be used as input in next steps of modeling workflow. The ideal map for tracing the atomic structure should include as many details and connections as possible and, at the same time, preserve the density areas of the initial map. In other words, we can use the best sharpened map (with higher resolution) corroborating that it does not make up new densities, absent in the starting map. Nevertheless, choose "the best" sharpened map could be difficult sometimes, especially if the map is very big or there are some regions optimized in one of the sharpened maps and other areas optimized in the other one. In that case, you can use several maps at the same time, having all of them perfectly aligned

according to the same origin of coordinates.

In the tiny example shown in this tutorial we are working with a high resolution map and there are almost no differences in resolution between the starting map and the two derived sharpened maps, although this is not usually the case in real life. In this quite uncommon case the initial unsharpened map would be enough to trace the atomic structure. However, in order to detail the method, the starting map and their two sharpened ones will be used simultaneously.

Extraction of the map asymmetric unit

Since smaller volumes usually include lower number of individual structural elements, making easier fitting models in maps and simplifying modeling process, the part of the map chosen to work with will always be the smaller asymmetrical subunit of the starting loaded map, also known as asymmetric unit (ASU). The size of the ASU thus depends on the symmetry order of the initial volume. The higher the symmetry order, the smaller the ASU. The atomic structure of the whole volume will be obtained straight forward by simply repetition of the ASU structure according to the symmetry order. Then, the first step to simplify the complexity of the initial volume is extracting the ASU. This task can be accomplished by using the *Scipion* protocol xmipp3 - extract unit cell that extracts the geometrical ASU of the map (Appendix 12).

Fig. 19 shows how to fill in this protocol form (1). Consider that in this particular case the protocol will be run three times, one with each map (the initial one and the two sharpened derived ones). Include each map in a protocol form parallel to that shown in Fig. 19 (2). Since `metHgb` macromolecule shows symmetry C2, we have selected cyclic symmetry (Cn) as type of symmetry (3), and 2 as symmetry order (4). The angle offset selected (5) turns -45º around the Z axis the mask used to create the ASU. The two wizards on the right (6, 7) help you to select the radii to delimit a fraction of the map comprised between the coordinate origin (inner radius 0.0) and the maximum radius (outer radius 50.0). The final extracted volume will be slightly higher than the ASU due to the expand factor 0.2 (8). The respective tutorial appendix 12 includes a comprehensive explanation of the meaning of parameters.

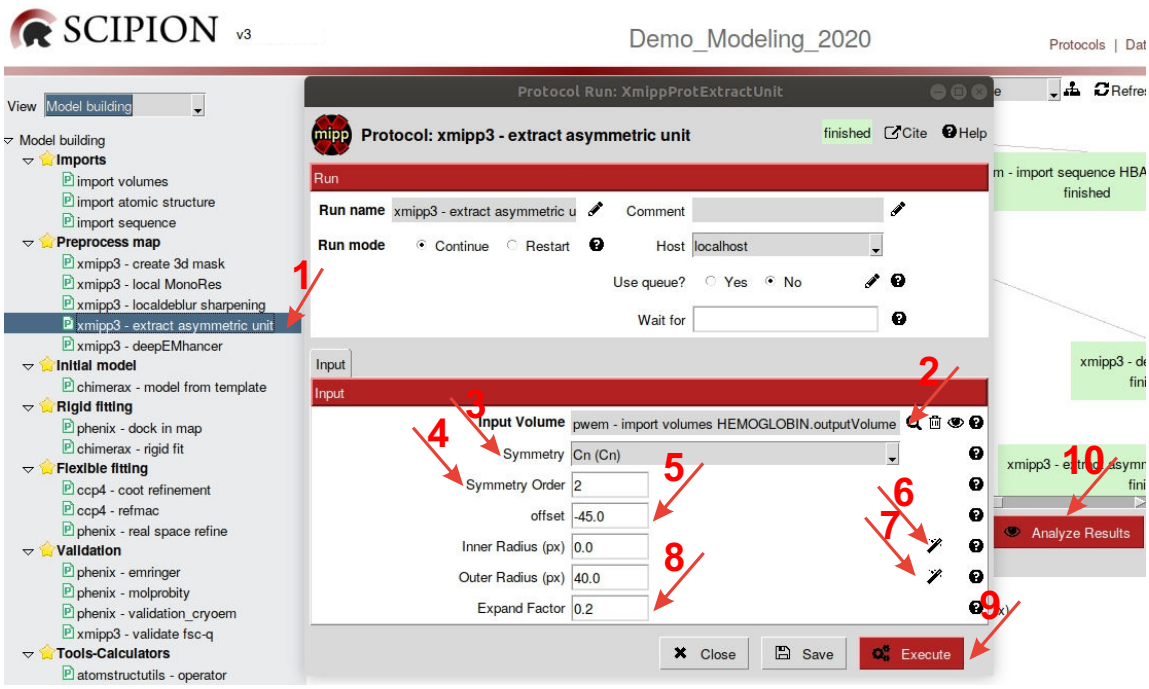


Figure 19: Extracting the map asymmetric unit (ASU).

After executing the protocol (Fig. 19 (9)), the resulting expanded ASU can be observed (10) with *ChimeraX* (Fig. 20). Note the additional expanded volume of the ASU on the left side of the figure. The ASU itself, on the right side, constitutes the half volume. Since the total volume contains the structure of four proteins, we can anticipate that this smaller asymmetrical subunit of the initial volume contains two proteins, one α and one β `metHbg` subunits. Then, the respective structures of these two proteins could be fitted in the map ASU simultaneously or in successive modeling workflow steps.

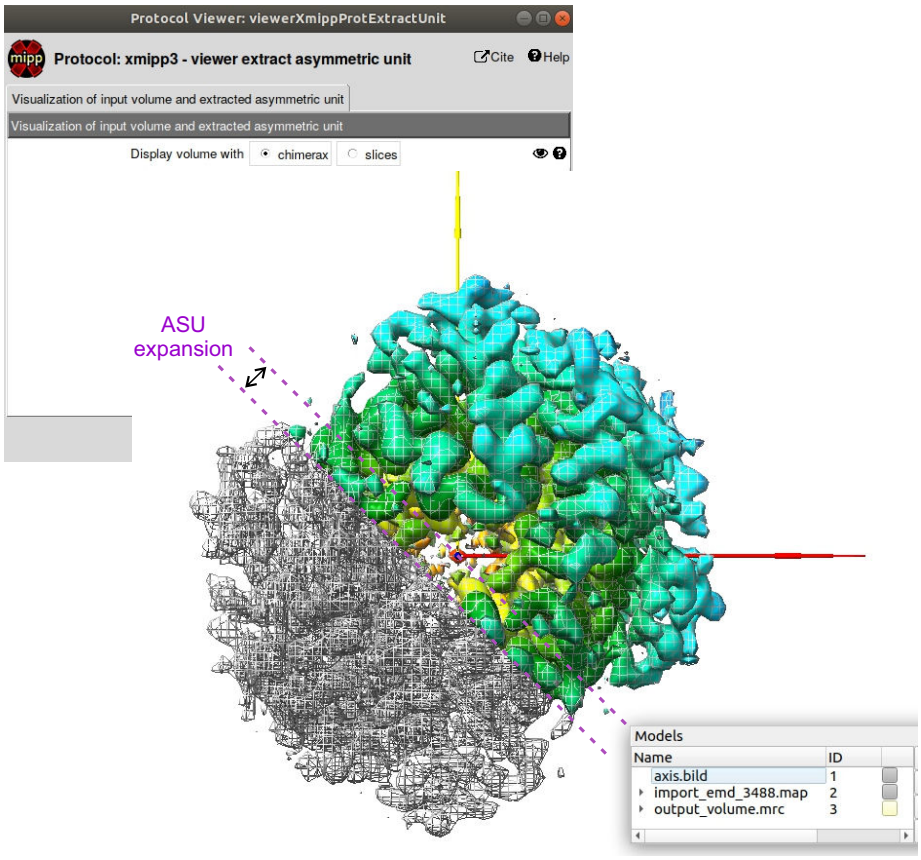


Figure 20: Expanded ASU (yellow-green-blue) and initial volume (gray) visualized with *ChimeraX*. The purple broken line on the right delimits the ASU (right) and its expanded volume (left).

6 Structure Prediction by Sequence Homology. Searching for Homologues

As we have mentioned above, in this tutorial we are going to use tools that allow to predict the atomic structure from sequence homology.

Structure prediction by sequence homology only requires the sequence itself of the specimen that we would like to model, from now ahead the *target sequence*, and the access to databases to seek structures or *templates* of homologous molecules. The

sequences of homologous molecules show statistically significant similarity because they share common ancestry. Since the sequence encodes the structural information, from high similar sequences necessarily follows high similar structures. Structures from the nearest homologous molecules will thus be preferred over remote relative ones. Remark that molecules containing several domains usually require independent searching for homologous templates of each domain. A small review about sequence similarity searching can be found in (Pearson, 2013), and in (Kryshtafovych et al., 2018) the assessment of current *template*-based modeling methods, many of them implemented as fully automated servers. Modeling tools appropriate to search for remote homologous *templates*, folding recognition and *template*-free methods (*ab initio*), as well as *de novo* modeling tools, which besides sequences use the volume itself, have still to be included in *Scipion* framework.

How to identify *templates* of the *target sequence*

Similarity searching programs like `BLAST` (Fig. 21) (Altschul et al., 1997), available in `https://blast.ncbi.nlm.nih.gov/Blast.cgi`, use the *target sequence* (1) to screen the structure-containing database `PDB` (2). Selecting or excluding a particular organism is an option (3). We usually start our searching selecting the organism in which we are interested or the closest evolutionarily related ones. If no similar sequences are found in these organisms, unrelated organisms may be selected or no one at all. Different searching algorithms are available (4) and one of them has to be selected. After executing `BLAST` (5) a list of score-ordered *templates* is retrieved.

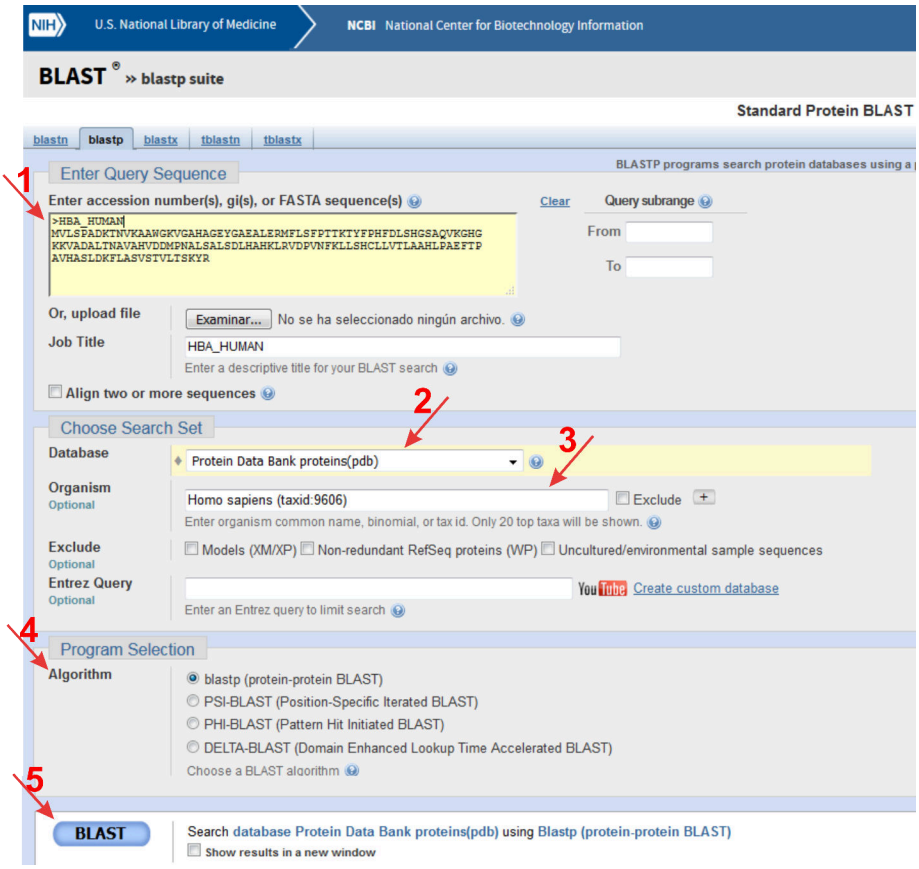


Figure 21: Form of the similarity searching program `BLAST`.

Of course, the closest relatives to human `Hgb` subunits, structurally characterized, will be their own structures contained in `PDB-5NI1`. However, in this tutorial we are going to assume that in our example the closest relatives to the human `Hgb` α and β subunits are the respective `Hgb` subunits (identity 49.3% and 45.21%) of the antarctic fish *Pagothenia bernacchii* (Camardella et al., 1992). The atomic structure associated to this *template* has `PDB` accession code `1PBX`. Information about the structure can be checked in `https://www.rcsb.org/structure/1PBX`. In general, it is a good idea to read the information related with the *template*, do it so and answer the following questions: (Answers in appendix 1; **Question 6_1**)

- How was this structure obtained (X-ray diffraction, EM, NMR)?
- What resolution does it have?
- How many chains does it include?

NOTE: *ChimeraX* also incorporates the possibility of running the BLAST algoritnm, although with lower number of options than those shown in Fig. 21. Nevertheless, if you know that there are high similar homologous sequences with associated structure, you can skip this searching step "outside" *Scipion* and go to the next step to get directly your *template* and your *target model*.

7 Moving from sequence to atomic structure scenario

In this section we are going to obtain the initial *model* of our *target* sequence, in this case the Hgb α subunit. To perform this task we are going to use the *Scipion* protocol chimerax - model from template. Although this protocol offers several different possibilities to get the right result (see Appendix 19 for details and use cases), we are going to consider that we already have a *template*, that we have found in the previous step and that will be used as protocol input. In addition, although the protocol also allows to get a model including two chains modeled simultaneously using the same *template* (multichain modeling), in our example we are going to model only one chain, the Hgb α subunit. Fig. 22 shows the *Scipion* workflow that we are going to detail in this section. Other possibilities of the protocol usage will be suggested at the end of the chapter.

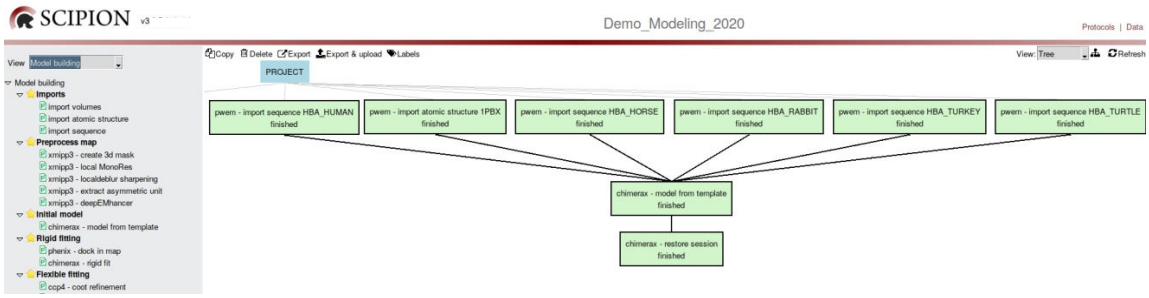


Figure 22: *Scipion* framework detailing the workflow to generate the first model of the human `Hgb` α subunit.

Downloading the atomic structure

Once identified the *template* that we are going to use as structural skeleton of our sequence, we import it into *Scipion* with the protocol import atomic structure (see Fig. 23 (1) and Appendix 13). Select the option for importing the atomic structure from ID (2), write the `PDB` accession code (3) and execute the protocol (4).

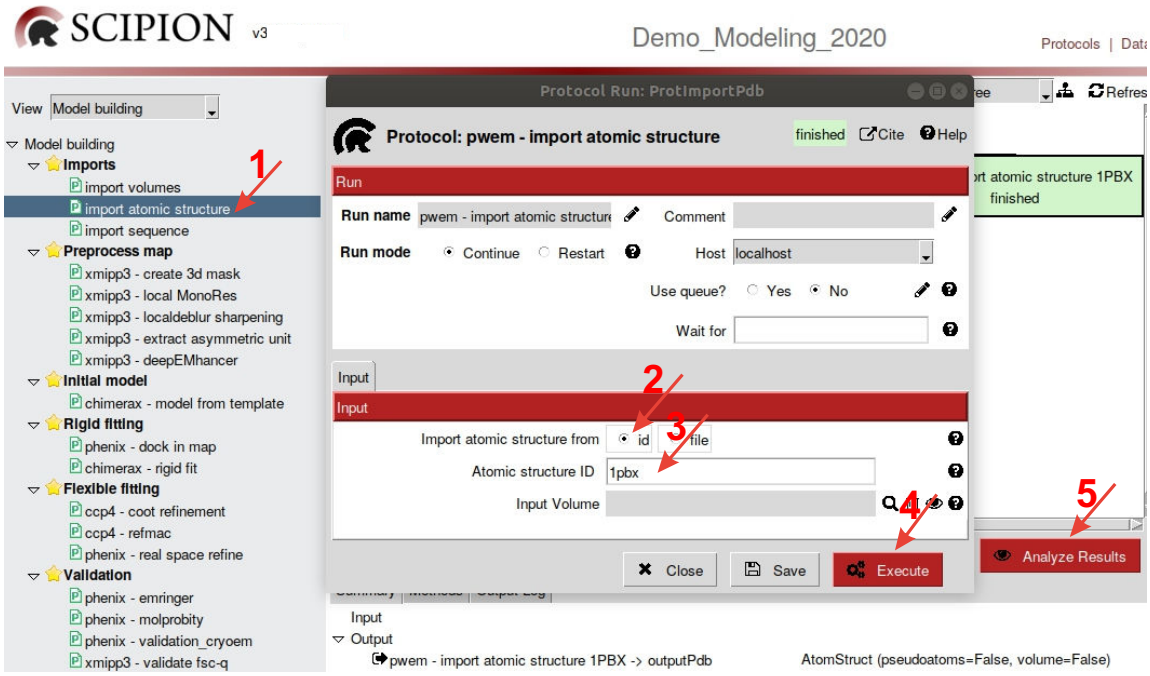


Figure 23: Importing the atomic structure `1PBX`.

You can visualize the imported structure (5) in *ChimeraX* (Fig. 24). By selecting chain `A` in the *ChimeraX* upper menu (1) you can distinguish the `Hgb` α subunit (2).

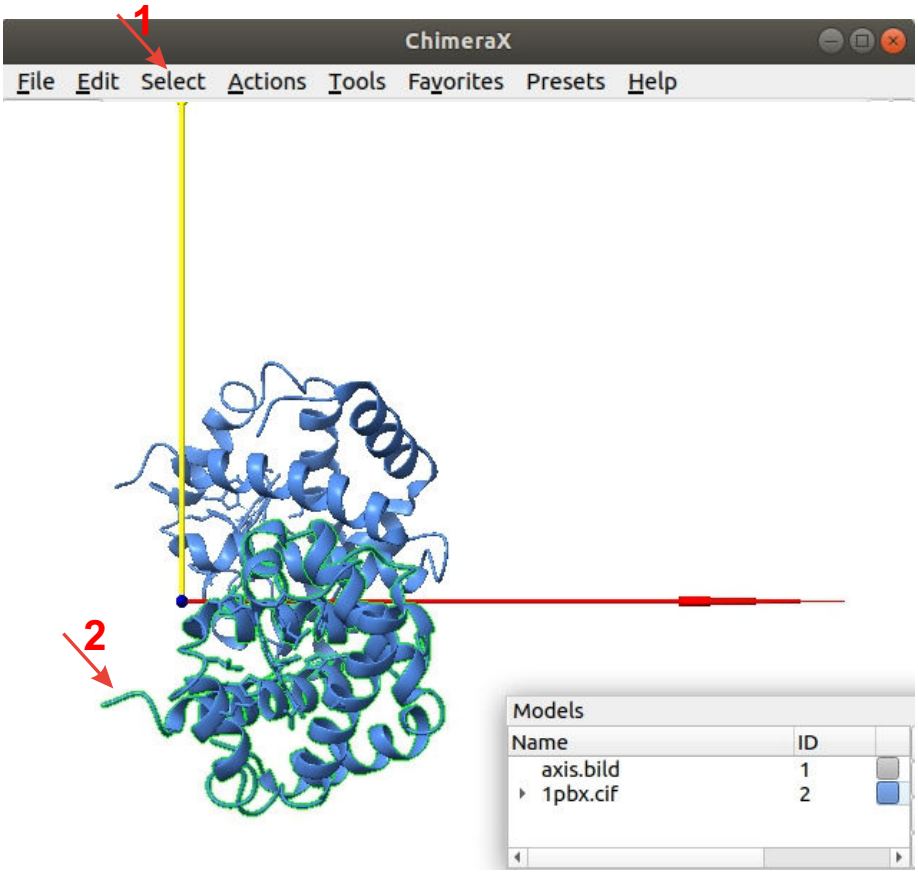


Figure 24: Atomic structure `1PBX` visualized with *ChimeraX*. `Hgb` α subunit (selected chain `A`) is shown green-highlighted.

Structural models of human metHgb subunits from templates

Modeller (Sali and Blundell, 1993) is one of the computational web services used by *ChimeraX*, which provides the interface to run the program. Working with *Modeller* requires a license key, which is provided free of charge for academic users. *Modeller* allows two types of modeling computations to generate theoretical models, *template*-based (sequence homology) and *template*-free (*de novo*, only for missing segments). In this tutorial we are going to consider the first one: structure prediction by sequence homology. Requirements for this type of modeling are the *template* structure and a sequence alignment including sequences of *target* and *template*.

`Note before starting!!!`: We are going to use a *ChimeraX*-derived protocol for the first time in this tutorial (chimerax - model from template, Appendix 19). Remark that this use of *ChimeraX* is completely different from the use of *ChimeraX* as a visualization tool, as we have done previously. By using the *ChimeraX* graphics window, opening it from the *Scipion* button `Analyze Results` we can observe protocol results but we CANNOT save anything. However, using *ChimeraX* as a tool, as it is the case in *Scipion ChimeraX*-derived protocols, we can perform different tasks, taking advantage of the available *ChimeraX* tools and, finally, we CAN save the obtained results and the working session.

- Preparing your sequence alignment:
  In addition to the ways to obtain the *target-template* sequence alignment using *ChimeraX*, this alignment can be also generated in the *Scipion* protocol chimerax - model from template (Appendix 19). This protocol allows selecting between pairwise and multiple sequence alignments. Besides producing more reliable alignments, especially for more distantly related sequences, multiple sequence alignments provide more structural information than pairwise alignments; they locate conserved regions in the molecule, thus improving predictions of structural arrangements due to mutant residues or residues that differ between *template* and *target* sequences (Pearson, 2013). For this reason, in this tutorial we are going to perform a multiple sequence alignment. Additionally, you can also test the available tools to perform pairwise alignments.

  Besides *target* and *template* sequences, other sequences are needed to accomplish a multiple sequence alignment. The type and number of the sequences included depends on the sequence conservation, although they have to allow differentiating conserved regions. As an example, our multiple sequence alignment will include four more `Hgb` α subunit sequences from organisms located between human and fish in the evolutionary scale: *Equus caballus* (Horse), *Oryctolagus cuniculus* (Rabbit), *Meleagris gallopavo* (Wild turkey), *Aldabrachelys gigantea* (Aldabra giant tortoise). Download these sequences one by one from `UniProtKB` database filling in the import sequence protocol form

with the appropriate accession codes, `P01958, P01948, P81023`, and `P83134`, respectively (Fig. 25). A similar process has to be followed for `Hgb` β subunit, importing `UniProtKB` sequences `P02062 (HBB_HORSE), P02057(HBB_RABIT), G1U9Q8 (G1U9Q8_MELGA)` and `P83133 (HBB_ALDGI)`.

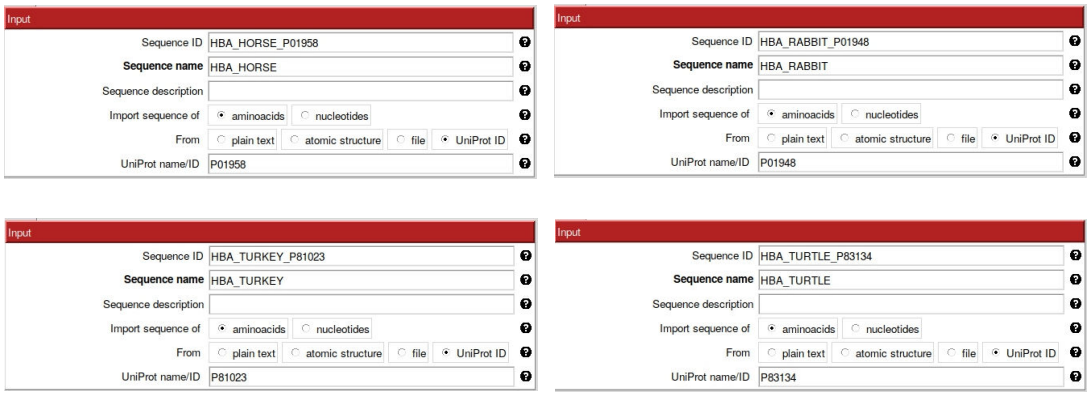


Figure 25: Importing additional sequences to perform the multiple sequence alignment.

- Access to *Modeller* in *ChimeraX*:
  The protocol chimerax - model from template allows direct opening of the multiple sequence alignment in *ChimeraX* and then, access to *Modeller* via web service. Fill in the protocol form (Fig. 26 (1)), including the *template* `1PBX` previously imported (2), the particular chain of interest (use the wizard to select it (3)) and the *target* sequence of human `Hgb` α subunit (4). Since we plan to improve the alignment by including additional sequences to align (5), they will have to be added next (6). Finally, select one of the multiple sequence alignment tools (7).

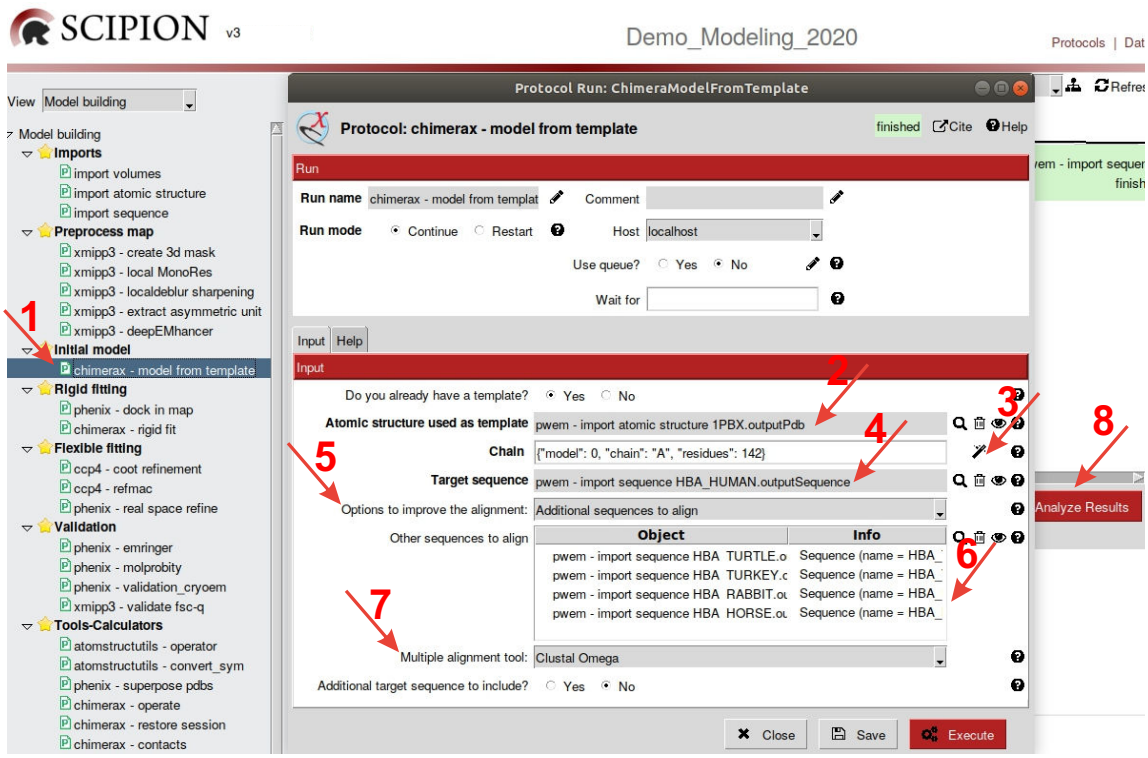


Figure 26: Importing the multiple sequence alignment in *ChimeraX*.

ChimeraX will be opened including this time the multiple sequence alignment together with the *ChimeraX* graphics window (Fig. 27). The *template* selected chain is shown green-highlighted in both windows. As you may observe in the alignment, `Hgb` α subunit is a quite conserved macromolecule; there is only one gap in the alignment because `PRO` (Proline) 47 residue disappeared throughout the evolutionary process.

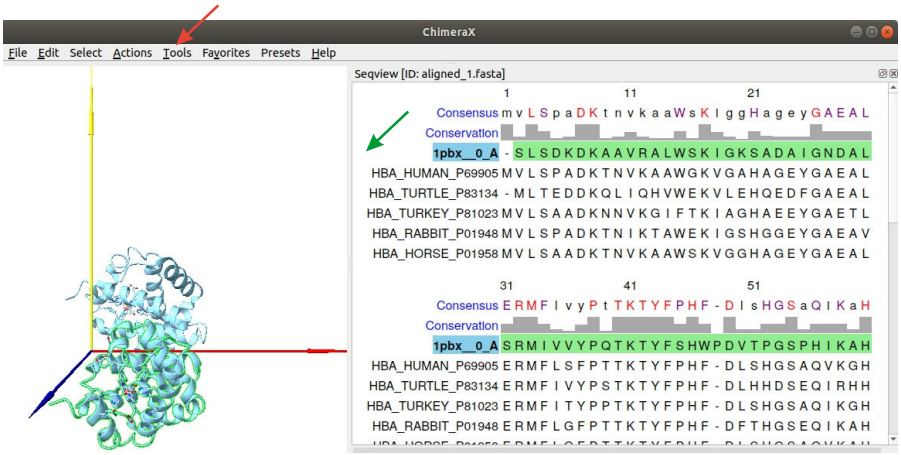


Figure 27: Opening the multiple sequence alignment in *ChimeraX*.

To complete the form that will allow us to get some atomic models of the *target* sequence in *Modeller* web service, we have two possibilities: a) to select `Tools` (Fig. 27, red arrow) `-> Sequence -> Modeller Comparative`, or b) clicking with the right mouse inside the `Seqview` box (Fig. 27, green arrow) and selecting `Structure -> Modeller Comparative Modeling...` in the pop up window. A new window of `Modeller Comparative` will be open (Fig. 28 (A)), that we have to fill in. `Sequence alignments` (1) should include the *template* sequence. In the `Target sequences` section (2) we should include the *target* sequence that we would like to model, `HBA_HUMAN_P69905` in this particular case. *Modeller* license key has to be included here (3). The number of output models, 5 by default, can be also specified (4). Since the target sequence that we would like to model should include `non-water HETATM residues` (`HEME` group) we are going to select this choice as `Advanced option` (5). Finally, press `OK` (6) to start the computation of potential models for your *target* sequence.

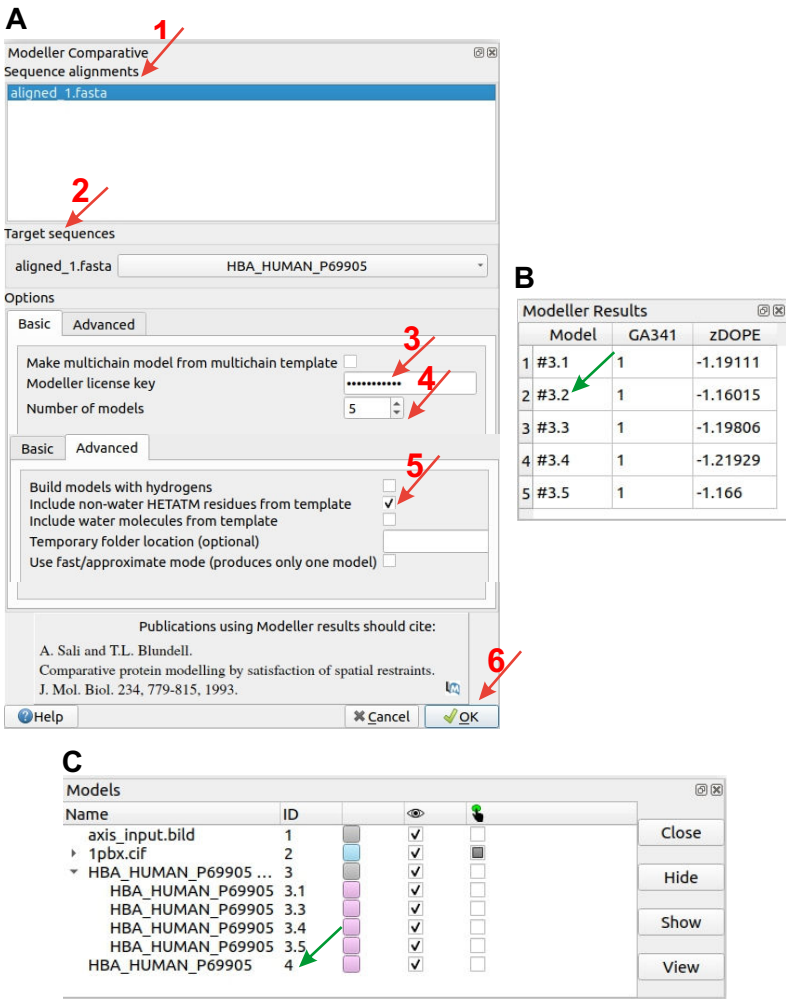


Figure 28: (A) Completing the form to access to homology modeling with *Modeller*. (B) Resulting *model scores*. (C) *ChimeraX* `Models` panel.

In *ChimeraX* main graphics window, lower left corner, you may see the status of your job. After a while, five possible atomic structures, from now ahead *models*, are retrieved for the *target* sequence (Fig. 28 (B)) together with their assessment scores. Column `GA341` of `Modeller Results` indicates the score derived from statistical potentials (values in `[0,1]`; `> 0.7` for reliable *models*). Column `zDOPE` (normalized Discrete Optimized Protein Energy) score depends on the atomic distance (negative

values for the better *models*). You can check every model numbers in *ChimeraX*'s main menu (`Tools -> Models` (C)).

For this tutorial we are going to select *model* `#3.2` (Fig. 28 (B), green arrow). Renaming this model is the first step to save it. We can rename the model by typing in the *ChimeraX* command line:

```
rename #3.2 id #4
```

The renamed atomic structure will appear in the `Models` panel (Fig. 28 (C), green arrow). To track this new atomic structure in the *Scipion* workflow, we can write in the *ChimeraX* command line:

```
scipionwrite #4 prefix model_from_modeller_3_2_
```

In case that the `Advanced option ''Include non-water HETATM residues from template''` ((Fig. 28 (A, 5)) didn't include the `HEME` group in the retrieved models, an alternative option to have the model with the `HEME` group (residue 144 from the atomic structure `#2` chain A) could be:

```
rename #3.2 id #4
save /tmp/chainA.cif format mmcif models #4
open /tmp/chainA.cif
select #2/A:144
save /tmp/HEME.cif format mmcif models #2 selectedOnly true
open /tmp/HEME.cif
scipioncombine #4,5

scipionwrite #6 prefix Hgb_alpha_
```

`Note`: We have saved the `HEME` group of the *template* `chain A` in a new file that will be opened as *model* `#5`. Finally, the combination of *models* `#4` (retrieved aminoacid *model* of the *target* sequence) and `#5` (`HEME` group of the *template* `chain`

`A`) generates a new model `#6` that will be saved in *Scipion*. A different `model ID` could be selected by the user adding to the last command line `modelid n`.

After closing *ChimeraX*, you can visualize (Fig. 26 (8)) your full predicted *model* (Fig. 29) that includes the `HEME` group (1). The string that we have included as `prefix` in the command line `scipionwrite` will allow us to follow the atomic structure in a more simple manner. You can check the `prefix` in the the name of the saved atomic structure (`Hgb_alpha_Atom_struct_6_006815`) in the `Models` panel of Fig. 29 (1). Interestingly, the suffix number of the saved atomic structure (`006815`) stands for the ID protocol number.

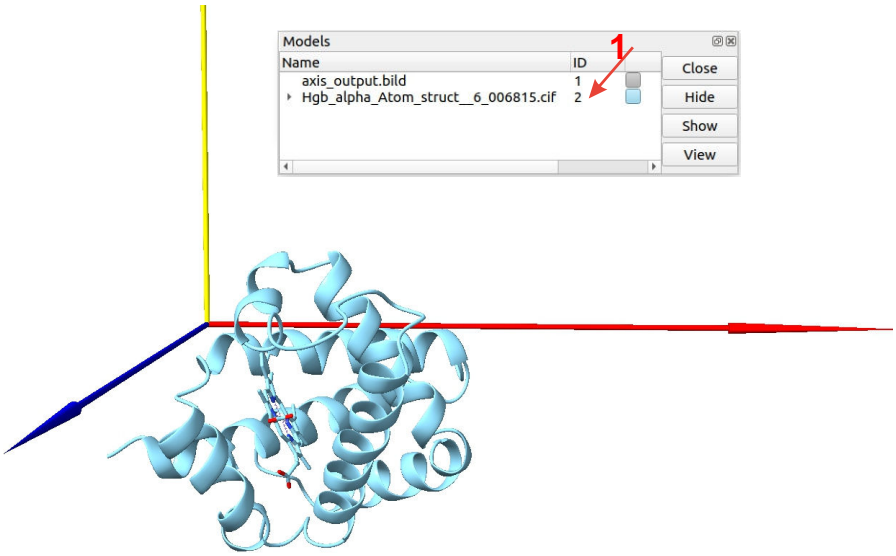


Figure 29: Initial *model* of human `metHgb` α subunit, including the `HEME` group (blue).

In a similar process, you can also obtain the initial atomic structure of the human `metHgb` β subunit. Take into account that in this last case the HEME group is the residue `148` of the chain `B`. The command lines are similar to the previous case of the `metHgb` α subunit if you also select the *model* `#3.2`.

```
rename #3.2 id #4
save /tmp/chainB.cif format mmcif models #4
```

```
open /tmp/chainB.cif
select #2/B:148
save /tmp/HEME_B.cif format mmcif models #2 selectedOnly true
open /tmp/HEME_B.cif
scipioncombine #4,5
setattr #6/A c chain_id B
scipionwrite #6 prefix Hgb_alpha_
```

In addition, we have included a command to change the chain `id` of the second polypeptide from `A` to `B`. In general, to change the chain `ID` you have to write:

```
setattr #model_number/old_ID c chain_id new_ID
setattr #model_number/old_ID r chain_id new_ID
```

Additional exercises for practising

Since the protocol chimerax - model from template allows to use other options, inspect by your own the possible result obtained by:

1. Using as input only the *target* sequence of the human `metHgb` α subunit.
2. Using as input the same atomic structure *template* and the *target* sequences of both the human `metHgb` α and β subunits. Improve the alignment of the human `metHgb` α subunit with additional sequences and improve the alignment of the human `metHgb` β subunit with your own sequence alignment that contains about 30 sequences.

Option of recovering the *ChimeraX* session

If for any reason you decide to go back and check a different *model* from the five *models* initially provided by *Modeller*, you can do it by using chimerax - restore session protocol (Appendix 6). This protocol may be used whenever *ChimeraX* session

had been saved, specifically after using protocols *ChimeraX* `rigid fit`, *ChimeraX* `operate`, *ChimeraX* `chimerax - model from template` and *ChimeraX* `map subtraction`. In addition to the *ChimeraX* command line `scipionss`, command lines `scipionwrite` and `scipioncombine` also save *ChimeraX* session by default. So, if you want to restore a previous session just open the form (Fig. 30, 1), and include the session that you'd like to restore (2).

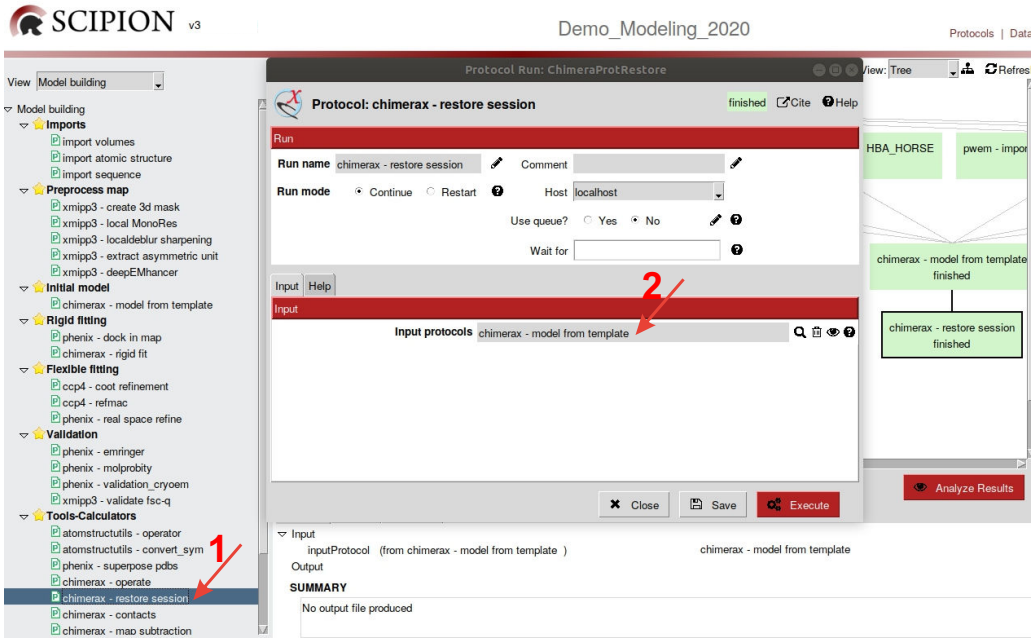


Figure 30: Restoring session in *ChimeraX*.

8 Merging 3D Maps and Atomic Structures: Rigid Fitting

Once we have the predicted *model* of any structural element included in our map, to fit that *model* in the volume constitutes the next step in the modeling workflow. Two protocols have been included in *Scipion* with this purpose, phenix - dock in map (Appendix 25, (Liebschner et al., 2019)) and chimerax - rigid fit (Appendix 7). The first one allows automatic fitting of *models* in *maps*, while the second one only does

it when *model* and *map* are quite close, thus requiring manual fitting in advance. Although there is no a general rule to fit *map* and *model*, because it will depend on the particular problem and on our previous knowledge, in this tutorial we are going to use *PHENIX* `dock in map` application first, followed by the final `Fit in Map` in *ChimeraX* `rigid fit`. Observe these two new steps in the modeling *Scipion* workflow in Fig. 31.

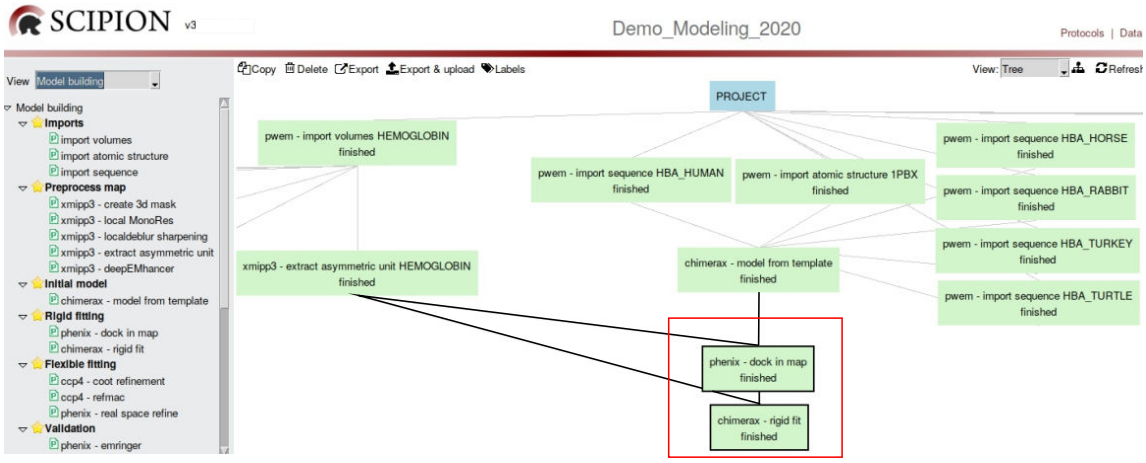


Figure 31: *Scipion* framework detailing the workflow to fit the first model of the human `Hgb` α subunit in the map asymmetric unit.

Initial rigid fit with *PHENIX* `dock in map`

Open phenix - dock in map protocol (Fig. 32 (1)), and complete the form with the the extracted map asymmetric unit (2), the map resolution (3), the *model* of atomic structure previously saved in *ChimeraX* (4), and the number of copies of this atomic structure that we'd like to fit in the map, 1 in this case (5). As an additional exercise you can check the result of fitting two copies of this structure in the initial input map.

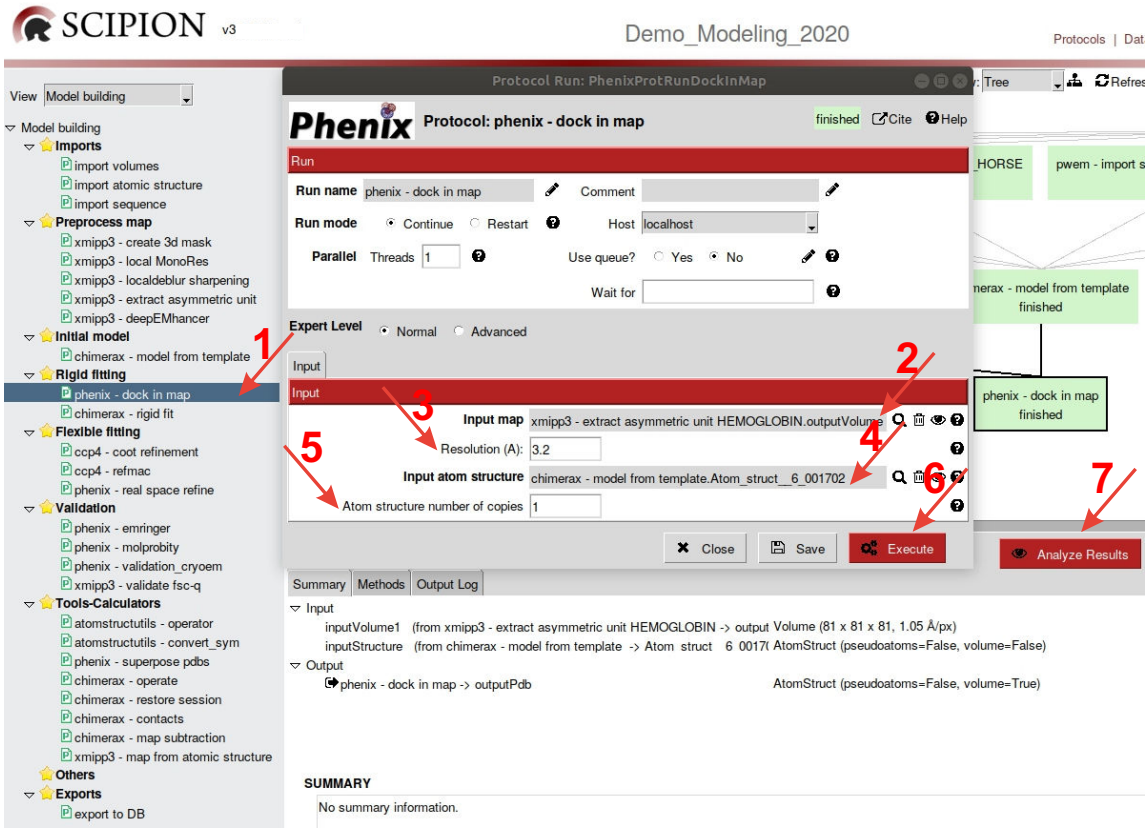


Figure 32: Rigid fit with phenix - dock in map protocol: Filling in the protocol form.

After executing the protocol phenix - dock in map (Fig. 32 (6)), you can check the docking results clicking in `Analyze Results` (7). *ChimeraX* graphics window will be opened (Fig. 33) showing the map and the atomic structure *modeled* in its initial location (pink) and fitted in the map (green) (Fig. 33 (1)).

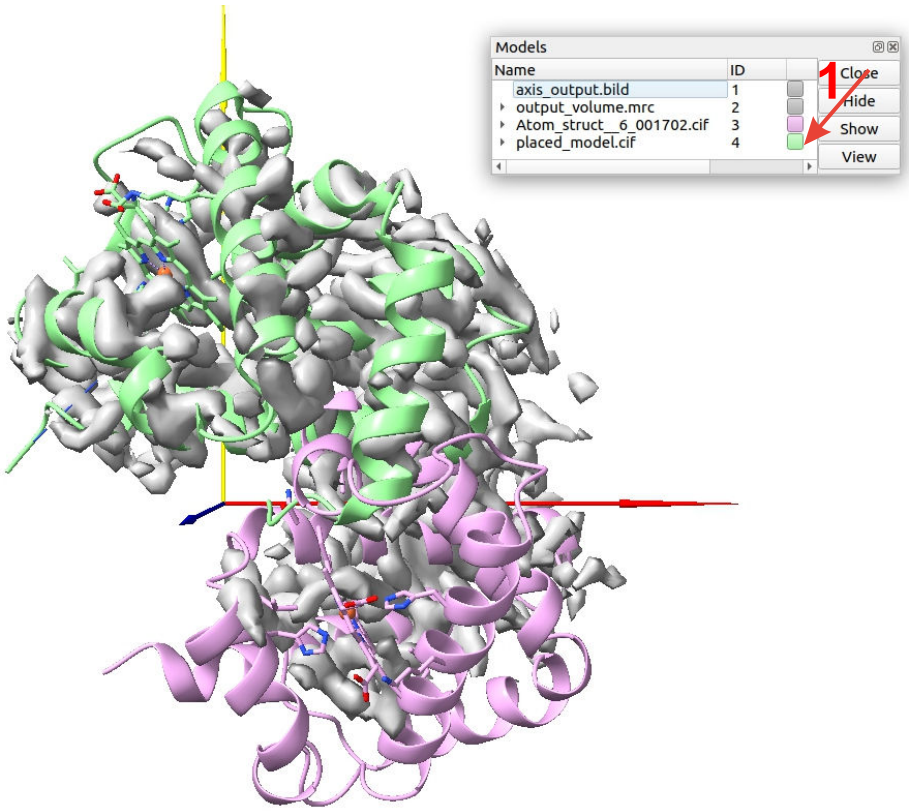


Figure 33: Rigid fit with phenix - dock in map: View of docking results in *ChimeraX*.

A rough inspection of the *placed_model* in Fig. 33 (remark the location of the `HEME` group, for example, which should be moved slightly to the right side) shows that the fitting could be improved a little. The second protocol, chimerax - rigid fit, will help in this purpose.

Completing rigid fit with *ChimeraX* `rigid fit`

`Note before starting!!!`: As we already advised previously, we are going to use a *ChimeraX*-derived protocol (chimerax - rigid fit, Appendix 7). Remark that this use of *ChimeraX* is completely different from the use of *ChimeraX* as a visualization tool. By using the *ChimeraX* graphics window, opening it from the *Scipion* button `Ana-`

`lyze Results`, we can observe protocol results but we CANNOT save anything in *Scipion*. However, using *ChimeraX* as a tool, as it is the case in *Scipion ChimeraX*-derived protocols, we can perform different tasks, taking advantage of the available *ChimeraX* tools and, finally, we CAN save the obtained results and the working session in *Scipion*.

To complete the rigid fitting of the *model* generated in the previous step, open the protocol chimerax - rigid fit, include again the map of the asymmetrical unit (2), and the just fitted *model* of the human `metHgb` α subunit (3), and execute the protocol (4).

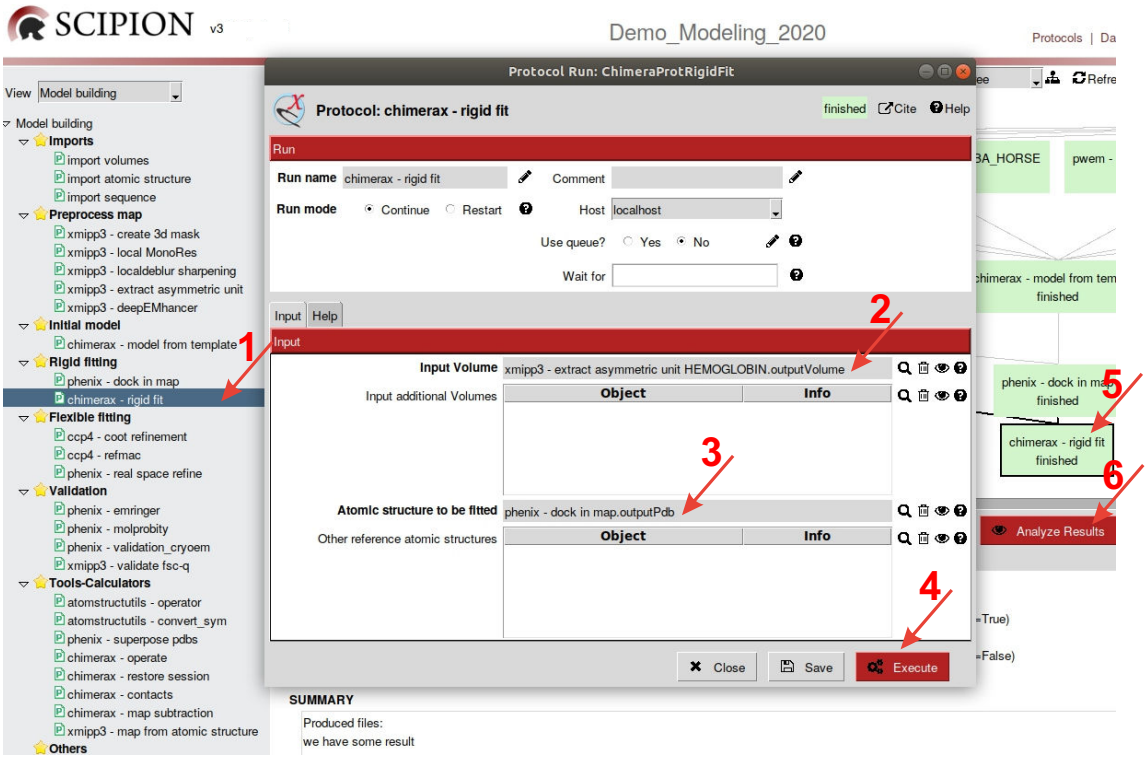


Figure 34: Completing the *ChimeraX* rigid fit protocol form.

Once opened the *ChimeraX* graphical window, we can complete the fitting of the *model* to the *map*, by *ChimeraX* command line or through the *ChimeraX* GUI.

- By *ChimeraX* command line, considering that *map* and *model* have *ID* numbers `#2` and `#3` (Fig. 35 (B)):
  `fitmap #3 inMap #2`

- By the *ChimeraX* GUI: Select in the upper main menu `Tools -> Volume Data -> Fit in Map`. A small window will be opened (Fig. 35 (A)). Select the appropriate *model* to fit in the *map* and press `Fit` (1) to allow the automatic rigid fitting.

A slight movement to the right perfectly fits *map* and *model*, as can be observed in (Fig. 35 (B)). To facilitate the visual inspection of the fitting we can replace the `surface` view of the map by `mesh` as indicated in (A). Observe this time the right placement of the `HEME` group in the *map* density.
To use the side view as additional tool to observe the fit, select in the upper main menu `Tools -> General -> Side View`.

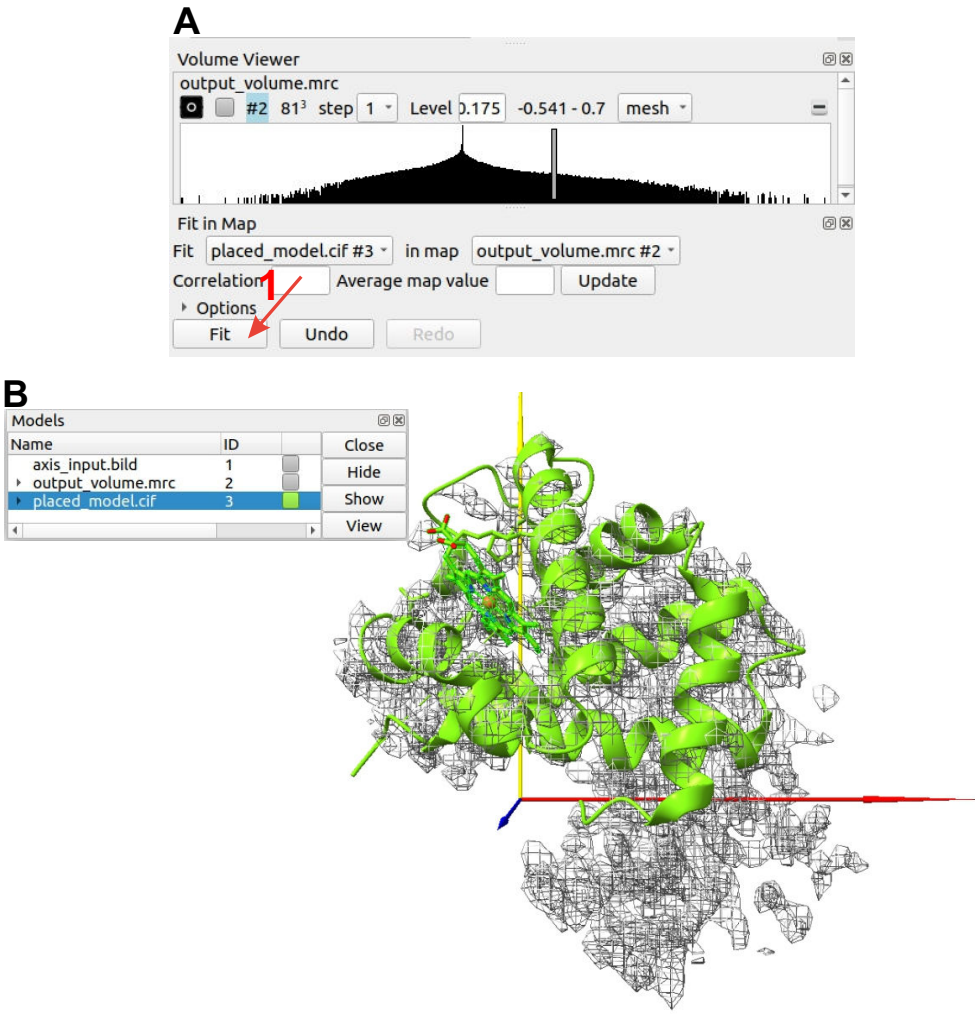


Figure 35: Fit in map with *ChimeraX*.

To track the *ChimeraX* fitted *model* in *Scipion* we have to save it as fitted *model* of the `metHgb` α subunit in the *ChimeraX* command line before closing the *ChimeraX* window:

```
scipionwrite #3 prefix Hgb_alpha_
exit
```

The string that we have included as `prefix` in the command line will allow us to follow the atomic structure in a more simple manner. In particular, if you click

`Analyze Results` (Fig. 34 (6)) the *ChimeraX* graphics window will open again and you can check the `prefix` in the the name of the saved atomic structure (`Hgb_alpha_Atom_struct_3_003753`) in the `Models` panel of Fig. 36 (1). Interestingly, the suffix number of the saved atomic structure (`003753`) stands for the ID protocol number and you can check it by simply surfing the mouse over the protocol (Fig. 34 (5)).

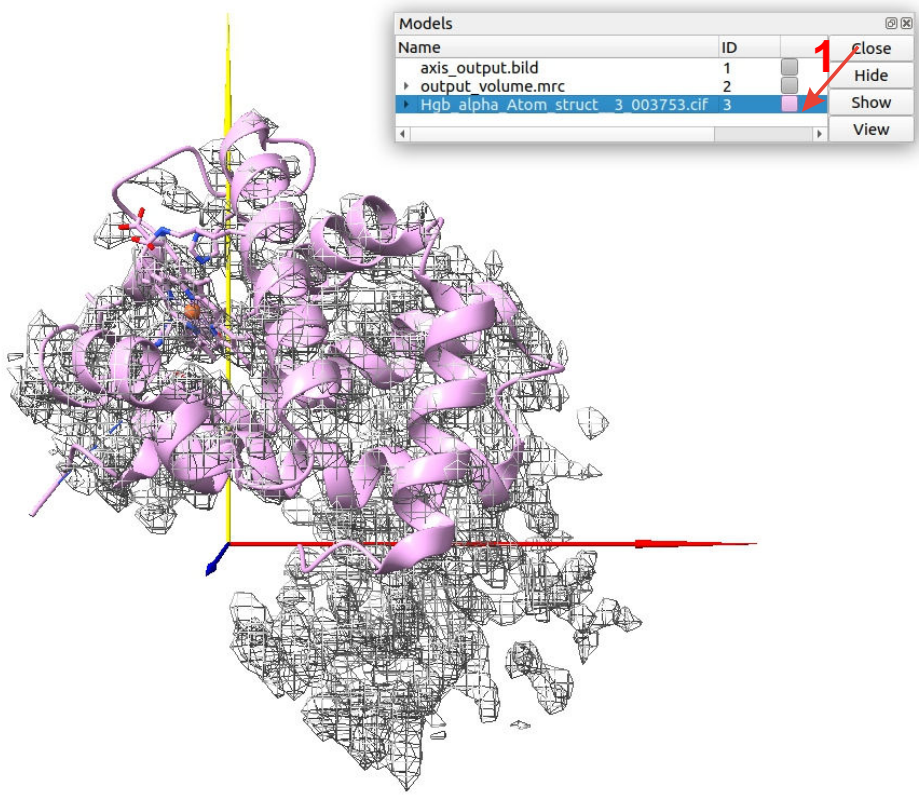


Figure 36: View in *ChimeraX* graphics window of the initial *model* of human `Hgb` α subunit fitted to the asymmetrical unit of the 3D map.

9 Refinement: Flexible fitting

Although the rigid fitting approximates *map* and atomic *model*, a detailed visual inspection of *map* and *model* reveals that some residues are not perfectly fitted. In

order to get a better fit, not only of the carbon skeleton but also of residue side chains, a flexible fitting or refinement has to be accomplished. Refinement can thus be defined as the optimization process of fitting *model* parameters to experimental data. Different strategies, categorized as refinement in the real space and refinement in the Fourier space, can be followed. Implemented in *Scipion* are two protocols for real space refinement, ccp4 - coot refinement (Appendix 8, (Emsley et al., 2010)) and phenix - real space refine (Appendix 23, (Afonine et al., 2018b), manual and automatic, respectively, and one automatic protocol to refine the *model* in the reciprocal space, ccp4 - refmac (Appendix 9, (Vagin et al., 2004)).

Observe the new steps in the modeling *Scipion* workflow in Fig. 37.

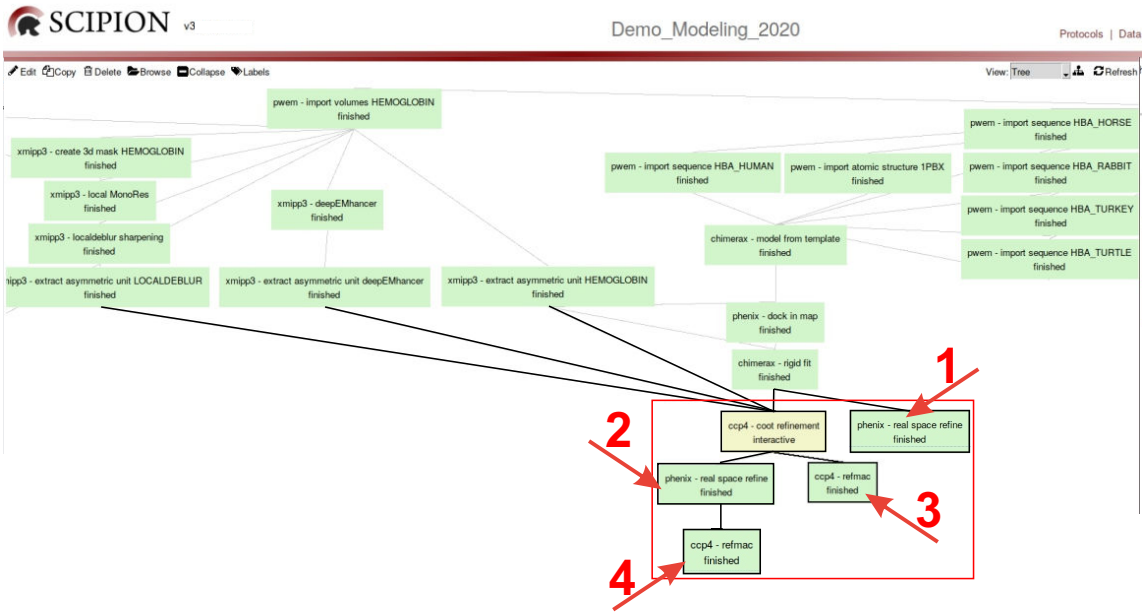


Figure 37: *Scipion* framework detailing the workflow to refine the model of the human `Hgb` α subunit in the map asymmetric unit.

CCP4 *Coot* Refinement

Initially devoted to atomic models obtained by X-ray crystallography methods, *Coot* (from Crystallopgraphic Object-Oriented Toolkit) is a 3D computer graphics tool

that allows simultaneous display of *map* and fitted *model* to accomplish mostly interactive modeling operations. Although this tutorial does not try to show every functionality of *Coot*, but indicate how to open, close and save partial and final *Coot* refined structures in *Scipion*, some of *Coot* basic relevant commands will be shown. Initially, we are going to refine our *moldel* with *Coot*. First of all, open the ccp4 - coot refinement protocol (Fig. 38 (1)), load the map asymmetric units (2), with electron density normalized to 1 (*Coot* performs this step by default), and the fitted structure *model* (3). To read the protocol Help is recommended. After executing the protocol (4), the *Coot* graphics window will appear to start working.

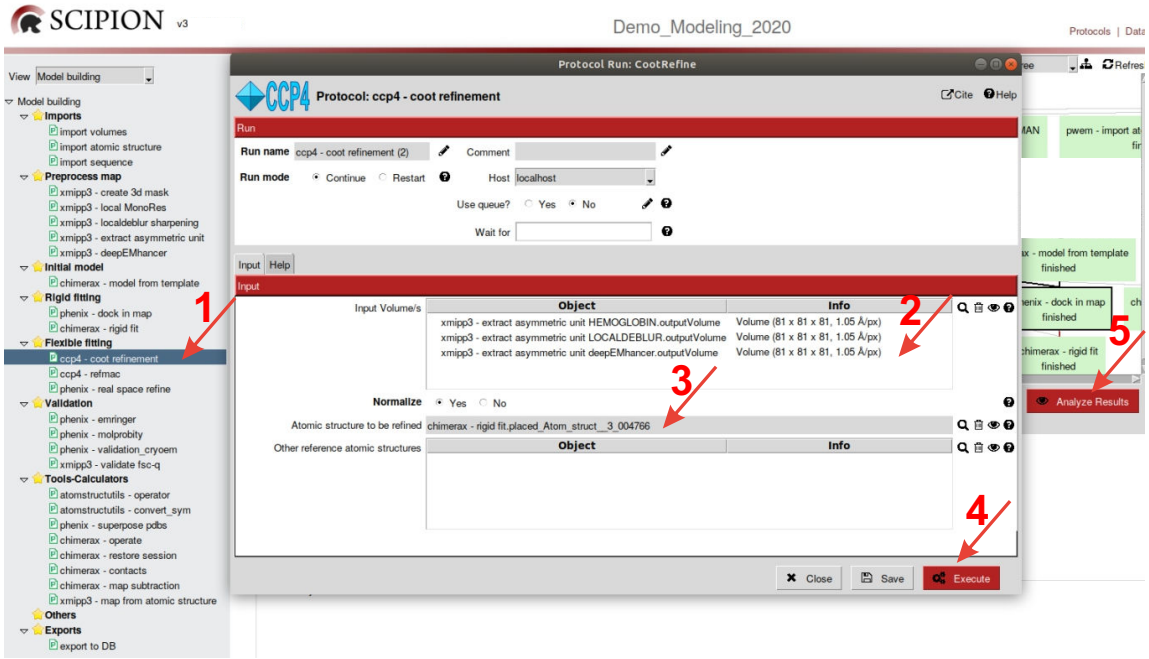


Figure 38: Filling in *Coot* refinement protocol.

To check the objects downloaded in *Coot*, go to the second bar of the main menu and select `Display Manager`. Maps (numbers `#1`, `#2` and `#3`) and model `Hgb_alpha_Atom_struct__3_007124.cif` (number `#0`) are displayed on the left (Fig. 39 (A)). Remark that you have buttons to display a particular map (1) and to increase or reduce map density scrolling it (2). In this case, since we have selected the display of the unsharpened map asymmetric unit, we can only observe this *map* together

with the *model*. If you want to check any of the sharpened maps, select it and scroll it. Note that all maps should be aligned. Try to see differences in details and connectivity of the map to assess if the sharpened maps really optimize the map density compared to the unsharpened one. If this is the case, try to follow the refinement according to the density of the best map (the most optimized one) checking the reliability of the density according to the unsharpened map, specially in the most controversial areas. Since you count on several sharpening maps you can also take advantage of the different map optimizations that you could have in the distinct areas of the map.

To start with the refinement process, we are going to identify the part of *model* misfitted to the density map. Visual inspection would clarify this point in some cases, although direct observation of the `Density fit analysis` might be a shorter way. With this aim, go to the main menu of *Coot* graphical window and select `Validate -> Density fit analysis`. The density fit will be analyzed regarding a specific map. To select any of them, go to the *Coot* right side menu (Fig. 39 (B)(3)) and open the `Select Map for Fitting` window (C). This density analysis, that you can see for the three map asymmetric units in (Fig. 39 (D)) shows that residues 1, 51, 73, 138-142 do not fit perfectly to the density map. The color range scale goes from green color (good fit) to red color (bad fit). There are some differences among maps and, as it was expected, the sharpened maps display higher restraints and show additional residues partially misfitted.

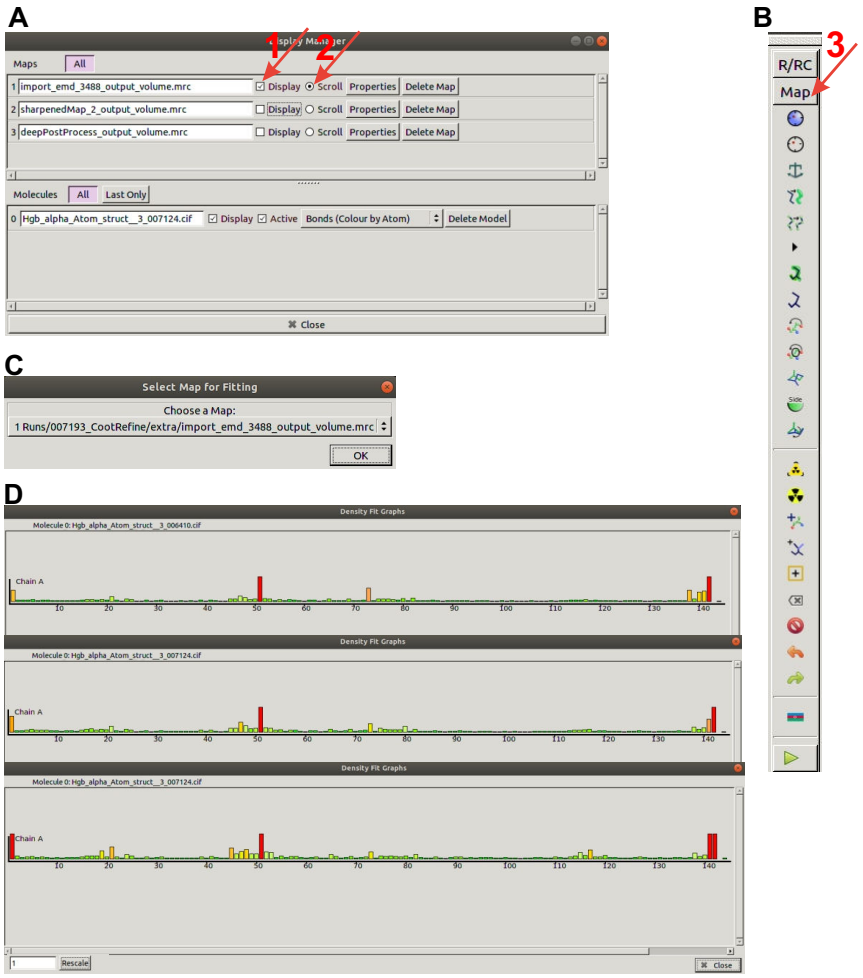


Figure 39: A. *Coot* `Display Manager`. B. *Coot* right side menu. C. *Coot* `Select Map for Fitting` window. D. Map density fit analysis of the *model* in *Coot* regarding the unsharpened map (upper), *LocalDeblur* sharpened map (middle) and *DeepEMhancer* sharpened map (lower).

According to Fig. 39 (B), `MET` residue of the new chain `A` does not fit to the map density. Maybe this residue has been processed post-translationally, as we have anticipated in **Starting Input data** section. To solve this question, go to *Coot* main menu and select `Draw -> Go To Atom... -> Chain A -> A 1 MET` (Fig. 40 (A)). `MET` residue will be located in the center of *Coot* graphics window. Check if

this residue is surrounded by any electron density. As Fig. 40 (B)(1) shows, no density associates to the first chain residue. `MET` will thus be deleted. Then go to the lower right side menu and select the symbol to delete items (B)(2). Select `Residue/Monomer` in the opened `Delete item` window, and click the `MET` residue that you want to delete. Go again to `Validate -> Density fit analysis` and check if the orange bar shown in `MET` residue Fig. 39 (B) disappeared.

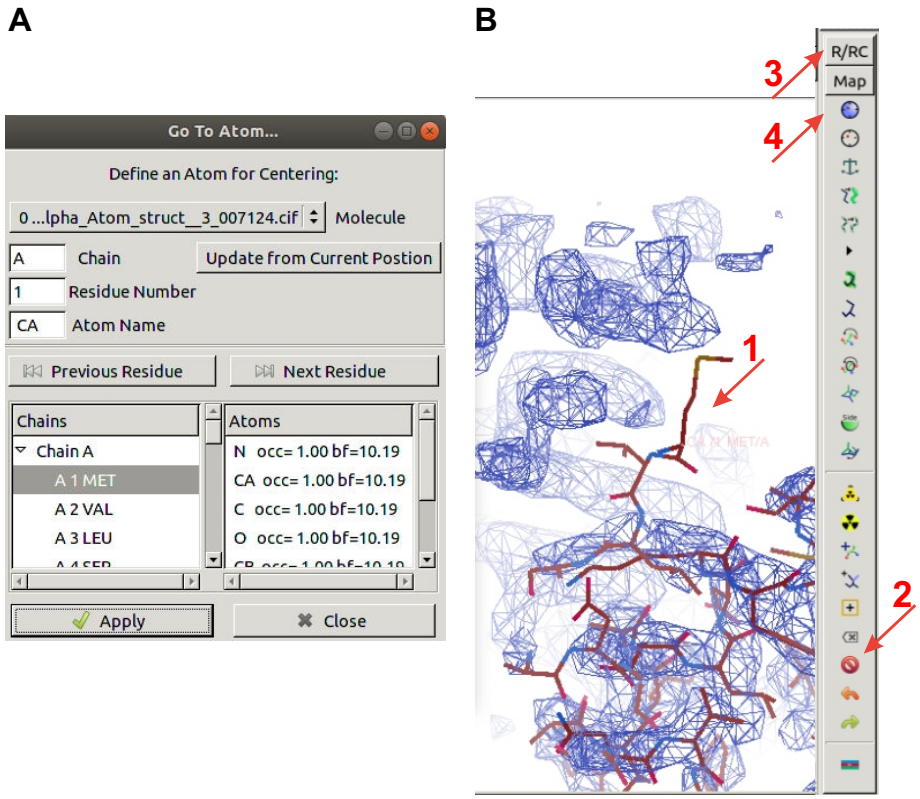


Figure 40: Removing post-translationally processed Methionine residue in *Coot*. Note that the icons shown in the image right side may be partially hidden if the screen is small.

Although in this particular example the most interesting manual refinement strategy could be repair only the misfitted residues because they are very few, in a more general case, in which we could have many misfitted residues, an initial quick refinement may be accomplished. With this purpose, first of all, go to the upper right

side menu (Fig. 40 (B)(3)) and select all four restrictions for `Regularization and Refinement` in the respective window of parameters. Secondly, open the *Scipion* browser (Fig. 41 (1)) and navigate to the `extra` directory, open the `coot.ini` text file (2), and modify the file so it matches the information shown below (3).

```
[myvars]
imol:  0
aa_main_chain:  A
aa_auxiliary_chain:  AA
aaNumber:  4
step:  10
```

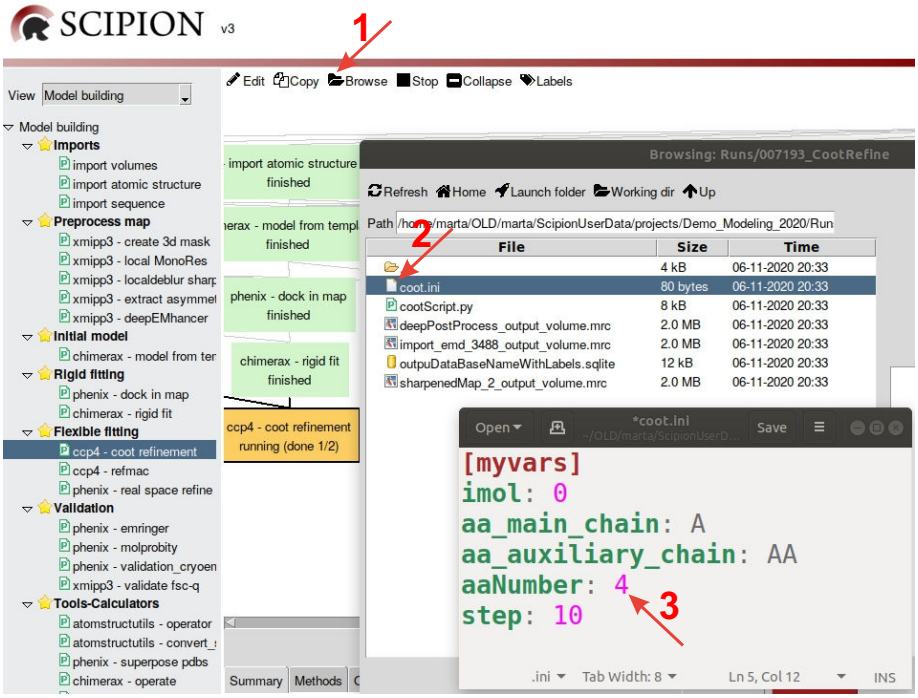


Figure 41: Edit coot.ini file.

Finally, go back to *Coot* window and press "U" to initiate global variables and "z" to refine the next upstream 10 residues. Go through those residues, one by one, and accept refinement if you agree with it. If you disagree with the refinement of

any residue, perform the interactive refinement, visualizing the residue side chain. Repeat the refinement process with "z" until the end of the molecule. Check that the red bar of residue number 53 (Fig. 39) goes missing at the end of this process.

After this partially automatic and partially interactive processing, go to `Draw -> Go To Atom... -> Chain A -> A 2 VAL` (`VAL` is now the first residue of the `metHgb` α subunit) and start the detailed interactive refinement of the initial residues of chain A. To accomplish this interactive refinement of a small group of 5 to 10 residues, select the blue circle in the upper right side menu and click the initial and final residues of the small group of residues (Fig. 40 (B)(4)). The group of selected residues gets flexible enough to look manually for another spatial distribution. Following these instructions, try to solve the misfit that you can find in `TYR` 141 residue at the end of the molecule. Specifically, try to improve the result of the `Validate -> Density fit analysis`, as you can see from (A) to (B) in Fig. 42, moving `TYR` 141 ((A)(1)) to the nearest empty map density ((A)(2)). Accept the refinement parameters after the displacement of `TYR` ((B)(3)). Finally, check the `Density Fit Graph`.

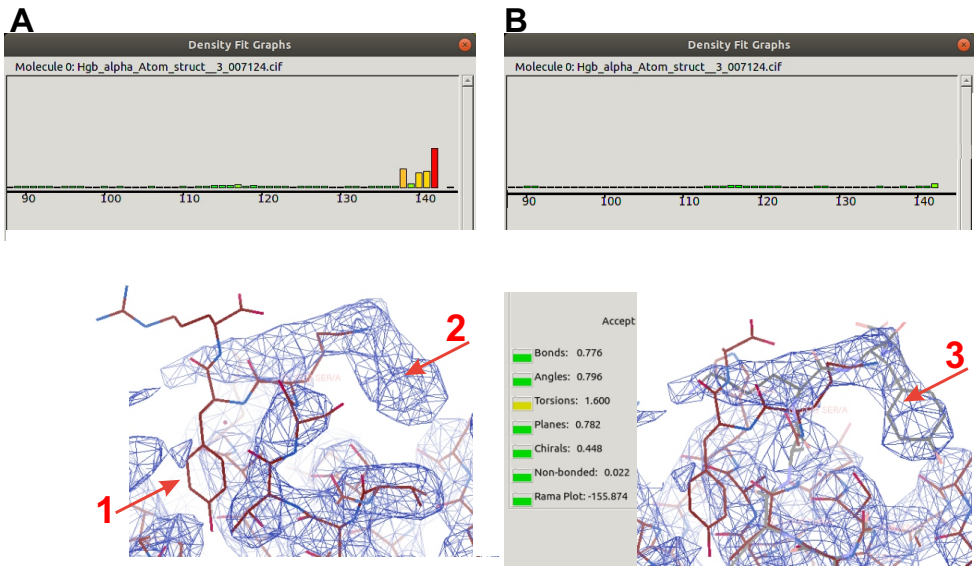


Figure 42: *Coot* fit in the map density of residue `TYR` 141.

Rotamer refinement is another refinement tool available in *Coot*. You can try to improve your current *model* modifying rotamers reported as incorrect in `Validate -> Rotamer analysis`. Otherwise, the next refinement program in modeling workflow (*PHENIX* `real space refine`) will perform rotamer refinement.

At the end of this interactive refinement with *Coot*, the refined atomic structure has to be saved in *Scipion*. You can save the atomic structure with its default name/label by pressing w. If you want to add a special label to identify the atomic structure in the *Scipion* workflow you can save that label in *Coot* main menu `Calculate -> Scripting -> Python` and the `Coot Python Scripting` window will be opened and you can write there your label name, for example `label1_HBA_HUMAN`. This label will appear in the `Summary` window of the *Scipion* framework (Fig. 43 (A)). Assuming that `0` is your *model* number, write in Command:

```
scipion_write (0, 'label1_HBA_HUMAN')↵
```

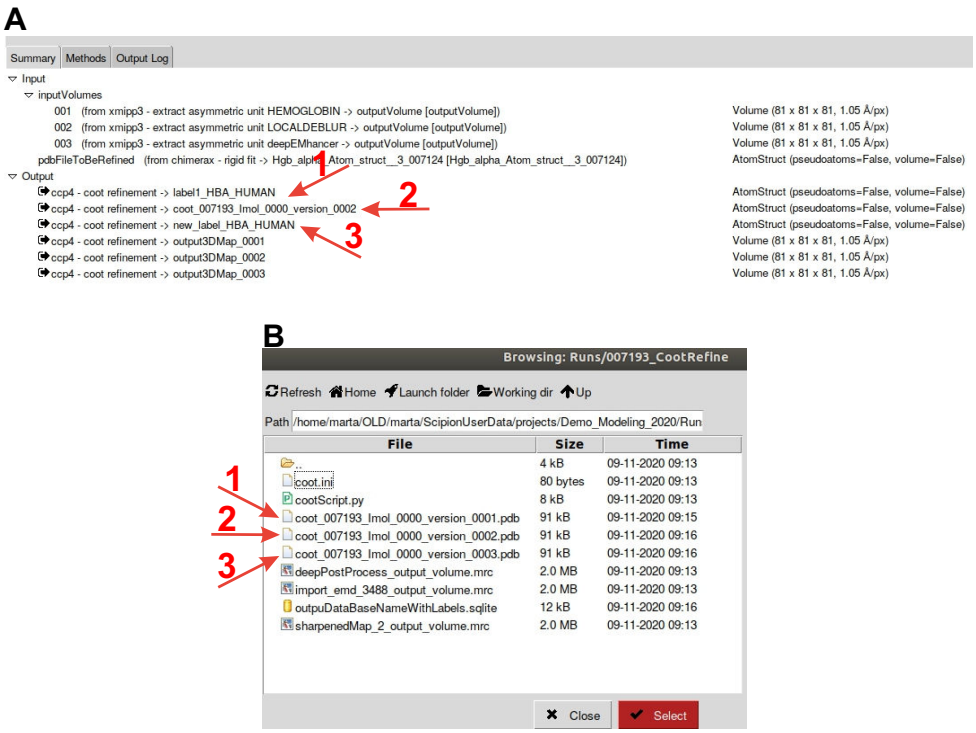


Figure 43: A. *Coot* `Summary` showing label names of each independent saved atomic structure (1, 3: user's chosen labels; 2: default label). B. (1, 2, 3) Respective atomic structure file names in the `extra` folder.

In its interactive way, ccp4 - coot refinement protocol can be launched again whenever you want in *Scipion*, and the last atomic structure saved will be loaded in *Coot* graphics window. This functionality of *Scipion* allows to stop the interactive refinement and continue the process in the last refinement step, maintaining each one of the intermediate refined structures saved in order in the *Scipion* tutorial folder `/Runs/000XXX_CootRefine/extra` (Fig. 43 (B)). Remark that if you want to continue with the refinement process you have to select the `Run mode` option `Continue` when you edit the *Coot* refinement protocol. In this way, to go again to intermediate refined structures is also possible. Finally, when you reach the final refined structure, save it, and you may press e to fully stop the *Coot* protocol.

A similar refinement process to that followed in *Coot* for `metHgb` α subunit chain

`A`, has to be carried out for the `metHgb` β subunit.

`Note about chain IDs`: Check the `id` of each chain. Although you have the possibility of changing this `id` in *ChimeraX*, as we have seen in the subsection "Structural models of human metHgb subunits from templates" (`metHgb` β subunit), you also have the possibility of performing this task in *Coot*, as it is shown in the next example in which we change the chain `id` from `A` to `B`. To change the name of the chain, go to the *Coot* main menu and select the option `Edit` (Fig. 44 (A)(1)) and then `Change chain IDs` and select the current name of the chain `A` (Fig. 44 (B)(2)) by the new one, `B` (3).

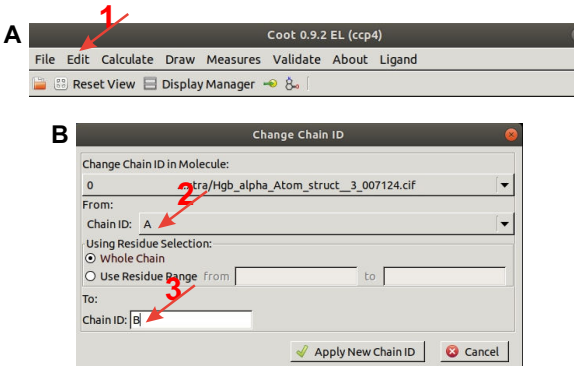


Figure 44: A. *Coot* main menu. B. *Coot* window to change chain IDs.

PHENIX Real Space Refine

In order to compare the previous *Coot* interactive refinement with an automatic refinement, we are going to use the phenix - real space refine protocol in parallel, as indicated in Fig. 37 (1). In addition, we can assess if the automatic refinement obtained with the protocol phenix - real space refine is able to complement and improve the result of the *Coot* manual refinement (Fig. 37 (2)). Protocol phenix - real space refine implements in *Scipion* the *phenix.real_space_refine* program developed to address cryo-EM structure-refinement requirements. Following a workflow similar to the *PHENIX* reciprocal-space refinement program *phenix.refine*, basically devoted to crystallography, *phenix.real_space_refine* program, mainly used in cryo-EM, is able to refine in

real space atomic models against maps, which are the experimental data.

Start working by opening phenix - real space refine protocol (Fig. 45 (1)), load as input volume the map asymmetric unit saved in *Coot* that you consider the most optimized one (2, the *deepEMhancer* sharpened map in this case), write the volume resolution (3), and load the atomic structure (*model* `Hgb_alpha_atom_struct__3_-007124` in the case 1 of Fig. 37 or *model* `new_label_HBA_HUMAN` in the case 2 (4)). After executing the protocol (6), results can be checked (7). Try to compare the `MolProbity statistics` that you can see in the `Summary` of the *Scipion* framework after changing the `Advanced` parameter `Local grid search` (5) from `Yes` to `No` (default value).

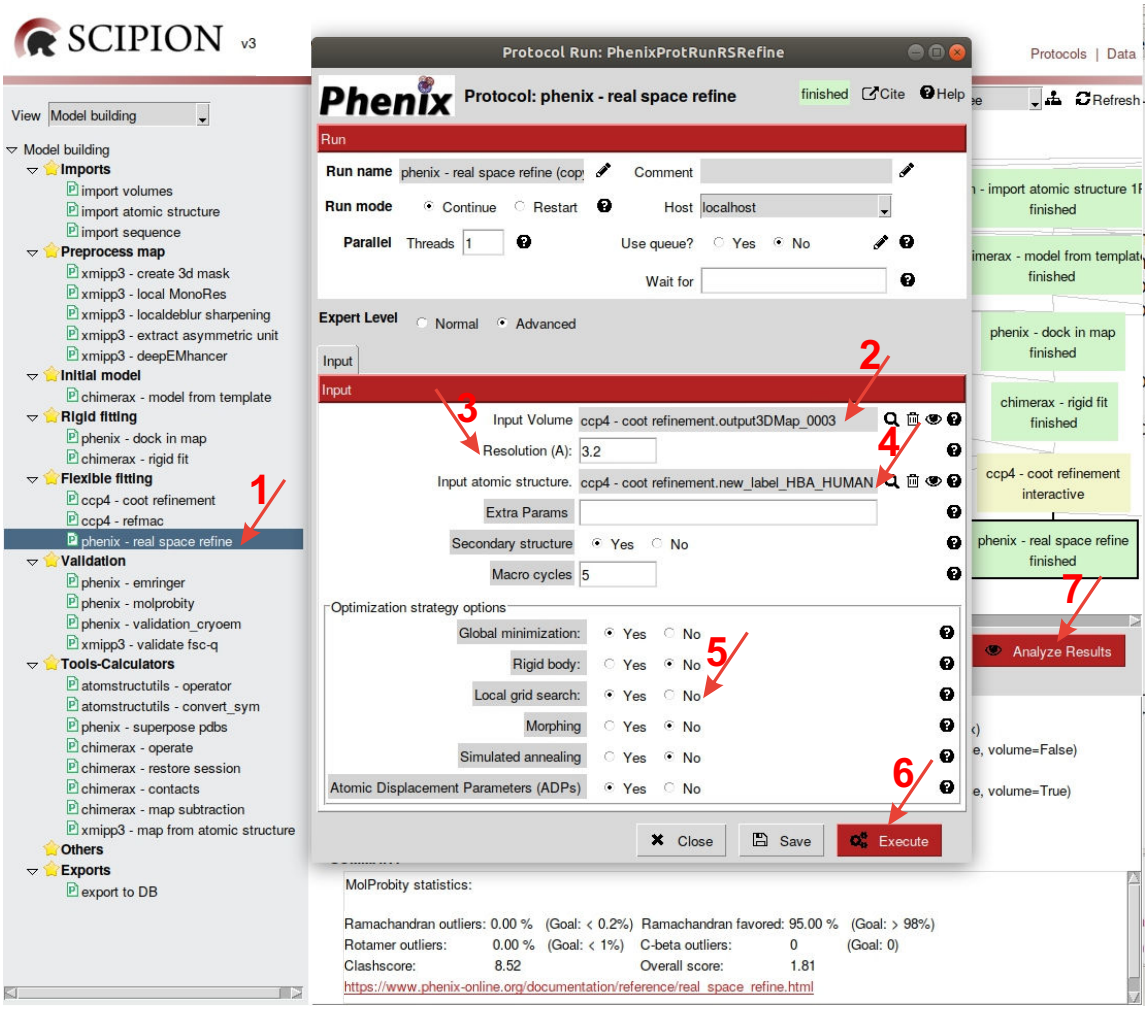


Figure 45: Completing *PHENIX* Real Space Refine protocol (Case 2 of Fig. 37).

The first tab of results shows the initial *model* atomic structure (Fig. 46 (pink)) as well as the refined one (green), both fitted to the normalized map asymmetric unit saved in *Coot*.

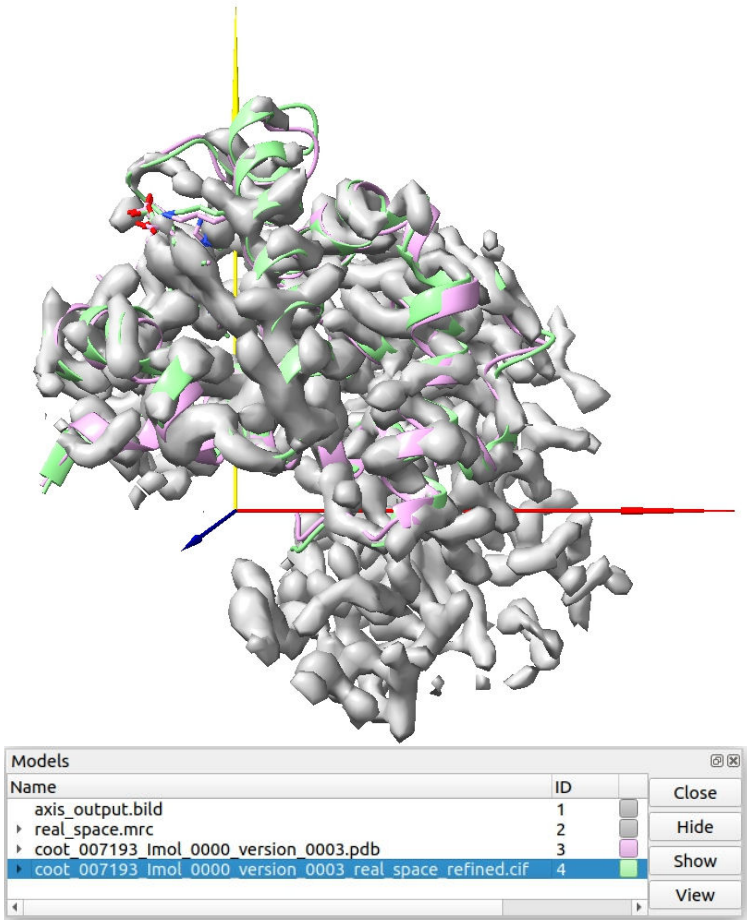


Figure 46: *ChimeraX* visualization of refined *model* of `metHgb` α subunit by *PHENIX* Real Space Refine protocol (Case 2 of Fig. 37).

The rest of tabs detail different statistics useful to compare the quality of distinct *models* such as *MolProbity* statistics and `Real-space` correlations. *MolProbity* results will be discussed in the next section of validation and comparison. Regarding `Real-space` correlations, different *models* can be compared by using the global number of CC_{MASK}, which indicates the correlation *model*-to-*map* calculated considering the map region masked around the *model*. You can check also individual correlation values for each residue. Remark that residues with lower correlation values might be susceptible to improve by additional refinement in *Coot*. Have a look to those corre-

lation values in the case 1 of Fig. 37 and answer the following questions: (Answers in appendix 1; **Question 9_1**)

- What is the CC_{MASK} value?
- Which one is the residue that shows the lower correlation value? Why?
- What is that correlation value?
- Which one is the second residue that shows the lower correlation value? Why?
- What is that correlation value?
- What is the correlation value of `HEME` group?

Now, compare these results with those obtained in the case 2 of Fig. 37, in which we have run *PHENIX* `real space refine` after *Coot*. Have the above values of correlation changed? (Answer in appendix 1; **Question 9_2**)

The conclusion of this part of refinement in real space is that *Coot* and *PHENIX* `real space refine` might perform complementary tasks. The usage of both protocols may improve the result, especially when partial processing or big rearrangements of molecules are involved.

Before finishing our refinement workflow with *Refmac*, we can ask ourselves how can we improve correlations in real space by modifying the `Advanced` parameters in the protocol form. Will the correlation values change if we set to "yes" optimization parameters previously set to "no", and increase the number of macro cycles from 5 to 30? Take into account that this process takes much more time (around 6 times more) than the previous one. (Answer in appendix 1; **Question 9_3**)

`Note`: An interesting application of the *PHENIX* `real space refine` visualiza-

tion tools is the possibility of load *Coot* from the *PHENIX* viewer and correct the structure of outliers residues and classhes. A recurively use of *PHENIX* `real space refine` and *Coot* protocols is thus possible.

CCP4 Refmac

As in the case of *Coot*, *Refmac* (from maximum-likelihood Refinement of Macromolecules) was initially developed to optimize models obtained by X-ray crystallography methods but, unlike *Coot*, automatically and in reciprocal space. The *models* refined in the real space with *Coot* and *PHENIX* `real space refine`, successively, will be used as inputs to perform a second refinement step in the Fourier space with *Refmac* protocol ccp4 - refmac. Firstly, open the *Refmac* protocol form (Fig. 47 (1)), load the volume generated by *Coot* (2), the atomic structure obtained with *Coot* (case 3 of Fig. 37) (3) or with *PHENIX* `real space refine` after *Coot* (case 4 of Fig. 37), and the volume resolution as maximum resolution (4). Execute the protocol (5) and when it finishes, analyze the results (6).

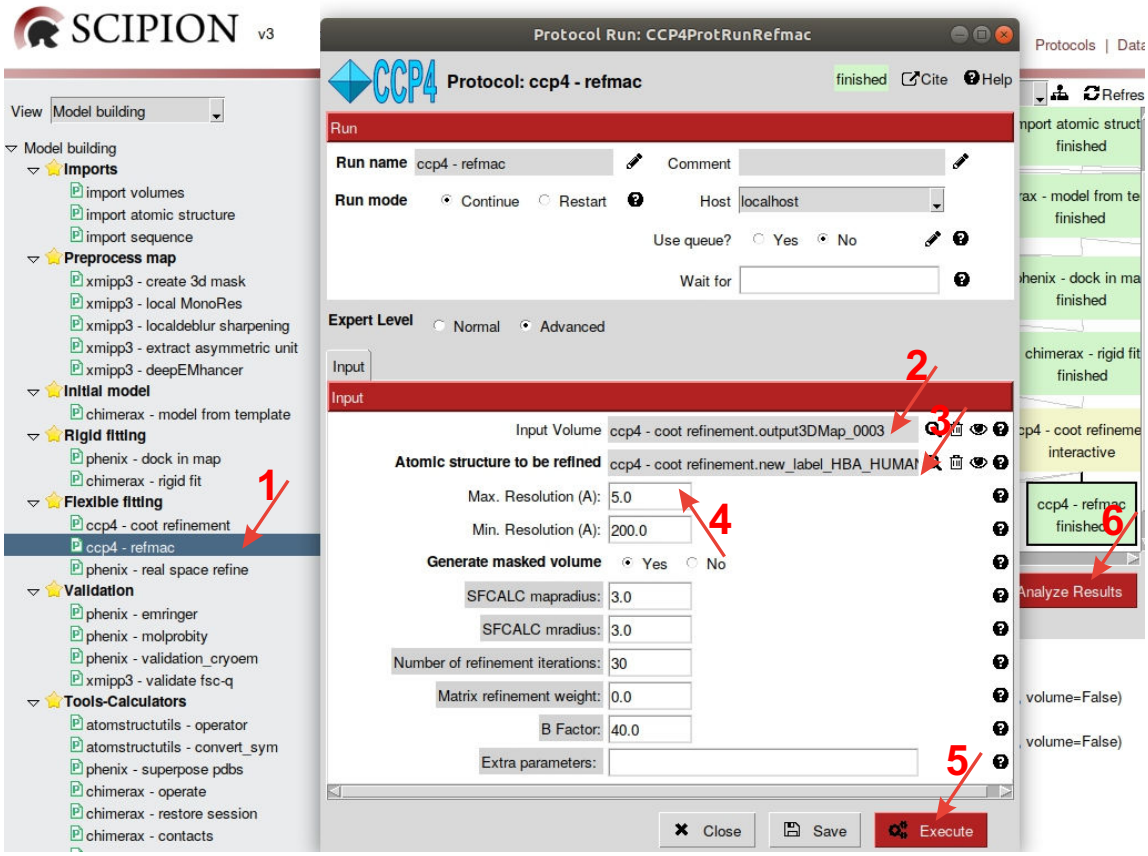


Figure 47: Filling in *Refmac* protocol (Case 3 of Fig. 37).

Clicking the first item in the display menu of results (Fig. 48 (1)), *ChimeraX* graphics window will be opened showing the input volume, the initial *model* (`new_label_HBA_HUMAN` obtained with *Coot* (Fig. 49, pink), and the final *Refmac* refined *model* (Fig. 49, green). By clicking the third item in the display menu of results (Fig. 48 (2)), a summary of *Refmac* results are shown. Check if values of `R factor` and `Rms BondLength` have improved with this refinement process in these three cases:

- Running *Refmac* after *Coot*:
  Can you see an improvement running *Refmac* immediately after *Coot*, thus ignoring *model* improvements generated by *PHENIX* `real space refine`? (Answers in appendix 1; **Question 9_4**)

- Running *Refmac* after *PHENIX* `real space refine` after *Coot*:
  Why the improvement seems to be very small? (Answers in appendix 1; **Question 9_5**)

- Running *Refmac* after *PHENIX* `real space refine` without a mask:
  Compare previous *Refmac* results (after *Coot* and *PHENIX* `real space refine`) with those obtained selecting the option `No` in the protocol form parameter `Generate masked volume`. Use two different volumes, the one generated by *Coot* protocol, and the one generated by the `extract asymmetric unit` protocol. Are there any differences? Why? (Answers in appendix 1; **Question 9_6**)

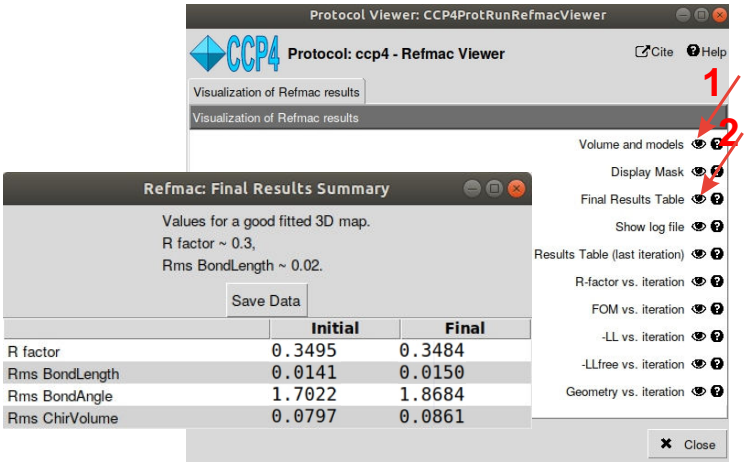


Figure 48: Display menu of *Refmac* results.

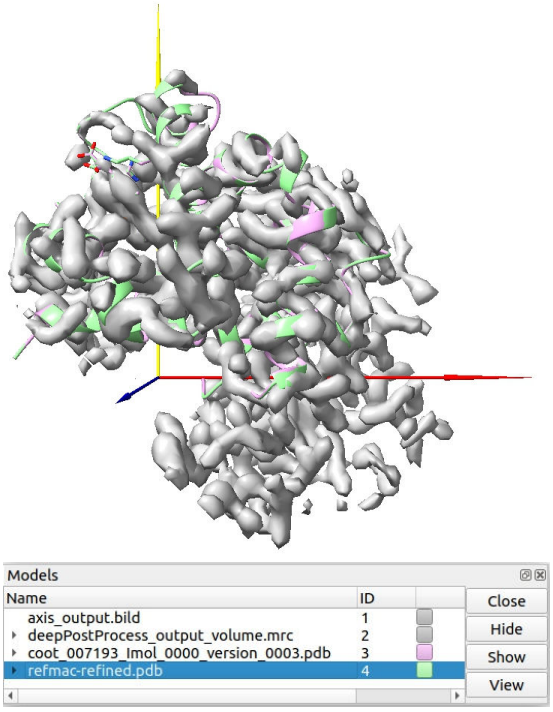


Figure 49: *ChimeraX* visualization of refined *model* of `metHgb` α subunit by *Refmac* (Case 3 of Fig. 37).

Have a look to the rest of items in the display window of results.

The best refinement workflow

At this point we wonder about the optimal steps to follow in the refinement process. Should we have to use *Coot* first, then *PHENIX*, then *Refmac*?, or maybe, with a different *map* and *model*, should we start with the automatic refinement and then go to the manual one? The right answer is that there is no a unique answer. The strategies and the number of steps of refinement might differ and the only requirement is that the next step in refinement should generate a better structure than the previous one. This premise requires to apply common validation criteria to assess the progressive improvement of our *model*.

10 Structure validation and comparison

At the end of the refinement process of `metHgb` α subunit (a similar one would be required for β subunit), we need to assess the geometry of our *model* regarding the starting volume to detect *model* controversial elements or *model* parameters that disagree with the map. Although each refinement program has their own tools to assess the progress of refinement (*Coot* `Validate` menu; *PHENIX* `real space refine` real space correlations; *Refmac* `R factor` and `Rms BondLength`), in this tutorial section, three assessment tools will be described to obtain comparative validation values after using any protocol in the workflow: Protocols *EMRinger* (phenix - emringer, Appendix 20, (Barad et al., 2015)), *MolProbity* (phenix - molprobity, Appendix 21, (Davis et al., 2004)), and *Validation CryoEM* (phenix - validation_cryoem, Appendix 22, (Afonine et al., 2018a)). *Validation CryoEM* protocol will show *MolProbity* validation values as well as correlation coefficients in real space. Old versions of *PHENIX* (v. 1.13) do not include this tool. Correlation values in real space will thus be computed if a map is provided in *MolProbity* protocol. Additionally, we are going to introduce the protocol phenix - superpose pdbs (Appendix 24, (Zwart et al., 2017)) useful to compare visually the geometry of two atomic structures.

Observe the first validation steps in the modeling *Scipion* workflow in Fig. 50 starting from output *models* generated by *PHENIX* `real space refine` and *Refmac*.

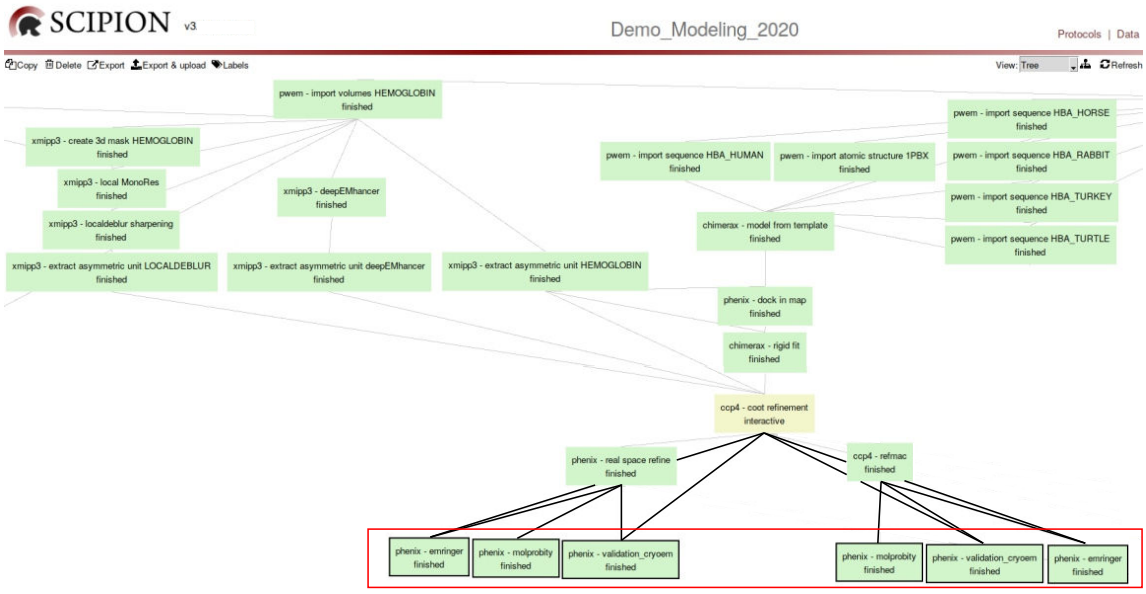


Figure 50: *Scipion* framework detailing the workflow to validate the model of the human `Hgb` α subunit.

`Note`: Structure validation is a model building step that you have to perform recursively during the refinement process to assess if you are improving your structure or not. Once you finish the refinement process you'll obtain the final assessment values. These values should be in a certain range if you want to submit the atomic structure to databases. These final validation scores should be computed regarding the density map that you submit as main map, although during the recursive process you might have used the sharpened maps for refinement/validation.

EMRinger

Specifically designed for cryo-EM data, *EMRinger* tool assesses the appropriate fitting of a model to a map, validating high-resolution features such as side chain arrangements. The placement of side chains regarding the molecule skeleton depends on the χ_1 dihedral angle (a dihedral angle is the angle between two intersecting planes), which is determined by atomic positions of (`N`, `C`α, `C`β) and (`C`α, `C`β, `C`γ) (see Fig. 51). The side chain dihedral angles tend to cluster near 180° and ±60°. The

lower deviations regarding these values, the better *model*, and the higher *EMRinger* value.

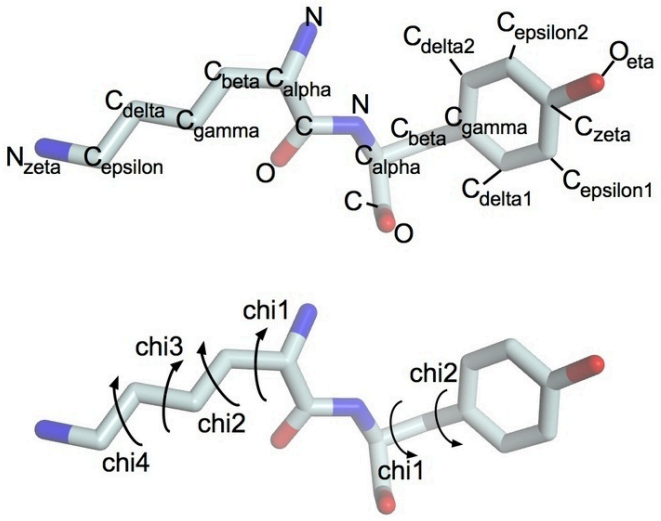


Figure 51: Naming convention in side chains explained in a lysine-tyrosine strand. Note that these two residues are within a protein and thus have no terminal region.

We can start assessing with *EMRinger* the `metHgb` α subunit *models* that we have generated along the modeling workflow. In each case, open the phenix - emringer protocol ((Fig. 52 (1)), load the extracted map asymmetric unit (initial or saved with *Coot*) (2) and the atomic structure that you'd like to validate in relation to the map (3), execute the program (4) and analyze results (5). A menu to check results in detail will be opened (bar `EMRinger results`). *Phenix EMRinger* plots with density thresholds, with rolling window for each chain, as well as dihedral angles for each residue are shown here. The most relevant results, especially the *EMRinger* score, will also be written in the protocol `SUMMARY` (6).

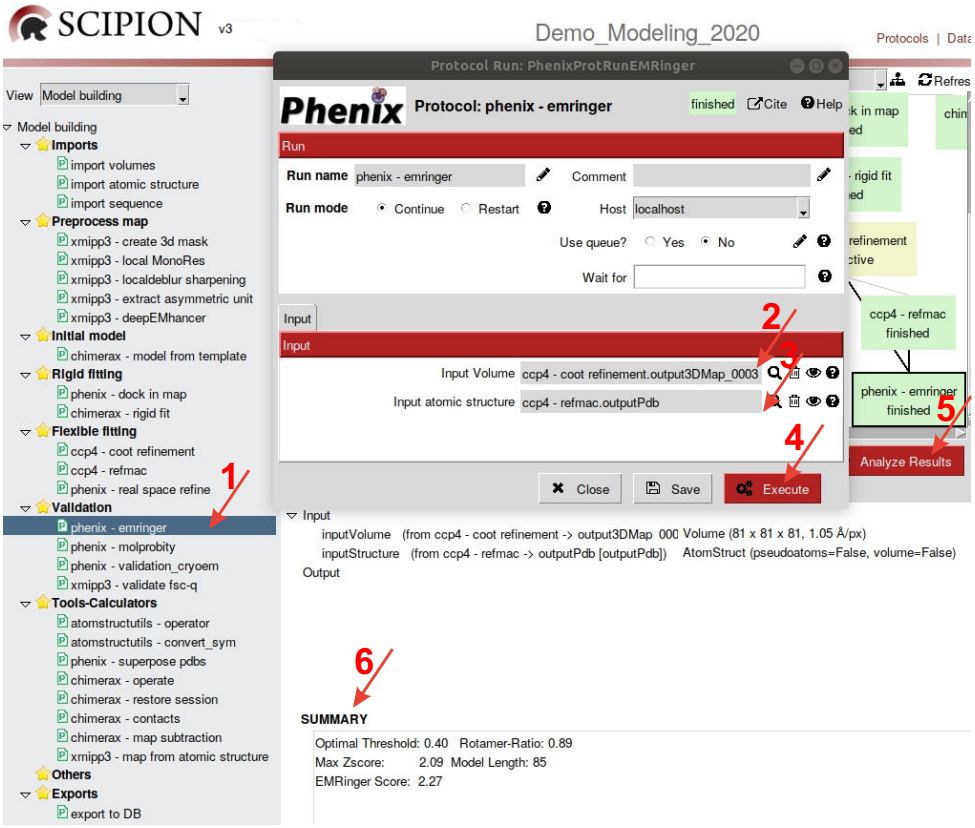


Figure 52: Completing *EMRinger* protocol form.

Run *EMRinger* protocol and determine the respective score after running *ChimeraX* `rigid fit`, *Coot* refinement, *PHENIX* `real space refine` (form parameters indicated in Fig. 45) after *Coot*, and *Refmac* refinement with MASK before and after *PHENIX* `real space refine`. Considering *EMRinger* `score`, does our `metHgb` α subunit *models* seem to be OK or, at least, did they improve? (Answers in appendix 1; **Question 10_1**). Try the same validation with β subunit *models*.

MolProbity

The atomic structure validation web service *MolProbity*, with better reference data has been implemented in the open-source CCTBX portion of *PHENIX* (Williams

et al., 2018). This widely used tool assesses *model* geometry and quality at both global and local levels. Originally designed to evaluate structures coming from X-Ray diffraction and NMR, it does not take into account the quality of the fitting with a 3D density map. The implementation of *MolProbity* in *PHENIX* v. 1.13, nevertheless, includes the possibility of adding a volume and assessing the correlation in the real space.

The assessment process that we have carried out with *EMRinger* can also be done with *MolProbity* in *Scipion*. We are going to validate the geometry of `metHgb` α subunit *models* that we have generated along the modeling workflow. In each case, open the phenix - molprobity protocol (Fig. 53 (1)), load the extracted unit cell volume (initial or generated by *Coot*) (2) with its resolution (3) only if your *PHENIX* version is 1.13 and you want to have real space correlation between map and *model*. For *PHENIX* versions higher than 1.13 simply load the *model* atomic structure (4) and execute the protocol (5). With `Analyze results` (6) menu bars are shown. *MolProbity* results bar include validation statistics. Protocol `SUMMARY` emphasizes the most relevant ones (7).

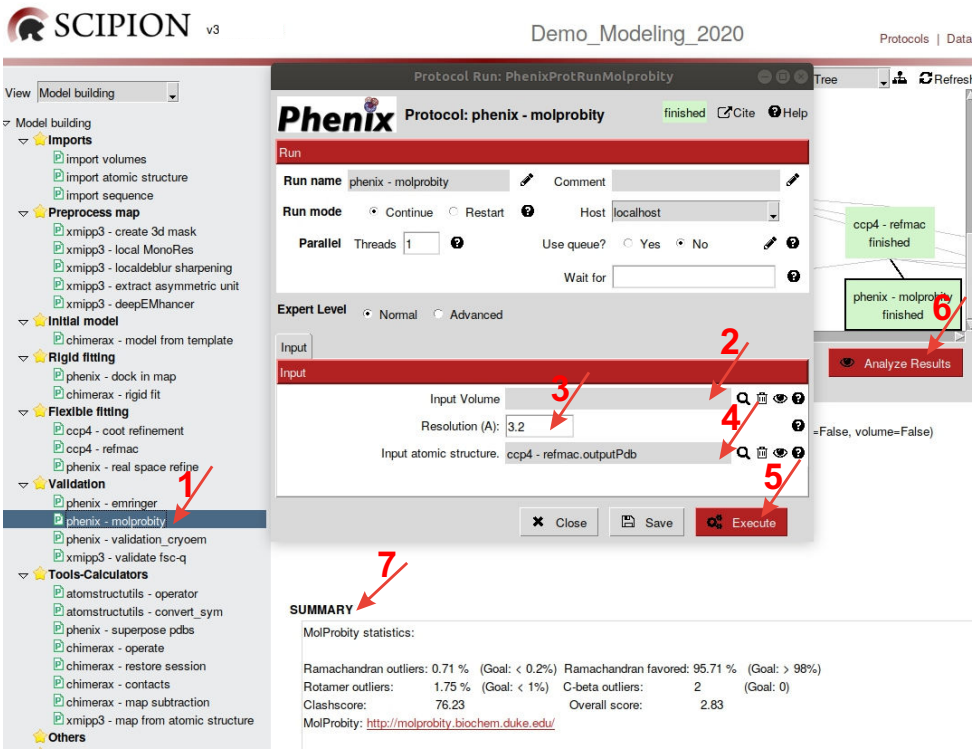


Figure 53: Completing *MolProbity* protocol form.

Run *MolProbity* protocol to obtain its statistics after running *ChimeraX* `rigid fit`, *Coot* refinement, *PHENIX* `real space refine` (form parameters indicated in Fig. 45) after *Coot*, and *Refmac* refinement with MASK before and after *PHENIX* `real space refine`.

Validation CryoEM

PHENIX versions higher than 1.13 combine multiple tools for validating cryo-EM maps and models into the single tool called *Validation CryoEM* ((Afonine et al., 2018a)). This tool has been implemented in *PHENIX* versions higher than 1.13.

To carry out the global validation of maps and models obtained from cryo-EM data, open the protocol phenix - validation_cryoem in *Scipion* (Fig. 54 (1)), load the

map (initial or generated by *Coot*) (2) with its resolution (3), load the *model* atomic structure (4) and execute the protocol (5). `Analyze results` (6) shows the same menu bars available in results section of *PHENIX* `real space refine` protocol. *MolProbity* results bar include validation statistics. Protocol `SUMMARY` (7) emphasizes the most relevant ones.

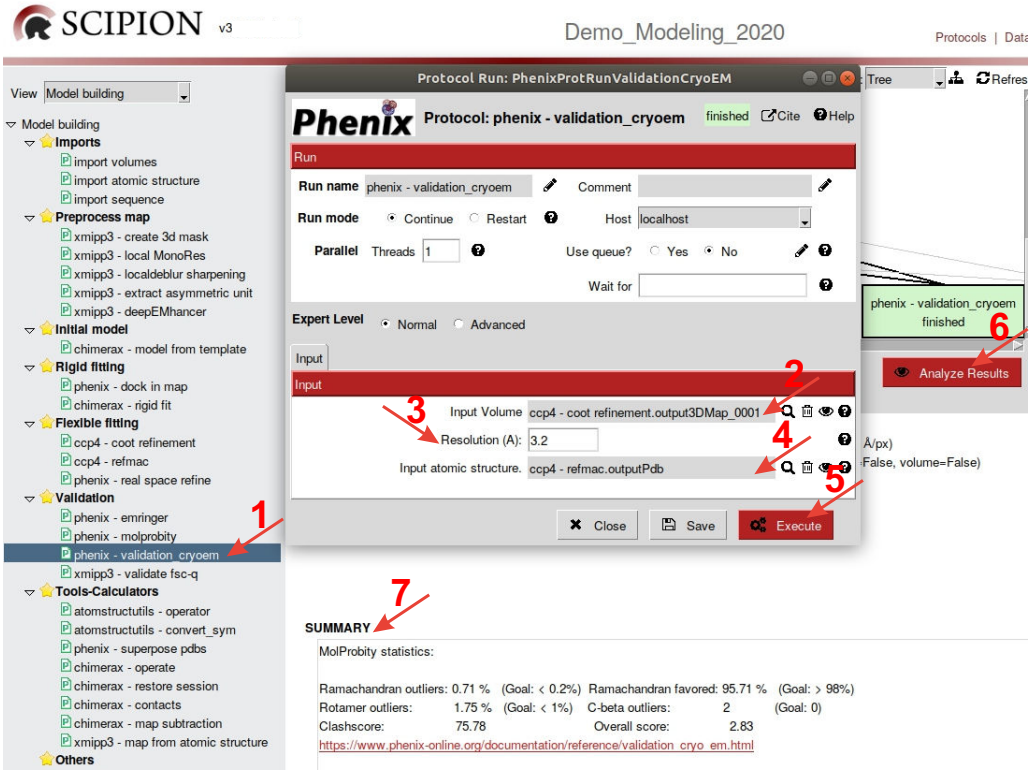


Figure 54: Filling in *PHENIX Validation CryoEM* protocol form.

In order to compare validation results of *models* obtained along the modeling workflow, fill in the next table (Table 2) including, in addition to *MolProbity* statistics, *EMRinger* scores and CC_{MASK} values obtained before. (Answers in appendix 1; **Question 10_2**). The same table (Table 2) can be completed for `metHgb` β subunit (Appendix 1; **Question 10_3**)

Table 2: Validation statistics of human `metHgb` α subunit *model*. `RSRAC` stands for `Real Space Refine` after *Coot*. `Rama` stands for `Ramachandran`.

Statistic	*ChimeraX*	*Coot*	*PHENIX* `RSRAC`	*Refmac* after *Coot*	*Refmac* after `RSRAC`	`5NI1`
CC_{MASK}						
EMRinger `score`						
`RMS` (Bonds)						
`RMS` (Angles)						
`Rama favored` (%)						
`Rama allowed` (%)						
`Rama outliers` (%)						
`Rotamer outliers` (%)						
`Clashscore`						
`Overall score`						
Cβ `deviations`						
`RMSD`						

Results compiled in this table indicate that statistics are uncorrelated. From the point of view of correlation in real space, the best *model* was obtained from *PHENIX* `real space refine` after *Coot*. Considering *EMRinger* `score`, the best *model* derives from the whole workflow *Coot* `->` *PHENIX* `real space refine`. With *MolProbity* `Overall score` as validation rule, the last step in the workflow could be suppressed because the best value was obtained after *Coot* `->` *PHENIX* `real space refine` (last modification of parameters). We'd like to select the best *model* and continue refining it in order to improve it as much as possible. Assuming that no one *model* is perfect, how can we select the best one?

Model Comparison

The question posed in the previous item does not have an easy answer in the real world, in which we do not know the final atomic structure. In this tutorial, nevertheless, we know the atomic structure already published for this cryo-EM map and

we may wonder how far we are from it. The question can be answered by comparing a) validation statistics that we have obtained for our *models* with the statistics computed for the available α subunit in `PDB` structure `5NI1`, and b) the atomic structures themselves by overlapping.

Comparison of validation statistics

Validation statistics of `metHgb` α subunit of `PDB` structure `5NI1` should be obtained as first step to compare them with validation statistics of our *models*. With this aim we are going to follow the workflow remarked in the Fig. 55:

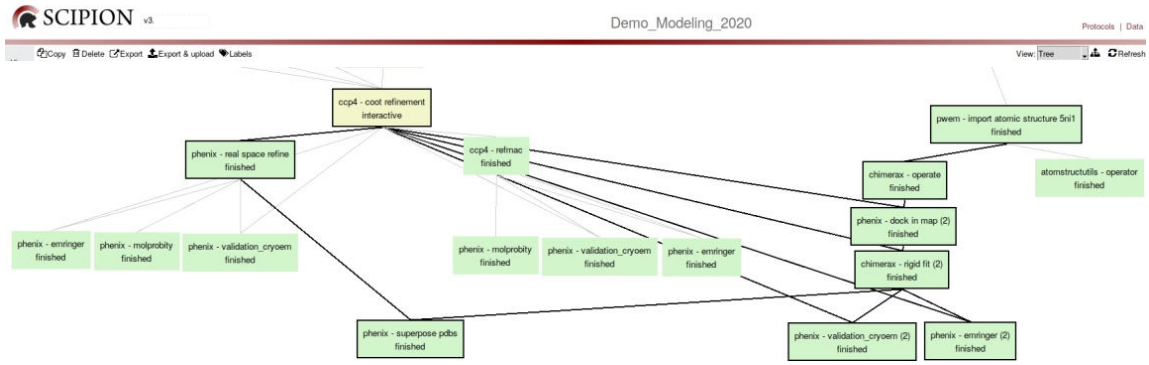


Figure 55: *Scipion* framework detailing the last part of the validation workflow.

- Protocol import atomic structure:
  Download from `PDB` structure `5NI1`

- Protocol chimerax - operate (Appendix 5):
  Similar to *ChimeraX* `rigid fit`, *ChimeraX* `operate` protocol allows to perform operations with atomic structures. We are going to use this protocol to save independently in *Scipion* the `metHgb` α subunit. Open the protocol (Fig. 56 (1)), complete the parameter `PDBx/mmCIF` including the atomic structure `5NI1` previously imported (2), and execute the protocol (3).

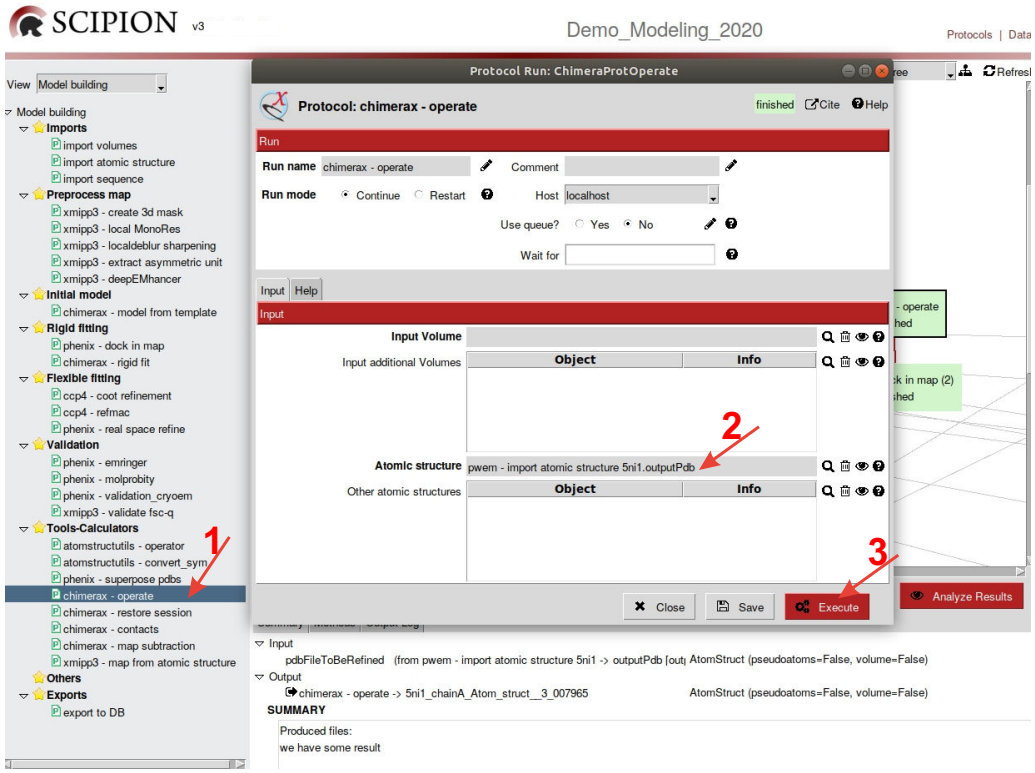


Figure 56: Filling in *ChimeraX* `operate` protocol form.

The *ChimeraX* graphics window will be opened with the structure `5NI1` as model number `#2`. To save independently the structure of human `metHgb` α subunit (chain A), write in *ChimeraX* command line:

```
select #2/A
save /tmp/5ni1_chainA.cif format mmcif models #2 selectedOnly true
open /tmp/5ni1_chainA.cif
scipionwrite #3 prefix 5ni1_chainA_
```

Remark that the model saved in *ChimeraX* command line includes both the aminoacid chain and the `HEME` group. In case you are interested in extracting only the aminoacid chain, you can use the protocol atomstructutils - operator,

specifically designed to extract/add individual chains from/to an atomic structure (Atomic Structure Chain Operator; Appendix 2). Compare the results of protocols *ChimeraX* `operate` and Atomic Structure Chain Operator in Fig. 57. The red arrow points at `HEME` group.

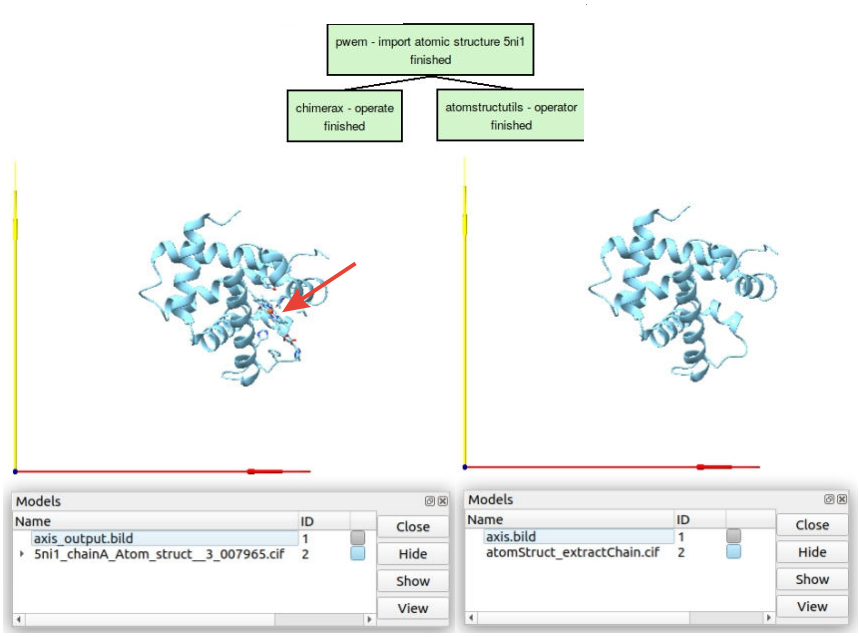


Figure 57: Comparison of results obtained with the protocols *ChimeraX* `operate` (left) and Atomic Structure Chain Operator (right).

- Protocol phenix - dock in map:

  Open *PHENIX* `dock in map` protocol and follow the instructions above indicated. The structure saved in *ChimeraX* `operate` will replace this time our previous *model*. Results can be observed in Fig. 58.

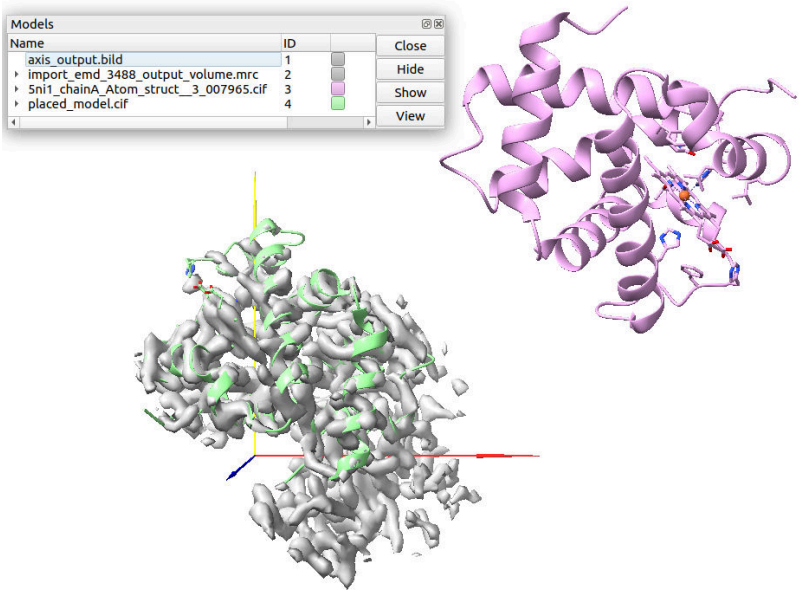


Figure 58: Results view of phenix - dock in map protocol.

- Protocol chimerax - rigid fit: Open again *ChimeraX* `rigid fit` protocol and, following the already indicated instructions, include this time the atomic structure `placed_model.cif` generated in the previous step. To fit the `metHgb` α subunit from 5NI1 structure in the extracted asymmetric unit and save the fitting write in *ChimeraX* command line:

  ```
  fitmap #3 inMap #2
  scipionwrite #3 prefix 5ni1_chainA_fitted_
  ```

- Validation protocols phenix - emringer and phenix - validation_cryoem:

  Compute validation statistics with these two protocols for `metHgb` α subunit from `PDB` structure `5NI1`, write respective values in the previous table (Table 2), and compare them with the statistics of our *models*.

  Considering results shown in appendix 1 (**Question 10_2**) for `metHgb` α subunit, we can conclude that published structures are not perfect and we are

not very far from this published one. In fact, we have overcome every statistic except CC_{MASK}. Nevertheless, the different *models* generated after *Coot* refinement can still be improved by iterative refinement processes. Validation statistics thus allow to follow the quality improvement of atomic models.

Comparison of atomic structures

PHENIX protocol phenix - superpose pdbs allows to compare two atomic structures by overlapping them. Root mean square deviation (RMSD) between the fixed structure (the published one) and one of our *models* supports the classification of *models* according to its proximity to the published model. Open *PHENIX* `superpose pdbs` protocol form (Fig. 59 (1)), include the published structure of the `metHgb` α subunit as fixed structure (2), each one of the *models* generated along the worflow (3), execute the protocol (4) and check results by pressing `Analyze results` (5). Arrows of Fig. 60 remark differing parts between the atomic structure of the `metHgb` α subunit from `PDB` structure `5NI1` (green) and our *model* generated by automatic refinement with *PHENIX* real space refine protocol (pink). By opening these structures in *Coot* you can see the differences between them. Finally, complete the Table 2 with the value of `RMSD (final)` (6) obtained for each *model*. (Answers in appendix 1; **Question 10_2**).

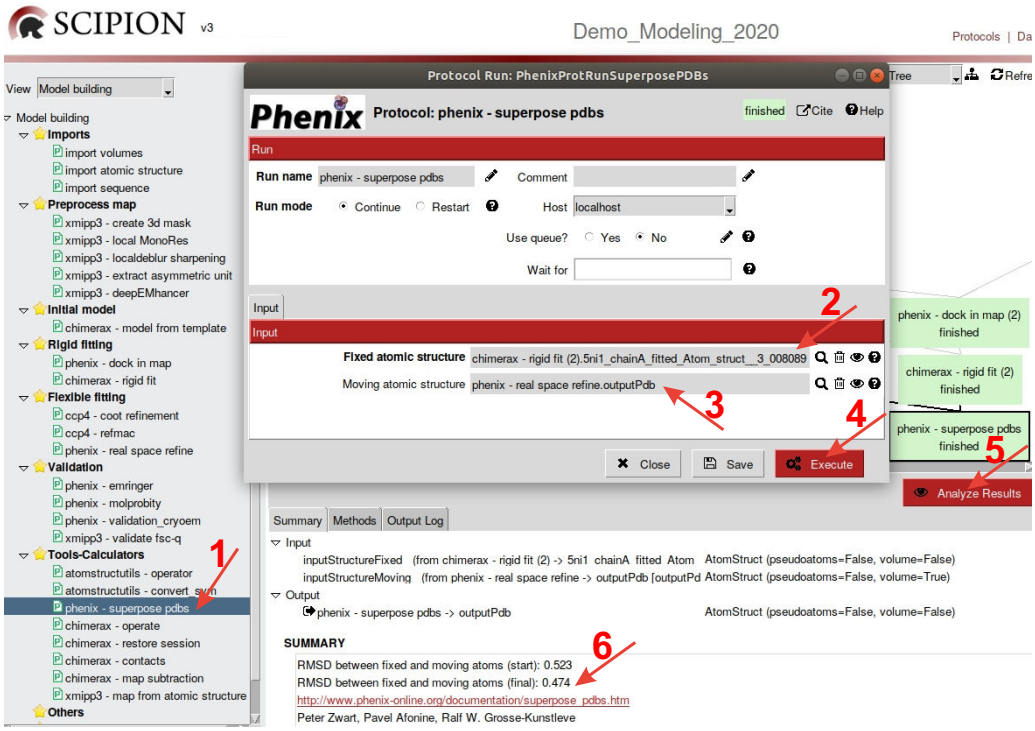


Figure 59: Completing *PHENIX* `superpose pdbs` protocol form.

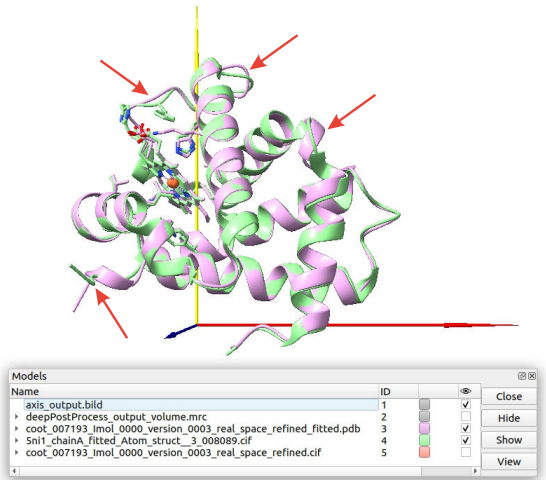


Figure 60: *Model* generated for `metHgb` α subunit superposed to the published α chain of `5NI1` structure.

A *model* for `metHgb` α subunit has to be selected at the end of the validation process. According to the statistics of Table 6 (Appendix 1; **Question 10_2**), select the *model* obtained in modeling workflow showing the smallest RMSD value, high value of *EMRinger* `score`, quite high value of CC_{MASK} and acceptable *MolProbity* statistics. Follow a similar process to validate and select the *model* generated for `metHgb` β subunit. Appendix 1 **Question 10_3** contains a statistics table for `metHgb` β subunit, similar to that obtained for `metHgb` α subunit.

In the real world the selected *models* usually are the starting point to improve specific validation parameters by additional refinement. Since the improvement of certain parameters normally implies worsening of other parameters, a final compromise solution has to be taken.

11 Building the asymmetric unit

Once we have selected the *models* for `metHgb` α and β subunits (see the workflow branches to have α and β subunits Fig. 61), we can regenerate the smallest asymmetrical element of the starting map. With this aim we are going to use protocols to operate with atomic structures (chimerax - operate or atomstructutils - operator), to refine them both manually (ccp4 - coot refinement) and automatically (phenix - real space refine) and to validate them (phenix - validation_cryoem and phenix - emringer). A brief schema of the main steps of this part of the workflow can be seen in Fig. 61. Take into account that in real live probably many more steps of refinement and validation will be required.

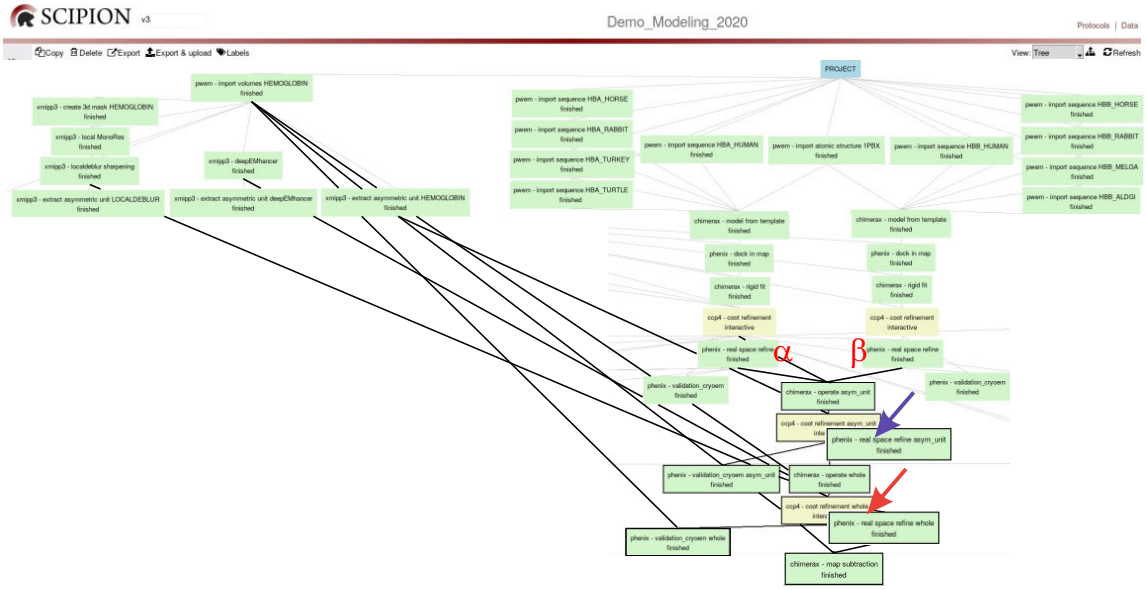


Figure 61: *Scipion* framework detailing the workflow to reconstruct the structure of the map asymmetric unit (blue arrow) and the whole atomic structure (red arrow).

- Protocol to join the `metHgb` α and β subunits in a unique atomic structure:
  Two protocols can be used in *Scipion* for this purpose (chimerax - operate or atomstructutils - operator) and the result should be identical.
  Before starting, nevertheless, be sure that you have two atomic structures and each one includes an only chain with a different `id`. Remember that chain `ids` may be changed for other chain `ids` in *ChimeraX* and *Coot*.
  Secondly, it could be very convenient to change the *Scipion* output label of each subunit, in order to follow them easily in *Scipion*. According to the Fig. 62 go to the Summary of the two final protocols that allow to generate those atomic structures and press the black arrow (C) to select the option `Edit`. Type the new output name of the structures (`HBA_refined` (D) and `HBB_refined` (E), respectively).

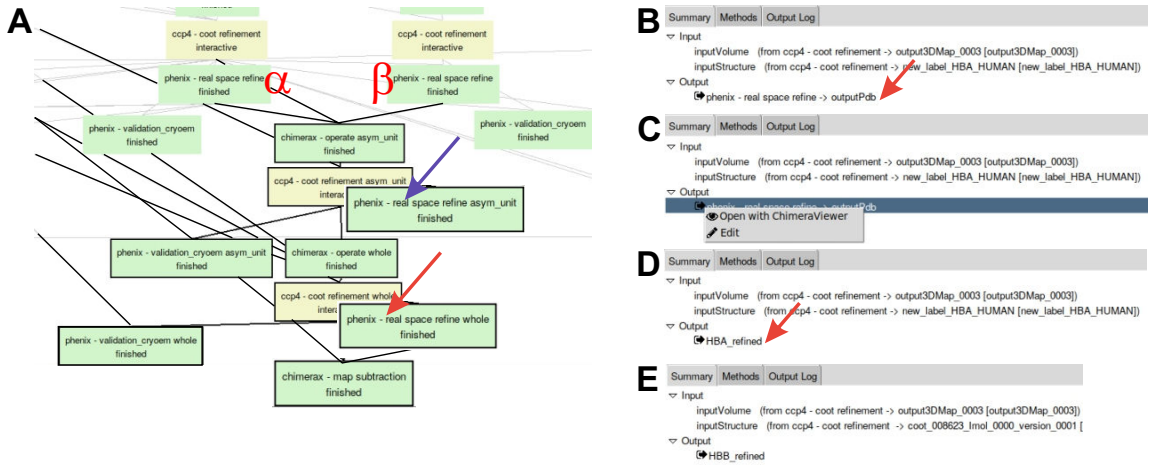


Figure 62: A. Zoom in on Fig. 61. B. Summary of the protocol box from *PHENIX* `real space refine` (α in (A)). Red arrow points at the *Scipion* output name. C. Menu opened pressing the output black arrow of the Summary. D. New name of the *Scipion* output in the Summary. E. Summary from the protocol box *PHENIX* `real space refine` (β) after applying the same edition process.

Then, open again *ChimeraX* `operate` protocol and following the already indicated instructions, include the *models* of `metHgb` α and β subunits in params `Atomic structure` and `Other atomic structures`, respectively (Fig. 63 (A)). Firstly, check that both `models` are perfectly fitted in the map asymmetric unit. Otherwise, apply the command `fit inMap`, as it was previously shown. Next, create a single atomic structure by joining models `#3` and `#4` in *ChimeraX* `Models` panel. To generate a combined `model` write in the command line:

```
scipioncombine #3,4
```

The new model `#5` is shown in *ChimeraX* `Models` panel (Fig. 63). Finally, save this fitted structure writing in *ChimeraX* command line:

```
scipionwrite #5 prefix asymmetric_unit_model_
```

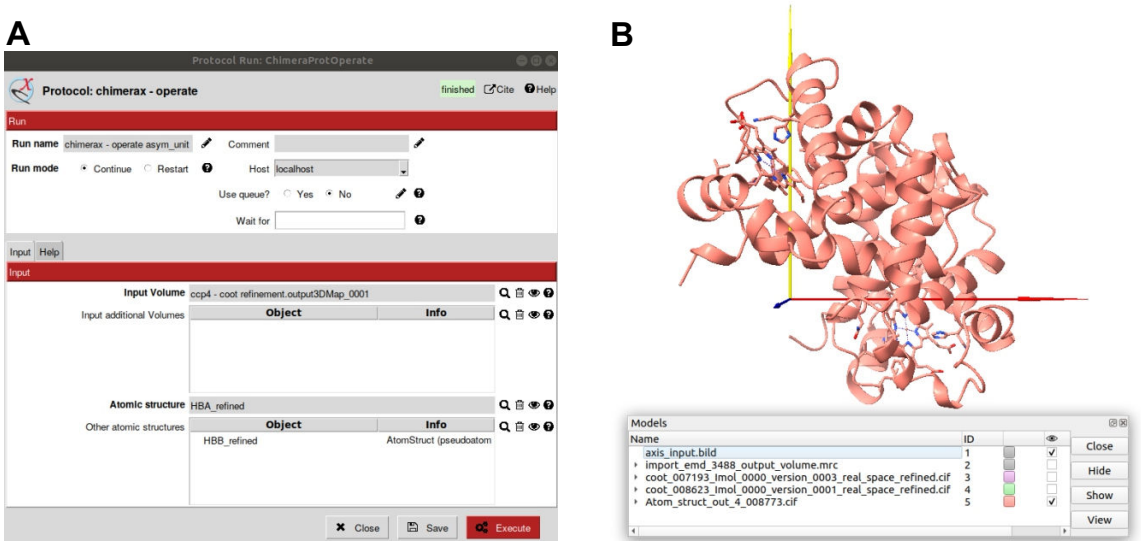


Figure 63: A. Completing the chimerax - operate protocol with the atomic structures `HBA_refined` and `HBB_refined` . B. *ChimeraX* graphics window showing the combined `model #5`.

- Protocols to refine the new combined structure generated:

  At this point refinements could cover specially the overlapping area between the two chains. Help yourself with the *Coot* tools of `Validate` in the main menu, as well as the visualization tools of *PHENIX* `real space refine` protocol.

- Validation protocols to select the best *model* of the human `metHgb` unit cell:

  Validate the new combined structure generated is recommendable before continuing with the next steps in the workflow.$EMRinger$ and $ValidationCryoEM(MolProbity)$ validation statistics should be computed for the new *model* of human `metHgb` asymmetric unit, generated by combining `metHgb` α and β subunits. Appendix 1 (**Question 11_1**) contains a statistics table for the unit cell *model* (Table

8). We can try to improve those statistics by additional refinement processes. By performing refinement in real space with *Phenix* some of the statistics could result improved. Table 8 contains also RMSD values computed in a similar way as we have seen for α and β subunits, considering as fixed structure chains A and B from `5NI1` atomic structure. To continue with the modeling process we can select the unit cell *model* generated by *Phenix* `real space refine` because most of its validation statistics show the best values (CC_{MASK}, *EMRinger* `score` and *MolProbity* values). Exceptionally, RMSD regarding the published structure yields the worst value.

12 The whole macromolecule

To regenerate the whole human `metHgb` macromolecule, we are going to follow basically the schema shown in Fig. 61. Starting from the symmetric unit, *ChimeraX* `operate` protocol allows to generate the whole molecule by symmetry. As in the previous step, validation programs drive to selection of the best *model* of the whole molecule after one or several rounds of assessment - refinement -assessment. A final validation step will be accomplished with *ChimeraX* `map subtraction` protocol to assess the volume density occupancy of the new macromolecule generated.

- Protocol chimerax - operate to generate the whole molecule of human `Hgb`:

  Following previous instructions, open *ChimeraX* `operate` protocol (Fig. 56 (1)), load the selected atomic structure *model* of `metHgb` asymmetric unit (2), and execute the protocol (3). *ChimeraX* graphics interface will show you the *model* of `metHgb` asymmetric unit. Considering the C2 symmetry of the whole molecule, write in *ChimeraX* command line to re-generate the whole molecule:

  ```
  sym #3 C2 copies true
  ```

  A symmetric image of the input *model* (Fig. 64; *model* `#3`) will be generated.

The new *model* `#4` contains both the input (Fig. 64, *model* `#4.1`)and the symmetric unit (*model* `#4.2`).

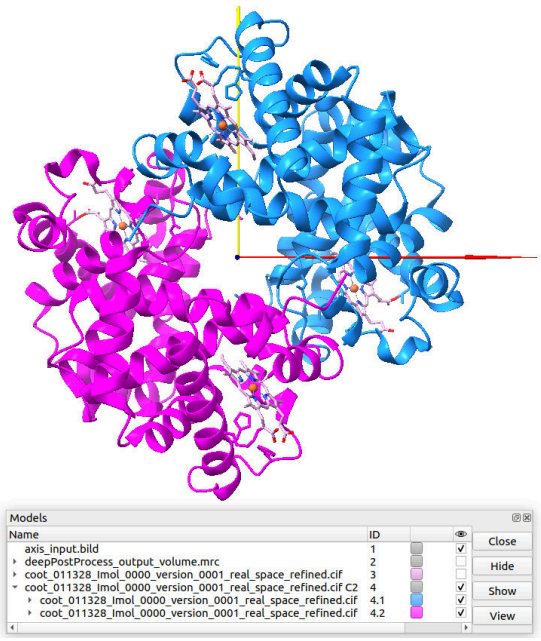


Figure 64: *Model* generated by symmetry for the whole human `metHgb`.

Although the whole structure can be saved by writing in *ChimeraX* command line `scipionwrite #4 prefix whole_model_`, in order to have only one *model* and not a group of two *models*, we will write in *ChimeraX* command line:

```
save /tmp/chains_C_D.cif format mmcif models #4.2
open /tmp/chains_C_D.cif
setattr #5/A c chain_id C
setattr #5/A r chain_id C
setattr #5/B c chain_id D
setattr #5/B r chain_id D
scipioncombine #3,5
scipionwrite #6 prefix whole_model_
```

Remark that we have changed the `ids` of symmetric chains `A` and `B` by `C` and `D`, respectively.

`Note`: In this small example selected for modeling it doesn't matter if we model the map asymmetric unit or the whole molecule. In real life, however, to model the whole molecule doesn't make sense because of its huge size. In that case, we will limit our modeling to the map asymmetric unit. The right modeling of this part of the molecule will require to add the adjacent asymmetric units in order to perform the appropriate modeling of the overlapping areas, avoiding steric classes in the reconstruction by symmetry of the whole molecule. In that case, the command lines would be:

- To generate the symmetry copies:
  `sym #3 C2 copies true`

- To remove in the new *model* `#4` the symmetry copies with centers within a certain range of distance `d` of the center of the molecule input *model*:
  `delete #4 & #3 #>d`

At this point we will continue with the refinement process of this asymmetric unit plus neighbors. The validation will focus only in the asymmetric unit, which will be recovered by removing the remaining adjacent asymmetric units. This cleaning or removing of the neighbor units can be performed with the protocol *ChimeraX* `operate` each time we would like to validate the structure.

- Protocols to refine the new combined structure generated:

As we said in the previous chapter regarding the building of the asymmetric unit, refinements should cover specially the overlapping areas, in this case between the two asymmetric units. Help yourself with the *Coot* tools of `Validate` in the main menu, as well as the visualization tools of *PHENIX* `real space refine` protocol.

- Validation protocols to select the best *model* of the whole human `Hgb`:

  EMRinger and $ValidationCryoEM(MolProbity)$ statistics have to be computed for the new *model* of the whole human `metHgb` obtained by using *ChimeraX* `operate` protocol (see results Table 9 in Appendix 1; **Question 12_1**). Because of high values of CC_{MASK} and *EMRinger* `score`, as well as acceptable *MolProbity* statistics, *model* generated by *ChimeraX* `operate` protocol is selected as *model* of the whole human `metHgb`. Additional refinement steps with *PHENIX* `real space refine` and *Refmac* do not seem to improve the result significantly. In this case, the RMSD value of the selected atomic structure *model*, regarding the published structure, yields an intermediate value between the best and the worst one.

- Protocol chimerax - map subtraction to assess volume density occupancy:

  We perform this analysis in order to identify parts of the density map that were not modeled previously, maybe unknown parts of the complex, although areas where the *model* doesn't fit the map can be also identified. Sometimes the density level or the resolution in these areas differ from the rest of the map and commonly are more blurry, which makes them much more difficult to identify and trace. Ideally, we would like to remove the map density associated to the already traced atomic structure to facilitate the modeling of the remnant density. Obviously, there are some limitations in this process because the structure-derived map might not be absolutely identical to the reconstructed map. As one possible aproximation, we will run a protocol based on *ChimeraX* (see Appendix 4 with use cases) to subtract the modeled part of the map from the whole map.

  First, open the *ChimeraX* `map subtraction` protocol (Fig. 65 (1)), load both the initial map obtained from the reconstruction process (2) and its resolution. Although in this case we are going to consider the nominal map resolution, in real life you should test different resolution values among which the half

value of the resolution obtained by FSC is recommended. Include also the refined atomic structure *model* of the whole human `metHgb` (3). As a control of the subtraction process we are going to remove 7 residues of the chain `A`. With this aim, use the three wizards on the right (4) to select that chain and residues located between positions `22` and `28`, both included. Since we are interested in observing differences in the whole map, the default option `No` will be maintained regarding the selection of a map fraction around the atomic structure (5). Then, execute the protocol (6).

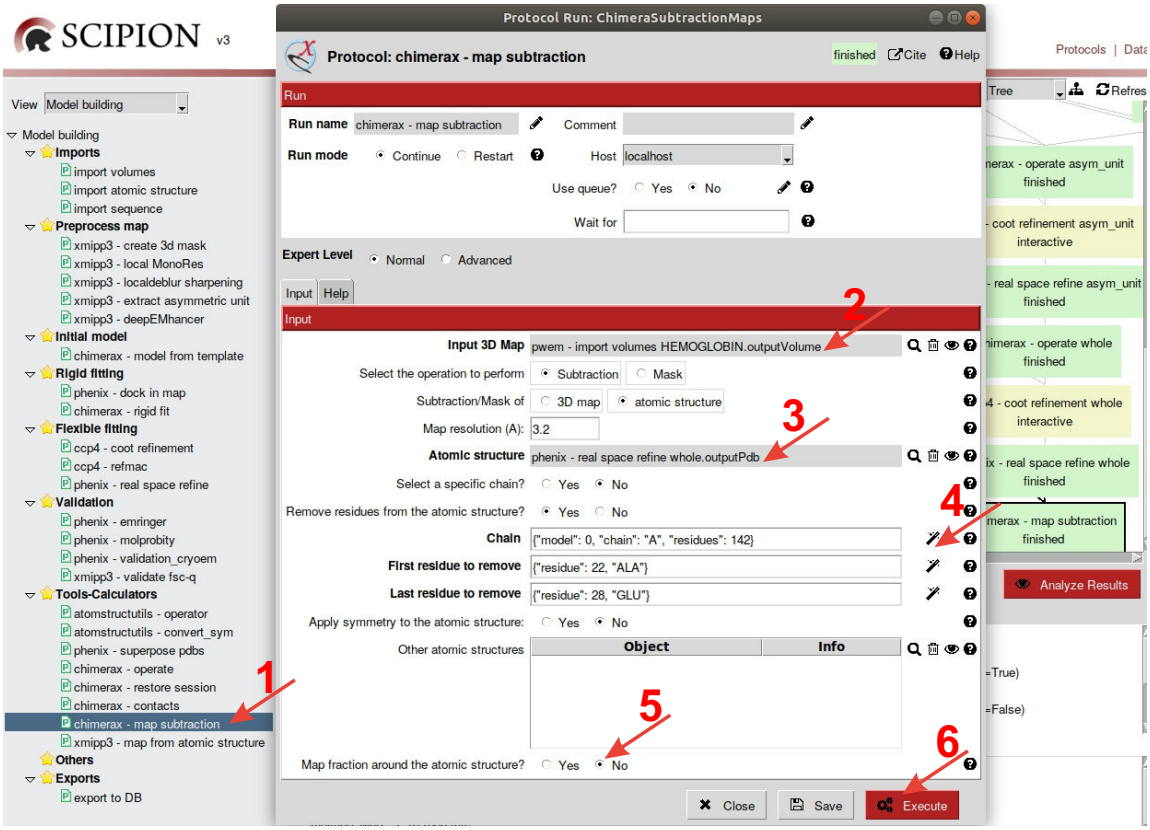


Figure 65: Completing the protocol chimerax - map subtraction.

ChimeraX graphics window will open and the commands driving the subtraction process will be applied. The Fig. 66 shows in blue the map resulting from subtracting the *model*-derived map from the starting map `EMD-3488` after

applying a Gaussian filter. Three main map bodies can be observed moving the density threshold of this map (model #9 in the Models panel). The red arrow number 1 points to the control map derived from removing 7 residues of the chain A of the atomic structure. The other two red arrows (number 2) point to two unexpected remnant densities.

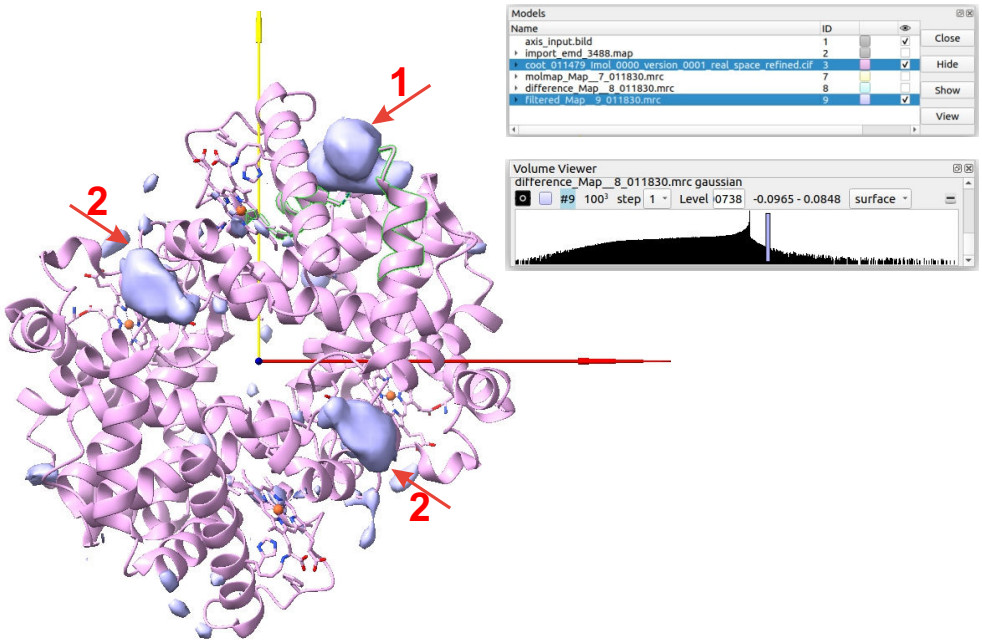


Figure 66: Filtered subtraction map (blue bodies) and refined atomic structure (pink) of the whole human Hgb.

The two additional bodies of density should not appear with an appropriate modeling of the human Hgb showing acceptable validation scores. However, in this final *model* of the whole human Hgb we didn't refine on purpose the C-terminal ends of chain A and its symmetric chain C. The ARG residues don't fit to the map density and the remnant densities identified in the subtraction protocol correspond to the C-terminal ends of chains A and C. A fair tracing of those parts of the molecule would avoid remnant densities others than the control. To check the right tracing of the human Hgb we have overlapped the above mentioned published atomic structure of the human Hgb (PDB ID 5NI1),

in green in the Fig. 67, and our final *model*, depicted in pink. The zoom in details the C-terminal end of our *model* (red arrow) and the published one (green arrow), which perfectly fits the body of density.

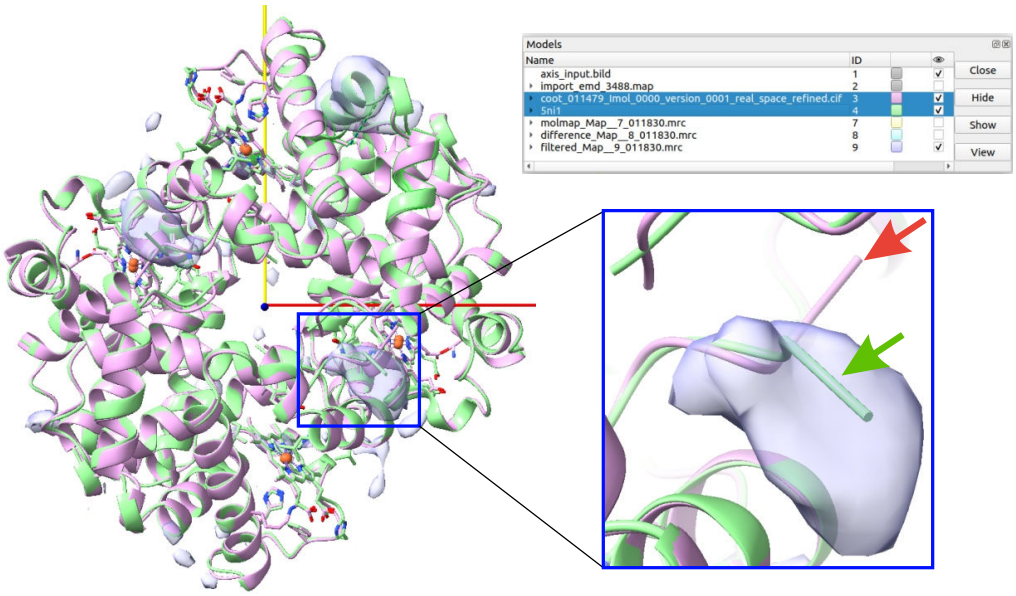


Figure 67: Overlapping structures of the models built (pink) and published (green) of the whole human `Hgb`. Zoom in to detail the C-terminal end of the chain `C`.

As a conclusion, if you do not have additional densities with the example of this tutorial, except the control one, you'd have performed a good modeling and you could use your atomic structure to perform other types of analyses and to publish it. Otherwise, you should still refine your *model*.

13 Summary of results and submission

Once we have selected the best *model* of the whole human `Hgb` and obtained good validation scores from *EMRinger*, *MolProbity* and other validation programs, and we have checked that we have the whole volume density modeled, we are ready to

submit the electron density map and its atomic interpretation to public databases and to make public our results.

Submission to public databases

Although submission of cryoEM maps and derived atomic structures to databases has to be done by direct online request (`https://deposit-pdbe.wwpdb.org/deposition/`), *Scipion* may contribute to organize the submission records. The protocol export to EMDB allows to perform this task (Appendix 27). By using this protocol we can save the files that you have/want to submit to databases in a labelled folder and in the appropriate format. Fig. 68 details the protocols of the modeling *Scipion* workflow involved in this task.

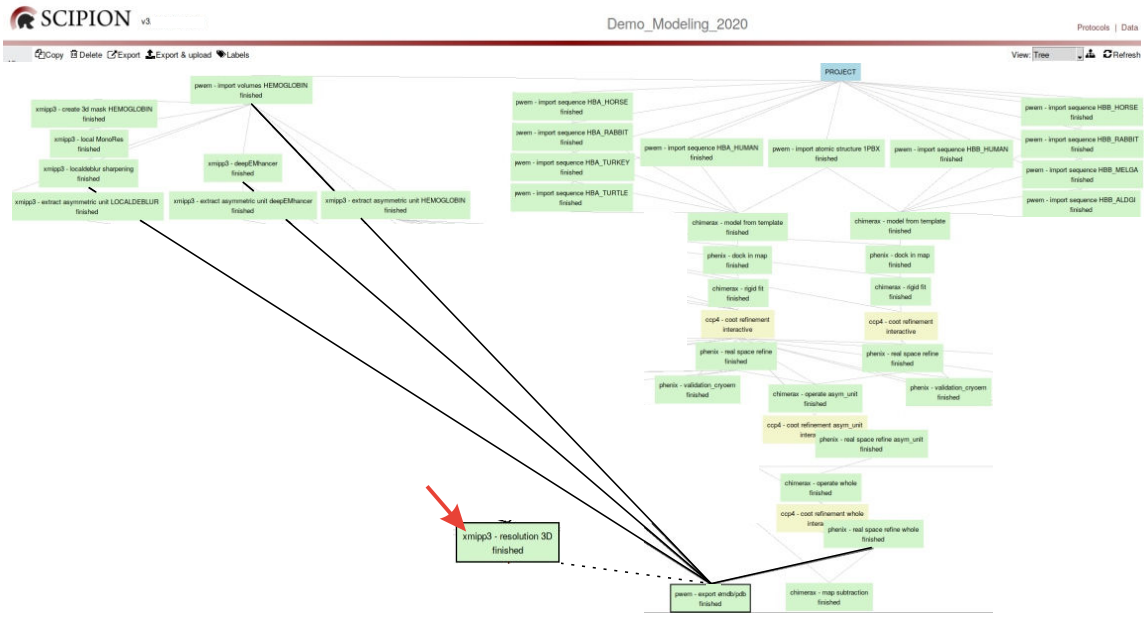


Figure 68: *Scipion* framework detailing the workflow to submit $cryo-EM$ results to databases.

When you submit the *map* and the *model* of a $cryo-EM$ experiment, besides these two records, an image of the *map* is also mandatory to submit. Other maps,

such as half maps or postprocessing-sharpening maps, as well as maks, are also recommended to submit. In addition, the FSC file is strongly encouraged. As you can see in Fig. 68, we can provide directly from the workflow the *map* and the *model*, as well as the two sharpening maps. The *map* image can be attached from a file. We lack, however, from the FSC file, since the FSC file is usually generated during the *map* reconstruction process starting from the half maps, for example with the xmipp3 - resolution 3D protocol(Fig. 68, red arrow). To compute the FSC file we could download the half maps from the database (`https://www.ebi.ac.uk/pdbe/entry/emdb/EMD-3488/index`) selecting the `zip` Bundle (Fig. 69 (red arrow)).

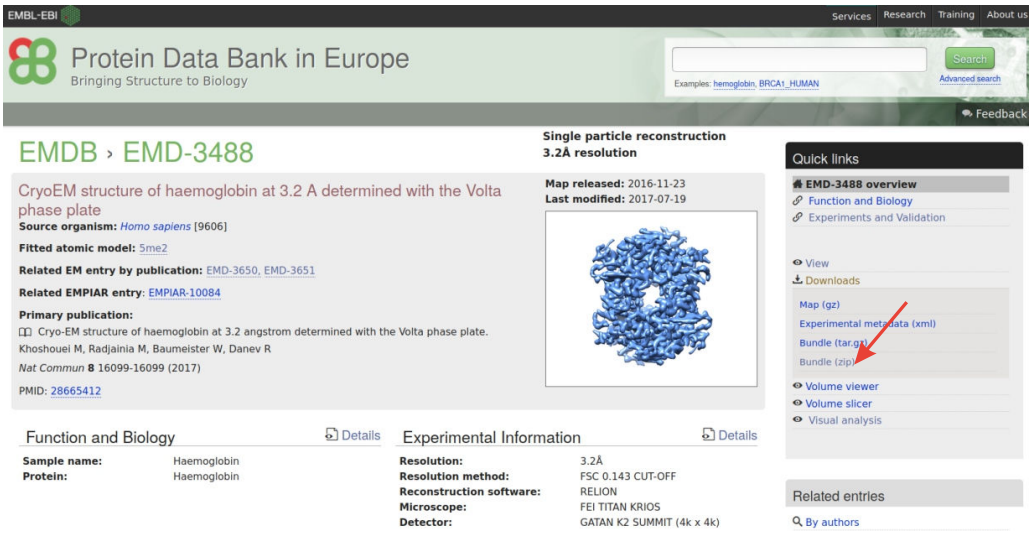


Figure 69: EMDB entry 3488 in PDBe

The `zip` folder contains the FSC file (`emd_3488_fsc.xml`) and the *map* image (`emd_3488.png`) but, unfortunately, lacks of half maps. Then, you can use any two half maps and compute the FSC file, just to submit it with the rest of the files.

To save all the relevant files in a single labelled folder, open the export to EMDB protocol (Fig. 70 (1)), and complete the form with the *Scipion* elements to export: `Main map` (2), `Additional maps:` `''Yes''` (3), the two sharpened maps as additional maps (4), the FSC file if you count on it (5), `Atomic structure` (6) and `Image` (7), previously saved in a known folder. Then, write the name of the exportation

directory path, or find it with the browser on the right. All submission files will be saved in the `directory` selected (8). A directory name related with the submission (number, date, project,...) is recommended.

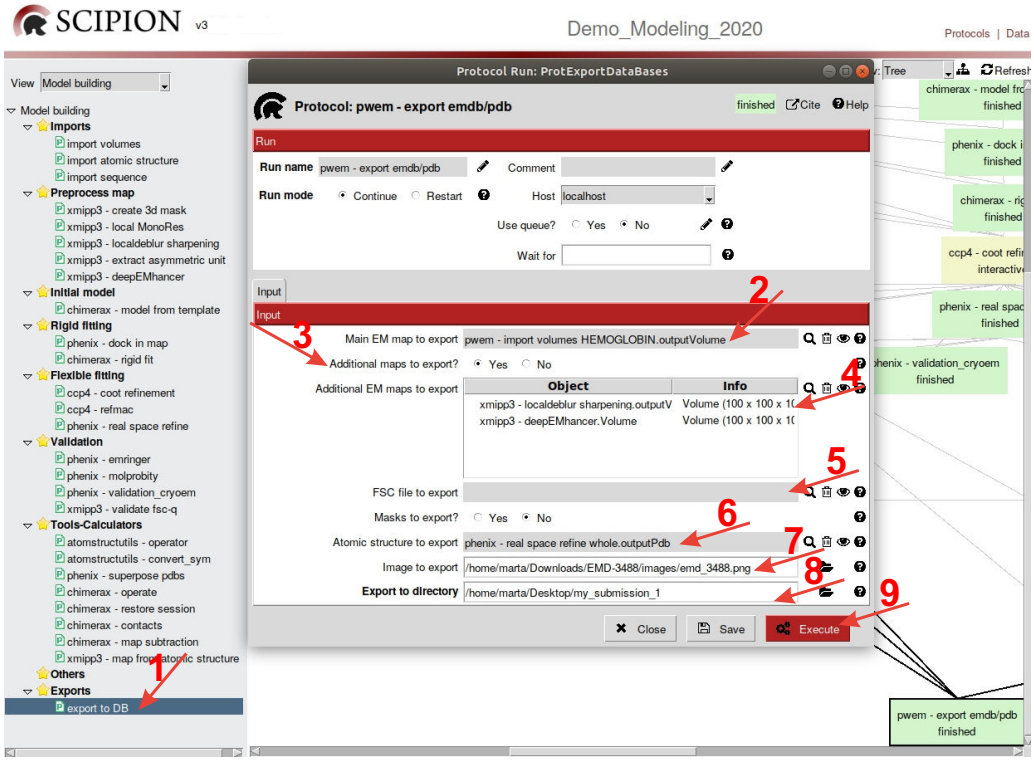


Figure 70: Saving files for submission to EMDB with protocol export to EMDB

After executing the protocol (9), you can check that all files are saved in the given directory. No additional visualization tools have been included in this protocol.

Publication of results

Since the atomic interpretation of a certain macromolecule will be probably the starting point of relevant mechanistic or biomedical studies, summaring and organizing our results constitutes the first step to draw the conclusions that will be made public by journals and talks. Many different questions can be posed based

on the atomic structure. Here we are wondering about interactions among members of the macromolecule. To answer this question we have included in *Scipion* the protocol chimerax - contacts to identify the residues involved in contacts between any couple of interacting molecules. "contacts" involve atoms within favorable interaction distances. Unfavourable contacts or severe clashes, in which atoms are too close together, although discarded by default in the final list of 'contacts", may also be shown by using appropriate advanced parameters, as you can see in Appendix 3.

As an example, in this tutorial we are going to learn how to get atom contacts of human haemoglobin `metHgb` atomic structure `5NI1`, associated to the starting map `EMD-3488`. This structure was already downloaded from `PDB` by using the protocol import atomic structure (Fig. 71 (1)). According to the aim of the analysis, two possible scenarios and the respective workflows can be considered to compute contacts: a) infering all contacts between any couple of members of the whole macromolecule (Fig. 71 (3)); b) infering all contacts between any couple of members of the asymmetric unit, and between one member of the asymmetric unit and another component from a neighbor asymmetric unit (Fig. 71 (5)).

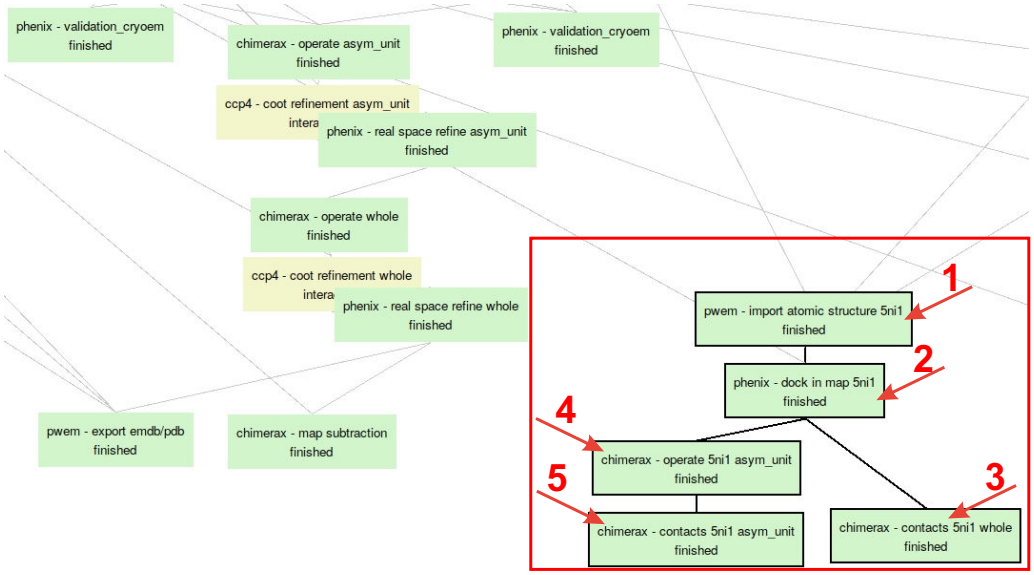


Figure 71: *Scipion* workflows inside the red box to get contacts between any two chains of a macromolecule (3) and between any two chains of the asymmetric unit, and between any chain of the asymmetric unit and a chain of a neighbor asymmetric unit (5).

Since the penultimate step of the second workflow (Fig. 71 (4)) requires applying symmetry, we are going to start moving the structure to match its symmetry center to the origin of coordinates using the protocol phenix - dock in map as we did previously (Fig. 32), including the whole starting map of the human `metHgb` and the imported atomic structure `5NI1` as `Input map` and `Input atom structure`, respectively.

Secondly, we are going to extract the structure of the asymmetric unit of the docked `5NI1` structure using the protocol chimerax - operator as it is indicated in Fig. 71 (4). Complete the protocol form including the last docked structure `5NI1` as `Atomic structure`. After executing the protocol, the *ChimeraX* graphics window will open. You can select and save the atomic structure of the map asymmetric unit writing in the *ChimeraX* command line:

```
select #2/A,B
```

```
save /tmp/chainAB.cif format mmcif models #2 selectedOnly true
open /tmp/chainAB.cif
scipionwrite #3 chainAB_
exit
```

- CASE A: Contacts between any couple of members of the whole macromolecule (Fig. 71 (3)):
  This option allows to get all contacts between all couples of members of the macromolecule. In the case of the human `metHgb` we have depicted all those possible contacts in the Fig. 72 (A).

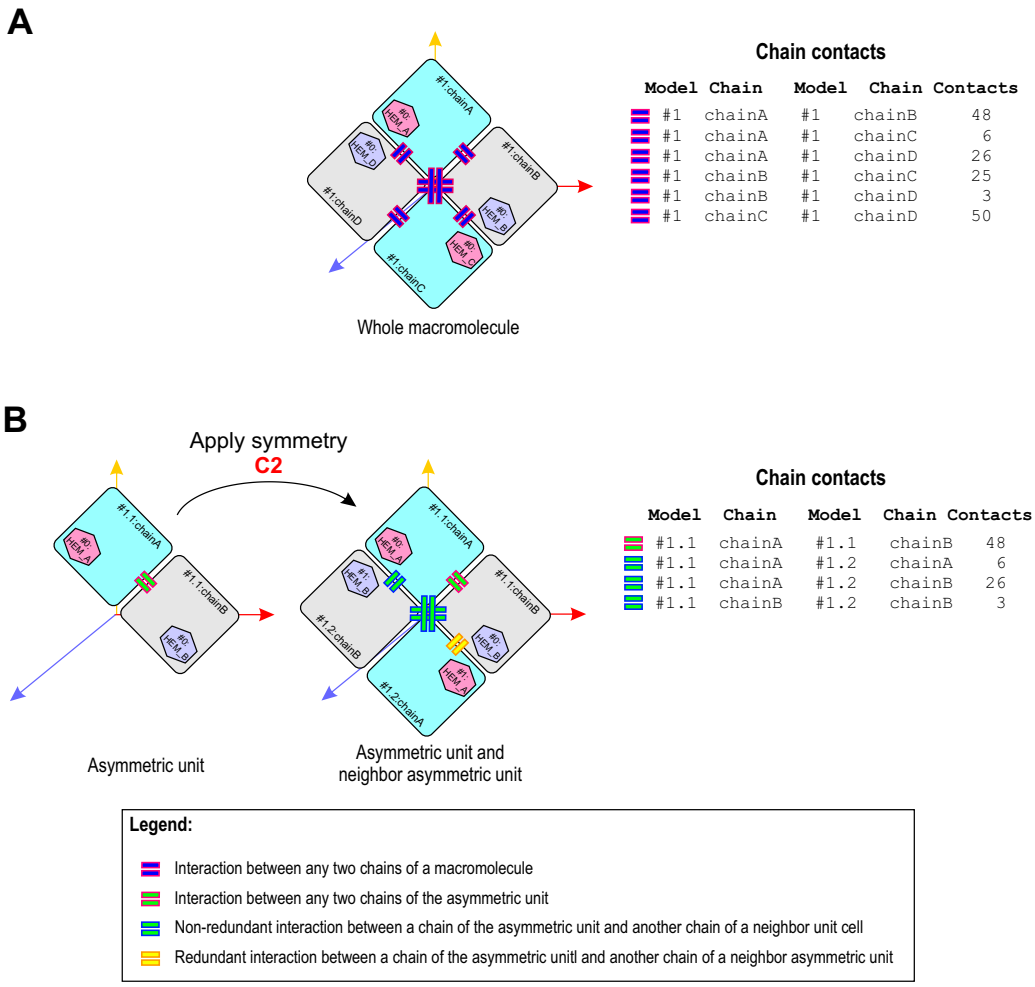


Figure 72: Schema of the human haemoglobin `metHgb` showing protein contacts between couples of chains of the whole macromolecule (A) and contacts obtained by applying symmetry to the asymmetric unit (B).

The protocol chimerax - contacts can be used to obtain the contacts depicted. Open this protocol (Fig. 73 (1)) and fill in the first `Input` (2) in which no symmetry will be applied. Include the docked `5NI1` structure (4) as `Atomic structure`. Use the wizard on the right to label the molecule chains (5) as they appear in the adjacent window, and execute the protocol.

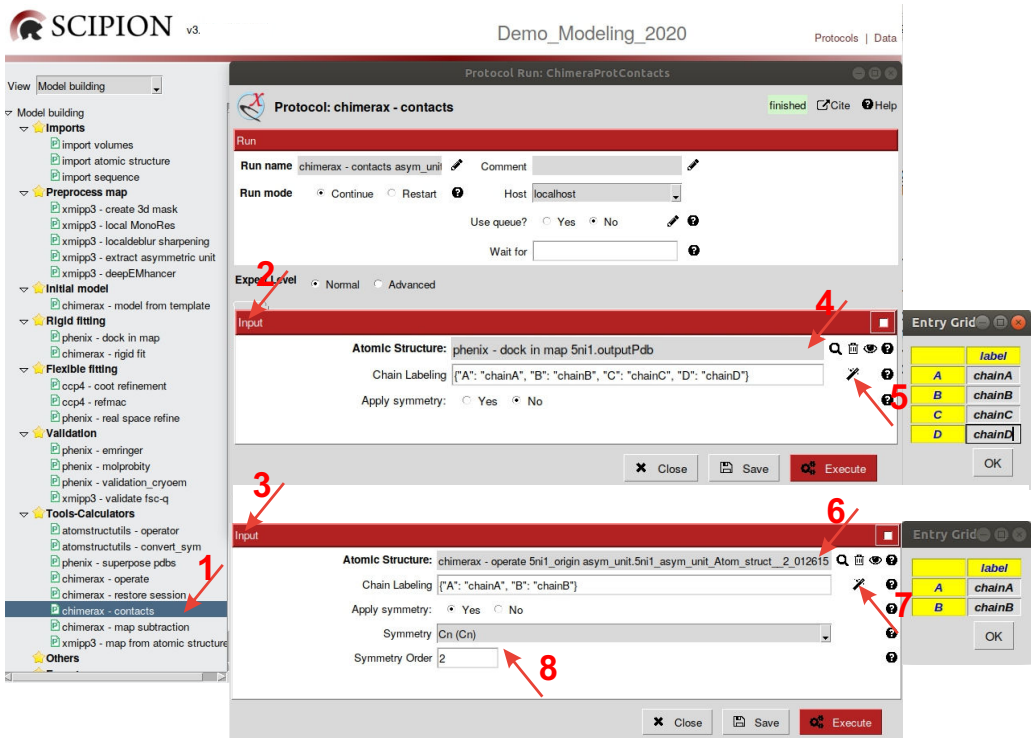


Figure 73: Filling in the chimerax - contacts protocol form with two different inputs: (2) to get atom contacts between couples of chains within the whole `metHgb`; (3) to get contacts between any couple of chains within the asymmetric unit, and "non-redundant" contacts between the asymmetric unit and another chain of a neighbor asymmetric unit of the human haemoglobin `metHgb`.

After executing the protocol, all atom contacts between the couples of proteins indicated in Fig. 72 (A) can be visualized by clicking Analyze Results (Fig. 74 (A)).

Figure 74: (A) Display of results of atom contacts between couples of chains within the whole `metHgb`; (B) Display of results of atom contacts between couples of chains within the asymmetric unit, and "non-redundant" contacts between a chain of the asymmetric unit and another chain from a neighbor asymmetric unit of the human haemoglobin `metHgb`.

The viewer window of the protocol *ChimeraX* `contacts` display different results (Fig. 74 (A)):

– `3D Visualization` box: Final atomic structure considered to compute

contacts that can be visualized with *ChimeraX*. Press the eye (1) to open the structure shown on the right.

- `Interacting chains` box: Summary list of all interacting chains, similar to the list shown on the right of the Fig. 72 (A). Press the eye to open it (2).
- `Contacts between interacting chains` box: In addition to the possibility of changing the order of the interacting chains in the display, as well as the maximal distance between residues to group them, this box allows to select couples of interacting chains (4) and inspect in detail the contacts between them pressing the eye on the right (3).

- CASE B: Contacts between any couple of members of the asymmetric unit and "non-redundant" contacts between one member of the asymmetric unit and another one from the neighbor asymmetric unit (Fig. 71 (5)). This second asymmetric unit has been obtained by applying symmetry with the protocol chimerax - contacts. Then, "non-redundant" interaction means any interaction that can not be inferred by symmetry. The Fig. 72 (B) shows the total number of interactions of our example. The interactions between the chain `B` of the asymmetric unit (model `#1.1`) and the chain `A` of the neighbor asymmetric unit (model `#1.2`) are symmetric to the interactions between chain `A` of the asymmetric unit (model `#1.1`) and chain `B` of the neighbor asymmetric unit (model `#1.2`). Since those interactions can thus be inferred by symmetry, they are "redundant" and are absent of the final list of contacts.

  Similarly to the case A, the protocol form has to be open (Fig. 73 (1)) and completed as indicated in the second `Input` (3). Include the asymmetric unit structure saved with the protocol *ChimeraX* `operate` (6), use the wizard on the right (7) to label the chains as it is shown on the right and, finally, include the respective type of symmetry of the human `metHgb` (8).

  Like in the case A, after executing the protocol all non-redundant atom contacts between any couple of proteins indicated in Fig. 72 (B) can be visualized

by clicking Analyze Results (Fig. 74 (B)). Besides the lower number of contacts displayed, remark that a relevant difference between the results of the case A and the case B is the final atomic structure visualized with *ChimeraX*, which discriminates between the starting asymmetric unit and the second one generated by symmetry.

`Note`: This second possibility of getting protein contacts observed in the case B is extremely useful when you have a big asymmetric unit, for example of a virus, and you are interested in contacts among proteins within the asymmetric unit and with other adjacent asymmetric units.

14 A Note on Software Installation

All the protocols shown in this document are available in the stable *Scipion* release `3.0.6` (code name *Eugenius*). This is a major release in which protocols are published as "plugins". Required plugins for each protocol are indicated in respective Appendices. Follow the instructions to install each plugin (`https://github.com/scipion-em/`).

In addition to the standard *Scipion* and `scipion plugins` installation, you need to install the following packages:

- **CCP4** (v. 7.0.056 or higher; protocols have been tested with v. 7.1): Connect to `http://www.ccp4.ac.uk/download/#os=linux` and follow instructions.
- **Phenix**: Connect to `https://www.phenix-online.org/download/` and follow instructions. Protocols have been tested for versions 1.13-2998, 1.16-3549, 1.17.1-3660 and 1.18.2-3874.
- **Clustal Omega**: `sudo apt-get install clustalo` (in ubuntu).
- **MUSCLE**: `sudo apt-get install muscle` (in ubuntu).

Finally, (1) edit the file `/.config/scipion/scipion.conf` and set the right values for the variables `CCP4_HOME` and `PHENIX_HOME`, and (2) execute `scipion config --update`

15 How to solve some problems that you can find during the execution of the modeling workflow

- *ChimeraX* command lines involving *Scipion* communication (`scipionwrite`, `scipionss`, `scipionrs` and `scipioncombine`) do not work:
  As indicated in `https://github.com/scipion-em/scipion-em-chimera/blob/devel/FAQ.rst`, these commands are *ChimeraX* plugins installed by the scipion-em-chimera setup. Firstly, check the right installation of the plugin just opening *ChimeraX* and executing the command line:

  ```
  help scipionwrite
  ```

  If it is installed, a help page will appear. Otherwise, type in the command line:

  ```
  devel install /path_to_scipion3_plugins/scipion-em-chimera/chimera/Bundles/scipio
  ```

  where path_to_scipion3_plugins is the path to the directory that contains Scipion3 plugins.
  Close the *ChimeraX* GUI and start the protocol again.

- `Maxit` installation

  – Remember: `Maxit` requires `flex` and `bison`. Install them with sudo apt-get

16 TODO

List of protocols in the process to be incorporated:

- **map_to_model:** (phenix) *de novo* model building.
- **buccaneer:** (ccp4) *de novo* model building.

References

Afonine, P. V., Klaholz, B. P., Moriarty, N. W., Poon, B. K., Sobolev, O. V., Terwilliger, T. C., Adams, P. D., Urzhumtsev, A., Sep 2018a. New tools for the analysis and validation of cryo-EM maps and atomic models. Acta Crystallogr D Struct Biol 74 (Pt 9), 814–840.

Afonine, P. V., Poon, B. K., Read, R. J., Sobolev, O. V., Terwilliger, T. C., Urzhumtsev, A., Adams, P. D., Jun 2018b. Real-space refinement in *PHENIX* for cryo-EM and crystallography. Acta Crystallographica Section D 74 (6), 531–544.
URL `https://doi.org/10.1107/S2059798318006551`

Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., Lipman, D. J., Sep 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25 (17), 3389–3402.

Barad, B. A., Echols, N., Wang, R. Y., Cheng, Y., DiMaio, F., Adams, P. D., Fraser, J. S., Oct 2015. EMRinger: side chain-directed model and map validation for 3D cryo-electron microscopy. Nat. Methods 12 (10), 943–946.

Brown, A., Long, F., Nicholls, R. A., Toots, J., Emsley, P., Murshudov, G., Jan 2015. Tools for macromolecular model building and refinement into electron cryo-microscopy reconstructions. Acta Crystallogr. D Biol. Crystallogr. 71 (Pt 1), 136–153.

Camardella, L., Caruso, C., D'Avino, R., di Prisco, G., Rutigliano, B., Tamburrini, M., Fermi, G., Perutz, M. F., Mar 1992. Haemoglobin of the antarctic fish Pagothe-

nia bernacchii. Amino acid sequence, oxygen equilibria and crystal structure of its carbonmonoxy derivative. J. Mol. Biol. 224 (2), 449–460.

Davis, I. W., Murray, L. W., Richardson, J. S., Richardson, D. C., Jul 2004. MOLPROBITY: structure validation and all-atom contact analysis for nucleic acids and their complexes. Nucleic Acids Res. 32 (Web Server issue), W615–619.

Emsley, P., Lohkamp, B., Scott, W. G., Cowtan, K., Apr 2010. Features and development of Coot. Acta Crystallogr. D Biol. Crystallogr. 66 (Pt 4), 486–501.

Goddard, T. D., Huang, C. C., Meng, E. C., Pettersen, E. F., Couch, G. S., Morris, J. H., Ferrin, T. E., 01 2018. UCSF ChimeraX: Meeting modern challenges in visualization and analysis. Protein Sci. 27 (1), 14–25.

Khoshouei, M., Radjainia, M., Baumeister, W., Danev, R., Jun 2017. Cryo-EM structure of haemoglobin at 3.2 Å determined with the Volta phase plate. Nat Commun 8, 16099.

Kovalevskiy, O., Nicholls, R. A., Long, F., Carlon, A., Murshudov, G. N., 03 2018. Overview of refinement procedures within REFMAC5: utilizing data from different sources. Acta Crystallogr D Struct Biol 74 (Pt 3), 215–227.

Kryshtafovych, A., Monastyrskyy, B., Fidelis, K., Moult, J., Schwede, T., Tramontano, A., Mar 2018. Evaluation of the template-based modeling in CASP12. Proteins 86 Suppl 1, 321–334.

Liebschner, D., Afonine, P. V., Baker, M. L., Bunk?czi, G., Chen, V. B., Croll, T. I., Hintze, B., Hung, L. W., Jain, S., McCoy, A. J., Moriarty, N. W., Oeffner, R. D., Poon, B. K., Prisant, M. G., Read, R. J., Richardson, J. S., Richardson, D. C., Sammito, M. D., Sobolev, O. V., Stockwell, D. H., Terwilliger, T. C., Urzhumtsev, A. G., Videau, L. L., Williams, C. J., Adams, P. D., Oct 2019. Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. Acta Crystallogr D Struct Biol 75 (Pt 10), 861–877.

Martínez, M., Jiménez-Moreno, A., Maluenda, D., Ramírez-Aportela, E., Melero, R., Cuervo, A., Conesa, P., DelCano, L., Fonseca, Y., Sanchez-García, R., Strelak,

D., Conesa, J., Fernandez-Giménez, E., de Isidro, F., Sorzano, C., Carazo, J., Marabini, R., 2020. Integration of Cryo-EM Model Building Software in Scipion. J Chem Inf Model 60 (5), 2533–2540.

Pearson, W. R., Jun 2013. An introduction to sequence similarity ("homology") searching. Curr Protoc Bioinformatics Chapter 3, Unit3.1.

Pérez-Illana, M., Martínez, M., Condezo, G. N., Hernando-Pérez, M., Mangroo, C., Brown, M., Marabini, R., Martín, C. S., 2020. Cryo-em structure of enteric adenovirus hadv-f41 highlights structural divergence among human adenoviruses. bioRxiv .
URL https://www.biorxiv.org/content/early/2020/07/01/2020.07.01.177519

Perutz, M. F., Rossmann, M. G., Cullis, A. F., Muirhead, H., Will, G., North, A. C., Feb 1960. Structure of haemoglobin: a three-dimensional Fourier synthesis at 5.5-A. resolution, obtained by X-ray analysis. Nature 185 (4711), 416–422.

Ramírez-Aportela, E., Vilas, J. L., Melero, R., Conesa, P., Martínez, M., Maluenda, D., Mota, J., Jiménez, A., Vargas, J., Marabini, R., Carazo, J. M., Sorzano, C. O. S., 2018. Automatic local resolution-based sharpening of cryo-em maps. bioRxiv .
URL https://www.biorxiv.org/content/early/2018/10/02/433284

Sali, A., Blundell, T. L., Dec 1993. Comparative protein modelling by satisfaction of spatial restraints. J. Mol. Biol. 234 (3), 779–815.

Sanchez-Garcia, R., Gomez-Blanco, J., Cuervo, A., Carazo, J., Sorzano, C., Vargas, J., 2020. Deepemhancer: a deep learning solution for cryo-em volume post-processing. bioRxiv .
URL https://www.biorxiv.org/content/early/2020/08/17/2020.06.12.148296

Vagin, A. A., Steiner, R. A., Lebedev, A. A., Potterton, L., McNicholas, S., Long, F., Murshudov, G. N., Dec 2004. REFMAC5 dictionary: organization of prior chem-

ical knowledge and guidelines for its use. Acta Crystallogr. D Biol. Crystallogr. 60 (Pt 12 Pt 1), 2184–2195.

Vilas, J. L., Gomez-Blanco, J., Conesa, P., Melero, R., Miguel de la Rosa-Trevin, J., Oton, J., Cuenca, J., Marabini, R., Carazo, J. M., Vargas, J., Sorzano, C. O. S., 02 2018. MonoRes: Automatic and Accurate Estimation of Local Resolution for Electron Microscopy Maps. Structure 26 (2), 337–344.

Williams, C. J., Headd, J. J., Moriarty, N. W., Prisant, M. G., Videau, L. L., Deis, L. N., Verma, V., Keedy, D. A., Hintze, B. J., Chen, V. B., Jain, S., Lewis, S. M., Arendall, W. B., Snoeyink, J., Adams, P. D., Lovell, S. C., Richardson, J. S., Richardson, D. C., 01 2018. MolProbity: More and better reference data for improved all-atom structure validation. Protein Sci. 27 (1), 293–315.

Zwart, P., Afonine, P., Grosse-Kunstleve, R., 2017. Superimposing two PDB files with superpose_pdbs. `https://www.phenix-online.org/documentation/reference/superpose_pdbs.html`, Accessed: 2018-10-31.

Appendices

1 Answers to Questions

- **Question 6_1**

  Method: X_Ray diffraction.
  Resolution: 2.5 Å
  Chains: 2; A (α chain) and B (β chain)

- **Question 9_1**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

CC_{MASK} value: 0.778
Residue with lower correlation value: 142 ARG (Misfit at the end of the chain)
Correlation value: 0.186172531458
Second residue with lower correlation value: 1 MET (Post-translationally processing)
Correlation value: 0.348504275208
Correlation value of `HEME` group: 0.81328813 (To get this value, Select Residue Type (Other) and Show CC below (0.9 or 1.0)).

- **Question 9_2**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

CC_{MASK} value has improved to 0.787.
A 142 ARG correlation has improved to 0.4282267700789.
`HEME` group correlation has not improved (0.81007005253).

- **Question 9_3**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

CC_{MASK} value has improved to 0.805.
A 142 ARG correlation has improved to 0.474205806292.
`HEME` group correlation has also improved to 0.821341112742.

- **Question 9_4:**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

Table 3: *Refmac* results:

Statistic	Initial	Final
`R factor`	0.3865	0.3441
`Rms BondLength`	0.0142	0.0165
`Rms BondAngle`	2.0081	1.9696
`Rms ChirVolume`	0.1401	0.0844

The improvement is quite remarkable.

- **Question 9_5:**

Table 4: *Refmac* results:

Statistic	Initial	Final
`R factor`	0.3506	0.3488
`Rms BondLength`	0.0137	0.0150
`Rms BondAngle`	1.6843	1.8655
`Rms ChirVolume`	0.0783	0.0783

Why: Because the starting values were already very good.

- **Question 9_6:**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

Table 5: *Refmac* results. `RSRAC` stands for `Real Space Refine` after *Coot*.

Refmac	`RSRAC`		*Coot*	
Statistic	Initial	Final	Initial	Final
`R factor`	0.4869	0.4855	0.4971	0.4825
`Rms BondLength`	0.0176	0.0212	0.0136	0.0193
`Rms BondAngle`	1.9186	2.3549	1.8053	0.2382
`Rms ChirVolume`	0.1112	0.1055	0.1470	0.1043

Starting and final `Rfactor` values seem to be worse when we do not generate a mask volume around the atomic structure when *Refmac* runs both after *Coot* + *PHENIX* `Real Space Refine` (compare with Table 4) and after *Coot* (compare with Table 3). Whitout using a delimiting mask, the whole volume is considered, even if the structure fits to a small part of the volume. The use of mask is thus especially indicated when map and model show different sizes. However, no differences are detected when the volume generated by the `extract unit cell` protocol or normalized volume generated by *Coot* are used (data not shown).

- **Question 10_1**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

EMRinger `score` after *ChimeraX* `rigid fit`: 3.86.
EMRinger `score` after *Coot*: 2.37; Manual refinement depends on each user and in this case, for instance, we did not pay attention to rotamers.
EMRinger `score` after *Phenix* `real space refine` after *Coot*: 5.38
EMRinger `score` after *Refmac* after *Coot*: 2.87; With *Refmac* parameters used, the improvement got with *Phenix* `real space refine` after *Coot* is clearly higher than the improvement got with *Refmac* after *Coot*.
EMRinger `score` after *Refmac* after *Phenix* `real space refine` after *Coot*:

5.34; *Refmac* does not improve very much the result (because it was already good). With the *EMRinger* statistic, we can say that the modeling workflow is helpful to get a quite good model.

- **Question 10_2:**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

Table 6

Table 6: Validation statistics of human `metHgb` α subunit *model*. `RSRAC` stands for `Real Space Refine` after *Coot*. `Rama` stands for Ramachandran.

Statistic	*ChimeraX*	*Coot*	*Phenix* `RSRAC`	*Refmac* after *Coot*	*Refmac* after `RSRAC`	`5NI1`
CC_{MASK}	0.569	0.725	0.787	0.803	0.801	0.843
EMRinger `score`	3.86	2.37	5.38	2.87	5.34	3.98
`RMS` (Bonds)	0.0188	0.0183	0.0090	0.020	0.0191	0.0126
`RMS` (Angles)	2.41	2.02	1.30	1.94	1.84	1.43
`Rama favored` (%)	97.14	97.12	95.68	97.12	95.68	94.24
`Rama allowed` (%)	2.15	2.88	4.32	2.88	4.32	5.76
`Rama outliers` (%)	0.71	0.00	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	1.75	24.78	0.00	23.01	1.77	0.88
`Clashscore`	70.34	26.24	1.81	21.73	1.36	2.26
`Overall score`	2.66	3.12	1.24	3.02	1.35	1.39
`C`β `deviations`	1	7	0	2	0	0
`RMSD`	0.841	0.447	0.456	0.414	0.384	0.0

- **Question 10_3:**

NOTE: Actual values depend on previous fitting and they may differ from the ones shown in this appendix.

Table 7

Table 7: Validation statistics of human `metHgb` β subunit *model*. `RSRAC` stands for `Real Space Refine` after *Coot*. `Rama` stands for Ramachandran.

Statistic	*ChimeraX*	*Coot*	*Phenix* `RSRAC`	*Refmac* after *Coot*	*Refmac* after `RSRAC`	`5NI1`
CC_{MASK}	0.524	0.690	0.776	0.765	0.767	0.830
EMRinger `score`	1.13	3.93	4.76	3.70	5.32	4.87
`RMS` (Bonds)	0.0313	0.0169	0.0078	0.0191	0.0183	0.0117
`RMS` (Angles)	2.17	1.97	1.33	1.95	1.87	1.40
`Rama favored` (%)	96.55	97.92	96.53	97.22	95.14	95.83
`Rama allowed` (%)	3.45	2.08	3.47	2.78	4.86	4.17
`Rama outliers` (%)	0.0	0.00	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	1.68	29.66	0.85	27.97	5.93	0.00
`Clashscore`	75.93	34.24	3.89	25.57	2.16	4.32
`Overall score`	2.75	3.16	1.40	3.14	1.92	1.50
`C`β `deviations`	0	8	0	1	0	0
`RMSD`	0.935	0.495	0.470	0.441	0.494	0.0

- **Question 11_1:**

Table 8: Validation statistics of human `metHgb` unit cell *model*. `RSR` stands for `Real Space Refine`. `Rama` stands for Ramachandran.

Statistic	*ChimeraX* `rigid fit`	*Phenix* `RSR`	*Refmac* after `RSR`	`5NI1`
CC_{MASK}	0.787	0.808	0.789	0.840
EMRinger `score`	4.64	4.58	4.35	4.11
`RMS` (Bonds)	0.0187	0.0093	0.0182	0.0122
`RMS` (Angles)	1.860	1.380	1.840	1.410
`Rama favored` (%)	95.41	95.41	94.70	95.05
`Rama allowed` (%)	4.59	4.59	5.30	4.95
`Rama outliers` (%)	0.0	0.0	0.00	0.00
`Rotamer outliers` (%)	3.90	0.00	3.90	0.43
`Clashscore`	5.31	3.54	3.32	3.53
`Overall score`	2.05	1.46	1.94	1.49
`C`β `deviations`	0	0	0	0
`RMSD`	0.494	0.509	0.537	0.00

- **Question 12_1:**

Table 9: Validation statistics of whole human `metHgb` *model*. `RSR` stands for `Real Space Refine`. `Rama` stands for Ramachandran.

Statistic	*ChimeraX* `operate`	*Phenix* `RSR`	*Refmac* after `RSR`	`5NI1`
CC_{MASK}	0.810	0.803	0.792	0.842
EMRinger `score`	4.95	4.70	4.05	4.18
`RMS` (Bonds)	0.0093	0.0076	0.0181	0.0122
`RMS` (Angles)	1.390	1.350	1.860	1.410
`Rama favored` (%)	95.41	95.41	95.41	95.23
`Rama allowed` (%)	4.59	4.59	4.59	4.77
`Rama outliers` (%)	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	0.00	0.00	5.41	0.43
`Clashscore`	4.97	3.21	2.54	3.53
`Overall score`	1.58	1.43	1.94	1.48
`C`β `deviations`	0	0	0	0
`RMSD`	0.579	0.642	0.454	0.00

2 Atomic Structure Chain Operator protocol

Protocol designed to perform two types of operations with chains from atomic structures in *Scipion*: a) Chain extraction: An individual chain will be extracted from a polymeric atomic structure. The extracted chain will be saved as monomer in a new atomic structure and will not include HETATM and water molecules. b) Chain addition: One or several chains will be added to a reference atomic structure. The resulting addition will be saved as a new polymeric atomic structure.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-atomstructutils`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Tools-Calculators` (Fig. 75 (A))

- Protocol form parameters (Fig. 75 (B and C)):

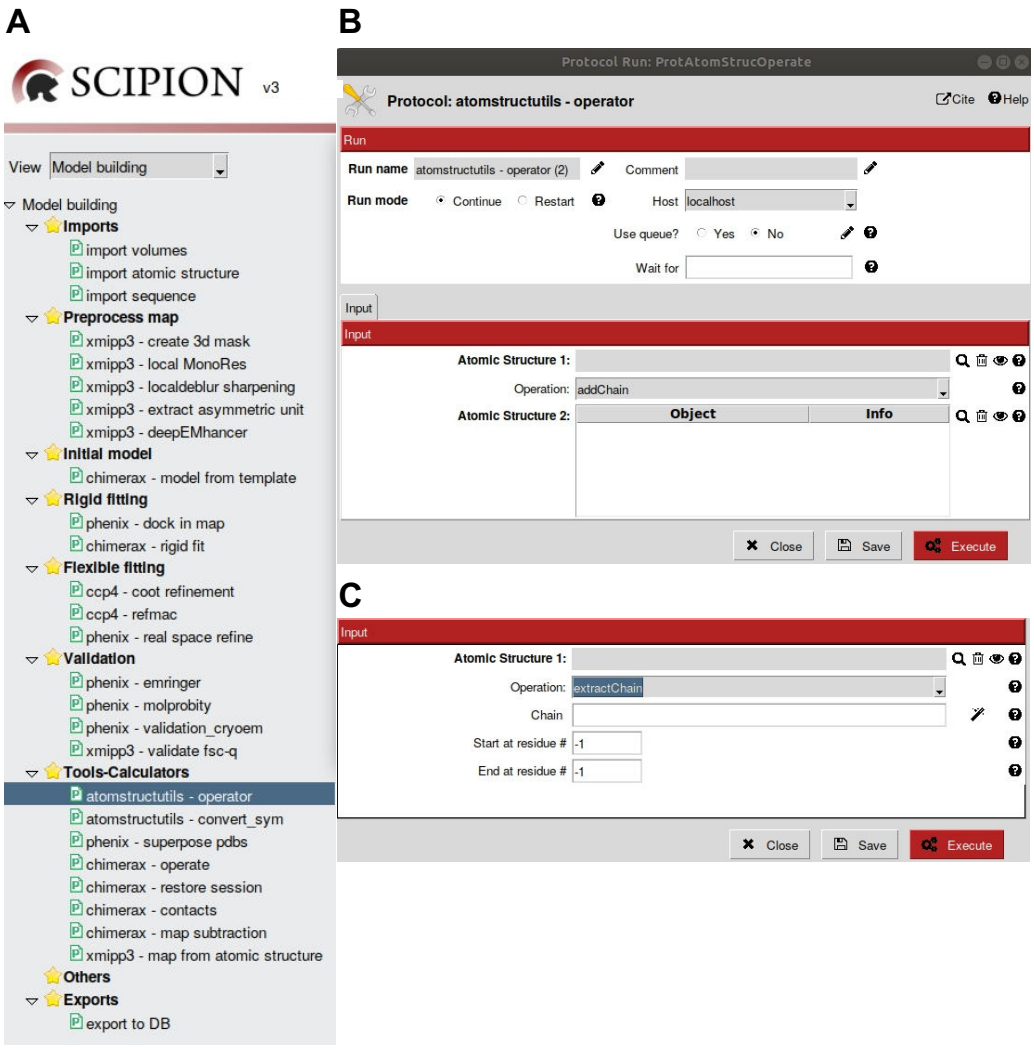


Figure 75: Protocol atomstructutils - operator. A: Protocol location in *Scipion* menu. B: Protocol form to extract a chain from an atomic structure. C: Protocol form to add one or several chains to an atomic structure.

  - `Atomic structure 1`: PDBx/mmCIF atomic structure, previously downloaded or generated in *Scipion*.
  - `Operation`: Two types of operations can be performed with this protocol:

    * `extractChain`: Extraction of only one chain from a polymeric atomic structure. By selecting this option, three additional params have to be completed (Fig. 75 (B)):
        · `Chain`: Specific chain that has to be extracted. The wizard on the right helps the user to select that chain showing the number of the starting model structure, the name of the chain, and its number of residues.
        · `Start at residue #`: The default value (-1) allows to extract the whole chain. In case you would like to extract only a fraction of the chain, the number of the initial required residue should be indicated.
        · `End at residue #`: The default value (-1) allows to extract the whole chain. In case you would like to extract only a fraction of the chain, the number of the last required residue should be indicated.
    * `addChain`: Addition of one or several chains to an initial atomic structure. By selecting this option, an additional param has to be completed (Fig. 75 (C)):
        · `Atomic structure 2`: One or several PDBx/mmCIF atomic structures, previously downloaded or generated in *Scipion*.

- Protocol execution: Adding specific structure/chain label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred

to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes and the new atomic structure generated are model numbers `#1` and `#2`, respectively, in *ChimeraX* `Models` panel. Write in *ChimeraX* command line:
`split #2`
to check the individual chains included in the new atomic structure generated.

- Summary content:
  Since an atomic structure is generated:

  - Protocol output (below *Scipion* framework):
    `atomstructutils - operator -> ouputPdb`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    No summary information.

3 ChimeraX Contacts protocol

Protocol designed to obtain contacts favorable and unfavorable (clashes or close contacts, where atoms are too close together) between any couple of chains of an atomic structure in *Scipion* by using *ChimeraX*.

- Requirements to run this protocol and visualize results:

  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu: `Model building -> Tools-Calculators` (Fig. 76 (A))
- Protocol form parameters (Fig. 76 (B)):

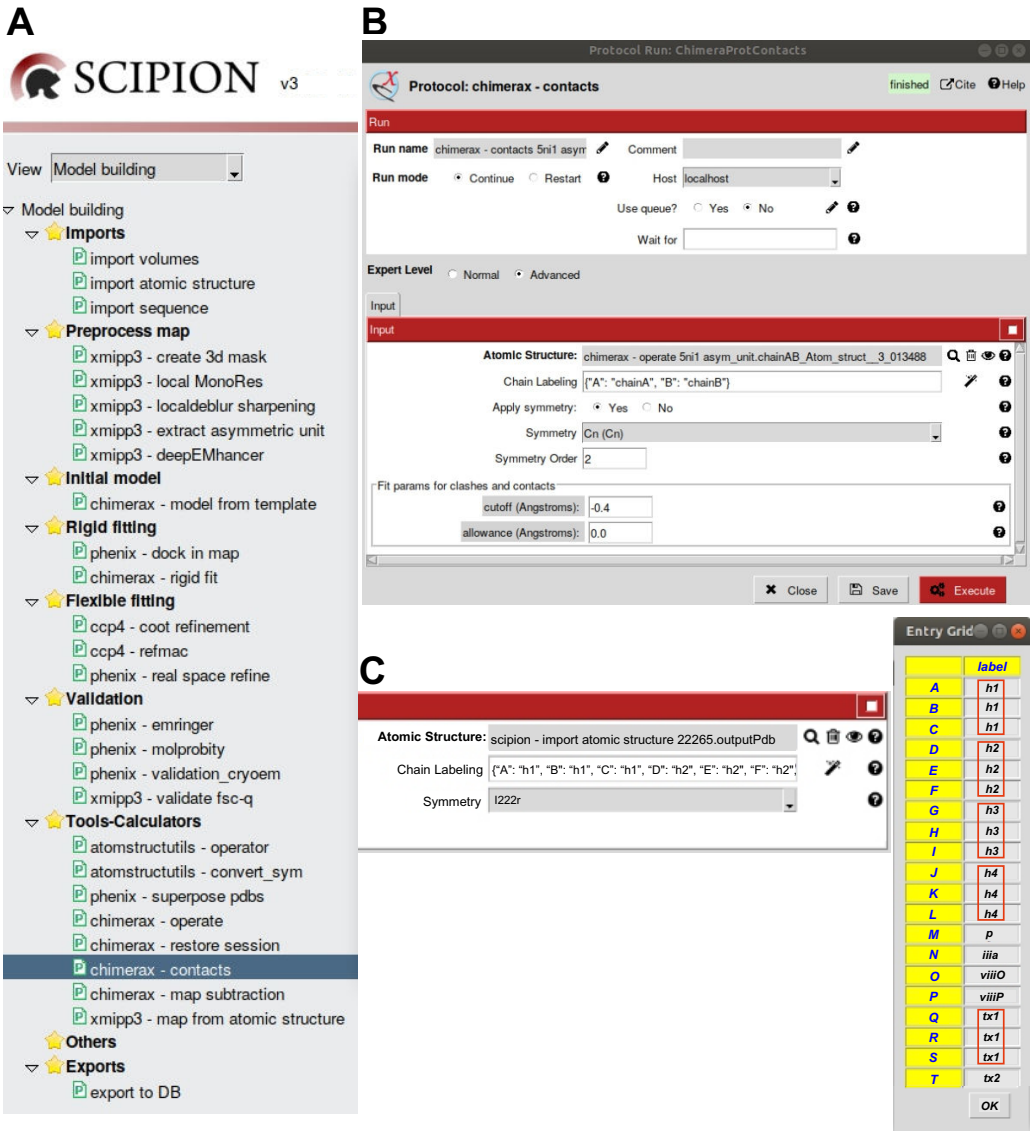


Figure 76: Protocol chimerax - contacts. A: Protocol location in *Scipion* menu. B: Protocol form. C: Protocol form detailing `Chain Labelling` for `I222r` symmetry.

- `Atomic Structure`: Param to select one atomic structure previously downloaded or generated in *Scipion* with the aim of calculating contacts between any couple of chains.
- `Chain Labeling`: Param to asign a specific label for each one of the chains of the atomic structure. Chain labeling allows to group chains in order to get only contacts among chains from different groups. When two chains show the same label, contacts between any of these chains and an independent chain, or a chain that belongs to a different group, will be calculated. However, no contacts will be computed between chains included in the same group. Fig. 76 (C) shows an example of chain grouping in four different groups. Each one of these groups includes three chains: $h1 : [A, B, C]$; $h2 : [D, E, F]$; $h3 : [G, H, I]$; $h4 : [J, K, L]$; $tx1 : [Q, R, S]$. The rest of chains remain as independent chains. There is a wizard on the right side of the `Chain Labeling` protocol form box to help the user to fill in the form since it specifies the names of the different chains included in the `Atomic Structure` input.
- `Apply symmetry`: Param that allows the user to select if symmetry has to be applied.
  - Set to `Yes` if the `Atomic Structure` input is the asymmetric unit of a macromolecule and you'd like to know the contacts between any two chains within the asymmetric unit as well as the contacts between any chain of the asymmetric unit and a chain from a neighbor asymmetric unit. Consider, in this case, that only neighbor unit cells located at less than 3Å of the input unit cell will be generated.
    WARNING: Be sure that the origin of coordinates equals the symmetry center of the input asymmetric unit, in order to generate adjacent asymmetric units able to interact with the input asymmetric unit.
  - Set to `No` if you'd like to know the contacts between any two chains within the `Atomic Structure` input.
- `Symmetry`: If the user selects `Yes`, an additional protocol param box will interrogate about the type of symmetry. In order to reconstruct a macro-

molecule from a unit cell, symmetries allowed are cyclic (Cn), dihedral (Dn), tetrahedral (T), octahedral (O), and eight icosahedral symmetries (I). Each icosahedral symmetry shows its respective *ChimeraX* orientation (https://www.cgl.ucsf.edu/chimerax/docs/user/commands/sym.html):

  * I222: *ChimeraX* orientation 222; two-fold symmetry axes along the X, Y, and Z axes.
  * I222r: *ChimeraX* orientation 222r; *ChimeraX* orientation 222 rotated 90°about Z.
  * In25: *ChimeraX* orientation n25; two-fold symmetry along Y and 5-fold along Z.
  * In25r: *ChimeraX* orientation n25r; *ChimeraX* orientation n25 rotated 180°about X.
  * I2n3: *ChimeraX* orientation 2n3; two-fold symmetry along X and 3-fold along Z.
  * I2n3r: *ChimeraX* orientation 2n3r; *ChimeraX* orientation 2n3 rotated 180°about Y.
  * I2n5: *ChimeraX* orientation 2n5; two-fold symmetry along X and 5-fold along Z.
  * I2n5r: *ChimeraX* orientation 2n5r; *ChimeraX* orientation 2n5 rotated 180°about Y.

- Symmetry Order: After selecting Cn or Dn symmetries, an additional protocol param box will interrogate about the symmetry order. A positive integer has to be written here. If the integer is 1 no symmetry will be applied.
- Tetrahedral orientation: After selecting T symmetry, an additional protocol param box will interrogate about the tetrahedral orientation. The two *ChimeraX* orientation have been included (https://www.cgl.ucsf.edu/chimerax/docs/user/commands/sym.html):

  * 222: Two-fold symmetry axes along the X, Y, and Z axes, a three-fold along axis (1,1,1).

    * `z3`: A three-fold symmetry axis along Z and another three-fold axis in the YZ plane.

  – `Fit params for clashes and contacts`: Advanced params that allow to modify interatomic distances in order to identify not only favorable interactions (by default), but also unfavorable ones (clashes) where atoms are too close together (`https://www.cgl.ucsf.edu/chimerax/docs/user/commands/clashes.html`).

    * `cutoff (Angstroms)`: Negative cutoff indicates favorable contacts; the default value to identify contacts is -0.4 (from 0.0 to -1.0). The default value to identify clashes is 0.6 (from 0.4 to 1.0). Large positive cutoff identifies the more severe clashes.

    * `allowance (Angstroms)`: The default value to identify contacts is 0.0, whereas the default value to identify clashes is 0.4.

- Protocol execution:

  Adding specific structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results: After executing the protocol, the chimerax - contacts viewer window will be opened. This window includes three boxes (Fig. 74):

  – `3D Visualization`: *ChimeraX* graphics window will be opened by selecting this option. The input atomic structure is shown, as well as the additional structure generated, if symmetry has been applied.

  – `Interacting chains`: A text file will be opened detailing the number of atomic contacts, the models and the chains involved in contacts. Two scenarios can be examined:

* If `Apply symmetry` was set to `Yes`: If no chain groups have been established, all contacts between any couple of chains within the input atomic structure will be shown. Besides,"non-redundant" contacts between any chain of the input unit cell structure and any chain of the neighbor unit cells will also be shown. By "non-redundant" contacts we define all those contacts that cannot be inferred by symmetry. An example of this type of contacts is shown in Fig. 72 (A). In addition, input atomic structure is model `#1.1`, whereas models generated by symmetry will be `#1.2, #1.3` and so on, if several models are generated. Each one of these models is supposed to be a neighbor unit cell located at less than 3 Å from the input one.
  WARNING: If no additional models are generated at less than 3 Å from the input one, consider the possibility that the symmetry center of the input structure does not coincide with the center of coordinates.
* If `Apply symmetry` was set to `No`: If no chain groups have been established, all contacts between any couple of chains within the input atomic structure will be shown (Example in Fig. 72 (A)). There is only one model in this case, model `#1`.

– `Contacts between interacting chains`: This box allows to select a particular interaction between two chains to identify the residues involved in that interaction. The summary of results will be displayed in a text file. It includes the number of atom contacts between the residues of chain 1, model 1 and the residues of chain 2, model 2.

* `Swap chain columns in the summary of contacts`: Select `Yes` to display in the text file the number of contacts between the residues of chain 2, model 2 and the residues of chain 1, model 1. Otherwise, selecting `No`, the default order of columns will be shown.
* `Distance to group residues (Number of residues)`: Maximum number of residues between two residues that allows to group these two residues. Then, if two residues are closer than this number of residues (distance), they will be grouped. In a long list of grouped

residues, the distance between two consecutive residues has to be lower than the set number of residues, 4 by default.

  * `Select two interacting chains and get the summary of contacts`: Select a particular interaction with the scroll arrow on the right and view the text file with the summary of contacts for that interaction.

- Summary content:

  - Protocol output (below *Scipion* framework): No output information.
  - `SUMMARY` box:
    No summary information.

4 ChimeraX Map Subtraction protocol

ChimeraX-based protocol designed to subtract two maps. These two maps can be two density maps experimentally obtained or derived from different computations, including the generation of a density map from an atomic structure. In the context of the *Scipion* modeling workflow this protocol helps to find out unmodeled densities in a map as a whole or in a specific part of it. In addition, wrong modeled regions can be also identified with this protocol since the atomic structure could doesn't fit to the density map.

- Requirements to run this protocol and visualize results:

  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu: `Model building -> Tools-Calculators` (Fig. 77 (A))

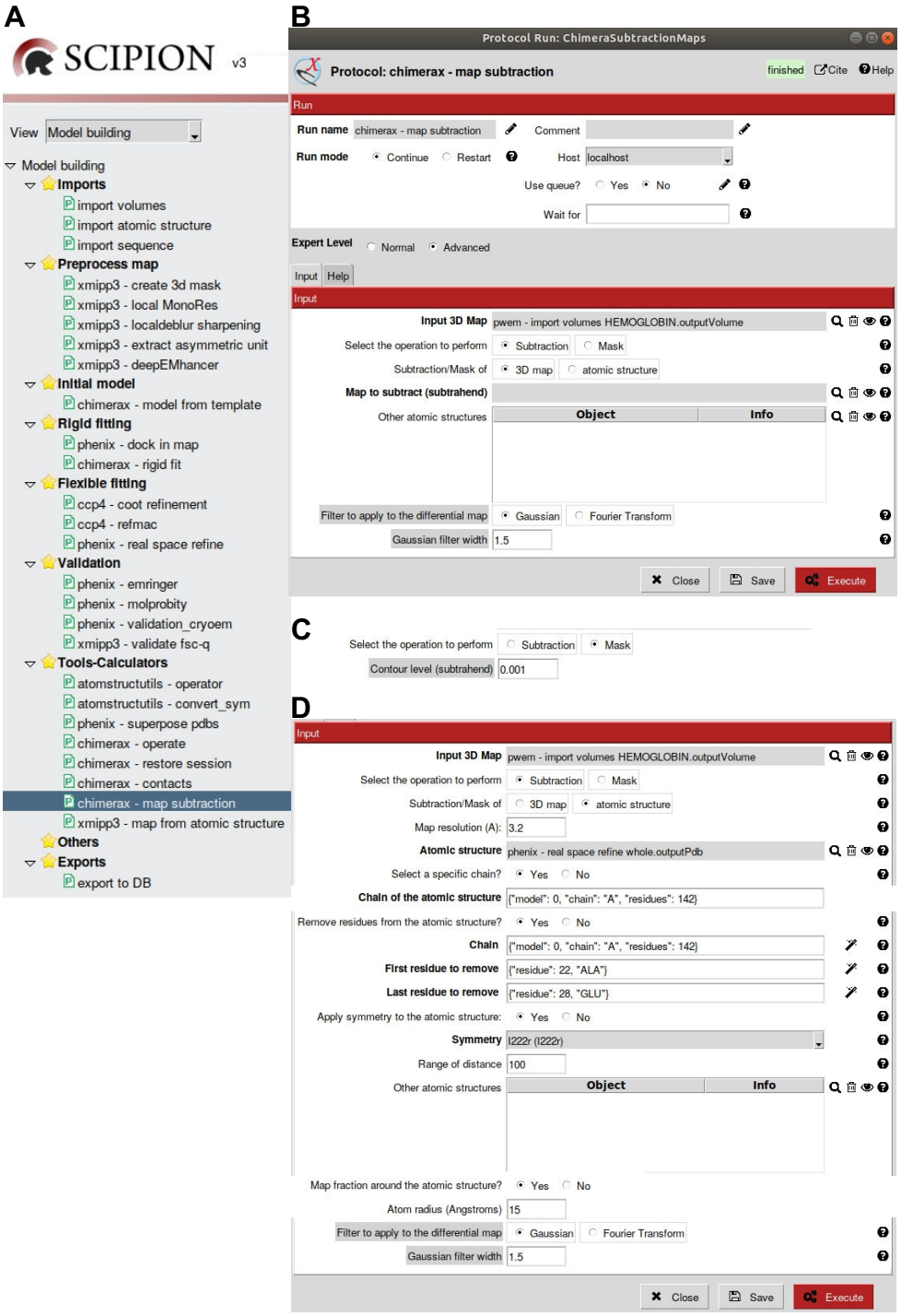


Figure 77: Protocol chimerax - map subtraction. A: Protocol location in *Scipion* menu. B: Protocol form to subtract two maps. C: Param option `Mask`. D: Protocol form to subtract an atomic structure from a map. All possible params are shown.

- Protocol form parameters (Fig. 77 (B,C,D)):
  `Input` section:

  - `Input 3D Map`: Include here any map previously downloaded or generated in *Scipion* that you would like to use as minuend of the subtraction operation.
  - `Select the operation to perform`: Two possibilities are allowed:
    - `Subtraction`: Between minuend and subtrahend maps, and you'll obtain the difference. WARNING: Both maps have to be perfectly aligned.
    - `Mask`: The voxel region of the subtrahend greater than a certain level will be masked (Fig. 77 (C)). The default level is `0.001` although can be modified with the `Advanced` param `Contour level (subtrahend)`. If no level is supplied, *ChimeraX* will compute that level value.
  - `Subtraction/Mask of`: Select the subtrahend of the subtraction operation. Two possibilities are allowed:
    - `3D map`: Any map previously downloaded or generated in *Scipion*. WARNING: The sampling rate of this map should be identical to the subtrahend's.
    - `atomic structure`: Previously downloaded or generated in *Scipion*. By selecting this option many new params will interrogate about the structure-derived map that you would like to generate (Fig. 77 (D)).
      - `Map resolution (Å)`: This is a tricky param and a uniform rule cannot be followed since, although its value is related with the minuend map resolution obtained by computing the `FSC` in the reconstruction process, local variations of this resolution seem to be involved. As a general rule, start with a resolution value half of the one obtained by the `FSC` and check your results. Then test other resolution values closer to the one computed by the `FSC` and compare the results with the previous one. At the end select

the resolution that maximizes the difference between the minuend and the subtrahend.

- `Atomic structure`: Select the atomic structure in the *Scipion* workflow to generate the called `molmap_Map`.
- `Select a specific chain?`: In case you are interested in generate the sustrahend `3D map` from the input atomic structure as a whole, answer `No` to this question. However, answer `Yes` if you want to derive that map from a specific chain of the atomic structure. If this is the case, a new param (`Chain of the atomic structure`) will interrogate you about the specific chain that you can select with the help of the wizard on the right.
- `Remove residues from the atomic structure?`: Select `Yes` to answer this question in case you'd like to count on a control of density levels to identify the differential density. The aim of this control is identify the density of the removed residues in the differential map. However, be cautious about discarding other densities that could appear in lower resolution areas and have density levels slightly different that the control one. After running the program the *ChimeraX* graphics window will open and the atomic structure won't show the removed residues. To make easier the localization of this area, ten residues both upstream and downstream of the removed aminoacids will be highlighted.
  Additional params to interrogate about the residues to be removed are `Chain`, `First residue to remove` and `Last residue to remove`. A wizard on the right helps to select this three elements. WARNING: In case you have already selected a specific chain of the structure to generate the `3D Map`, this chain will appear by default in the param `Chain` since the selection of a different chain wouldn't make sense.
- `Apply symmetry to the atomic structure`: In case your input atomic structure to derive the subtrahend `3D map` corresponds to

the asymmetric unit and you'd like to have the whole atomic structure or at least several adjacent asymmetric units together with the input one, select the option `Yes`. Otherwise, the subtrahend derived map will only correspond to the asymmetric unit. All *ChimeraX* symmetries will be available (`https://www.cgl.ucsf.edu/chimerax/docs/user/commands/sym.html`). In case you select symmetries cyclic or dihedral, an additional param will interrogate you about the `Symmetry Order`. Pay attention to the param `Range of distance`, set to `100` by default. This is the distance (in Å) from the center of the input asymmetric unit to the center of additional allowed asymmetric units, in order to select only the closer ones. You should probably modify the default value to regenerate big maps by applying symmetry.

  - `Map fraction around the atomic structure?`: Select the option `Yes` if you want to limit the input minuend `3D Map` to a certain area around the atomic structure. This is the option recommended if you have a big starting map and you'd like to substract a much smaller subtrahend structure-derived map since the visualization of results will be much easier. An additional param, `Atom radius (Å)` asks you about the distance around the input structure used to crop the input `3D Map`. `15` is the default value. The *ChimeraX*-generated map is called `zone_Map`.

* `Other atomic structures`: Additional atomic structures previously downloaded or obtained in *Scipion* can be included here to help you identify particular areas of the map or structure. Then, those structures are only informative and won't be used to generate the subtrahend map.
* `Filter to apply to the differential map`: `Advanced` parameter to clean the background of the differential map by applying a filter in order to maximize the differences between the minuend and the subtrahend maps, since the differential map usually results quite blurry. This `filtered_Map` will always appear together with the `differ-`

ence_Map when the *ChimeraX* graphics window opens. To filter the differential map you can choose between two different filters, Gaussian (with variable width) and based on the Fourier Transform.

- Protocol execution:

  Adding specific volume label is recommended in Run name section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of Run name box, complete the label in the new opened window, press OK, and finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to Restart the Run mode.
  Press the Execute red button at the form bottom.
  After executing the protocol the *ChimeraX* graphics window will open and show the different inputs (maps and atomic structures), as well as the maps generated by the *ChimeraX* commands such as molmap_Map, zone_Map, difference_Map and filtered_Map. Most of the outputs are already saved in *Scipion*, however you can perform any operation of your preference and save the new results before closing *ChimeraX*. Common commands of *ChimeraX-Scipion* communication are allowed in this case: scipionwrite, scipionss and scipioncombine.

- Visualization of protocol results:

  After exiting the protocol, press Analyze Results and the *ChimeraX* graphics window will open with every saved elements, inputs and outputs, which might be distinct acording to the inputs. In addition to items mentioned in the previous paragraph, the atomic structure without the removed residues used as a control, called mutated_Atom_structure will be also shown overlapping the input structure.

  By pressing the left black arrow shown in the Summary Output the saved maps can be also opened with *ShowJ*, the default *Scipion* viewer that shows each map's slices (https://github.com/I2PC/scipion/wiki/ShowJ).

- Summary content:

  - Protocol output (below *Scipion* framework):
    For each map: `chimerax - map subtraction -> ouput map name`;
    `Volume (x, y, and z dimensions, sampling rate)`.
    For each atomic structure: `chimerax - map subtraction -> output atomic structure name`;
    `AtomStruct (pseudoatoms=False, volume=False)`.

  - `SUMMARY` box:
    `Produced files`:
    List of output map names
    `we have some result`

USE CASES

- `Use Case 1: Detection of remnant density in the penton region of the human adenovirus HAdV-F41 density map (EMDB ID EMD-10768, (Pérez-Illana et al., 2020))`
  Aim: Run the *Scipion* workflow depicted in Fig. 78 (A). The output of protocols 1, 2 and 3 can be seen in the *ChimeraX* viewer by pressing `Analyze Results`.

  - In the Fig. 78 (B) appears the whole adenovirus map, output of protocol 1. To visualize this map write in the *ChimeraX* command line:

    ```
    volume #2 region all showOutline false
    ```

    and adjust level densities according to level indicated in the shown `Volume Viewer`.

  - In the Fig. 78 (C) the extracted asymmetric unit is shown, overlapped to the whole map, as output of protocol 2. To visualize these maps, in addition to the previous *ChimeraX* command line and the adjustment of map

levels indicated below, modify the transparency of the whole map writing:

```
volume #2 transparency 0.8
```

– Finally, the Fig. 78 (D) details the atomic structure of the biological asymmetric unit obtained by modeling as output of protocol 3 (`PDB ID 6YBA`). Select `Atoms -> Hide` and `Cartoons -> Show` to change to ribbons the view of the structure. The overlapping of this structure to the geometrical map asymmetric unit allows to observe the area (5, dotted blue circle) where the penton is located and we will try to see a remnant density. To visualize the map, write in the command line:

```
volume #2 transparency 0.9
```

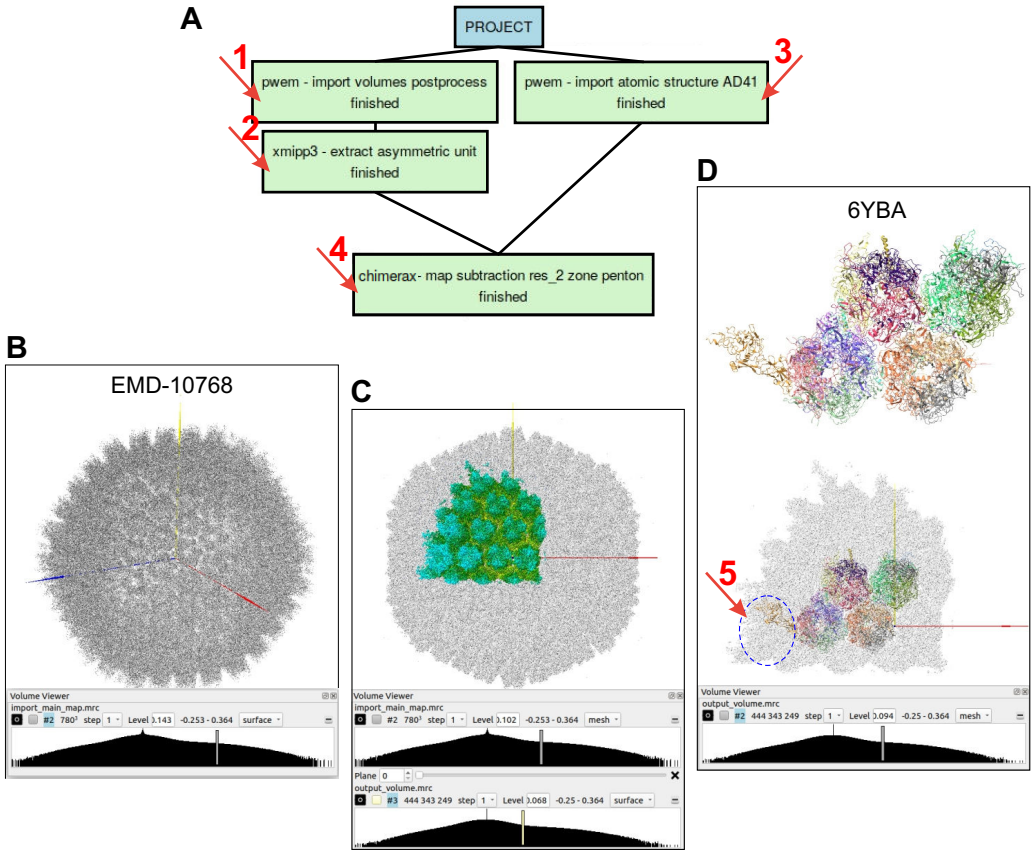


Figure 78: (A) *Scipion* workflow showing protocols 1, 2, 3 and 4. (B) Adenovirus HAdV-F41 map image. (C) Map geometrical asymmetric unit extracted from the adenovirus map. (D) Adenovirus atomic structure of the biological asymmetric unit overlapped to the geometrical map unit.

To look for remnant densities in the penton area we have to complete the chimerax - subtraction protocol with the indicated params (Fig. 79. Remark that in this case we have selected half of input `3D Map` resolution (4 Å) although other values could be tested. The only chain of the penton in the atomic structure of the asymmetric structure is the chain `M`, inside the dotted blue circle of Fig. 78 (D), and 8 residues will be removed as a control of density levels. In addition, icosahedral `I222r` symmetry will be applied to the selected chain in order to complete the five units of the penton. In order to visualize better the

map difference, a map fraction around the atomic structure is selected.

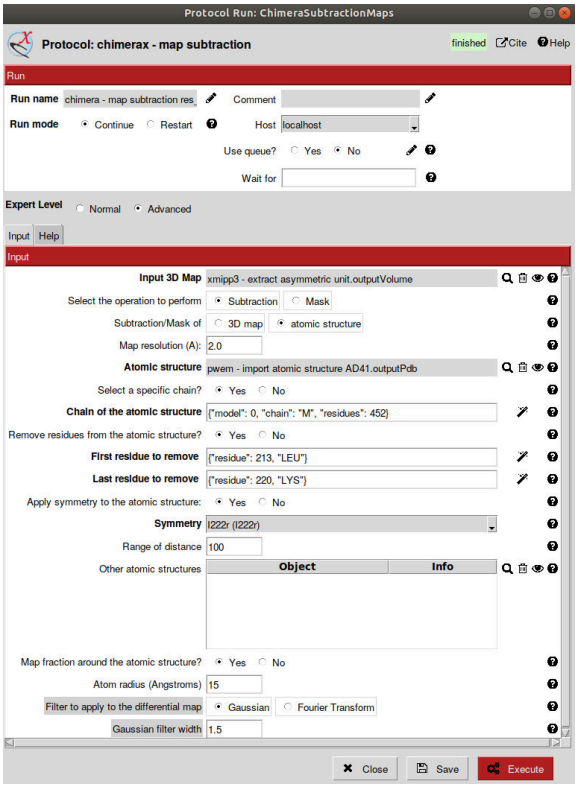


Figure 79: Completing the protocol chimerax - subtraction to find renmant densities in the penton area of the adenovirus map.

Protocol execution: Follow the general procedure shown above (Protocol execution section) and the *ChimeraX* graphics window will open. At this point several maps and atomic structures will be shown, as the `Models` panel indicates (Fig. 80 (C, top)). Have a look to each map and structure to identify them and play with the density levels to maximize the differences between the input `3D Map` restricted to the penton area (`zone_Map`) and the penton atomic structure-derived map (`molmap_chainM_Map`). The Fig. 80 images A and B show the difference `filtered_Map` in the penton side (A) and upper (B) views, respectively, according to the density level observed on the `Volume Viewer` panel (C, middle). Red arrows point at the densities associated to the removed

residues used as a density control. The adjacent ten residues to the removed ones upstream and downstream are green-highlighted. The penton upper view (B) was obtained opening the *ChimeraX* main menu (`Tools -> General -> Side View` and setting the view as indicated (C, botton). From these results we can conclude that a remnant density in the upper part of the penton, if exits, it is not so evident.

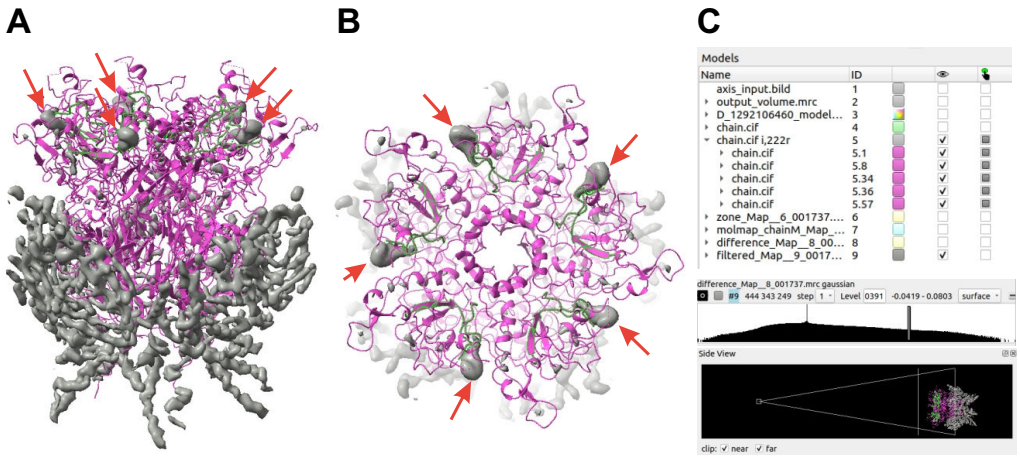


Figure 80: (A) Side view of the adenovirus penton atomic structure (magenta) and the gaussian filtered difference map (grey). (B) Upper view. (C) From top to bottom, *ChimeraX* `Models` panel, `Volume Viewer` panel, specified for the gaussian filtered difference map, and `Side View` panel, respectively.

With the exception of the input adenovirus biological asymmetric unit atomic structure, the rest of elements shown in the *ChimeraX* graphics window will also appear in the *ChimeraX* viewer that opens pressing `Analyze results`. Consider then the possibility of performing additional operations and saving them with the `scipionwrite` command before closing the graphics window. After exiting the protocol you can visualize your results.

- `Use Case 2: Since the asymmetric unit of the human adenovirus HAdV-C5 atomic structure contains a small chain called X (PDB ID 6B1T),`

`we'd like to check if there is a remnant density in the previous human adenovirus HAdV-F41 density map (EMDB ID EMD-10768) that could be interpreted as the HAdV-C5's chain X.`
Aim: Run the *Scipion* workflow depicted in Fig. 81 (A) to inspect for remnant densities around the area covered by the hexons in the biological asymmetric unit area of the adenovirus map (A, 6). The output of all protocols can be seen in the *ChimeraX* viewer by pressing `Analyze Results`.

- In the Fig. 78 (B, C, D) you also have the output of protocols 1, 2 and 3.
- The output of the protocol 4 shows the atomic structure of human adenovirus HAdV-C5. Select `Atoms -> Hide` and `Cartoons -> Show` to change to ribbons the view of the structure. This structure was fitted to the map asymmetric unit of adenovirus HAdV-F41 by using the protocol chimerax - operator (6) and the result of this oputput, overlapped to the geometrical map asymmetric unit, is shown in Fig. 81 (B). To visualize this map write in the *ChimeraX* command line:

  ```
  volume #2 transparency 0.8
  ```

  and adjust level densities according to level indicated in the shown `Volume Viewer` (Fig. 81 (D)). Select `Atoms -> Hide` and `Cartoons -> Show` to change to ribbons the view of the structure.
- The output of protocol 5 details some small chains of `ALA` residues previously traced in the remnant density of the adenovirus HAdV-F41. They are used as a control to be sure that we identify new densities previously unmodeled. Since they are very small we have depicted them selecting `Styles -> Stick` and overlapped to the geometrical map asymmetric unit (Fig. 81 (C) with the same transparency and map adjustment shown in (B).

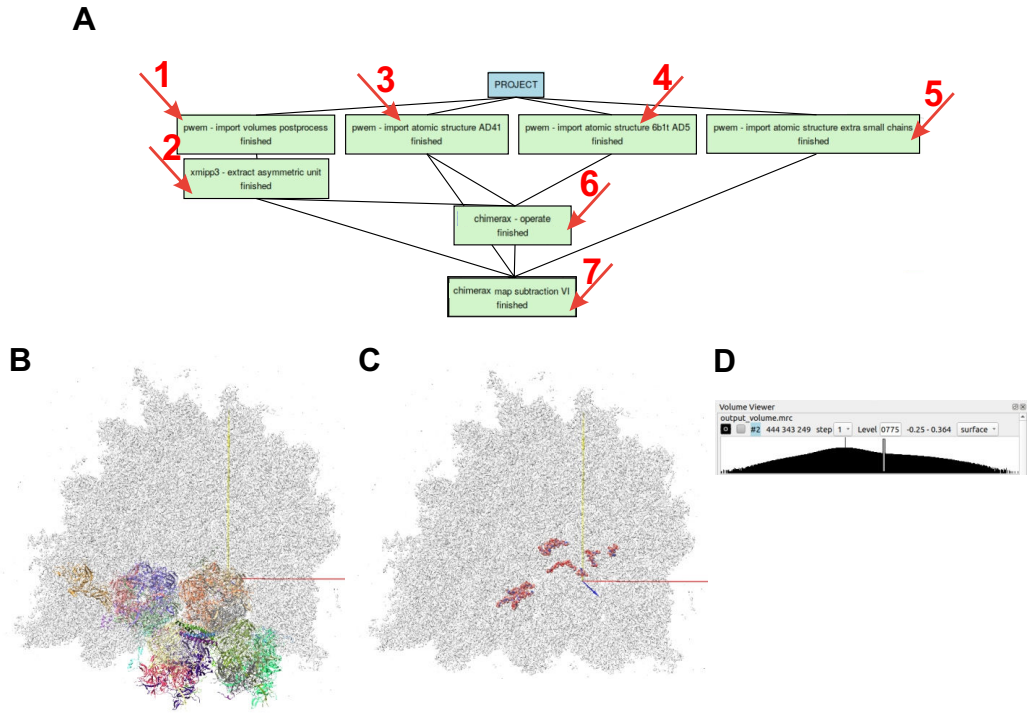


Figure 81: (A) *Scipion* workflow showing protocols 1-7. (B) HAdV-F41 adenovirus geometrical map asymmetric unit (grey) and, fitted to it, the atomic structure of the biological asymmetric unit atomic structure of HAdV-C5 adenovirus (colored). (C) HAdV-F41 adenovirus geometrical map asymmetric unit (grey) and some small aminoacid chains previously traced in the remnant density, imported in the protocol 5 (colored). (D) Level of density selected to visualize the map in B and C.

To look for remnant densities in the area of hexons we have to complete the chimerax - subtraction protocol with the indicated params (Fig. 82. As in the previous use case, we have selected half of input `3D Map` resolution (4 Å) although other values could be tested.

Taking into account that the remnant densitities could be quite inconspicuous we are going to use two different controls this time. One of them will be, as in the previous use case, the deletion of 5 residues of hexon chain `D`, in a region presumed to be quite close to the remnant density that we are looking for. The

second control will be some small aminoacid chains previously traced in the remnant density since we'd like to discriminate between this density and other new one and unmodeled. These extra small chains have to be included in the form param `Other atomic structures`.

Although this time we do not have to consider a specific chain or applying symmetry, since we have to look for a chain similar to HAdV-C5 adenovirus chain `X`, it is quite recommendable to include in the form param `Other atomic structures` the structure 6B1T fitted to the geometrical map asymmetric unit, as shown in Fig. 81 (B), and obtained from protocol 6 (A).

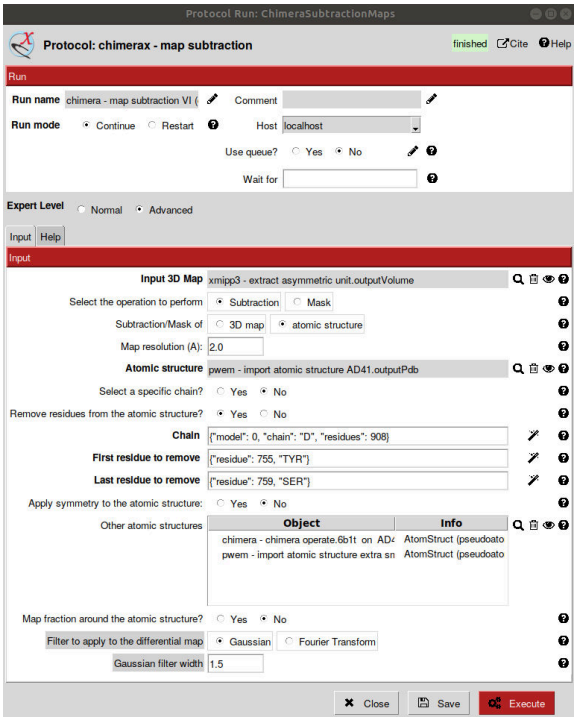


Figure 82: Completing the protocol chimerax - subtraction to find renmant densities in the biological asymmetric unit area of the adenovirus map.

Protocol execution: Follow the general procedure shown above (Protocol execution section) and the *ChimeraX* graphics window will open. At this point several maps and atomic structures will be shown, as the `Models` panel indi-

cates (Fig. 83 (A, bottom)), except the *model* `#6`. Have a look to each map and structure to identify them and play with the density levels to maximize the differences between the input `3D Map` (`output_volume.mrc`) and the `6YBA` atomic structure-derived map (`molmap_Map`). The Fig. 83 (A) show the difference `filtered_Map` obtained. The zoom in on the framed area displays in detail the difference considering two different map levels (B and C). To have this view, besides select the molecules to show according to the `Models` panel (A, bottom), write in *ChimeraX* command line:

```
volume #2 transparency 0.8
select #4/X
save /tmp/6b1t_chainX.cif format mmcif models #4 selectedOnly true
open /tmp/6b1t_chainX.cif
```

HAdV-C5 adenovirus chain `X` can be visualized as *model* `#6`.

The result, described in Fig. 83, doesn't demonstrate a clear continuous density in the proximity of the HAdV-C5 adenovirus chain `X`. Although not very evident, it could be there. Then we cannot conclude that it doesn't exit, only that we were unable to identify it.

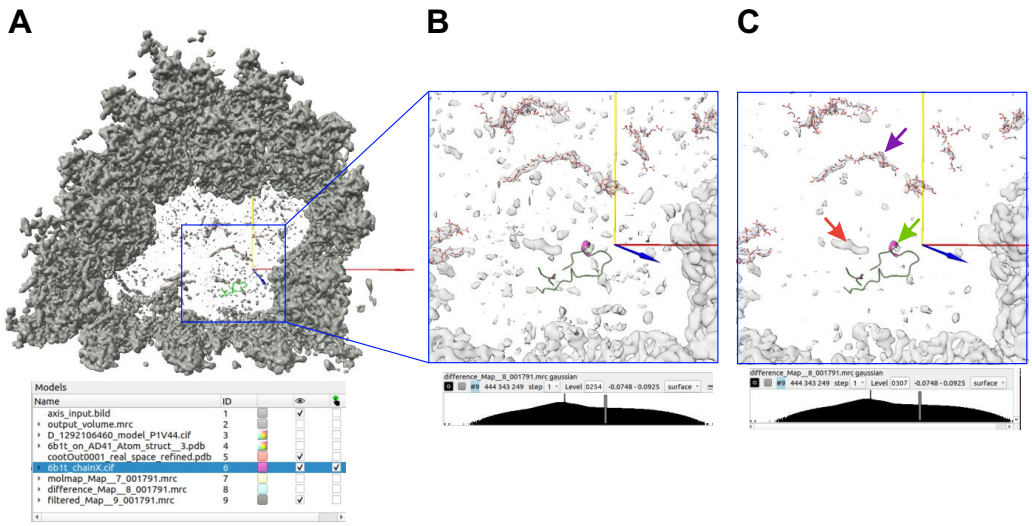


Figure 83: (A) Gaussian filtered difference map (grey) of adenovirus HAdV-F41 asymmetric unit (top) and `Models` panel of items loaded in *ChimeraX* including the *model* `#6` (bottom). (B) Zoom in on the subtracted area with the map density level indicated in the `Volume Viewer` below . (C) Idem with a higher cleaning of the background. The red arrow points at the control density. The green arrow points at the HAdV-C5 adenovirus chain `X`. The purple arrow points at one of the adenovirus HAdV-F41 small chains previously traced.

5 ChimeraX Operate protocol

Protocol designed to perform operations with atomic structures in *Scipion* by using *ChimeraX*. A volume or set of volumes can also be included. Structures or maps generated by using this protocol can be saved in *Scipion* after executing specific *ChimeraX* commands. *ChimeraX* `rigid fit` protocol constitutes a particular case of this protocol to perform rigid fitting in *Scipion* by using *ChimeraX* (Appendix 7).

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`

  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu:
  `Model building -> Tools-Calculators` (Fig. 84 (A))

- Protocol form parameters (Fig. 84 (B)):

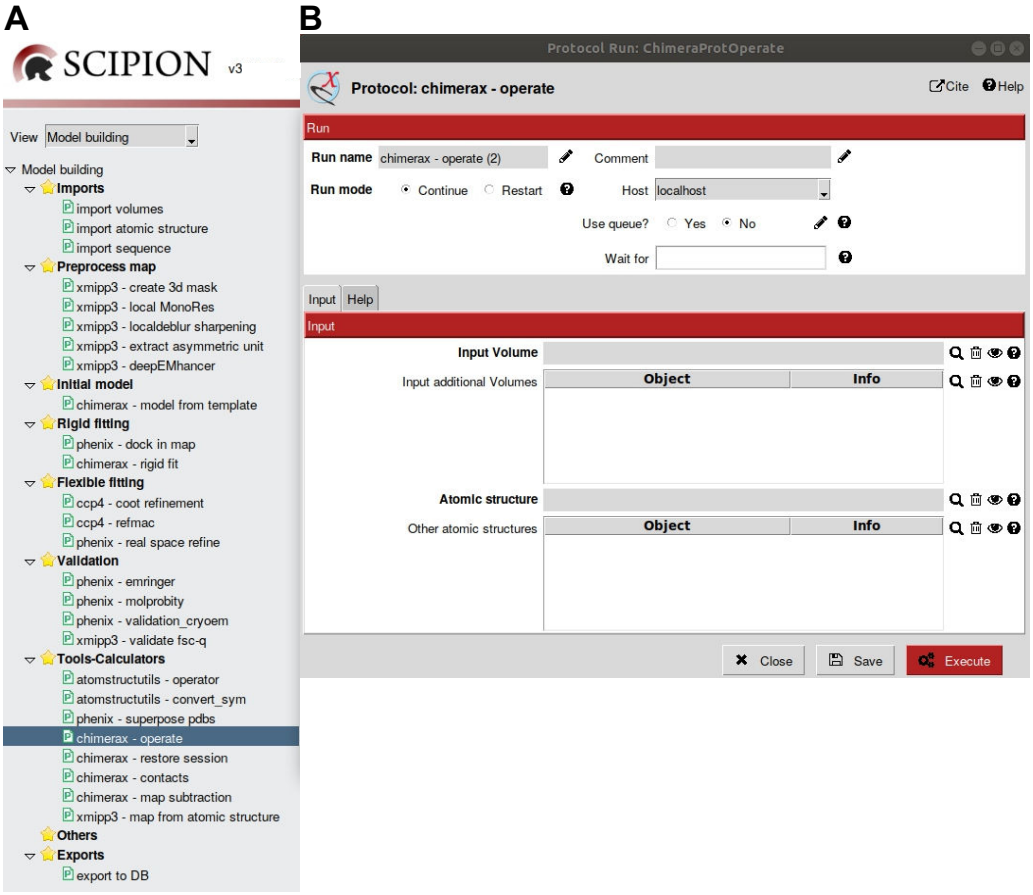


Figure 84: Protocol chimerax - operate. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input` section

    * `Input Volume`: Optional parameter to be completed with the electron density map previously downloaded or generated in *Scipion*.

    * `Input additional Volumes`: Optional parameter to include other electron density maps previously downloaded or generated in *Scipion*.
    * `Atomic structure`: Atomic structure previously downloaded or generated in *Scipion*.
    * `Other atomic structures`: Additional atomic structures.

  – `Help` section

    This section contains *ChimeraX* commands required to save *models* according to their reference volumes, which can also be saved if required. Remark that using `scipionwrite` command, *ChimeraX* session will be saved by default, without prejudice that it may be saved with `scipionss` command. `scipioncombine` command allows to merge in only one atomic structure two or more. *ChimeraX* sessions can be restored by using chimerax - restore session protocol.

- Protocol execution:

  Adding specific protocol label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

  ChimeraX graphics window will be opened after executing the protocol. Electron density map(s), if loaded, and the atomic structure(s) are shown. Steps to follow depend on the specific operation to carry out. Usually, new volumes or structures are generated, sometimes by combination of others, each one with a specific *model* number displayed in the `Models` panel, and they have to be saved in *Scipion*.

  - To combine two or more atomic structures:
    Write in *ChimeraX* command line:
    `scipioncombine #n1,n2`

    `#n1` and `#n2` are the respective *model* numbers of two different atomic structures. Optionally you can set the *model* number of the output combined structure `#n3`:
    `scipioncombine #n1,n2 modelid n3`

  - To save a map or an atomic structure generated with this *ChimeraX* protocol with *model* number `#n`:
    Write in *ChimeraX* command line:
    `scipionwrite #n`

    Optionally you can write a prefix to easily recognize that map or structure. Then, the prefix depends on the user. Example:
    `scipionwrite #n prefix my_favorite_model_`

  - Close *ChimeraX* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structures and electron density volumes, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, volume, and atomic structure are model numbers `#1`, `#2`, and `#3`, respectively, in *ChimeraX* `Models` panel. If no volumes have been included, coordinate axes and each atomic structure are model numbers `#1` and `#2`, respectively.

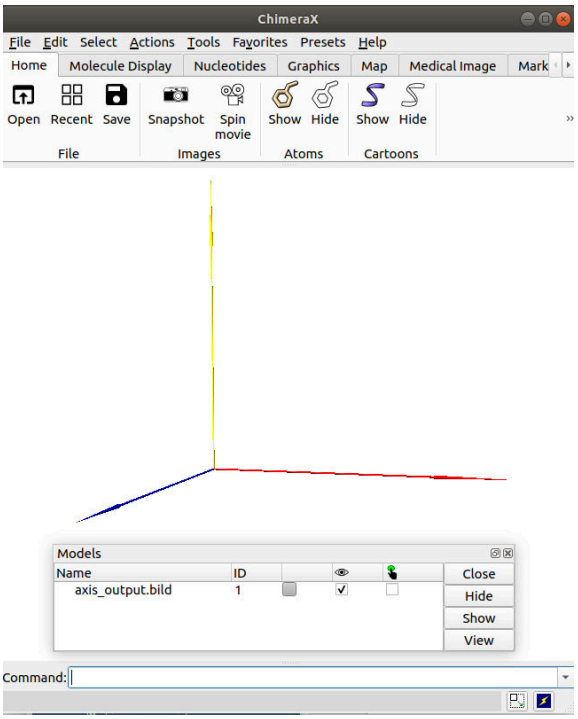


Figure 85: Default *ChimeraX* graphics window with coordinate axes.

- Summary content:

  - If an atomic structure is generated:

    * Protocol output (below *Scipion* framework):
      `chimerax - operate -> output atomic structure name, starting with the prefix;`
      `AtomStruct (pseudoatoms=True/ False, volume=True/ False).`
      Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
    * `SUMMARY` box:
      Produced files:

output atomic structure name, starting with the prefix (.cif file)
we have some result

– If a volume is generated:

* Protocol output (below *Scipion* framework):
`chimerax - operate -> output 3D map name`; `Volume (x, y, and z dimensions, sampling rate)`.

* `SUMMARY` box:
Produced files:
output 3D map name, starting with the prefix (.mrc file)
we have some result

6 ChimeraX Restore Session protocol

Protocol designed to restore *ChimeraX* session, provided that this session has been saved previously in *Scipion*. Currently, four protocols save *ChimeraX* sessions when *ChimeraX* commands `scipionwrite`, `scipionss` or `scipioncombine` are used, chimerax - rigid fit, chimerax - operate, chimerax - model from template and chimerax - map subtraction (Appendices 7, 5, 19 and 4, respectively). Restored sessions allow inspect any element contained in a previously saved *ChimeraX* session, perform *ChimeraX* operations, and finally save maps or atomic structures.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
`Model building -> Tools-Calculators` (Fig. 86 (A))
- Protocol form parameters (Fig. 86 (B)):

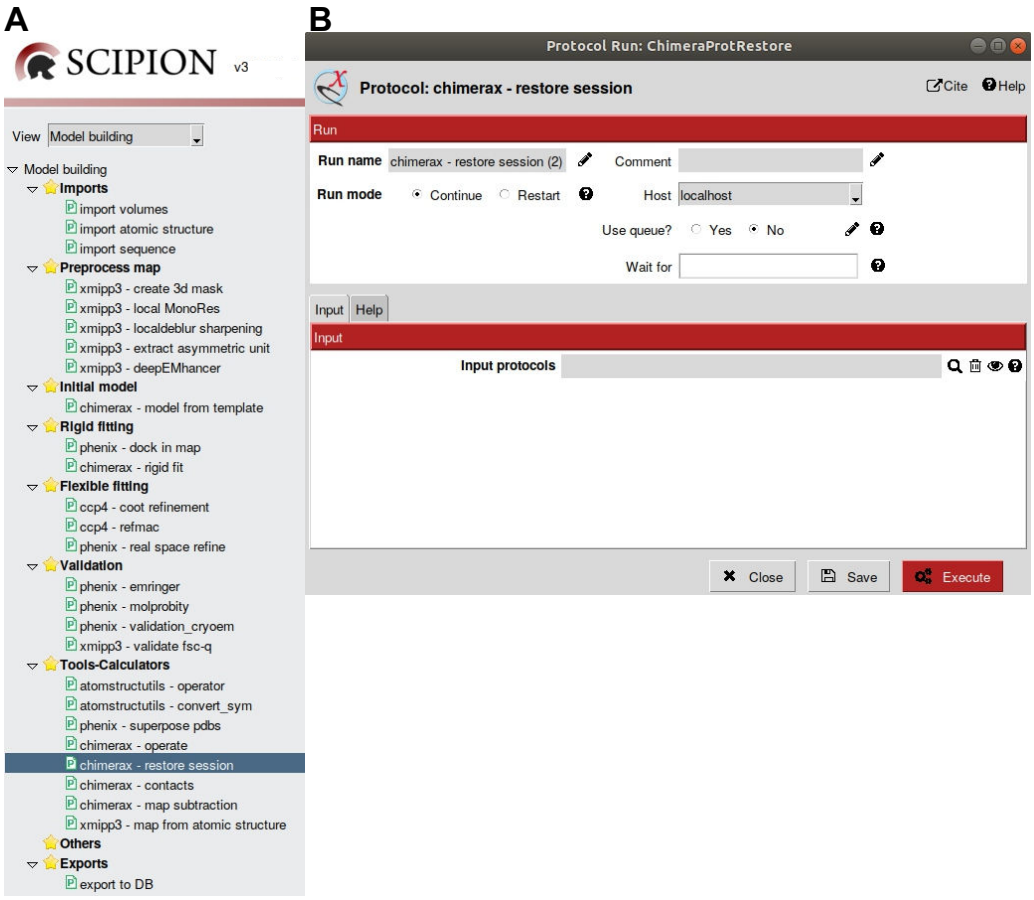


Figure 86: Protocol chimerax - restore session. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Input protocols`: Parameter that allows to select a particular protocol in which *ChimeraX* session has been saved in *Scipion*. As it was mentioned before, four protocols support this possibility (*ChimeraX* `rigid fit`, *ChimeraX* `operate`, *ChimeraX* `model from template` and *ChimeraX* `map subtraction`).
- `Help` section

  This section contains *ChimeraX* commands required to save *models* according to their reference volumes, which can also be saved if required.

Remark that using `scipionwrite` command, *ChimeraX* session will be saved by default, without prejudice that it may be saved with `scipionss` command. *ChimeraX* sessions can be restored again by using this same chimerax - restore session protocol.

- Protocol execution:

  Adding specific protocol label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

  ChimeraX graphics window will be opened after executing the protocol showing the complete list of elements that appeared in *ChimeraX* graphics window when the session was saved, coordinate axes, electron density maps, and atomic structures. Steps to follow depend on the specific operation to carry out. New volumes or structures may be generated as usual in *ChimeraX*, and they can be combined or saved in *Scipion* in the common way.

  – To combine two or more atomic structures:
    Write in *ChimeraX* command line:
    `scipioncombine #n1,n2`

    `#n1` and `#n2` are the respective *model* numbers of two different atomic structures. Optionally you can set the *model* number of the output combined structure `#n3`:
    `scipioncombine #n1,n2 modelid n3`

  – To save a map or an atomic structure generated with this protocol: Write in *ChimeraX* command line:

`scipionwrite #n prefix userString_`.
Replace `#n` by model numbers shown in *ChimeraX* `Models` panel. `prefix` + string preferred by the user to easily identify the atomic structure is optional, although quite recomendable.
Replace `#n` by model numbers shown in *ChimeraX* `Models` panel.

- Close *ChimeraX* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structures and electron density volumes, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). In this particular case a *ChimeraX* graphics window identical to the input session will be opened and it will include every element saved lately.

- Summary content:

  - If an atomic structure is generated:

    * Protocol output (below *Scipion* framework):
      `chimerax - operate -> output atomic structure name, starting with the prefix`;
      `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
      Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
    * `SUMMARY` box:
      Produced files:
      output atomic structure name, starting with the prefix (.cif file)
      we have some result

  - If a volume is generated:

* Protocol output (below *Scipion* framework):
  `chimerax - operate -> output 3D map name`; `Volume (x, y, and z dimensions, sampling rate)`.

* `SUMMARY` box:
  Produced files:
  output 3D map name, starting with the prefix (.mrc file)
  we have some result

7 ChimeraX Rigid Fit protocol

Protocol designed to manually fit atomic structures to electron density maps in *Scipion* by using *ChimeraX*. If *map* and *model* are quite close, e.g. after running *PHENIX* `dock in map` protocol, automatic fitting is also possible. Fitted atomic structures generated by using this protocol can be saved in *Scipion* after executing specific *ChimeraX* commands.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Rigid fitting` (Fig. 87 (A))
- Protocol form parameters (Fig. 87 (B)):

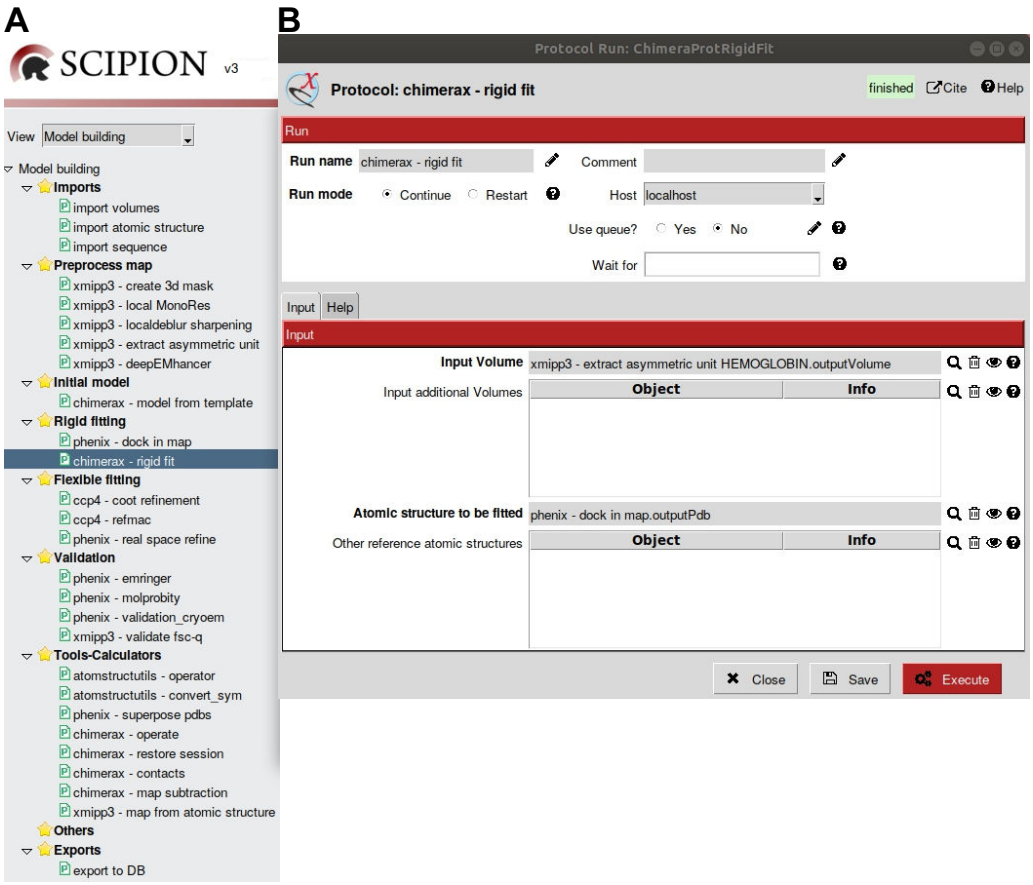


Figure 87: Protocol chimerax - rigid fit. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Input Volume`: Mandatory param to load the electron density *map* previously downloaded or generated in *Scipion* to fit the atomic structure.
  - `Input Volume`: *Idem*. If additional maps, others than the previous *map*, are needed.
  - `Atomic structure to be fitted`: Mandatory param to load the atomic structure previously downloaded or generated in *Scipion* to be fitted to an electron density map.

    * `Other reference atomic structures`: Atomic structures others than the *model* that can help in the rigid body fitting process.

  – `Help` section

    This section contains *ChimeraX* commands required to save *models* according to their reference volumes, which can also be saved if required. Remark that using `scipionwrite` command, *ChimeraX* session will be saved by default, without prejudice that it may be saved with `scipionss` command. *ChimeraX* sessions can be restored by using chimerax - restore session protocol. In addition `scipioncombine` allows to combine several atomic structures in a unique *model*.

- Protocol execution:

  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

  ChimeraX graphics window will be opened after executing the protocol. The electron density map and the atomic structure are shown. Main steps to complete the rigid body fitting are:

  – If density map and atomic structure are quite close to each other:
    Go to *ChimeraX* main menu and select `Tools -> Volume Data-> Fit in Map`. A small `Fit in Map` window will be opened. Once the right atomic structure and the electron density volume have been selected, fit them by clicking `Fit`.
    `Note`: The same result can be obtained by typing in the command line `fit #n1 inMap #n2`, with `#n1` and `#n2` *ChimeraX* model numbers of *model* and *map*.

  - If *model* and *map* are far from each other, start the fitting process interactively activating and inactivating *ChimeraX* objects alternatively to finally get *map* and *model* close enough to go to the previous step. Otherwise, consider the possibility of running before the *PHENIX* `dock in map` protocol.
  - To combine two or more atomic structures:
    Write in *ChimeraX* command line:
    `scipioncombine #n1,n2`

    `#n1` and `#n2` are the respective *model* numbers of two different atomic structures. Optionally you can set the *model* number of the output combined structure `#n3`:
    `scipioncombine #n1,n2 modelid n3`

  - Save fitted *model* regarding the *map* by writing in *ChimeraX* command line:
    `scipionwrite #n prefix userString_`.
    Replace `#n` by model numbers shown in *ChimeraX* `Models` panel. `prefix` + string preferred by the user to easily identify the atomic structure is optional, although quite recomendable.
  - Close *ChimeraX* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of *model* and *map*, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, map, and atomic structure are model numbers `#1`, `#2`, and `#3`, respectively, in *ChimeraX* `Models` panel in the most simple case.

- Summary content:

- If only the atomic structure has been saved by `scipionwrite` command:
  - Protocol output (below *Scipion* framework):
    `chimerax - operate -> output atomic structure name, starting with the prefix`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    Produced files:
    output atomic structure name, starting with the prefix (.cif file)
    we have some result
- If both the atomic structure and its reference electron density map have been saved by `scipionwrite` command:
  - Protocol output (below *Scipion* framework):
    `chimerax - rigid fit -> Map_name`; `Volume (x, y, and z dimensions, sampling rate)`.
    `chimerax - rigid fit -> Atom_struct_name`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    Produced files:
    Map_name file, starting with the prefix (.mrc)
    we have some result

8 CCP4 Coot Refinement protocol

Protocol designed to interactively fit and refine atomic structures, in real space, regarding electron density maps in *Scipion* by using *Coot* (Emsley et al., 2010). This

protocol integrates *Coot* 3D graphics display functionality in *Scipion*, supporting accession to *Coot* input and output data in the general model building workflow. *Coot*, acronym of Crystallopgraphic Object-Oriented Toolkit, gathers several tools useful to perform mostly interactive modeling procedures and is integrated in CCP4 software suite (`www.ccp4.ac.uk/ccp4\_projects.php`). Initially applicable to X-ray data, some modifications of *Coot* also allow to model atomic structures regarding electron density maps obtained from cryo-EM ((Brown et al., 2015)). Additional instructions to use *Coot* can be found in `https://www2.mrc-lmb.cam.ac.uk/personal/pemsley/coot/`. Remark in `https://www2.mrc-lmb.cam.ac.uk/personal/pemsley/coot/web/docs/coot.html#Mousing-and-Keyboarding` mouse requirements to get the *Coot* best functioning.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite (from version 7.0.056 to 7.1)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Flexible fitting` (Fig. 88 (A))
- Protocol form parameters (Fig. 88 (B)):

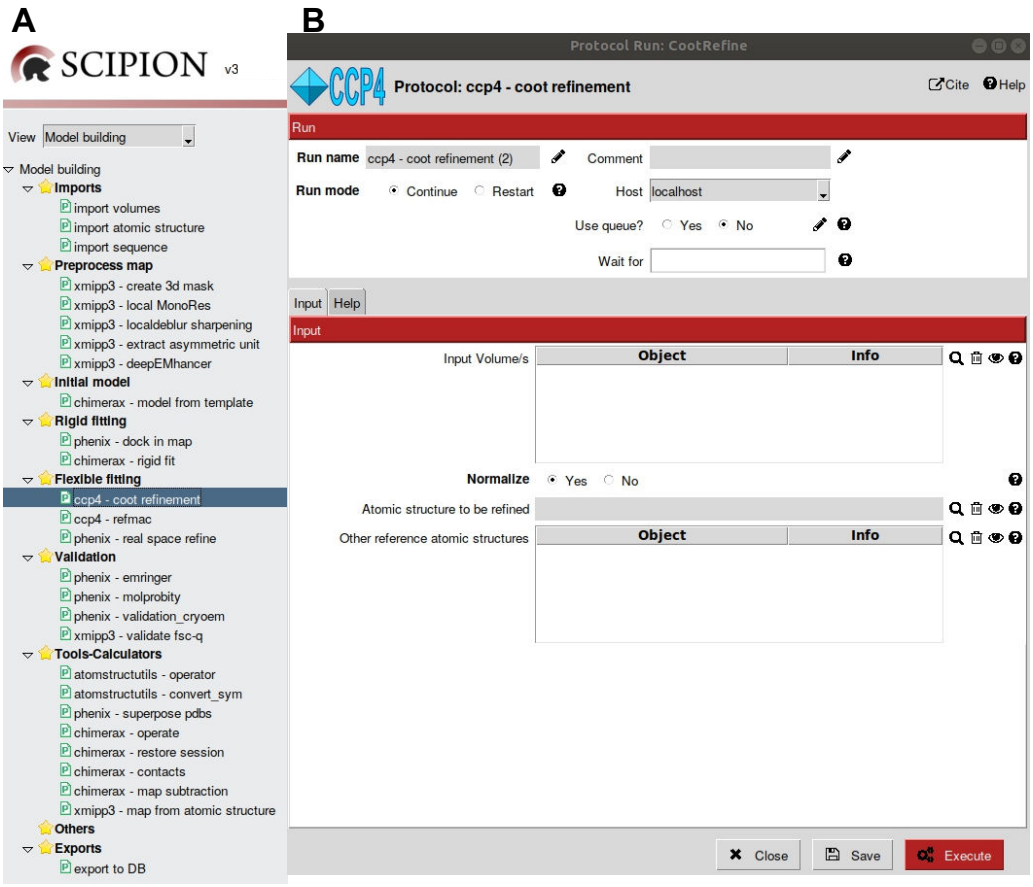


Figure 88: Protocol ccp4 - coot refinement. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Input Volume/s`: At least one or several electron density maps previously downloaded or generated in *Scipion*. The density volume regarding to which an atomic structure has to be modeled has to be included in this volume list.
  - `Normalize`: Parameter set to "Yes" by default to perform normalization of map electron density levels according to *Coot* requirements ([0, 1]). This normalization approximates cryo-EM density data to maps obtained from X-ray crystallography because it diminishes Z-score

(number of standard deviations) variation of map values.

* `Atomic structure to be refined`: Mandatory param to load an atomic structure previously downloaded or generated in *Scipion*. This structure will be fitted and refined according to a particular density volume.
* `Other reference atomic structures`: Additional atomic structures previously downloaded or generated in *Scipion* that may be helpful in the refinement process.

– `Help` section

This section contains *Coot* commands to make easier some interactive refinement steps and to save refined atomic structures. Their reference volumes will be saved by default with the refined atomic structures. Here you are an overview of these commands:

* Automatically moving from one chain to another in an atomic structure:
  · Press ‘‘`x`’’ in the keyboard to move from one chain to the previous one.
  · Press ‘‘`X`’’ to change from one chain to the next one.
* Initializing global variables:
  Press ‘‘`U`’’ in your keyboard.
* Semi-automatic refinement of small groups of residues (10 to 15):
  As soon as *Coot* protocol is executed, the text file `coot.ini` will be saved in the project folder `/Runs/00XXXX_CootRefine/extra/` (Fig. 93 (1, 2)). This file content has to be modified according to our atomic structure model in this way:
  · `imol`: `#0` has to be replaced by the number of the molecule that has to be refined. This number appears detailed in *Coot* main menu `Display Manager` (Fig. 89 (B, red arrow)).
  · `aa_main_chain`: `A` has to be replaced by the name of the molecule chain that has to be refined.

- `aa_auxiliary_chain`: `AA`, name of the small chain of 10-15 residues, can be optionally replaced by other name.
- `aaNumber`: `#100` has to be replaced by the position of the residue from which the refinement has to start.
- `step`: `#10` will be replaced by the desired small step of residues that gets flexible enough to select other conformation of this auxiliary chain.

Save `coot.ini` text file after its modification. Go to the residue position indicated in `aaNumber`, initialize global variables with ‘‘`U`’’, and pres ‘‘`z`’’ or ‘‘`Z`’’ in the keyboard to refine those `aaNumber` residues upstream or downstream, respectively.

* Printing *Coot* environment:
  Press ‘‘`E`’’ in the keyboard.
* Saving an atomic structure after an interactive working session with *Coot*:
  `Coot Python Scripting` window will be opened with *Coot* main menu `Calculate -> Scripting... -> Python...` (Fig. 89 (A)). By writing `scipion_write()`, molecule `#0` will be saved by default in *Scipion*. Molecule number can be checked in *Coot* main menu `Display Manager` (Fig. 89 (B, red arrow)). Saving the molecule this way is equivalent to press ‘‘`w`’’ in the keyboard.
  The number `#n` of the specific molecule has to be written in brackets to save any other molecule than `#0`.
  Although the name of the saved atomic structure is `coot_XXXXXX_Imol_YYYY_version_ZZZZ.pdb` by default (`XXXXXX` is the protocol box number, `YYYY` the *model* number and `ZZZZ` the number of times that the molecule has been saved), other names/labels of your preference are also allowed. That name/label has to be introduced with `scipion_write()` command, as it is detailed in the example (Fig. 89 (A)). The addition of `.pdb` extension is not required.
  If no more interactive sessions with *Coot* are planned, after saving the

atomic structure, *Coot* will be definitively closed by pressing ‘‘e’’ in the keyboard.

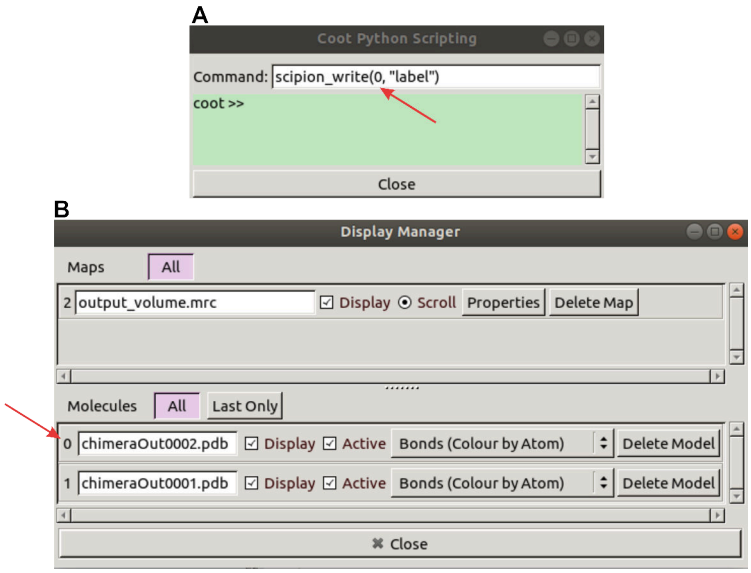


Figure 89: Protocol ccp4 - coot refinement. A: Saving labeled atomic structure with `Coot Python Scripting` window. B: `Display Manager` window.

- Protocol execution:

  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`. However, if you want to restart the protocol in the last point that you let it before and continue working with the last file saved in *Coot*, set to `Continue` the `Run mode`.
  Press the `Execute` red button at the form bottom.

  Coot graphics window will be opened after executing the protocol. Electron density maps and atomic structures are shown. Although steps to follow depend

on the specific operation to carry out, a list of basic initial tasks and tools could be helpful:

- Check maps and atomic structures definitively loaded in *Coot*:
  By opening `Display Manager` window (*Coot* main menu) (Fig. 89 (B)).
- Set parameters appropriate to visualize them:
  Electron density maps are sometimes more difficult to visualize. Moving mouse scroll-wheel forward and backward increases or reduces, respectively, map contour level. If the volume is still invisible, check if map and atomic structures are properly fitted. The radius of the density sphere can be modified in *Coot* main menu `Edit -> Map Parameters ... -> Global map properties window`.
- Check chain names of each atomic structure, and edit them if needed in *Coot* main menu `Edit -> Change Chain IDs...`.
- Set the text file `coot.ini` (Fig. 93 (2)), edit it and save it if needed.
- Set refinement conditions:
  Click `Refine/Regularize control` button (upper right side of *Coot* graphics window) (Fig. 90 (1)) and select the four restriction types in `Refinement and regularization Parameters` window (2).

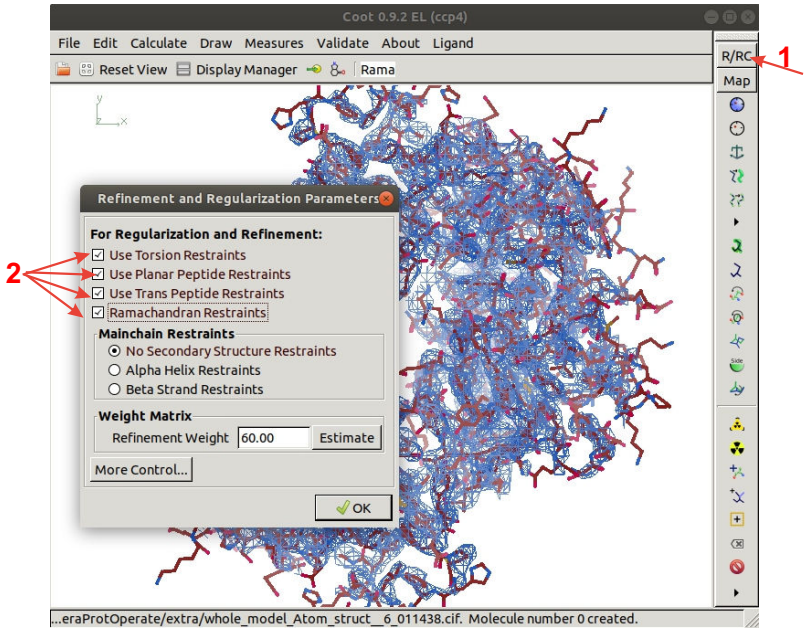


Figure 90: Protocol ccp4 - coot refinement. `Refinement and regularization Parameters` window.

Once those basic parameters are set, some steps to follow in refinement process are:

- Check validation parameter windows to have an idea of controversial areas and quality of the fitting:
  Go to *Coot* main menu `Validation -> Ramachandran Plot`, `Validation -> Density fit analysis` and `Validate -> Rotamer analysis`. Validation windows have to be checked throughout the refinement process.
- Refine the ends of each chain. Basic interactive refinement process requires several steps:
  - First, go to an atom included in the area that is going to be refined: Go to *Coot* main menu `Draw -> Go To Atom...` and select chain and atom.
  - Assess electron density in that area, and consider the possibility of processing part of the residues.

* Click the button `Real Space Refine Zone` (upper right side of *Coot* graphics window) (Fig. 91 (A) (1)) to put it active. Next, click two residues of the chain (2 and 3). A second flexible grey chain overlaps the starting chain. That grey chain can be moved in order to get a different conformation according to the density map (hidden in Fig. 91 (A)).
* If refinement parameters get acceptable values, press `Accept` in `Accept Refinement?` window (Fig. 91 (B)).

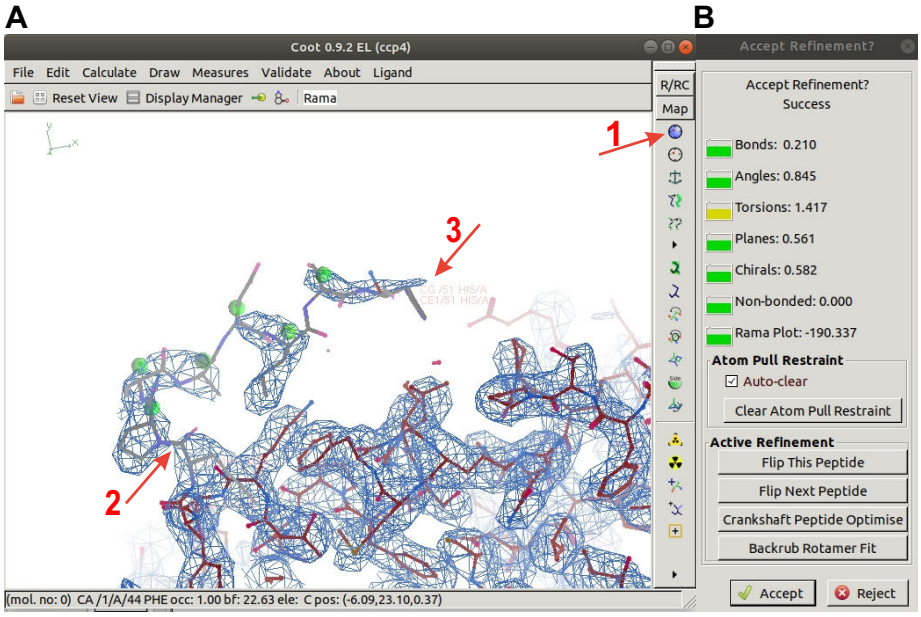


Figure 91: Protocol ccp4 - coot refinement. (A) Interactive refinement of the chain fragment between residues 2 and 9. (B) Accepting refinement window.

– Refine each chain following instructions from Help section:

  * Go to the residue `aaNumber` (*Coot* main menu `Draw -> Go To Atom...`).
  * Initialize global variables.
  * Repeat this loop until reaching the end of the chain:

    1.- Press ‘‘`z`’’ in the keyboard.

2.- Inspect one by one, and fit to the volume density, every residue from the small auxiliary chain.
3.- Accept the refinement.

* Check validation parameters to focus refinement in specific chain areas (*Coot* main menu `Validate -> Density fit analysis`).

– After finishing refinement of every chain, save the structure (press ‘‘`e`’’ if *Coot* has to be definitively closed and not interactive anymore).

– Close *Coot* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structures and electron density volumes, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, volume, and first atomic structure are model numbers `#1`, `#2`, `#3`, respectively, in *ChimeraX* `Model Panel`. Every atomic structure saved during *Coot* refinement process will appear in `Model Panel` (Fig. 92). If you want to visualize results in *Coot* graphics window you only have to open the protocol in the last point that you let it before and set to `Continue` the `Run mode`. Close the *Coot* protocol without saving anything in this case.

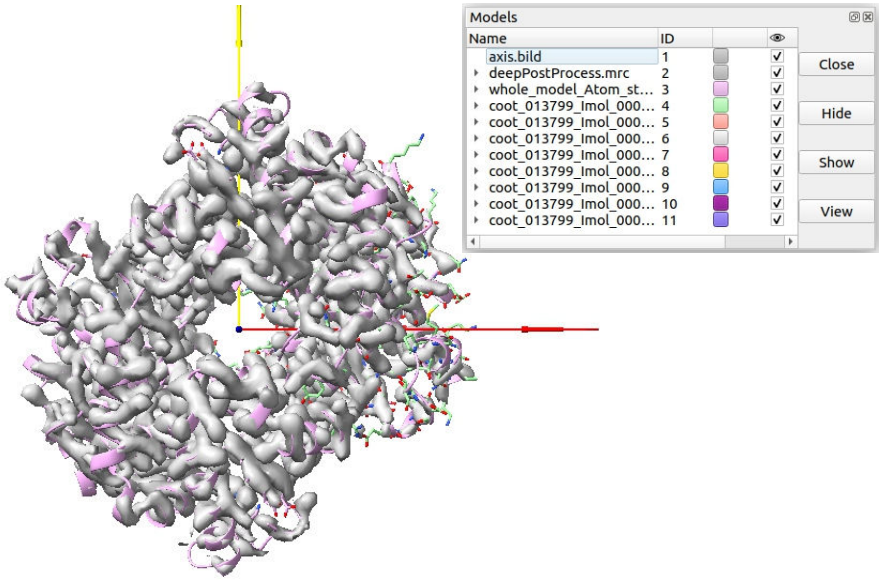


Figure 92: Protocol ccp4 - coot refinement. *Coot* results visualized in *ChimeraX*.

Since *Scipion* projects keep every intermediate atomic structure partially refined (Fig. 93 (1, 3), users can include any of them in successive following modeling workflow steps performed in *Scipion* (Fig. 94).

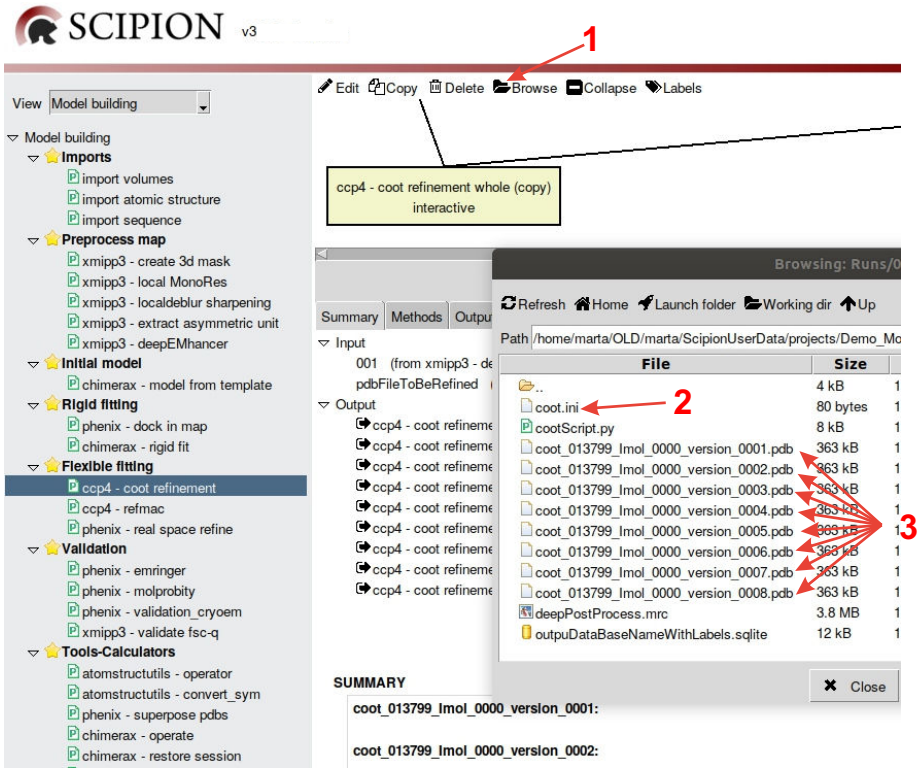


Figure 93: Protocol ccp4 - coot refinement. Browse content after several runs of interactive *Coot* protocol.

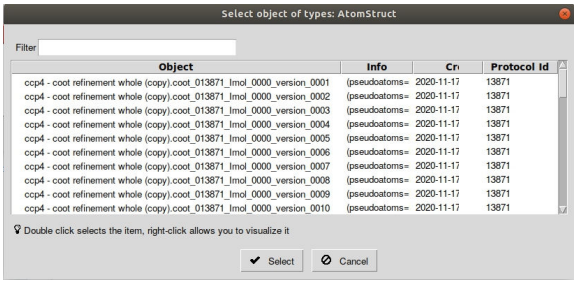


Figure 94: Protocol ccp4 - coot refinement. *Scipion* window that allows to select any of *Coot* partially refined structures.

- Summary content:

- Protocol output (below *Scipion* framework):
  - Each *Coot* intermediate atomic structure partially refined (`#n`):
    `ccp4 - coot refinement ->` label name selected by the user or common output *Coot* name (`coot_XXXXXX_Imol_YYYY_version_ZZZZ.pdb`);
    `AtomStruct (pseudoatoms=False, volume=False)`.
    Pseudoatoms is set to `False` because the structure is made of atoms instead of pseudoatoms. Volume is set to `False` because no electron density map is associated to the atomic structure.
  - Each *Coot* input map (saved by default):
    `ccp4 - coot refinement -> output3DMap_XXXX`;
    `Volume (x, y, and z dimensions, sampling rate)`.
  - `SUMMARY` box for each *Coot* intermediate atomic structure partially refined:
    label name selected by the user or common output *Coot* name (`coot_-XXXXXX_Imol_YYYY_version_ZZZZ.pdb`)
    Idem for maps.

9 CCP4 Refmac protocol

Protocol designed to refine atomic structures, in reciprocal space, regarding electron density maps in *Scipion* by using *Refmac* (Vagin et al., 2004), (Kovalevskiy et al., 2018). This protocol integrates *Refmac* functionality in *Scipion*, supporting accession to *Refmac* input and output data in the general model building workflow.
Refmac, Refinement of Macromolecular Structures by the Maximum-Likelihood method, allows the refinement of atomic models against experimental data, and is integrated in CCP4 software suite (`www.ccp4.ac.uk/ccp4_projects.php`). Initially applicable to X-ray data, some modifications of *Refmac* also support optimal fitting of atomic structures into electron density maps obtained from cryo-EM (Brown et al., 2015). Particullarly, *Refmac* considers a five-Gaussian approximation for electron scattering factors because, unlike of X-rays crystallography, cryo-EM scattering is modified

by each atom electric charge and ionization state. In addition, *Refmac* computes structure factors only for the model-explained part of the map. These structure factors are complex because they include, not only amplitude data, but also phase information. *Refmac* will try to minimize the difference between the "observed" and calculated structure factors, computed from cryo-EM maps and from atom coordinates (structure), respectively. Additional instructions to use *Refmac* can be found in `http://www.ysbl.york.ac.uk/refmac/`.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite (from version 7.0.056 to 7.1)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Flexible fitting` (Fig. 95 (A))
- Protocol form parameters (Fig. 95 (B)):

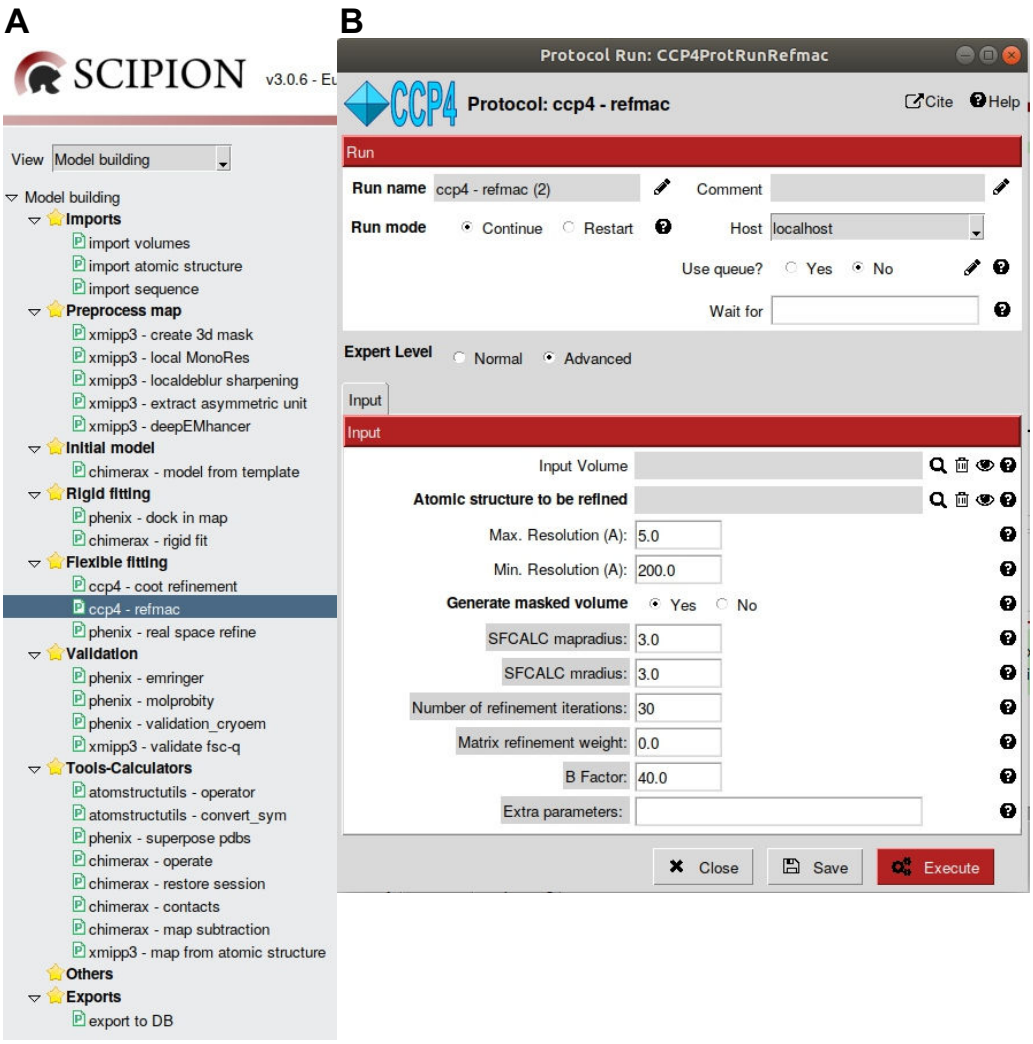


Figure 95: Protocol ccp4 - refmac. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input Volume/s`: An electron density map previously downloaded or generated in *Scipion*. An atomic structure should be refined regarding to this volume.
- `Atomic structure to be refined`: Atomic structure previously downloaded or generated in *Scipion*. This structure will be refined according to the electron density volume.

- `Max. Resolution (Å)`: Upper limit of resolution used for refinement, in Angstroms. Using double value of sampling rate is recommendable.
- `Min. Resolution (Å)`: Lower limit of resolution used for refinement, in Angstroms.
- `Generate masked volume`: Parameter set to "Yes" by default. With this option, structure factors will be computed for the map around model atomic structure. Otherwise (option "No"), structure factors will be computed for the whole map.
- `SFCALC mapradius`: Advanced parameter that indicates how much around the model atomic structure should be cut. 3Å is the default value.
- `SFCALC mradius`: Radius to compute the mask around the model atomic structure. 3Å is the default value.
- `Number of refinement iterations`: Cycles of refinement. 30 cycles is the default value.
- `Matrix refinement weight`: Weight parameter between electron density map (experimental data) and model atomic structure geometry. Increase this value if you want to give more weight to experimental data. If the value is set to 0.0, bond root mean square deviation from optimal values will be between 0.015 and 0.025.
- `B factor`: Geometrical restriction applied to bonded and nonbonded atom pairs. This B factor value set the initial B values.
- `Extra parameters`: This parameter gives the opportunity to add some extra *Refmac* parameters. Use "|" to separate the next parameter from the previous one.

- Protocol execution:

  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the

output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and a window panel will be opened (Fig. 96). Results can be visualized by selecting each menu element.

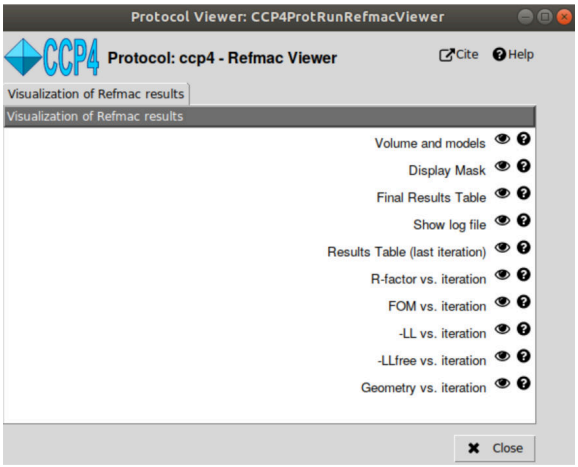


Figure 96: Protocol ccp4 - refmac. Menu to visualize *Refmac* results.

Options to visualize *Refmac* results:

  – Volume and models: *ChimeraX* graphics window displays coordinate axes, selected input volume, starting atomic structure generated by *Coot*, and final *Refmac* refined structure (Fig. 97).

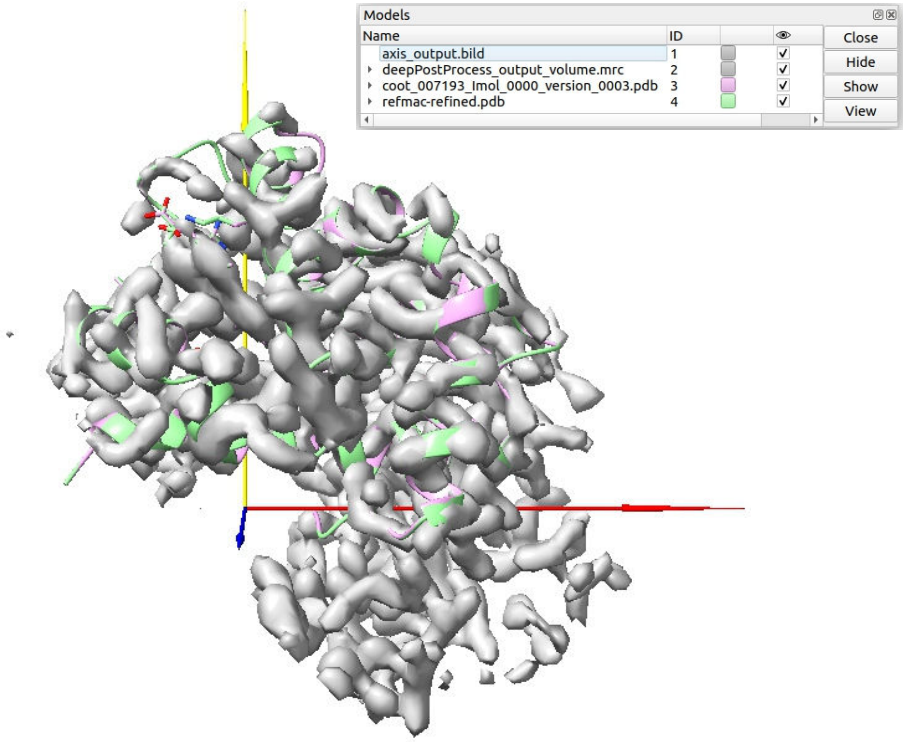


Figure 97: Protocol ccp4 - refmac. Map and models visualized with *ChimeraX*.

– Display Mask: *ChimeraX* graphics window displays the mask generated around the model atomic structure that has to be refined (Fig. 98).

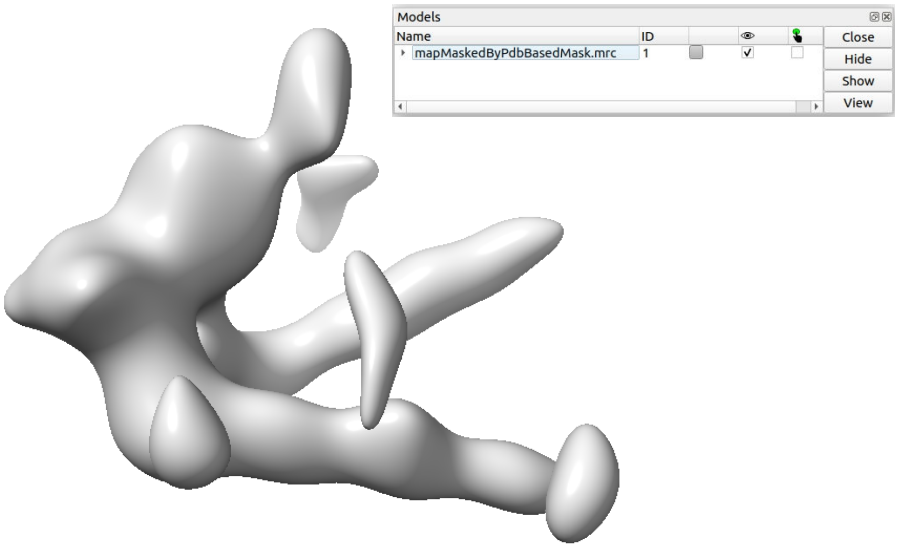
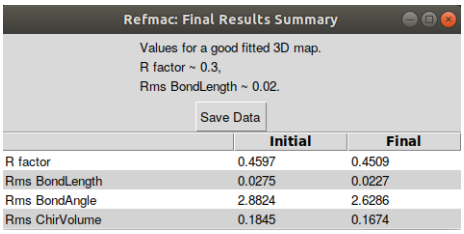


Figure 98: Protocol ccp4 - refmac. Mask visualized with *ChimeraX*.

– Final Results Table: Table showing the basic statistics of *Refmac* results. Comparison between initial and final refinement values allows to follow the refinement process. Lower final values than initial ones indicate that discrepancy indices between experimental data and ideal values are disminishing with refinement, which is desirable. R factor and Rms BondLength fair values should be around 0.3 and 0.02, respectively (Fig. 99).



	Initial	Final
R factor	0.4597	0.4509
Rms BondLength	0.0275	0.0227
Rms BondAngle	2.8824	2.6286
Rms ChirVolume	0.1845	0.1674

Figure 99: Protocol ccp4 - refmac. *Refmac* final results table.

– Show log file: *Refmac*-generated text file containing statistics of every *Refmac* running cycle (Fig. 100).

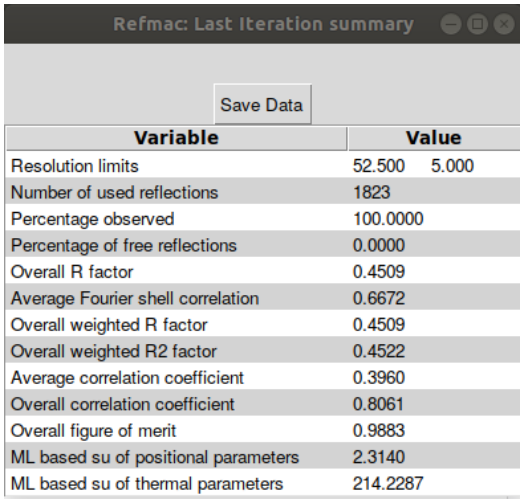
```
<B><FONT COLOR="#FF0000"><!--SUMMARY_BEGIN-->
<html> <!-- CCP4 HTML LOGFILE -->
<hr>
<!--SUMMARY_END--></FONT></B>
<B><FONT COLOR="#FF0000"><!--SUMMARY_BEGIN-->
<pre>

 ##############################################################
 ##############################################################
 ##############################################################
 ### CCP4 7.0.078: Refmac          version 5.8.0258 : 09/10/19##
 ##############################################################
 User: marta  Run date:  9/11/2020 Run time: 18:13:58


 Please reference: Collaborative Computational Project, Number 4. 2011.
 "Overview of the CCP4 suite and current developments". Acta Cryst. D67, 235-242.
 as well as any specific reference in the program write-up.
```

Figure 100: Protocol ccp4 - refmac. *Refmac* raw log file.

– Results Table (last iteration) (Fig. 101):



Variable	Value
Resolution limits	52.500 5.000
Number of used reflections	1823
Percentage observed	100.0000
Percentage of free reflections	0.0000
Overall R factor	0.4509
Average Fourier shell correlation	0.6672
Overall weighted R factor	0.4509
Overall weighted R2 factor	0.4522
Average correlation coefficient	0.3960
Overall correlation coefficient	0.8061
Overall figure of merit	0.9883
ML based su of positional parameters	2.3140
ML based su of thermal parameters	214.2287

Figure 101: Protocol ccp4 - refmac. *Refmac* last iteration results table.

* Resolution limits: 0.0 and the resolution value provided as input.
* Number of used reflections: Each reflection is defined as the common direction that the scattered waves follow, considering all the atoms included in a crystallographic unit cell. A structure factor will be computed for this common direction. The number of reflections is thus identical to the number of structure factors.

* Percentage observed: Percentage of observed reflections.
* Percentage of free reflections: Percentage of reflections observed and not included in the refinement process. These reflections are used to compute the `R factor free`.
* Overall `R factor`: Fraction of total differences between observed and computed amplitudes of structure factors, previously scaled, regarding total observed amplitudes of structure factors.

  $$Rfactor = \frac{\sum ||F_o| - |F_c||}{\sum |F_o|}$$

  where $|F_o|$ is the observed amplitude of the structure factor and $|F_c|$ is the calculated amplitude of the structure factor.
* Average Fourier shell correlation: `FSC`, cross-correlation between shells of two 3D volumes in Fourier space, calculated using complex Fourier coefficients, divided by the number of structure factors in a particular frequency (resolution) shell. $FSC_{average}$ has the advantage over `FSC` of being independent on weight (related with inverse variances of cryo-EM density maps) whenever resolution shells are thin enough that the number of structure factors in each shell is almost equal (Brown et al., 2015).
* Overall weighted `R factor`: Overall `R factor` that applies a weight factor to differences between observed and computed amplitudes of structure factors, and also applies that weight factor to the observed amplitudes of structure factors. As in the $FSC_{average}$, the weight is related with inverse variances of cryo-EM density maps.

  $$weightedRfactor = \frac{\sum (w|F_o| - |F_c||)}{\sum (w|F_o|)}$$

  where `w` is the weight factor.
* Overall weighted `R2 factor`: Also known as generalised `R factor`, this factor is computed as the root square of the fraction of total

squares of weighted differences between observed and computed amplitudes of structure factors, previously scaled, regarding the total of weighted squares of observed amplitudes of structure factors.

$$weightedR^2factor = \frac{\sum(w(|F_o|-|F_c||)^2)}{\sum(w(|F_o|)^2)}$$

* Average correlation coefficient:
* Overall correlation coefficient: Correlation between observed and calculated structure factor amplitudes, taking into account only reflections included in the refinement process.
* Cruickshank's `DPI` for coordinate error: Diffraction precision index, useful to estimate atomic placement precision. This factor is a function of the number of atoms and reflections included in the refinement, of the overall `R factor`, of the maximum resolutions of reflections included in the refinement, as well as the completeness of the observed data.
* Overall figure of merit: *Cosine* of the error of phases in radians; 1 indicates no error.
* `ML` based su of positional parameters: Comprehensive standard uncertainties of positional parameters based on the maximum likelihood function.
* `ML` based su of thermal parameters: Comprehensive standard uncertainties of thermal parameters (B values) based on the maximum likelihood function.

– `R factor` vs. iteration: Plot to visualize `R factor` and `R factor free` regarding iterations (Fig. 102):

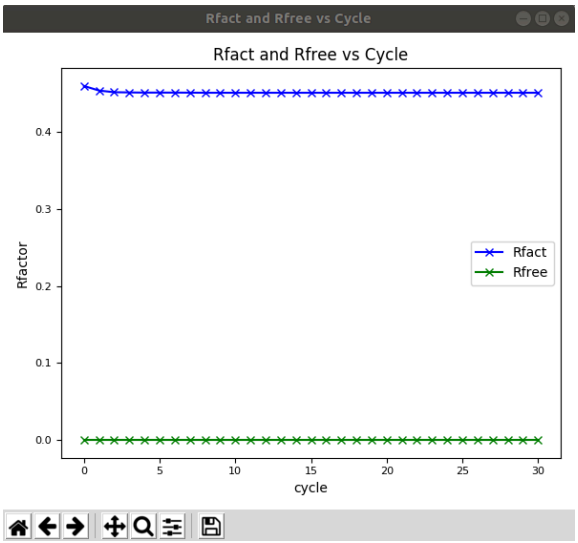


Figure 102: Protocol ccp4 - refmac. `R factor` vs. cycle plot.

– `FOM` vs. iteration: Plot to visualize Figure Of Merit regarding iterations (Fig. 103):

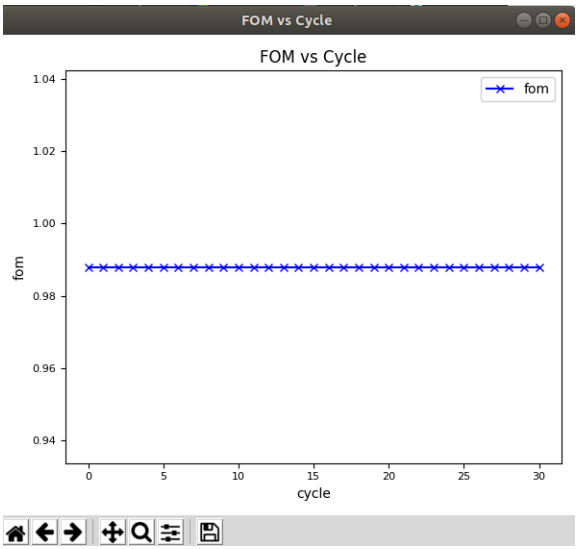


Figure 103: Protocol ccp4 - refmac. `Figure Of Merit` vs. cycle plot.

– `-LL` vs. iteration: Plot to visualize the log(Likelihood) regarding itera-

tions. Likelihood indicates the probability of a refined model, given the specific observed data (Fig. 104):

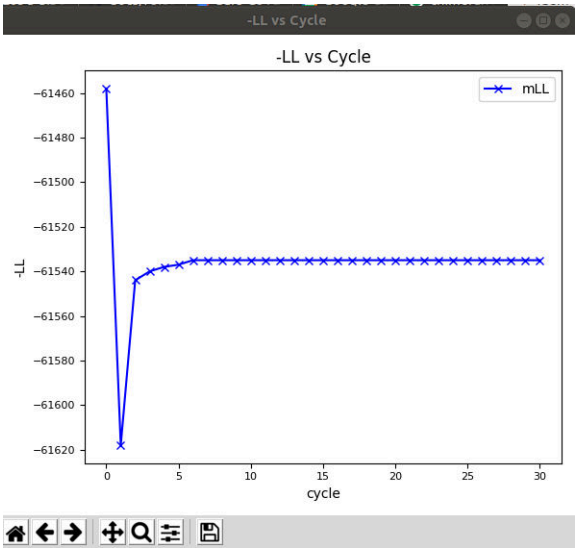


Figure 104: Protocol ccp4 - refmac. log(Likelihood) vs. cycle plot.

– `-LLfree` vs. iteration: Same definition as -LL vs. iteration, although considering only "free" reflections not included in refinement (Fig. 105):

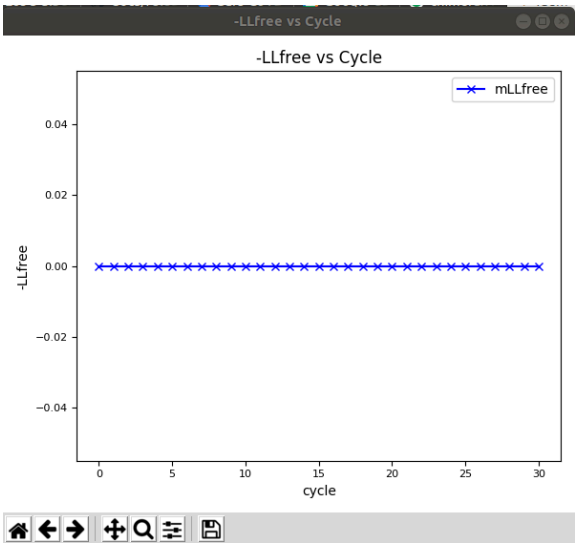


Figure 105: Protocol ccp4 - refmac. log(Likelihood) for "free" reflections vs. cycle plot.

– Geometry vs. iteration: Plot to visualize geometry parameter statistics regarding iterations (Fig. 106):

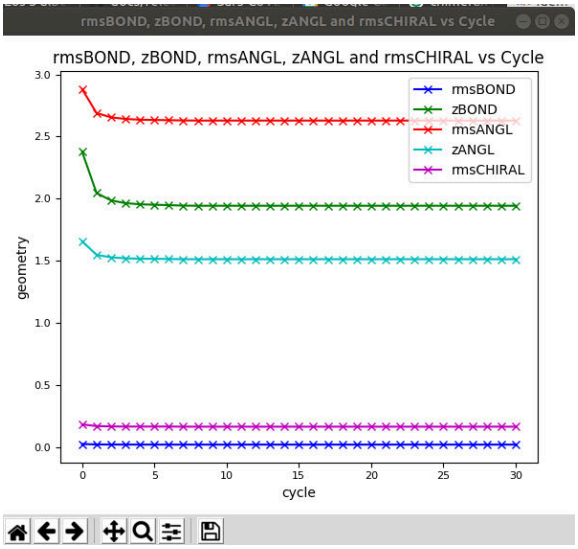


Figure 106: Protocol ccp4 - refmac. Geometry parameter statistics vs. cycle plot.

  - `rmsBOND`: Root mean square of structure atom covalent bond lengths, computed in Å, regarding ideal values of bond lengths. Selecting default weighting, `rmsBOND` values will be around 0.02.
  - `zBOND`: Number of standard deviations from the mean of covalent bond lengths. Selecting default weighting, `zBOND` values will be between 0.2 and 1.0.
  - `rmsANGL`: Root mean square of bond angles from refined structure, computed in degrees, regarding their ideal values. `rmsANGL` values should converge around 0.1.
  - `zANGL`: Number of standard deviations from the mean of bond angles.
  - `rmsCHIRAL`: Root mean square of chiral volumes from refined structure regarding their ideal values. Chiral volumes are determined by four atoms that form a piramid, and may show positive or negative values.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `ccp4 - refmac -> ouputPdb`;
    `PdbFile(pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    Statistics included in the above Final Results Table (Fig. 107):

Figure 107: Protocol ccp4 - refmac. Summary.

.

10 Create 3D Mask protocol

Protocol designed to create a mask, *i.e.*, a wrapping surface able to delimit a volume or subunit of interest, in order to modify the density values within or outside it. This mask can be created with a given geometrical shape (sphere, cube, cylinder...) or obtained from operating on a 3D volume or a previous mask.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-xmipp`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 108 (A))

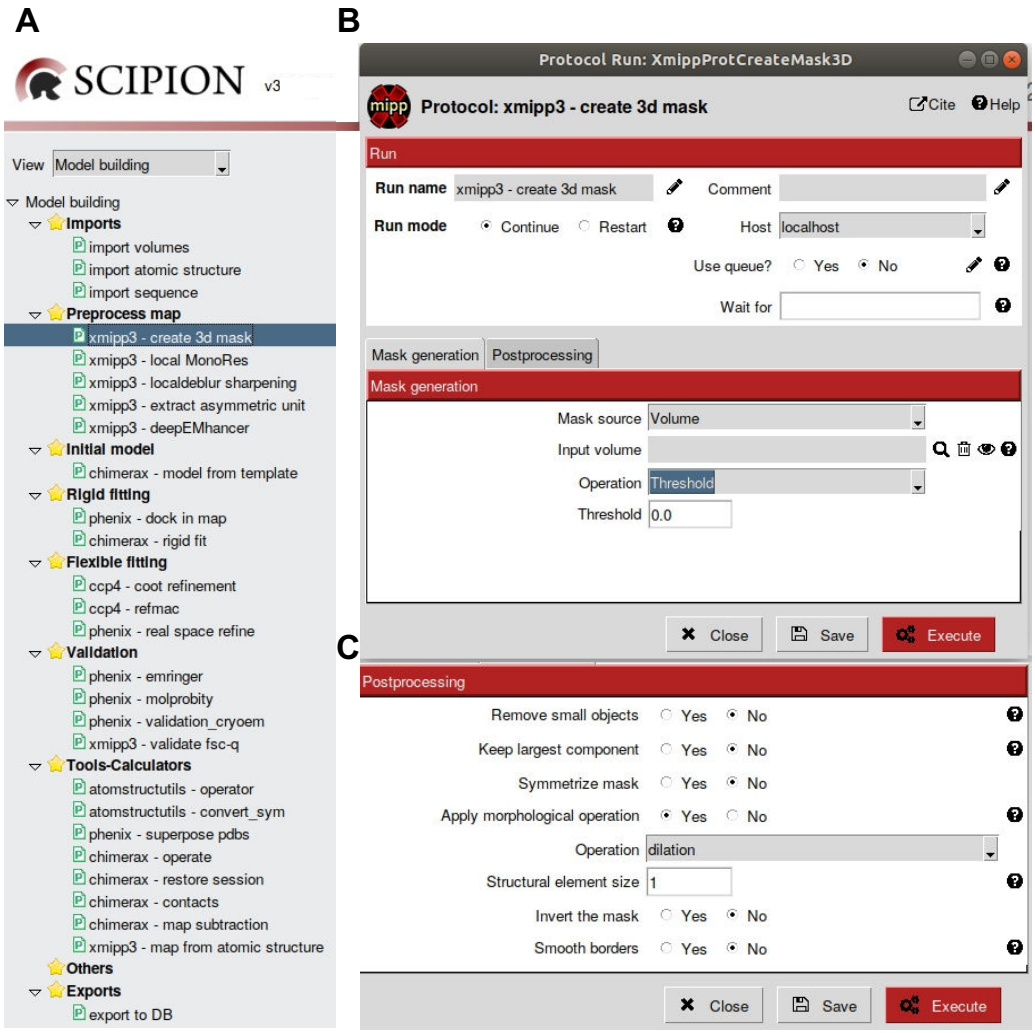


Figure 108: Protocol xmipp3 - create 3d mask. A: Protocol location in *Scipion* menu. B, C: Protocol form.

- Protocol form parameters (Fig. 108 (B: Mask generation; C: Postprocessing)):
  - `Mask generation`
    - `Mask source`: Selection of one of the next three possible types of sources for the mask, the map volume provided by the user, a specific geometrical design or a feature file.

1. `Volume`
   - `Input volume`: Electron density map previously downloaded or generated in *Scipion*.
   - `Operation`: Approach applied to generate the mask:
   (a) `Threshold`: By establishing a particular density `Threshold` (write here this threshold value).
   (b) `Segment`: Segmentation process according to:
   (c) ** `Number of voxels`(write here that value).
   (d) ** `Number of aminoacids`(write here that value).
   (e) ** `Dalton mass`(write here that value).
   (f) ** `Automatic`
   - `Only postprocess`: Use only the methods described in the tap `Postprocessing` (see below).
2. `Geometry`
   - `Sampling Rate (Åpx)`: Size of voxel dimensions in Å.
   - `Mask size (px)`: Mask dimensions in number of pixels.
   - `Mask type`: Sphere, box, crown, cylinder, Gaussian, raised cosine and raised crown. Dimensions of each one of these geometric shapes have to be assigned in pixels: Radius of the sphere (half size of the mask by default); box size; inner and outer radius of the crown, raised cosine and raised crown (half size of the mask by default); height of cylinder (mask size by default); Gaussian sigma (mask size/6 by default); and border decay or fall-off of the two borders of the crown (0 by default).
   - `Shift center of the mask?`: By selecting "Yes", the mask will be shifted to a new origin of coordinates `X, Y, Z`.
3. `Feature File`: Select with the browser the feature file in your computer.

– `Postprocessing`

  - * `Remove small objects`: Selection of "Yes" allows to ignore ligands of the map volume below a certain size (in voxels).
  - * `Keep largest component`: By selecting "Yes" a mask will be generated considering only the largest element of the map volume, ignoring the rest.
  - * `Symmetrize mask`: By selecting "Yes" a symmetrized mask will be generated according to a specific symmetry group (look at http://xmipp.cnb.csic.es/twik `c1` symmetry indicates no symmetry, by default.
  - * `Apply morphological operation`: Slight modifications of the mask can be applied by dilation or erosion of the density region (`Structural element size`: One voxel by default). Combinations of dilation and erosion allow closing or opening empty spaces of density in the map volume.
  - * `Invert the mask`: This option allows to invert the values of density regarding the wrapping surface of the mask, masking the outer part instead the inner part.
  - * `Smooth borders`: Mask borders can be smoothed by applying a convolution of the mask with a Gaussian. The Gaussian sigma (in pixels) has to be supplied.

- Protocol execution:
  Adding specific mask label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`), the default *Scipion* viewer, will

open the mask by slices (Fig. 10). The *ShowJ* window menu (`File -> Open with ChimeraX`) allows to open the mask volume in *ChimeraX* graphics window.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `xmipp3 - create 3d mask -> ouputMask`;
    VolumeMask (x, y, and z dimensions, sampling rate).
  - `SUMMARY` box:
    Details about `Mask creation` and `Mask processing`.

11 DeepEMhancer Sharpening protocol

Protocol designed to apply *DeepEMhancer*, the automatic map postprocessing method that sharpens and masks part of the noise at medium/high resolution (Sanchez-Garcia et al., 2020), in *Scipion*. Detailed information of this method can be also obtained in `https://github.com/rsanchezgarc/deepEMhancer`.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-xmipp`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 109 (A))
- Protocol form parameters (Fig. 109 (B)):

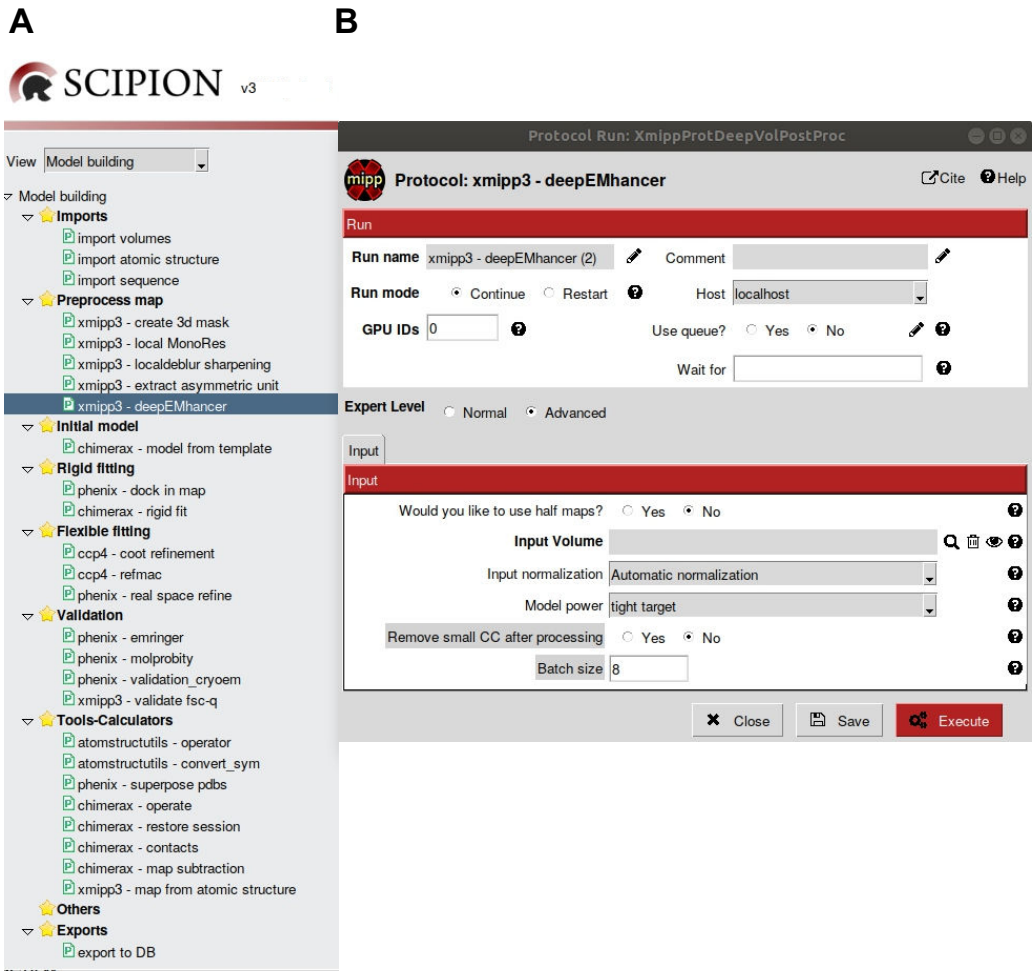


Figure 109: Protocol xmipp3 - deepEMhancer. A: Protocol location in *Scipion* menu. B: Protocol form.

– `Would you like to use half maps?`: Although the result will be the same if you decide to use half maps or a non-sharpened non-masked map, a way to ensure that you have selected the right input map is providing half maps. Then, using half maps is the preferred option over a non-sharpened non-masked map. In addition, the algorithm has been trained with half maps. The first step performed by the protocol is to compute the average map. Then, select `Yes` whenever you can provide half maps, otherwise

be sure you have the right average map, usually generated during the reconstruction process (map refinement). However, the algorithm does not work correctly using post-processed maps. So try to not use postprocessed maps. Since the two half maps can be obtained directly as independent maps or being associated to the average map, if you select `Yes` a new question will appear:

  * `Are the half maps included in the volume?`: Select `Yes` if your input average map has the two half maps associated. If this is the case, complete the next form param. Otherwise, you should provide both half maps as independent preexisting *Scipion* objects. To add those half maps a couple of form params will apper to fill in with each half map:
    - `Volume Half 1`
    - `Volume Half 2`

- `Input Volume`: Unsharpened unmasked electron density map previously downloaded or generated in *Scipion*.
- `Input normalization`: We need apply normalization to accomodate the intensity values of the map to the specific range of values of the trained neural network. Three possible normalization methods are suggested:
  * `Automatic normalization`: Default normalization mode that forces the noise average to be zero and the standard deviation 0.1 in an spheric shell around the specimen. Since then the noise always displays a similar distribution, the network gets easier to distinguish noise from signal. This method usually works correctly in almost any case. Exceptions could be very long specimens (fiber proteins) or those having big empty spaces (big viruses).
  * `Normalization from statistics`: Similar to the first one, though in this case users provide their own statistics of the noise (average and standard deviation). Using *ChimeraX* could be a good option to compute statistics of the noise such as min and max values, mean, standard deviation from the mean (SD) and root-mean-

square deviation from zero (RMS) (command line `measure mapstats` with the option `subregion`; `https://www.cgl.ucsf.edu/chimerax/docs/user/commands/measure.html#mapstats`).

    * `Normalization from binary mask`: Select this option only if your input map is a masked map, which otherwise is not recomendable when this algorithm is used. The binary mask assigns 1 to the specimen and 0 to the remaining density.

- `Model power`: Deep learning model to use, three options are available:
    * `tight target`: This default model is a equidistant balanced solution that works properly for maps that show resolution areas between 3.8 and 6 Å(wide range of resolution values).
    * `highRes`: This model allows a deep sharpening and it is recommended for high resolution maps (lower than 4 Å).
      (Note: In case your map shows a high heterogeneity with parts of high resolution, as well as areas of low resolution, using `tight target` and `highRes` is recommendable, studying which areas are better sharpened by each model).
    * `wide target`: This is the most conservative model. It is recommended when you have areas in which signal and noise are almost identical. Whereas `tight target` and `highRes` might delete those regions considering them only noise, `wide target` will surely preserve them.
- `Remove small CC after processing`: Advanced param that improves the sharpening result by removing small connected components (usually noise). The default value of this param is `No` because the improvement usually does not make up for the additional time and computational resources invested.
- `Batch size`: Since *DeepEMhancer* processes maps by dividing them in portions or smaller cubes that will be sent to GPUs, the value of batch size indicates the number of cubes that GPUs can simoultaneously process.

Increase or reduce the default number according to the performance of the GPUs.

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  (Remark: In this case you have the option `GPU IDs` that you have to complete according to your GPU core indexes.) Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Both the input map(s) and the sharpened map generated by *DeepEMhancer* (`deepPostProcess.mrc`) are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue). Coordinate axes, input volume, and sharpened map are model numbers `#1`, `#2` , and `#3`, respectively, in *ChimeraX* `Model Panel`. In case that half maps have been included, the respective additional model numbers will be applied.

  The possibility of visualizing the sharpened map by slices with *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`) is also opened as commonly in *Scipion*, selecting in the `Output` of the `Summary` box, black arrow `xmipp3 - deepEMhancer -> Volume`, the right mouse option `Open with DataViewer`.

- Summary content:

  – Protocol output (below *Scipion* framework):

`xmipp3 - deepEMhancer -> Volume`;
Volume (x, y, and z dimensions, sampling rate).

– `SUMMARY` box:
`Input`: type of map
`Normalization`: normalization method.

12 Extract asymmetric unit protocol

Protocol designed to obtain in *Scipion* the smallest asymmetric subunit of an electron density map having certain types of rotational symmetry.
WARNING: This protocol requires the starting volume located in the center of coordinate axes to equal the center of symmetry with the origin of coordinates.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-xmipp`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 110 (A))
- Protocol form parameters (Fig. 110 (B)):

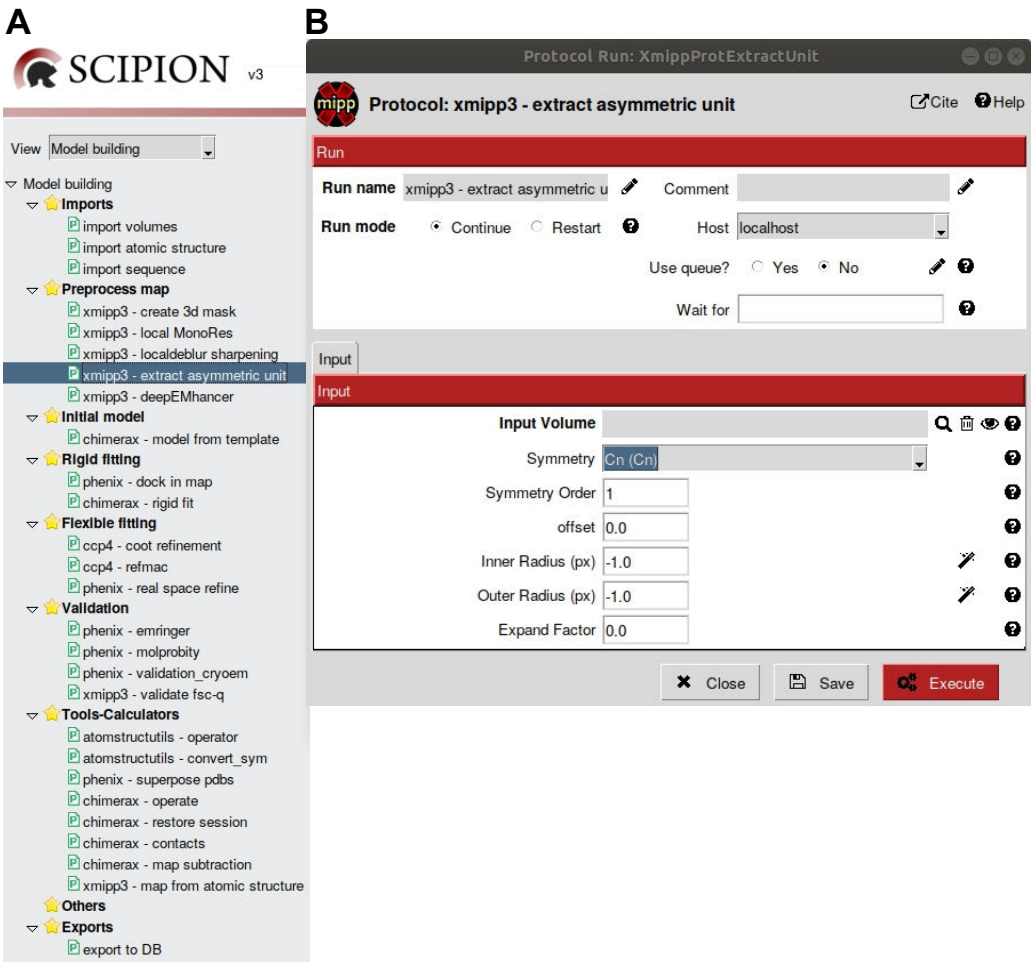


Figure 110: Protocol xmipp3 - extract asymmetric unit. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input Volume`: Volume already downloaded in *Scipion* from which the asymmetric unit will be extracted.
- `Symmetry`: In this protocol, symmetry refers only to rotational symmetry, also known in biology as radial symmetry. This symmetry is the property of volumes to preserve their shape after a partial turn around a symmetry axis.

  Types of rotational symmetry included in this protocol are shown in Fig.

111. Two names appear in each case, the first one corresponds to `XMIPP` nomenclature of symmetry because we are using `XMIPP` package, and the second one (in brackets) follows the general *Scipion* nomenclature. Current *Scipion* nomenclature is *ChimeraX*'s nomenclature, which is, in turn, the same symmetry nomenclature of the International Union of Cristallography.

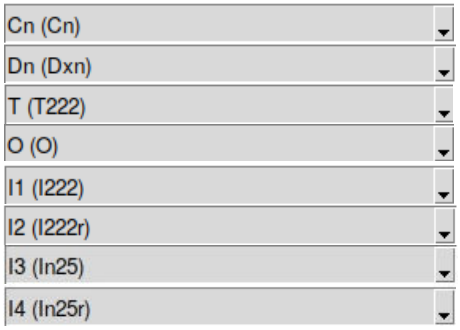


Figure 111: Protocol xmipp3 - extract asymmetric unit. Types of rotational symmetry.

* Cyclic symmetry `Cn (Cn)`: Only one symmetry axis goes through the geometric center of the volume. Two more form parameters are shown when this type of symmetry is selected:
  - `Symmetry Order`: Number of times (`n`) in which a volume shows the same shape when the volume rotates around the symmetry axis from 0 to 360º. If the same shape is only obtained after turning 360º, then `n = 1`. This means that the volume has no symmetry. `360°/n` determines the rotation angle.
  - `offset`: Starting angle around Z axis.
* Dihedral symmetry `Dn (Dxn)`: Two perpendicular symmetry axes go through the geometric center of the volume. As in the case of cyclic symmetry, two more form parameters are shown when this type of symmetry is selected:
  - `Symmetry Order`: Number of times (`n`) in which a volume shows

the same shape when the volume rotates around both symmetry axes from 0 to 360º. Analogously, `360º/n` determines the rotation angle.

· `offset`: Starting angle around Z axis.

* Tetrahedral symmetry `T (T222)`: Four symmetry axes go from each vertex to the opposing face center (order 3), and three symmetry axes join opposing edges (order 2). `Symmetry order = 12`.
* Octahedral symmetry `O (O)`: Three symmetry axes join opposing vertices (order 4), four symmetry axes join opposing face centers (order 3), and six symmetry axes join opposing edges (order 2). `Symmetry order = 24`.
* Icosahedral symmetries `I1 (I222), I2 (I222r), I3 (In25), I4 (In25r)`: Six symmetry axes join opposing vertices (order 5), 10 symmetry axes join baricenters of opposing faces (order 3), and 15 symmetry axes join opposing edges (order 2). `Symmetry order = 60`. Each type of icosahedral symmetry depends on its initial orientation. Check in *ChimeraX* (`https://www.cgl.ucsf.edu/chimerax/docs/user/commands/shape.html`) each icosahedral symmetry by writing in the command line: `shape icosahedron radius 50 orientation` (`222`: default, order 2 axes follow XYZ coordinate axes; `222r`: idem rotated 90º around Z axis; `n25`: an order 2 axis and an order 5 axis follow Y and Z axes, respectively, `n25r`: idem rotated 90º around Z axis.

– `Inner Radius (px)`: Minimal distance from the geometric center that delimits inwards the part of the electron density map that will be included in the extracted volume. A wizard symbol on the right side of this parameter can be helpful to select this radius.

– `Outer Radius (px)`: Maximal distance from the geometric center that delimits outwards the part of the electron density map to be included in the extracted volume. In other words, the part extracted of the map electron density will be between the `Inner` and the `Outer Radius`. Again, the wizard symbol on the right side of this parameter can be helpful to

select this radius.

- `Expand Factor`: Additional fraction of the asymmetrical unit cell that will be included in the extracted volume.

- Protocol execution:

  Press the `Execute` red button at the form bottom.
  Adding specific extracted volume label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and a small window will be opened (Fig. 112). This window allows you to select between `chimerax` (*ChimeraX* graphics window) and `slices` (*ShowJ*, the default *Scipion* viewer), to visualize the volume.

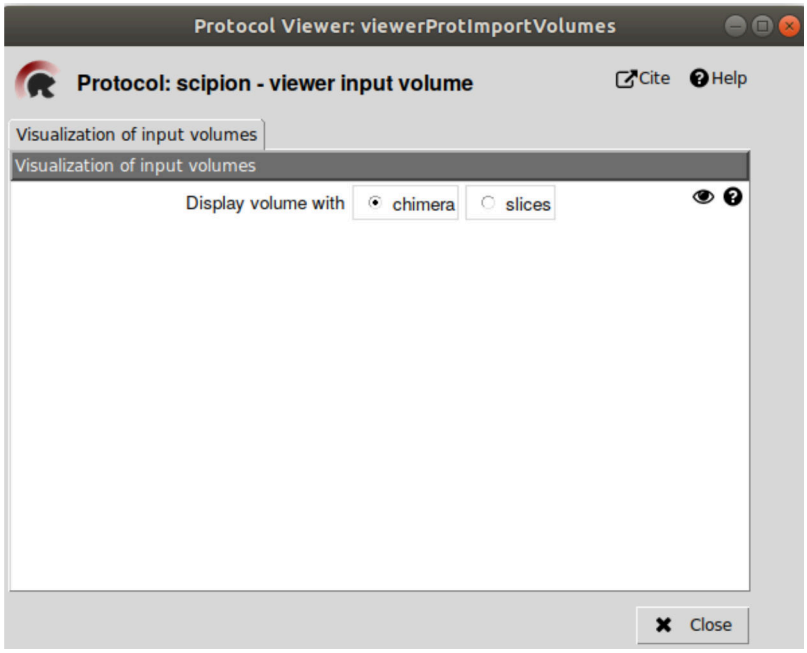


Figure 112: Menu to select a visualization tool.

– `chimerax`: *ChimeraX* graphics window

  Initial whole volume and extracted volume appear referred to the origin of coordinates in *ChimeraX*. To show the relative position of the volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, initial volume, and extracted map asymmetric unit are model numbers `#1`, `#2` and `#3`, respectively, in *ChimeraX* `Models` panel. Volume coordinates and pixel size can be checked in *ChimeraX* main menu `Tools -> Volume Data -> Map Coordinates: Origin index/ Voxel size`. WARNING: Take into account that coordinates appear in pixels while they have been introduced in Å.

– `slices`: *ShowJ*

  `https://github.com/I2PC/scipion/wiki/ShowJ`

  Each volume can be independently visualized by selecting it in the upper menu as the arrow indicates in Fig. 113.

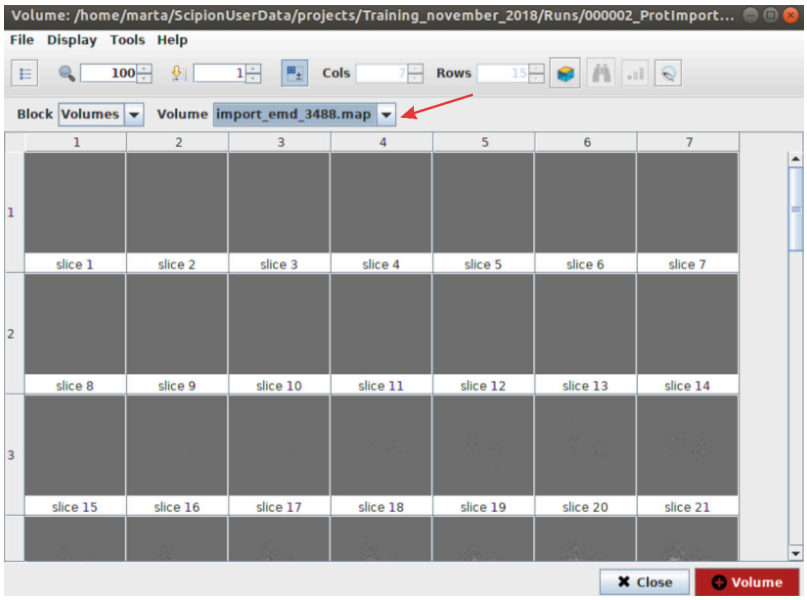


Figure 113: Protocol xmipp3 - extract asymmetric unit. Volume selection with ShowJ.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `xmipp3 - extract asymmetric unit -> ouputVolume`;
    Volume (x, y, and z dimensions, sampling rate).
  - `SUMMARY` box:
    Empty.

13 Import atomic structure protocol

Protocol designed to import an atomic structure in *Scipion* from PDB database or from a file of the user's computer.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`

– *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu: `Model building -> Imports` (Fig. 114 (A))
- Protocol form parameters (Fig. 114 (B)):

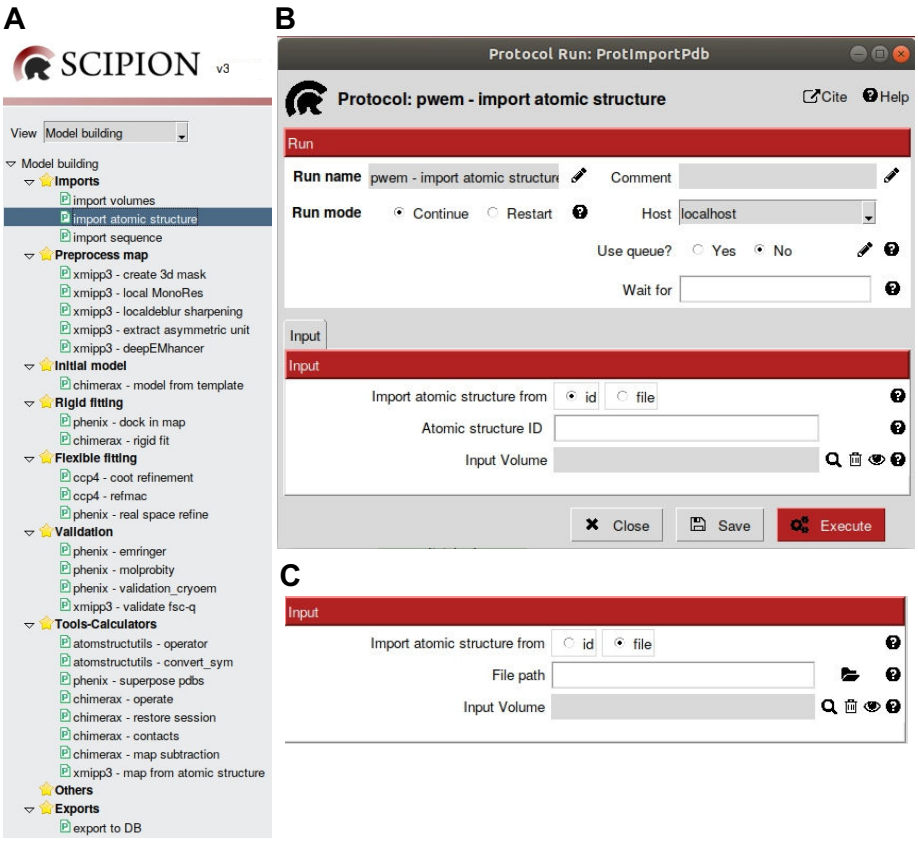


Figure 114: Protocol import atomic structure. A: Protocol location in *Scipion* menu. B: Protocol form to import the atomic structure from PDB. C: Protocol form to import the atomic structure from a file.

– `Import atomic structure from`: Parameter to select the origin of the atomic structure that you want to import. Two options are indicated:

  * `id`: Select this option if you want to import the atomic structure from PDB database. Associated to this option is the next form parameter:

· `Atomic structure ID`: Box to write the accession ID of the desired PDB structure. Structure extension .cif/ .pdb. is not required.

* `file`: Select this option if you want to import the atomic structure from a file. A new parameter appears associated to this option (Fig. 114 (C)):

· `File path`: Box to be completed with the file path. The browser located at the right side of the parameter box helps to look for the file in the user's computer.

– `Input Volume`: If you want to associate a previously downloaded volume in *Scipion* to the atomic structure, select that volume here.

- Protocol execution:

  Adding specific atomic structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default (Fig. 85). Atomic structures are referred to the origin of coordinates in *ChimeraX*. To show the relative position of the atomic structure, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue). Coordinate axes and imported atomic structure are model numbers `#1` and `#2`, respectively, in *ChimeraX* `Models` panel. If a volume has been associated to the atomic structure, coordinate axes and imported atomic structure are model numbers `#2` and `#3`, respectively, in *ChimeraX* `Models` panel, whereas structure-associated volume has model number `#1`. Volume coordinates and pixel size can be checked in *ChimeraX*

main menu `Tools -> Volume Data -> Map Coordinates: Origin index/ Voxel size`. WARNING: Take into account that coordinates appear in pixels.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `pwem - import atomic structure -> ouputPdb`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    Atomic structure imported from ID: / file: `PDB accession ID / path`

14 Import sequence protocol

Protocol designed to import aminoacid or nucleotide sequences in *Scipion* from four possible origins (plain text, atomic structures from PDB database or a file in your computer, text file of the user's computer, and `UniProtKB/ GeneBank` databases).

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
- *Scipion* menu: `Model building -> Imports` (Fig. 115 (A))
- Protocol form parameters (Fig. 115 (B)):

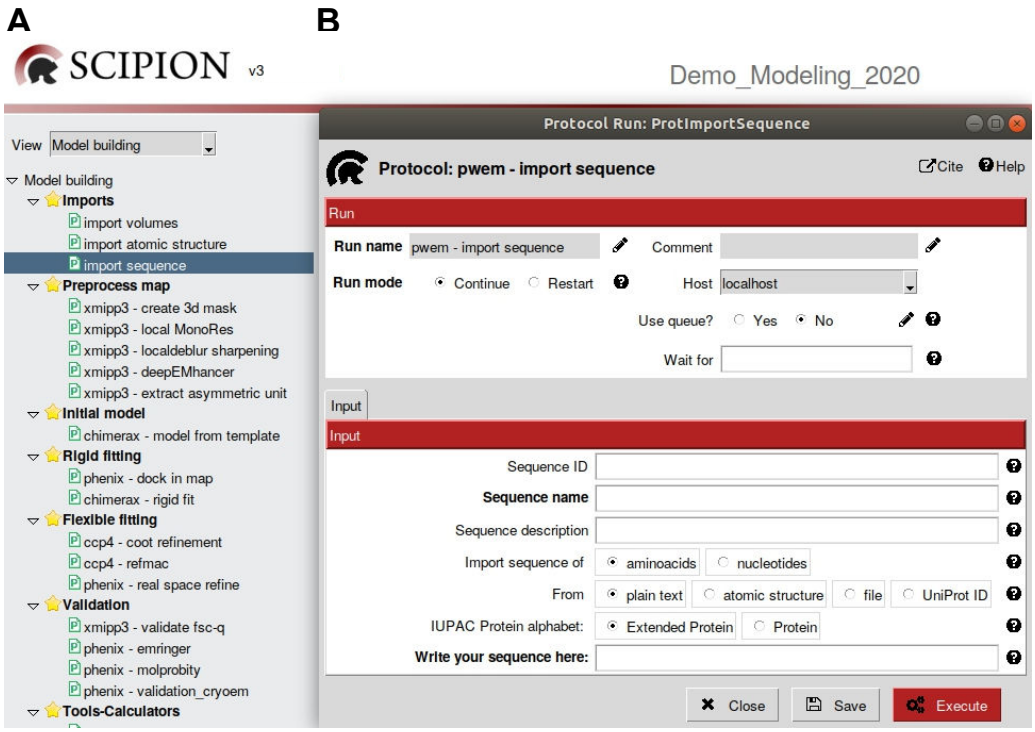


Figure 115: Protocol import sequence. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Sequence ID`: Optional short name to identify your sequence (acronym or number, e. g. `Q05769`). If no ID is assigned by the user, and the sequence has been downloaded from `GeneBank/UniProtKB/PDB` database, the database ID will be selected as `Sequence ID` (Read `Help` section (question mark) to see some examples). Otherwise, `Sequence name` will be set as `Sequence ID`. The `Sequence ID` will be included in sequence alignments in *ChimeraX* to identify the sequence.
- `Sequence name`: Compulsory short name to identify your sequence (example: `PGH2_MOUSE`). Names with certain meaning are recommended. The `Sequence name` will appear in the summary box of the *Scipion* protocol.
- `Sequence description`: Optional description of your sequence. It can include functionality, organism, size, etc... (e.g. `Prostaglandin G/H syn-`

thase 2). If no description is assigned by the user, and the sequence has been downloaded from GeneBank/UniProtKB/PDB database, the database description will be selected as Sequence description. Otherwise, no description will be included.

- Import sequence of: Selection parameter to choose between aminoacids and nucleotides. After selecting one of them, a new selection menu will be opened:
  - aminoacids: Parameter to select one of these four options:
    - plain text: Select this option if you want to introduce your own single letter aminoacid sequence. Since your sequence will be cleaned according to the standard protein alphabet of 20 aminoacids (Protein) or to an extended alphabet that includes 6 additional aminoacids or aminoacid groups (Extended Protein), you have to select one of these IUPAC Protein alphabets. Read Help section (question mark) to know the aminoacids included in each alphabet. Not only non-canonical aminoacids will be cleaned, but also wildcard characters such as *, #, ?, -, etc... Write your sequence here indicates the place where your single letter aminoacid sequence has to be written or paste.
    - atomic structure: Select this option if you want to download the sequence from an atomic structure (Fig. 116 (A)). Select id to download your sequence from PDB database. Then, write the PDB ID (Atomic structure ID) and select the chain sequence of your preference (Chain). Use the wizard on the right side of Chain parameter to select that chain. Follow an analogous process to download the sequence from an atomic structure that you already have in your computer. This time, the File path will replace the Atomic structure ID. By pressing the folder symbol, a browser will help you to find the structure file.
    - file: Select this option if your sequence is written in a text file that you already have in your computer (Fig. 116 (B)). By

pressing the folder symbol, a browser will help you to find the sequence file.

· `UniProtID`: Select this option if you want to download the sequence from `UniProtKB` database (Fig. 116 (C)). Write the name/ID of the respective sequence in the parameter box `UniProt name/ID`. An error message appears in case you introduce a wrong ID.

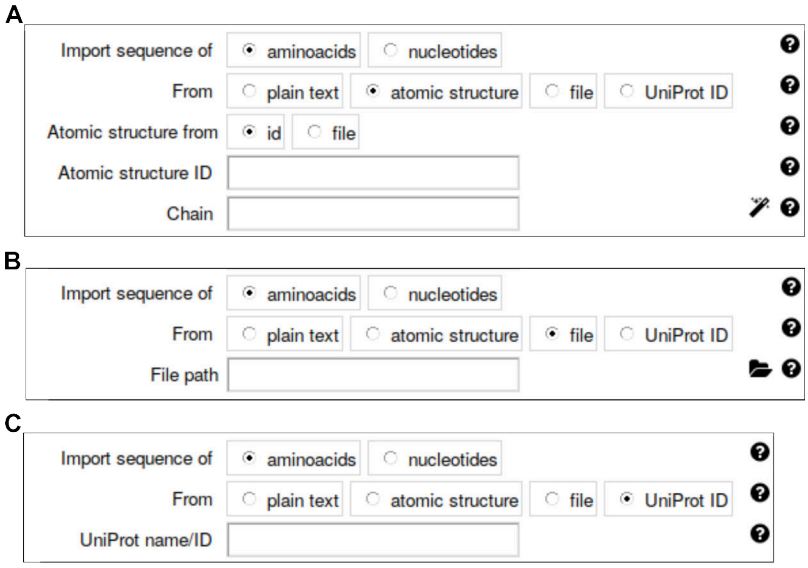


Figure 116: Protocol import sequence. Protocol form to import aminoacid sequences from the `PDB` database by indicating its respective `ID` (A), from a file (B), or from `UniProtKB` by writing the database `ID/name` (C).

∗ `nucleotides`: Analogously to `aminoacids` parameter, select one of these four options:

· `plain text`: Parameter to introduce your own single letter nucleotide sequence (Fig. 117 (A)). Since your sequence will be cleaned according to the standard nucleic acid alphabet, you have to select one of the next five alphabets. The first three are DNA alphabets and the last two ones are RNA alphabets. Read `Help`

section (question mark) to understand each alphabet. The most restricted ones are `Unambiguous DNA` ("A, C, G, T") and `Unambiguous RNA` ("A, C, G, U") for DNA and RNA, respectively. The cleaning process also involves wildcard characters such as `*`, `#`, `?`, `-`, etc...

- `atomic structure`: Information described for aminoacids is valid for nucleotides (Fig. 117 (B)).
- `file`: Information described for aminoacids is valid for nucleotides (Fig. 117 (C)).
- `GeneBank`: Information described for aminoacids is valid for nucleotides, this time replacing `UniProtKB` by `GeneBank` (Fig. 117 (D)).

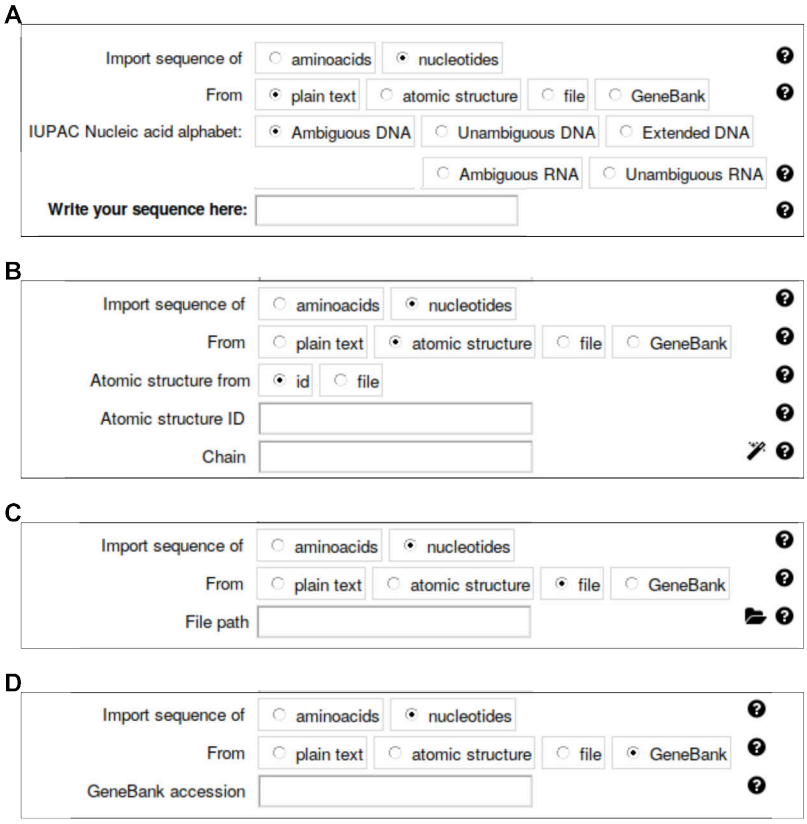


Figure 117: Protocol import sequence. Protocol form to write (A) or import nucleotide sequences from PDB database by indicating its respective ID (B), from a file (C), or from GeneBank by writing the database accession number (D).

- Protocol execution:

  Adding specific sequence label is recommended in Run name section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of Run name box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to Restart the Run mode.
  Press the Execute red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and a text editor will be opened in which you can read the sequence in `fasta` format. `Sequence ID` and `Sequence description` are included in the header.

- Summary content:

  - Protocol output (below *Scipion* framework):
    `pwem - import sequence -> ouputSequence`;
    `Sequence name`
  - `SUMMARY` box:
    Sequence of aminoacids/ nucleotides:
    Sequence `Sequence name` imported from plain text/ atomic structure/ file/ UniProt ID.

15 Import mask protocol

Protocol designed to import a mask in *Scipion* from a file of user's computer. Modifying the size of a previous mask is possible simply by changing the mask's sampling rate.

- Requirements to run this protocol and visualize results:

  - *Scipion* plugin: `scipion-em`

- *Scipion* menu: It does not appear in `Model building` view. Press Ctrl + f and the pop up window to search a protocol will be opened ((Fig. 152 (A)). Write any word related with the title of the protocol that you are looking for in the `Search` box. In this particular case we have written `mask`. Several protocols have been found related with this search word. Select the first one dessigned for the purpose that we are interested in (`pwem - import mask`).

- Protocol form parameters (Fig. 152 (B)):

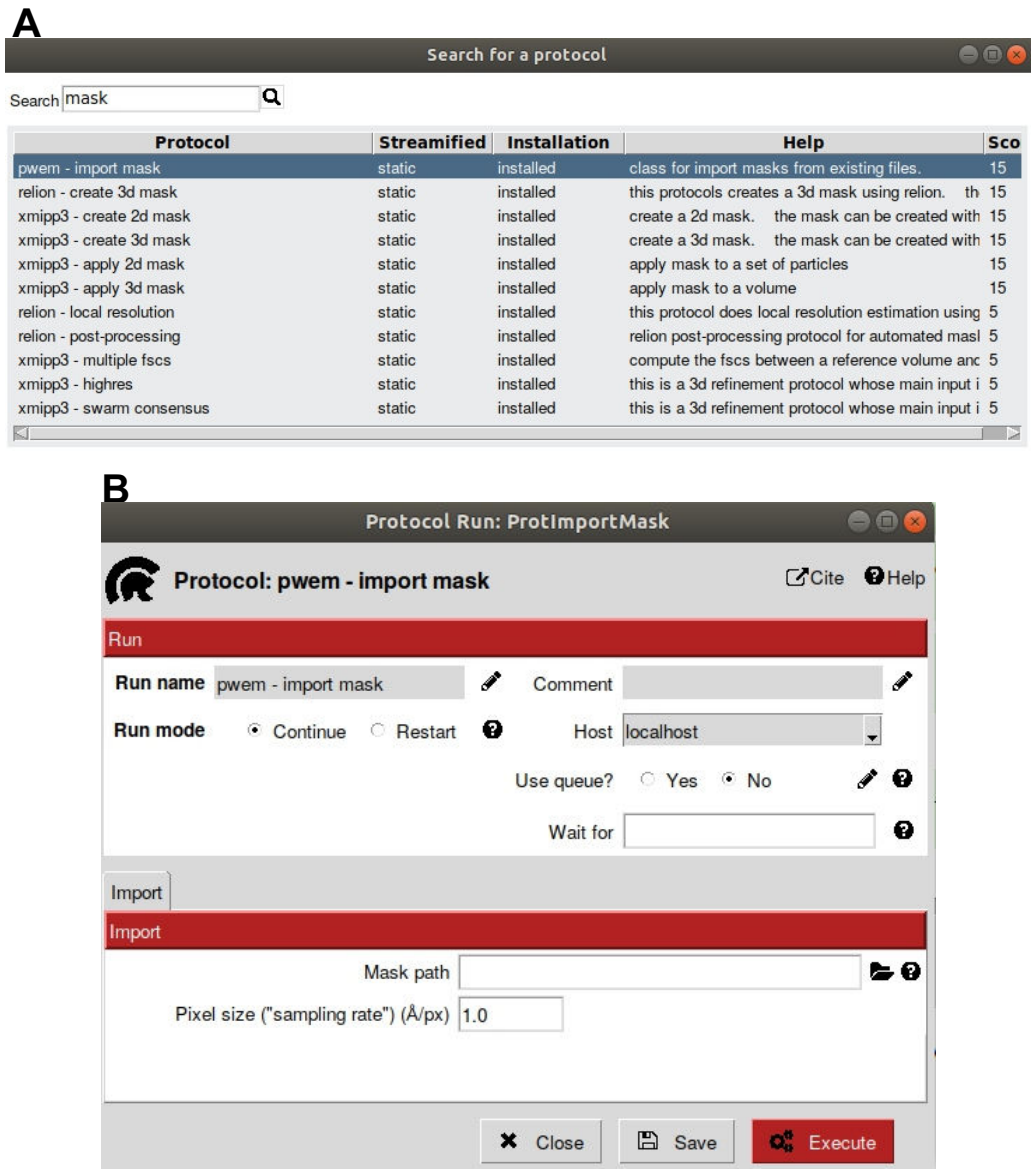


Figure 118: A. Protocol import mask. A: Window to search the protocol. B: Protocol form.

- `Input` section
  - `Mask path`: Open the browser on the right to select in your computer the

path to the previously saved mask.

- `Pixel size (‘‘sampling rate’’)` (Å/px): Write the new sampling rate rate value in the box.

- Protocol execution:

  Adding specific mask label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK, and finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ShowJ*, the default *Scipion* viewer, will allow you to visualize the `slices` window of the mask (Fig. 10). The *ShowJ* window menu (`File -> Open with ChimeraX`) allows to open the selected map in *ChimeraX* graphics window.

  - `slices`: *ShowJ*
    `https://github.com/I2PC/scipion/wiki/ShowJ`

- Summary content:

  - Protocol output (below *Scipion* framework):
    `pwem - import mask -> ouputMask`;
    `VolumeMask (x, y, and z dimensions, sampling rate)`.
  - `SUMMARY` box:
    `Mask file imported from`: The specific selected path to the mask in your computer should appear here.

16 Import volume protocol

Protocol designed to import electron density maps in *Scipion* from a file of user's computer.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Imports` (Fig. 119 (A))
- Protocol form parameters (Fig. 119 (B)):

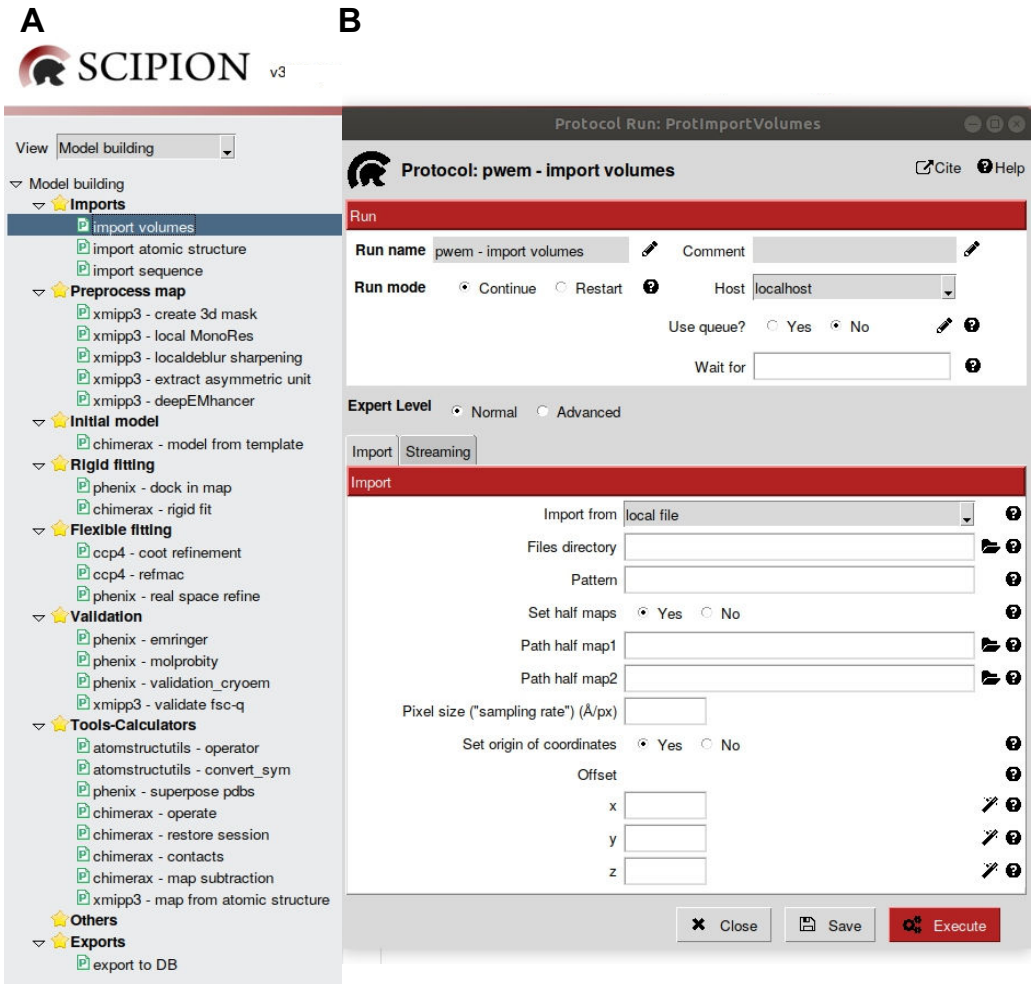


Figure 119: Protocol import volumes. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Import from`: Maps can be stored in EMDB or in your own computer. Select thus between these two options:
    - `local file`: Select this option if the map is stored in your computer. Several params will appear in this case:

- `Files directory`: Folder that contain one or several volumes (a set of volumes) that you'd like to import. By clicking the folder symbol on the right of `Files directory` box, a browser will be opened to allow you to look for the volume(s)-containing file in your computer. Click the volume that you want to select, if only one volume is going to be loaded. If a set of volumes from the same folder are going to be loaded, click the respective folder.
- `Pattern`: In case you'd like to import a set of volumes, you can include here the common name pattern to all of them. Read `Help` section (question mark) of this parameter and the previous parameter `Files directory` to know about wildcard characters that can be used to generalize patterns.
- `Set half maps`: In case you wouldn't like to associate half maps as attributes to your map, select the default option `‘‘No’’`. Otherwise select the option `‘‘Yes’’`. If this is the case, complete the next couple of params by looking for each half map in the browser on the right:
- ** `Path half map1`
- ** `Path half map2`
- `Pixel size (‘‘sampling rate’’)` (Å/px): The size of building blocks (the smallest units) of images depend on the microscope camera and magnification conditions used to get the data.
- `Set origin of coordinates`: You have to choose between setting the default origin of coordinates (option "No") or another origin of coordinates ("Yes"). The option by default sets the center of the electron density map in the origin of coordinates. This is the preferred option in case you want to run afterwards programs that require symmetry regarding the origin of coordinates, like the extract asymmetric unit protocol. If the selected origin of coordinates differs from the map header's, then a copy of the original map will be generated with the new origin of coordinates

in its header. If you decide to set your own origin of coordinates (option "Yes"), a new form parameter (`Offset`) will appear below:

- ** `Offset`: Write here x, y, and z coordinates of your preference (in Å). Suggestions for coordinates can be obtained by pressing the wizard symbol located on the right side of the `Offset` parameter. In map files with format .mrc, suggested coordinates will be read from the map header.

* `EMDBid`: Select this option if you want to import the density map directly from EMDB. A couple of params will appear in this case:

  - `EMDB map ID (integer)`: Write the number of the map EMDB accession.
  - `Offset`: Write here x, y, and z coordinates of your preference (in Å). Suggestions for coordinates can be obtained by pressing the wizard symbol located on the right side of the `Offset` parameter. In map files with format .mrc, suggested coordinates will be read from the map header.

– `Copy files?`: Advanced parameter set to "No" by default because copy density maps unnecessarily duplicates disk space occupied by them, space that could be quite big. Then, by default, volumes will be downloaded by a symbolic link to the file location in your computer. Set this parameter to "Yes" only if you plan to transfer the project to other computers in order to preserve map data in the *Scipion* project.

- `Streaming` section

  Go to this section if you plan simultaneous data acquisition and processing, and select the option "Yes". By default, *Scipion* considers that you run your processes once you have finished data acquisition (option "No").

- Protocol execution:

Adding specific volume label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK, and finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and a small window will be opened (Fig. 112). This window allows you to select between `chimerax` (*ChimeraX* graphics window) and `slices` (*ShowJ*, the default *Scipion* viewer), to visualize the volume.

  - `chimerax`: *ChimeraX* graphics window

    Volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of the volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes and the imported volume are model numbers `#1` and `#2`, respectively, in *ChimeraX* `Models` panel. Volume coordinates and pixel size can be checked in *ChimeraX* main menu `Tools -> Volume Data -> Map coordinates: Origin index/ Voxel size`. WARNING: Take into account that coordinates appear in pixels while they have been introduced in Å.

  - `slices`: *ShowJ*

    `https://github.com/I2PC/scipion/wiki/ShowJ`

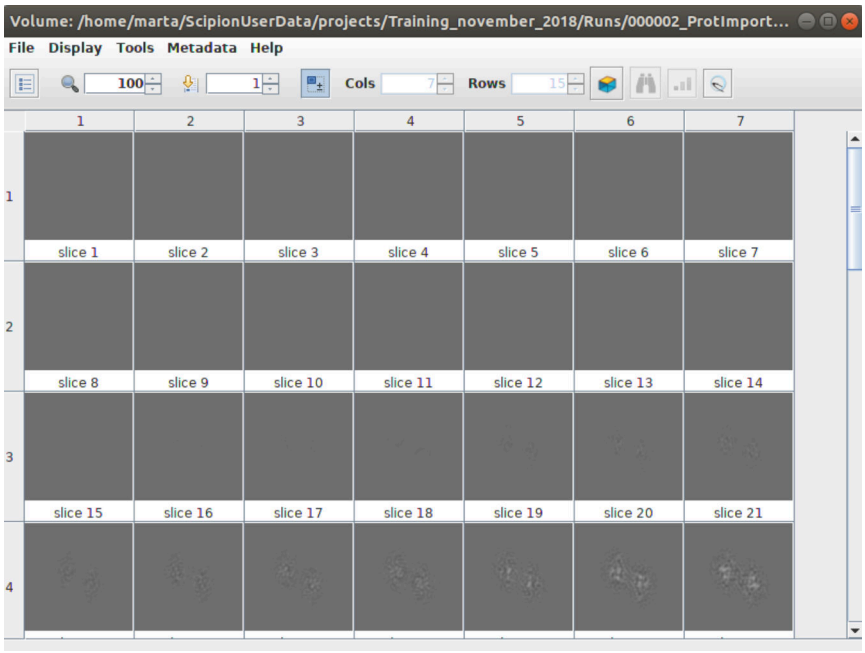


Figure 120: Protocol import volumes. Gallery model of *ShowJ* to visualize volume slices.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `pwem - import volumes -> ouputVolume`;
    `Volume (x, y, and z dimensions, sampling rate)`.
  - `SUMMARY` box:
    Path from which the volume has been downloaded.
    Sampling rate.

17 Local Deblur Sharpening protocol

Protocol designed to apply *LocalDeblur*, the automatic local resolution-based method that increases map signal at medium/high resolution (Ramírez-Aportela et al., 2018), in *Scipion*. Unlike similar approaches, *LocalDeblur* does not need any prior atomic

model, avoiding artificial structure factor corrections. Since the map gets much more interpretability, the modeling process results much easier. This type of sharpening is recommended for maps showing a broad range of resolutions, as it is usually common with membrane proteins or macromolecules highly flexible. In all those cases, applying a global sharpening method, like *Relion* `postprocess`, does not optimize the result when you have very different local resolution values because a global operation cannot improve all parts of the map. However, when changes in resolution are small there isn't almost difference between applying a global or a local sharpening method. For example, in maps with high symmetry, like viruses, resolution is quite homogenous. The absence of high differences in resolution determines that at high resolution the results of global or local sharpening methods are almost the same.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-xmipp`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 121 (A))
- Protocol form parameters (Fig. 121 (B)):

Figure 121: Protocol xmipp3 - localdeblur sharpening. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input map`: Unfiltered electron density map previously downloaded or generated in *Scipion*.
- `Resolution Map`: Resolution map generated by protocols like xmipp3 - local MonoRes. The resolution value in the corresponding voxel of the `Input map` is assigned to each voxel of the `Resolution Map`.
- `lambda`: Since *LocalDeblur* is based on an iterative formula repeated

until a convergence criterion is reached, *lambda* is the step size advanced parameter that modulates the speed of convergence. The default value, $lambda = 1$, indicates that the method itself establishes automatically the value of *lambda*. Although the default value is small enough to guarantee the convergence and large enough to speed it up, the *lambda* value can be increased by the user to accelerate the convergence process. Unlike the default value, that grows along the convergence process, the *lambda* value selected by the user will be maintained constant. Falling into a local minimum is a risk derived of increasing the convergence speed.

- `K`: Weight assigned to the difference between the local resolution and the spatial frequency of the center of each bandpass filter. This difference weighted by `K` is the base to compute the local weight of each channel in the filter bank, that correlates the input map with the sharpened map. The bigger the value of `K`, the lower the weight of each channel in the filter bank. Maximum weights are obtained when local resolution and spatial frequency of the center of each bandpass filter show identical values. As it has been empirically observed K=0.025 produces good results for most of tests performed. No big differences have been detected with K values ranging between 0.01 and 0.05. In the particular case of low resolution maps (lower than 6 Å), 0.01 seems to be a good choice.

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  To visualize the total number of sharpened maps generated according to the

number of iterations until convergence, press with the right mouse the black arrow placed in the lower part of the *Scipion* framework (Protocol output: `xmipp3 - localdeblur sharpening -> ouputVolumes`). The option `Open with DataViewer` will appear, select it. The sharpened map epochs will be detailed. The sharpening algorithm stops when the difference between two successive iterations is lower than 1%, thus generating a variable number of maps before stopping. You can select any of them and visualize the slices with *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`), the default *Scipion* viewer. To visualize the slices press the symbol that appears below `File` in the main menu. The *ShowJ* window menu (`File -> Open with Chimera`) allows to open the selected map in *ChimeraX* graphics window.

Nevertheless, the *ChimeraX* graphics window can also be opened to compare the input and the last iteration sharpened map. After executing the protocol, press `Analyze Results` and the *ChimeraX* graphics window will be opened. Volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of the volumes, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, the imported volume and the last sharpened one (`sharpenedMap_last.mrc`) are model numbers `#1`, `#2` and `#3`, respectively, in *ChimeraX* `Models` panel. Volume coordinates and pixel size can be checked in *ChimeraX* main menu `Tools -> Volume Data -> Map coordinates: Origin index/ Voxel size`.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `xmipp3 - localdeblur sharpening -> ouputVolumes`;
    SetOfVolumes (number of items, x, y, and z dimensions, sampling rate).
  - `SUMMARY` box:
    `LocalDeblur Map`.

18 Local MonoRes protocol

Protocol designed to apply the *MonoRes* method (Vilas et al., 2018) in *Scipion*. *MonoRes* is an automatic accurate method developed to compute the local resolution of a 3D map based on the calculation of the amplitude of the monogenic signal after filtering the map at different frequencies.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-xmipp`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 122 (A))
- Protocol form parameters (Fig. 122 (B)):

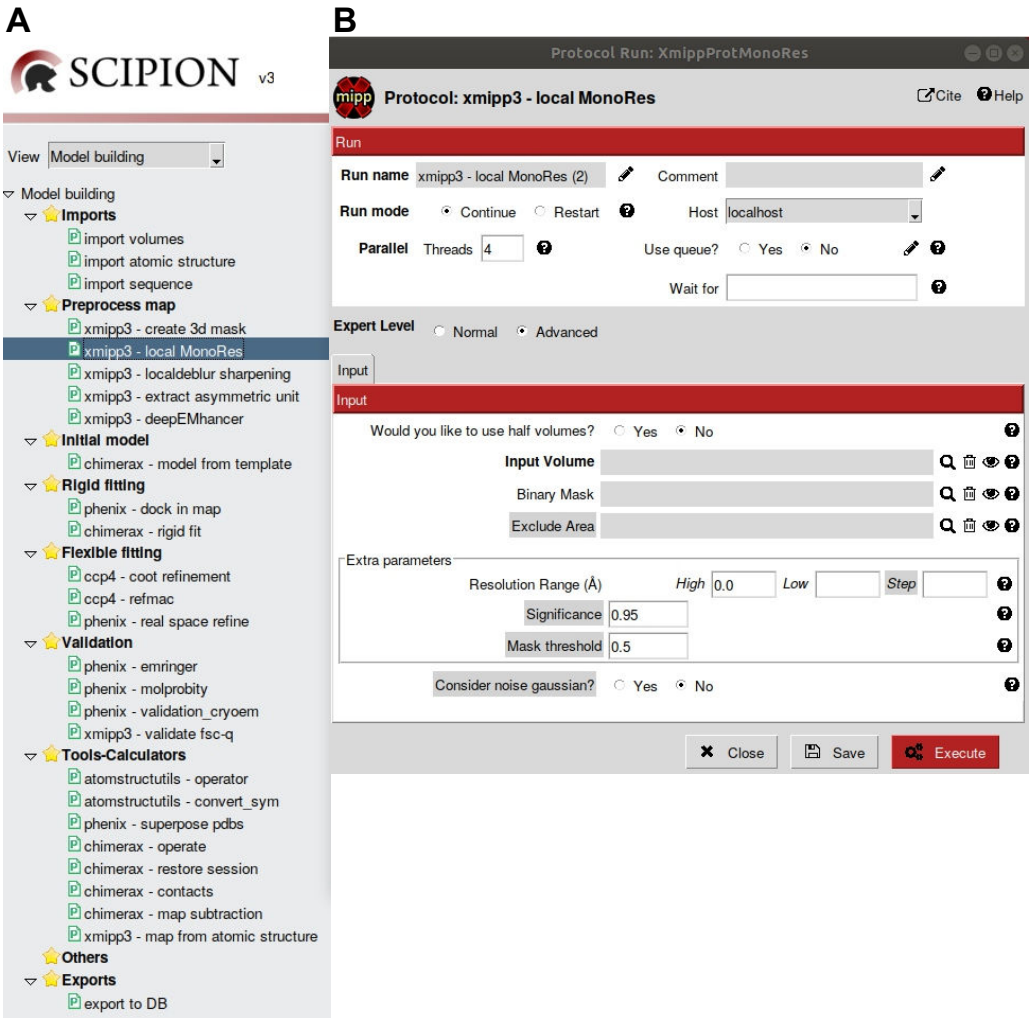


Figure 122: Protocol xmipp3 - local MonoRes. A: Protocol location in *Scipion* menu. B: Protocol form.

– `Would you like to use half volumes?`: Option "No" has been selected by default, with the box `Input Volume` to fill in with the volume, imported or generated in *Scipion*. However, since the noise estimation needed to determine the local resolution is based on half volumes, select "Yes" whenever half volumes are available. A couple of boxes will thus be opened to select both half volumes, `Volume Half 1` and `Volume Half 2`. If you

want an appropriate computation of local resolution try to use half maps, or raw average maps otherwise. Avoid using postprocessed or sharpened maps.

- `Binary Mask`: Mask that will be overlapped to the map volume in order to indicate which points of the map are specimen and which are not.
- `Exclude area`: Advanced parameter to select part of the specimen that should be excluded from the estimation of local resolution, for example in viruses to exclude the inner genetic material or in membrane proteins to exclude fosfolipids. Remark that we are talking about part of the specimen (signal and not noise) in which we are simply not interested.
- `Extra parameters`:
  - `Resolution Range (Å)`: Interval of resolution expected, from the maximum resolution (`High` = 0.0 by default), to the minimal resolution (`Low`) of the map volume. This parameter is empty by default and *MonoRes* will try to estimate it. `Step` is an advanced parameter that indicates the fraction of resolution of each interval in the range contained between the max and min resolution.
  - `Significance`: Advanced parameter that determines the significance of the hypothesis test computed to calculate the resolution (0.95 by default).
  - `Mask threshold`: Advanced parameter that indicates the density value required to get a binary mask in case there is none (0.5 by default). Density values below the threshold will be changed to 0 and values above the threshold will be changed to 1.
  - `Consider noise gaussian?`: "No" by default has to be changed to "Yes" if you prefer to establish the premise that the noise follows a gaussian distribution.

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol

at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and a menu window will be opened (Fig. 123):

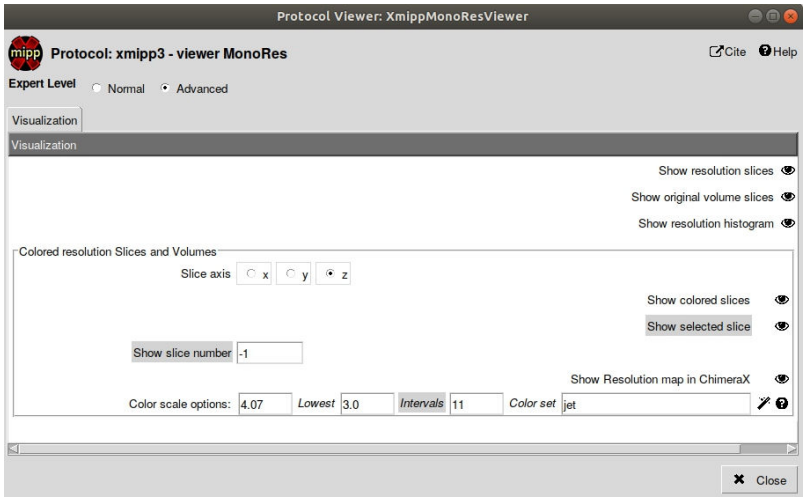


Figure 123: Protocol xmipp3 - local MonoRes. Menu to visualize results.

  - `Show resolution slices`: Map resolution slices are opened with *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`), the default *Scipion* viewer.
  - `Show original volume slices`: Original map slices are opened with *ShowJ*.
  - `Show resolution histogram`: Number of map voxels that show a certain resolution.
  - `Colored resolution Slices and Volumes`: Box that allows to display local resolution of map and slices according to a specific color code.

    * `Slice axis`: Select the perpendicular axis to visualize the slices. The `Z` axis is perpendicular to the screen.
    * `Show colored slices`: Map slices 34, 45, 56 and 67 of local resolution along the axis selected previously.
    * `Show selected slice`: Advanced parameter to show by default the 51 local resolution slide, or any other selected along the axis selected previously.
    * `Show slice number`: Advanced parameter to select the slice number to be shown by `Show selected slice`.
    * `Show Resolution map in ChimeraX`: The resolution map is shown using *ChimeraX*. Left hand bar indicates resolution colour code.
    *
        · `Color scale options`: Highest value of the resolution range.
        · `Lowest`: Lowest value of the resolution range.
        · `Intervals`: Number of resolution intervals from the highest to the lowest range value.
        · `Color set`: Color to apply to the local resolution map (`http://matplotlib.org/1.3.0/examples/color/colormaps_reference.html`).

    `Note`: Remark that on the right side you have a wizard to control color params.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `xmipp3 - local MonoRes -> resolution_Volume`;
    Volume (x, y, and z dimensions, sampling rate).
  - `SUMMARY` box:
    `Highest resolution` and `Lowest resolution`.

19 Model from Template protocol

Protocol designed to obtain a structure model for a target sequence in *Scipion*. Target structure is predicted by sequence homology using *Modeller* (Sali and Blundell, 1993) web service in *ChimeraX*.
WARNING: Working with *Modeller* requires a license key, which can be requested free of charge for academic users. Try to have this license key before starting the protocol execution.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-chimera`
  - Multiple sequence alignment tools: `Clustal Omega`, `MUSCLE`
- *Scipion* menu: `Model building -> Initial model` (Fig. 124 (A))
- Protocol form parameters (Fig. 124 (B)):

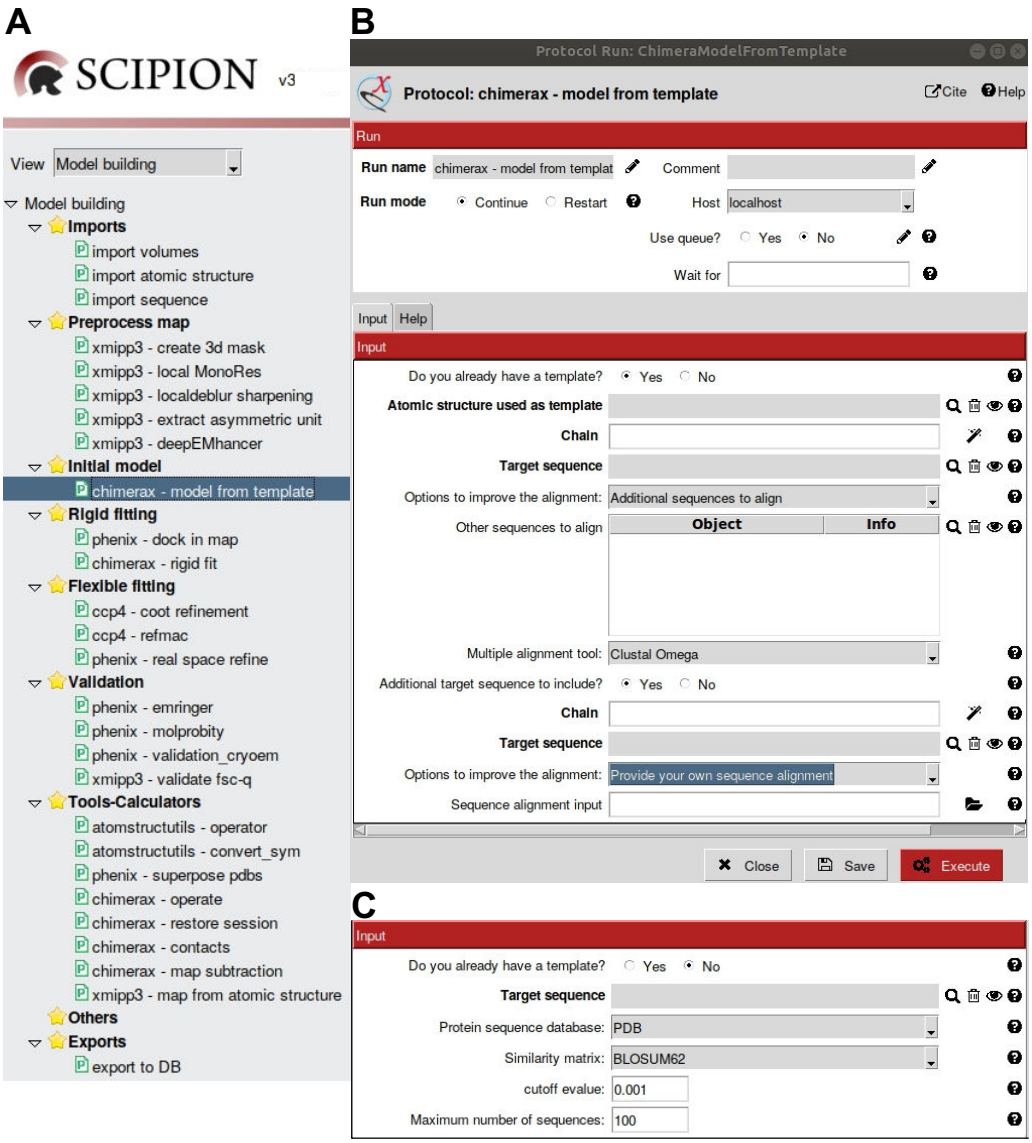


Figure 124: Protocol chimerax - model from template. A: Protocol location in *Scipion* menu. B: Protocol form: Option "template available". C: Protocol form: Option "looking for template".

- `Input` section
  - `Do you already have a template?`: Select ``Yes'' if you have found your *template* in a previous similarity searching step. Select ``No''

if you do not have any *template* to start the homology modeling and you would like to search for one.

* Option ``Yes'' (Fig. 124 (B))
  - `Atomic structure used as template`: Atomic structure previously downloaded in *Scipion*. This structure was selected by sequence homology, i.e. by looking for the structurally characterized sequence more similar (with higher identity) to the target sequence.
  - `Chain`: Specific monomer of the macromolecule that has to be used as structure *template* of the `target sequence`. Use the wizard on the right side of `Chain` parameter to select that chain.
  - `Target sequence`: Sequence previously downloaded in *Scipion*. This sequence has to be modeled following the structure skeleton of the selected *template*.
  - `Options to improve the alignment`: *Modeller* provides structural models of the *target sequence* based on a sequence alignment, in which at least sequences of *template* and *target* have to be included. Three options can be considered to improve this alignment:
    1. `None`: No more sequences are going to be included in the alignment except *model* and *target* sequences. Correlative param:
       ** `Alignment tool for two sequences`: Select one of the three available alignment methods, `Bio.parirwise2` (by default), `Clustal Omega`, `MUSCLE`.
    2. `Additional sequences to align` if you want to perform a multiple sequence alignment adding other sequences already downloaded in *Scipion*. Additional sequences, others than `template` and `target` sequences, are required to accomplish this multiple alignment. Correlative params:
       ** `Other sequences to align`: Box to complete with the additional sequences used to perform the multiple sequence align-

ment. All of them were previously downloaded in *Scipion*.
** `Multiple alignment tool`: Select between `Clustal Omega` and `MUSCLE` methods.

3. `Provide your own sequence alignment`: If you want to include other sequences in the alignment by providing your own sequence alignment. Correlative param:
** `Sequence alignment input`: Complete this box with the help of the right side browser including the sequence alignment file that you already have saved in your computer. Different alignment formats are available (`https://www.cgl.ucsf.edu/chimerax/docs/user/commands/open.html`). An example of alignment in `fasta` format can be seen below (Use case 3).

· `Additional target sequence to include?`: Select ‘‘`Yes`’’ if you'd like to obtain a multimer *model* by using two *target* sequences and the same multimer *template*. The params to complete the option ‘‘`Yes`’’ are identical to those already shown, with the exception of the `Atomic structure used as template`, already completed. However, no one of those params will appear in case you select ‘‘`No`’’ in order to obtain a *model* by using only one *target* sequence.

∗ Option ‘‘`No`’’ (Fig. 124 (C))

· `Target sequence`: Sequence previously downloaded in *Scipion*. This sequence has to be modeled following the structure skeleton of the *template* that you are going to select among the retrieved entries found by the similarity searching tool.

· `Protein sequence database`: Select one of the two suggested protein sequence databases, PDB and NR. Press the ‘‘`?`’’ symbol on the right to see the meaning of each one. Remark that the NR database allows you to get entries with, as well as without, atomic structure associated. These ones, which do not provide *templates*, could be useful to build a better sequence alignment.

· `Similarity matrix`: Select one of the "substitution matrix" to assign a score to any couple of residues in the alignment (`https://www.ncbi.nlm.nih.gov/blast/html/sub_matrix.html`).

· `cutoff evalue`: Maximum statistic value required to include a retrieved element in the hit list.

· `Maximum number of sequences` that you'd like to retrieve from the database.

– `Help` section

Follow this section steps to run *Modeller* via web service in *ChimeraX* and to select and save one of the retrieved models in *Scipion* framework.

- Protocol execution:

Adding specific template-target label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol on the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

Several *ChimeraX* windows will be opened after executing the protocol with different contents according to the distinct form param options. Although we are going to detail some of them through several use cases (see below), designed to ilustrate different applications of this protocol, as well as the procedure to follow in each case, in general we can predict the opening of *ChimeraX* graphics window and a sequence alignment window. Usually, in both windows the *template* sequence is green highlighted (see an example of these windows in Fig. 27). Main steps to follow ahead are:

– Ask for model(s) to *Modeller* by selecting `Tools -> Sequence -> Modeller Comparative` in the main menu of *ChimeraX* graphics window.

- Complete the new window opened for `Modeller Comparative` with the sequence alignment that includes the *template* and with the *target*(s) sequence(s), *Modeller* license key, multichain model, number of models retrieved by *Modeller*, and `Advanced` options like the building of models with hydrogens, as well as *model* inclusion of heteroatoms or water molecules. An example of completed *Modeller* window can be observed in Fig. 28 (A). By pressing `OK` the computation starts. The status of the job can be checked in the lower left corner of *ChimeraX* graphics window.
- After a while a new panel window will show the retrieved models of the `target` sequence (Fig. 28 (B)). Two statistics assess these models: `GA341`, statistical potentials derived-score, and `zDOPE`, normalized Discrete Optimized Protein Energy, atomic distance depending-score. Reliable models show `GA341` values higher than 0.7, and negative `zDOPE` values correspond to better models. Retrieved models can be checked in *ChimeraX* `Tools -> Models`. One of them should be selected (Fig. 28 (C)).
- Rename the selected model, for example `#n_initial` to `#n_final` with the command line:
  `rename #n_initial id #n_final`
- Save the retrieved model selected according to the new model number in the *Scipion* track system (`#n_final`) shown in *ChimeraX* `Tools -> Models` by writing in *ChimeraX* command line:
  `scipionwrite #n_final prefix user_defined_name_`

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures are referred to the origin of coordinates in *ChimeraX*. To show the relative position of the atomic structure, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes and selected atomic structure `model` are model numbers `#1` and `#2`, respectively, in *ChimeraX* `Models` panel if only one structure has been saved.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `chimerax - model from template -> name of the new atomic structure`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.
  - `SUMMARY` box:
    Produced files:
    we have some result

USE CASES

- `Use Case 1: Input atomic structure as template, 1 target sequence, Option ‘‘None’’ to improve the alignment`
  Aim: To model a *target* sequence using one chain of a homologous atomic structure as *template*, using only the sequences of *target* and *template* in the sequence alignment.

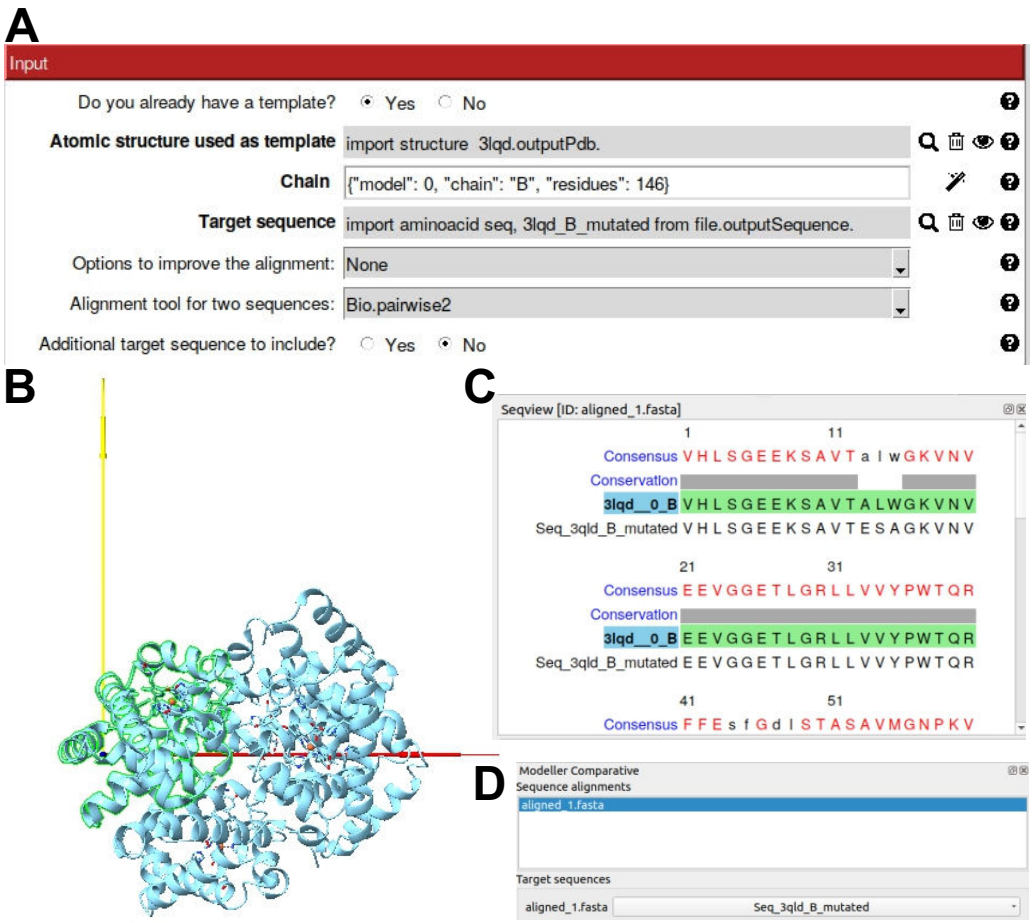


Figure 125: (A) Protocol form of chimerax - model from template. (B) *ChimeraX* view of the *template* atomic structure, highlighted in green the `chain B` selected to perform the modeling. (C) *ChimeraX* sequence view panel showing the sequence alignment between the *template* `chain B` sequence, greeen highlighted, and the *target* sequence. (D) Upper part of the *ChimeraX* `Modeller comparative` panel showing the sequence alignment selected, shown in (C), and the selected *target* sequence.

Protocol execution: Complete the protocol form as indicated in Fig. 125 (A). Follow the general procedure shown above (Protocol execution section). Windows (B) and (C) will appear. Open and complete the `Modeller Comparative`

panel as indicated in (D) and wait for a while. After getting the retrieved models, if you want to select, for example, the *target* model `#3.2`, write in the command line:
`rename #3.2 id #4`
`scipionwrite #4 prefix model_3_2_`
And *ChimeraX* `Quit` to close the protocol. Visualize your results.

- `Use Case 2: Input atomic structure as template, 2 target sequences, Option ‘‘Additional sequences to align’’ to improve the alignment, multichain modeling option in Modeller`
  Aim: To model simultaneously two *target* sequences to obtain a multichain model using two chains of a homologous atomic structure as *templates*, using other additional sequences than *target* and *template* in the sequence alignment.

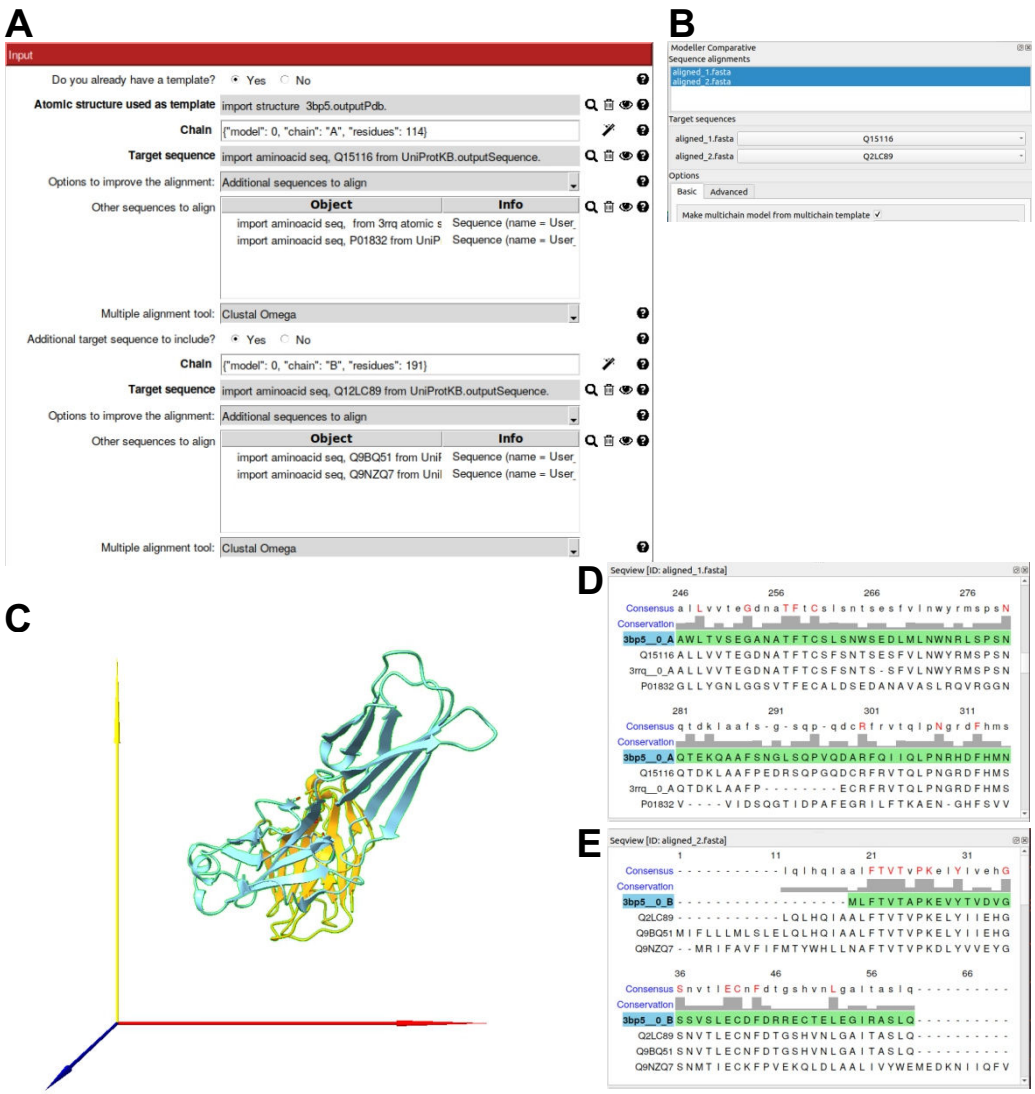


Figure 126: (A) Protocol form of chimerax - model from template. Remark that two additional sequences other than *template* and *target* sequences have been selected to improve both alignments. (B) Upper part of the *ChimeraX* `Modeller comparative` panel showing the two sequence alignment selected, shown in (D) and (E), and the two selected *target* sequences.(C) *ChimeraX* view of the *template* atomic structure, highlighted in yellow the `chain A` and in green the `chain B` selected to perform the modeling. (D) *ChimeraX* sequence view panel showing the sequence alignment between the *template* `chain A` sequence, greeen highlighted, the *target* sequence (`Q15116`) and two more additional sequences. (E) *ChimeraX* sequence view panel showing the sequence alignment between the *template* `chain B` sequence, greeen highlighted, the *target* sequence (`Q12LC89`) and two more additional sequences.

Protocol execution: Complete the protocol form as indicated in Fig. 126 (A). Follow the general procedure shown above (Protocol execution section). Windows (C), (D) and (E) will appear. Open and complete the `Modeller Comparative` panel as indicated in (B) and wait for a while. After getting the retrieved models, if you want to select, for example, the *target* model `#3.2`, write in the command line:

```
rename #3.2 id #4
scipionwrite #4 prefix model_3_2_
```

And *ChimeraX* `Quit` to close the protocol. Visualize your results.

- `Use Case 3: Input atomic structure as template, 2 target sequences, Option ‘‘Provide your own sequence alignment’’ to improve the alignment, multichain modeling option in Modeller`

  Aim: To model simultaneously two *target* sequences to obtain a multichain model using two chains of a homologous atomic structure as *templates*, using your own sequence alignment.

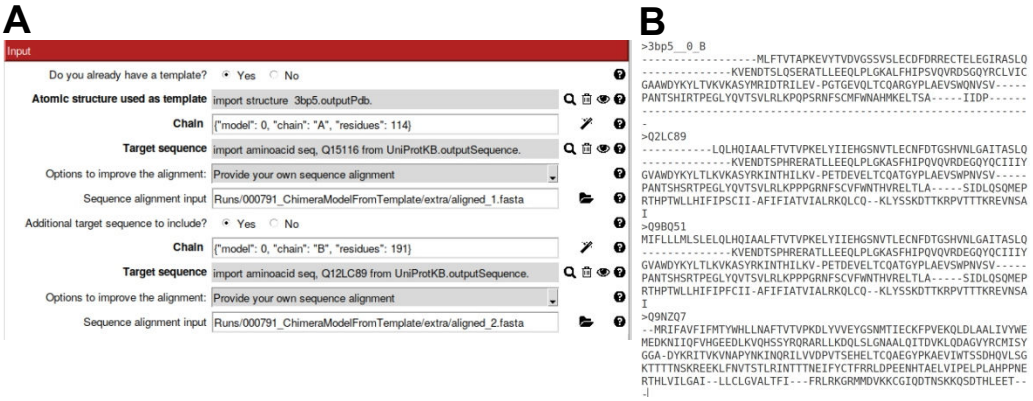


Figure 127: (A) Protocol form of chimerax - model from template. Remark that your own sequence alignment containing the sequences of *template* and *target*, among others, have been selected. (B) Example of sequence alignment in `fasta` format (file “aligned_2.fasta”) that includes the *template* `chain B` sequence, the *target* sequence (`Q12LC89`) and two more additional sequences.

Protocol execution: Complete the protocol form as indicated in Fig. 127 (A).

Follow the general procedure shown above (Protocol execution section). Remark that you already have a sequence alignment file for each *target* saved in your computer. An example can be seen in Fig. 127 (B). Windows (C), (D) and (E) of previous Fig. 126. Open and complete the `Modeller Comparative` panel as indicated in (B) and wait for a while. After getting the retrieved models, if you want to select, for example, the *target* model `#3.2`, write in the command line:

```
rename #3.2 id #4
scipionwrite #4 prefix model_3_2_
```

And *ChimeraX* `Quit` to close the protocol. Visualize your results.

- `Use Case 4: Input 1 target sequence, PDB searching database`
  Aim: To model a *target* sequence without previous information of a possible atomic structure *template*.

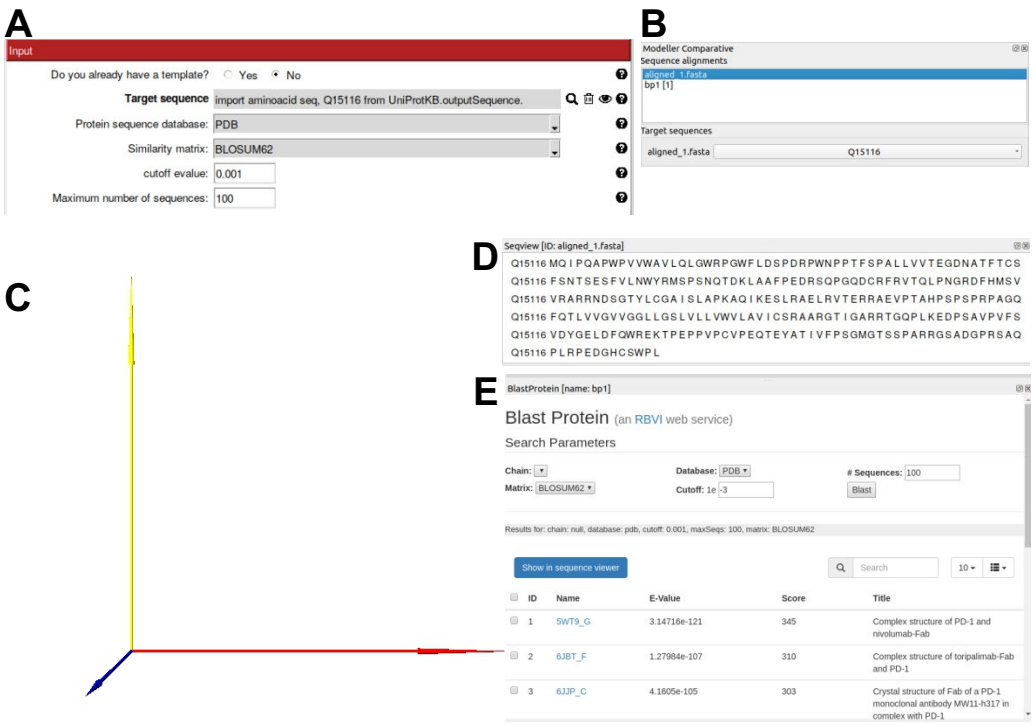


Figure 128: (A) Protocol form of chimerax - model from template that includes the *target* sequence that we'd like to model. (B) Upper part of the *ChimeraX* `Modeller comparative` panel showing the sequence alignment and the *target* sequence selected. (C) *ChimeraX* graphics window empty before opening the *template* atomic structure. (D) *ChimeraX* sequence view panel showing the *target* sequence. (E) *ChimeraX* BlastProtein panel showing the retrieved results from PDB database.

Protocol execution: Complete the protocol form as indicated in Fig. 128 (A). Follow the general procedure shown above (Protocol execution section). Remark that in this case there is no *template* atomic structure. Instead, an empty *ChimeraX* graphics window will appear (C) together with the *target* sequence that we'd like to model and `BLASTP` retrieved results (E). Note that it could take some seconds the opening of the `BlastProtein` panel. Have a look to these results and select one of them as possible `template` of your `target` sequence. In this particular case, for example, we are going to choose the first one

(`5WT9`). Open this atomic structure en *ChimeraX* by writing in the command line:

```
open 5wt9
```

At this point, open the `Modeller Comparative` panel and complete it as indicated in Fig. 128 (B). Wait for a while. After getting the retrieved models, if you want to select, for example, the *target* model `#3.2`, write in the command line:

```
rename #3.2 id #4
scipionwrite #4 prefix model_3_2_
```

And *ChimeraX* `Quit` to close the protocol. Visualize your results.

20 Phenix EMRinger protocol

Protocol designed to assess the geometry of refined atomic structures regarding electron density maps in *Scipion* by using *EMRinger* (Barad et al., 2015). Integrated in cryo-EM validation tools of *Phenix* software suite (`https://www.phenix-online.org/`) and created as an extension of the X-ray crystallography validation tool *Ringer*, *EMRinger* tool computes the amount of rotameric angles of the structure side chains as a function of map value to assess the goodness of the fitting to the cryo-EM density map.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-phenix`
  - PHENIX software suite (tested for versions 1.13-2998, 1.16-3549, 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Validation` (Fig. 129 (A))
- Protocol form parameters (Fig. 129 (B)):

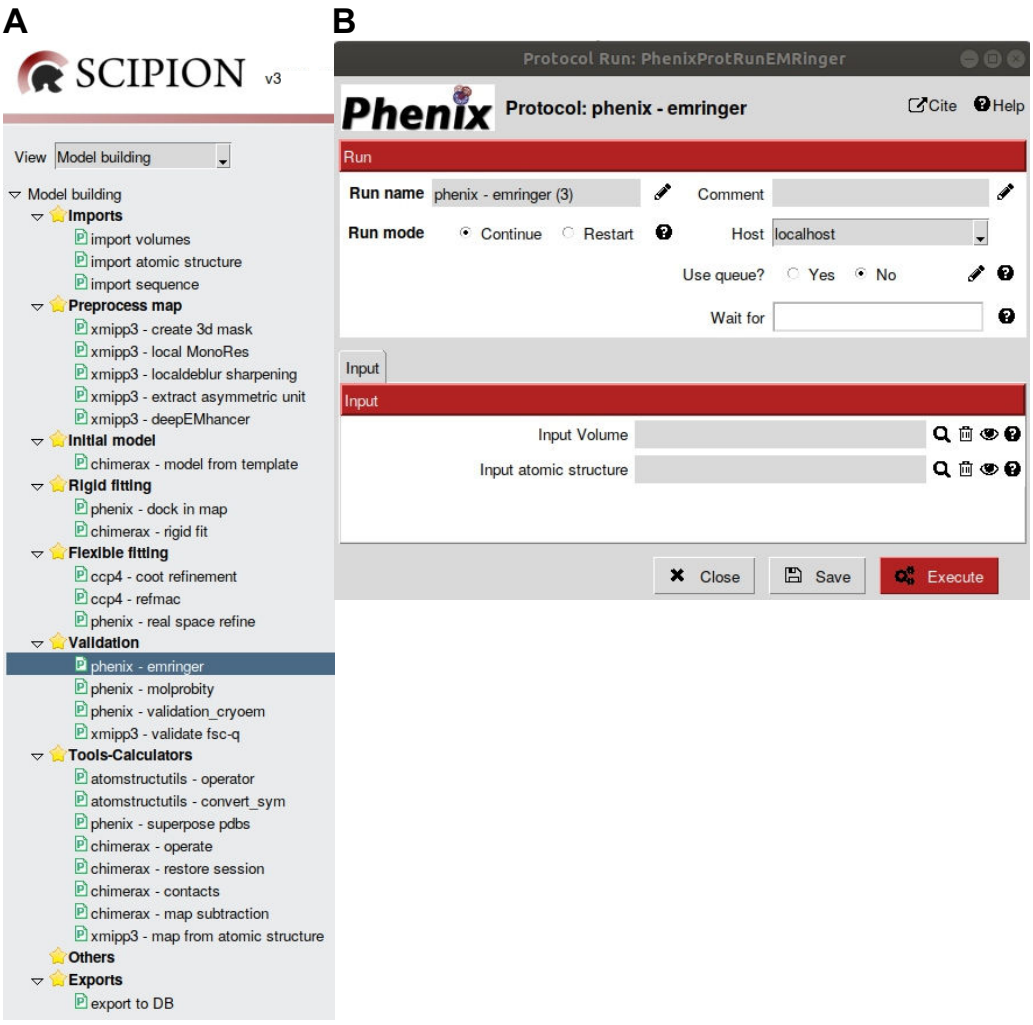


Figure 129: Protocol phenix - emringer. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input Volume`: Electron density map previously downloaded or generated in *Scipion*.
  - `Input atomic structure`: Atomic structure previously downloaded or generated in *Scipion* and fitted to the input electron density map.

- Protocol execution:

Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol on the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and the results window will be opened (Fig. 130).

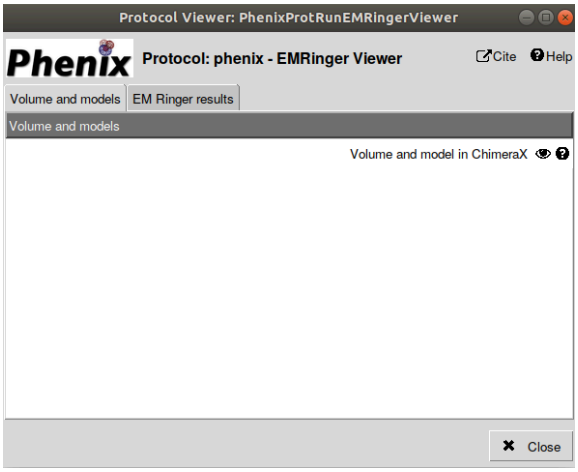


Figure 130: Protocol phenix - emringer. Taps to visualize Volume and models and *EMRinger* results.

Two taps are shown in the upper part of the results window:

  - `Volume and models`: *ChimeraX* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85).

– `EMRinger Results` (Fig. 131):

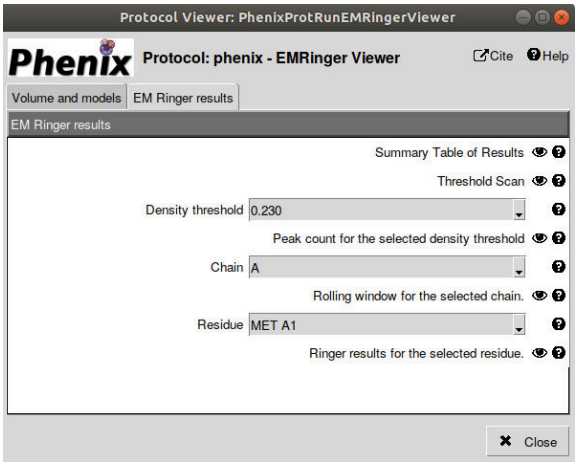


Figure 131: Protocol phenix - emringer. Menu to visualize *EMRinger* results.

∗ `Summary Table of Results` (Fig. 132):

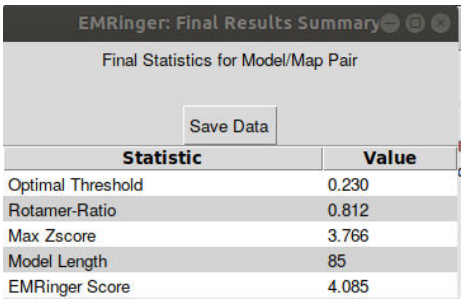


Figure 132: Protocol phenix - emringer. Final *EMRinger* results.

· `Optimal Threshold`: `Electron Potential` cutoff value of the volume, in a range of 20, at which maximum values of `EMRinger Score` and `Percentage of Rotameric Residues` are reached.

· `Rotamer-Ratio`: `Percentage of Rotameric Residues` at the `Optimal Threshold`.

· `Max Zscore`: Z-score indicating the significance of the distribution at the `Optimal Threshold`; in other words, probability of

finding a certain number of rotameric residues at a specific side chain dihedral angle, among the total number of map peaks found above the `Optimal Threshold`, assuming a binomial distribution of rotameric residues B(`n, p`) (n: total number of map peaks found above the `Optimal Threshold`; p: 39/72; with map sampling every 5º, 39 angle binds are considered rotameric from a total of $360/5 = 72$).

- `Model Length`: Total number of residues of the model considered in EMRinger computation, non-γ-branched, non-proline aminoacids with a non-H γ atom.
- `EMRinger Score`: Highest Z-score, rescaled regarding model length, across the range of `Electron Potential` thresholds. Since the Z-score is rescaled to the `EMRinger Score` according model length, `EMRinger Score` allows suitable comparisons among different model-map pairs. `EMRinger Score` of 1.0 is usual for initial models refined regarding 3.2-3.5Å resolution maps. For high-quality models with high resolution, `EMRinger Score` values higher than 2 are expected.

* `Threshold Scan`: Plots of `EMRinger Score` (blue line) and `Percentage of Rotameric Residues` (red line) regarding the `Electron Potential` threshold. The maximum value of `EMRinger Score` stablishes the `Optimal Threshold`.
* `Density threshold`: Box to select one of the 20 volume density cutoff values at which the `Percentage of Rotameric Residues` has been computed. The `Optimal Threshold`, at which the `EMRinger Score` was obtained, is shown by default.
* `Peak count for the selected density threshold`: Histograms counting rotameric (blue) and non-rotameric (red) residues at the selected `Electron Potential Threshold`.
* `Chain`: Box to select one of the chains of the model. By default, the name of the first chain is shown.

    * `Rolling window for the selected chain`: The analysis of EMRinger rolling window, performed on rolling sliding 21-residue windows along the primary sequence of monomers, allows to distinguish high quality regions of the model.
    * `Residue`: Box to select one residue, with at least one `Chi` angle (non-H γ atom-containing), located in the specific position indicated in the primary sequence of one of the monomer chains indicated.
    * `Ringer results for the selected residue`: Individual plots for each `Chi` angle of the selected residue. Detailed numeric values are shown in the `extra/*.csv` file.

- Summary content:

  `SUMMARY` box:
  Statistics included in the above `Summary Table of Results` (an example can be observed in Fig. 52 (6).

21 Phenix MolProbity protocol

Protocol designed to assess in *Scipion* the geometry of refined atomic structures without considering electron density maps by using *MolProbity* (Davis et al., 2004). Integrated in cryo-EM validation tools of *Phenix* software suite (`https://www.phenix-online.org/`), *MolProbity* tool validates geometry and dihedral-angle combinations of atomic structures. *MolProbity* scores can guide the refinement process of the atomic structure to get a good fitting of the atomic structure to the cryo-EM density map. Adding a volume as input in `Phenix MolProbity` protocol is possible for *PHENIX* v. 1.13 and `Real Space Correlation` coefficients between map and model-derived map will thus be calculated. Additionally, experimental electron density maps give sense to the interpretation of geometry outliers.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`

  - *Scipion* plugin: `scipion-em-phenix`
  - PHENIX software suite (tested for versions 1.13-2998, 1.16-3549, 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu: `Model building -> Validation` (Fig. 133 (A))
- Protocol form parameters (Fig. 133 (B)):

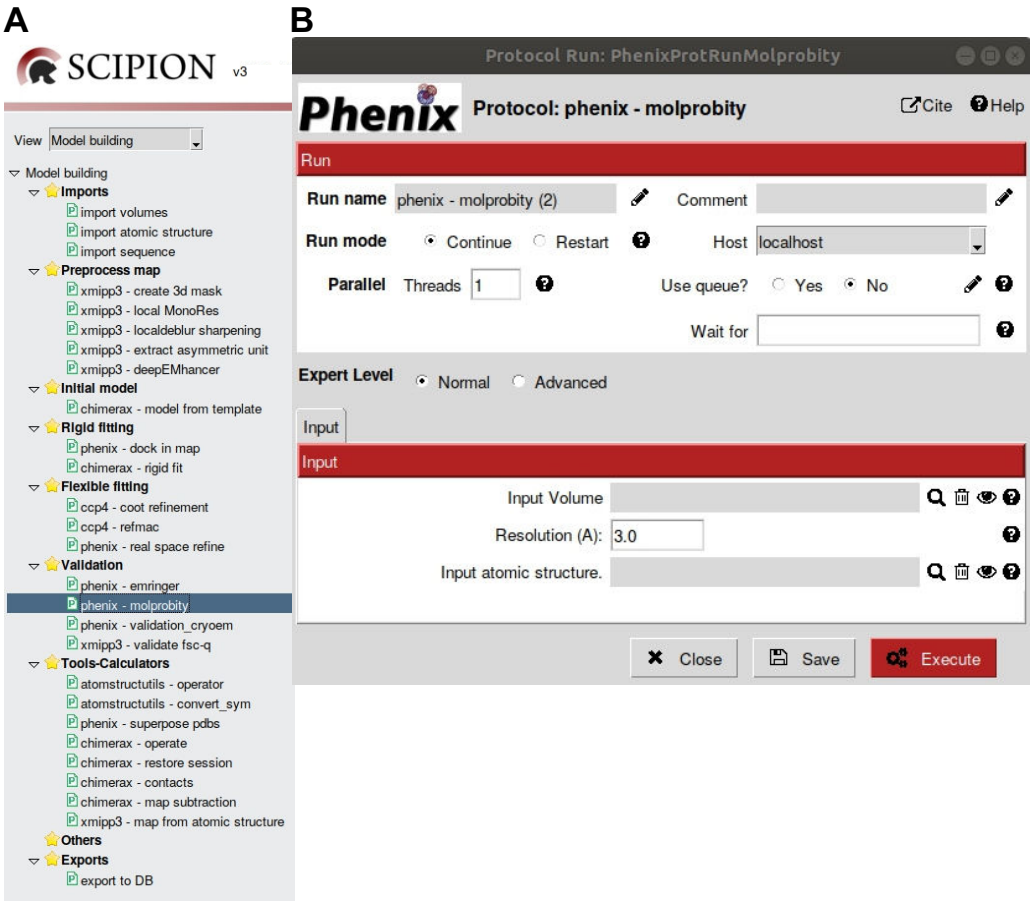


Figure 133: Protocol phenix - molprobity. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input Volume`: (Optional) Electron density map previously downloaded or generated in *Scipion*. Only with *PHENIX* v. 1.13 `Real Space Correlation` coefficients between map and model-derived map will be calculated.
- `Resolution (Å)`: Map resolution of the volume included in the `Input Volume` parameter.
- `Input atomic structure`: Atomic structure previously downloaded or generated in *Scipion* and fitted to the electron density map.

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and the results window will be opened (Fig. 134).

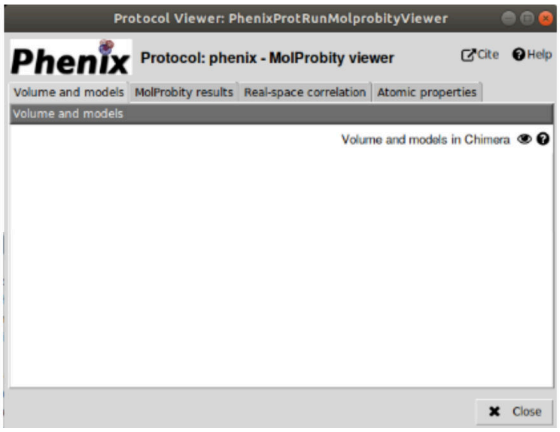


Figure 134: Protocol phenix - molprobity. Taps to visualize *MolProbity* and Real space correlation results. `Real-space correlation` tap only appears with *PHENIX* v. 1.13.

Four taps are shown in the upper part of the results window:

  - `Volume and models`: *ChimeraX* graphics window will be opened by default. Volume and atomic structure, if it is present, are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic

structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85).

– `MolProbity results` (Fig. 135):

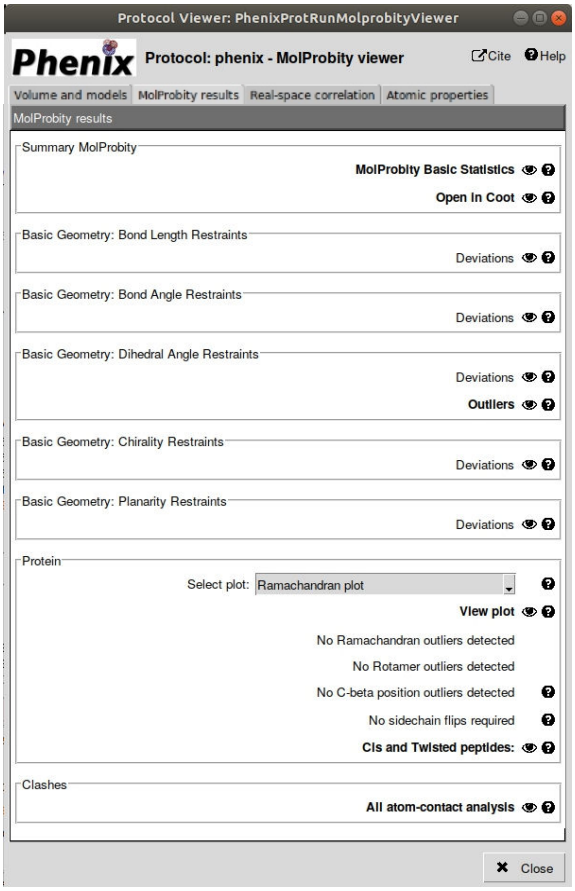


Figure 135: Protocol phenix - molprobity. *MolProbity* results.

∗ `Summary MolProbity`:

· `MolProbity Basic Statistics`: Statistics computed by the *Phenix* package to assess protein geometry using the same distributions as the MolProbity server:

`Ramachandran outliers`: Percentage of residues assessed that show an unusual combination of their ϕ (C-N-CA-C) and ψ (N-

CA-C-N) dihedral angles.

`Ramachandran favored`: Percentage of residues assessed that show a normal combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles. Ramachandran outliers and favored residues are detailed in the `Ramachandran plot`, shown below. Allowed residues are included in the small region comprised between favored and outlier regions of that plot.

`Rotamer outliers`: Percentage of residues assessed that adopt an unusual conformation of χ dihedral angles. Rotamer outliers, commonly used to characterize the conformation of protein sidechains, are detailed in Chi1-Chi2 plot, shown below.

`C-beta outliers`: Number of residues showing an unusual deviation (higher than 0.25 Å) of the Cβ from its ideal position. This deviation is an indicator of incompatibility between sidechain and backbone.

`Clashscore`: Score associated to the number of pairs of non-bonded atoms unsually close to each other, showing probable steric overlaps. Clashscore is calculated as the number of serious clashes per 1000 atoms. This value has to be as low as possible.

`RMS (bonds)`: Root-mean-square deviation of molecule bond lengths.

`RMS (angles)`: Root-mean-square deviation of molecule bond angles.

`Overall score`: *MolProbity* overall score represents the experimental resolution expected for the structure model. This value should be lower than the actual resolution. The lower the value, the better quality of the structure model.

· `Open in Coot`: Interactive visualization and structure modification tool for Ramachandran, Rotamer and Cβ outliers, as well as severe clashes. Coot graphics window will be centered on the specific atom or residue outlier when it is clicked. Improvements of the atomic structure are allowed in *Coot* and any modification can

be saved in *Scipion* as usual (look at `Help section: Saving an atomic structure after an interactive working session with Coot` (Appendix 8). The interactive *Coot* protocol box will appear hanging out of *MolProbity*.

- `Missing atoms`: For clarity, hydrogen atoms are not included.

* `Basic Geometry: Bond Length Restraints`: Bonded pairs of atoms outliers according to the bond restraints between pairs of bonded atoms. The `Deviations` table indicates the number of outliers and the number of restraints (in accordance with the geometry restraints library). Those outliers appear sorted by deviation (higher than 4 sigmas) in the `Outliers` list.
* `Basic Geometry: Bond Angle Restraints`: Bonded triplets of atoms outliers according to the angle restraints between triplets of bonded atoms. The `Deviations` table indicates the number of outliers and the number of restraints (in accordance with the geometry restraints library). Those outliers appear sorted by deviation (higher than 4 sigmas) in the `Outliers` list.
* `Basic Geometry: Dihedral Angle Restraints`: Bonded tetrads of atoms outliers according to the dihedral angle restraints between tetrads of bonded atoms. The `Deviations` table indicates the number of outliers and the number of restraints (in accordance with the geometry restraints library). Those outliers appear sorted by deviation (higher than 4 sigmas) in the `Outliers` list.
* `Basic Geometry: Chilarity Restraints`: Bonded tetrads of atoms outliers according to the chilarity restraints between tetrads of bonded atoms. The `Deviations` table indicates the number of outliers and the number of restraints (in accordance with the geometry restraints library). Those outliers appear sorted by deviation (higher than 4 sigmas) in the `Outliers` list.
* `Basic Geometry: Planarity Restraints`: Bonded groups of atoms outliers according to the planarity restraints between groups of bonded

atoms. The `Deviations` table indicates the number of outliers and the number of restraints (in accordance with the geometry restraints library). Those outliers appear sorted by deviation (higher than 4 sigmas) in the `Outliers` list.

* `Protein`: Box validating protein geometry:
  - `Select plot`: Box to select a plot to visualize: The Ramachandran plot or the Chi1-Chi2 plot.
  - `View plot`: Visualization of the plot previously selected.
  - `Ramachandran outliers`: List of Ramachandran residue outliers with their respective ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angle values.
  - `Rotamer outliers`: List of Rotamer residue outliers with their respective χ dihedral angles.
  - `C-beta outliers`: List of Cβ residue outliers with their respective angles (angular position of `C-beta` atom in radial space).
  - `Recommended Asn/Gln/His sidechain flips`: Asn, Gln and His residues, harboring asymmetric sidechains, recommended to be flipped to form favourable van der Waals contacts and hydrogen bonds.
  - `Cis and Twisted peptides`: Residues showing *cis* or *twisted* conformations that could be modeling errors.
* `Clashes`: Box to detail `All atom-contact analysis`, the list that contains all severe clashes (non-H atoms overlaping more than 0.4 Å) and that can be checked in *Coot*.

– `Real-space correlation`: (This tap will only appear with *PHENIX* v. 1.13 in case you include a electron density volume as input of the protocol) (Fig. 136):

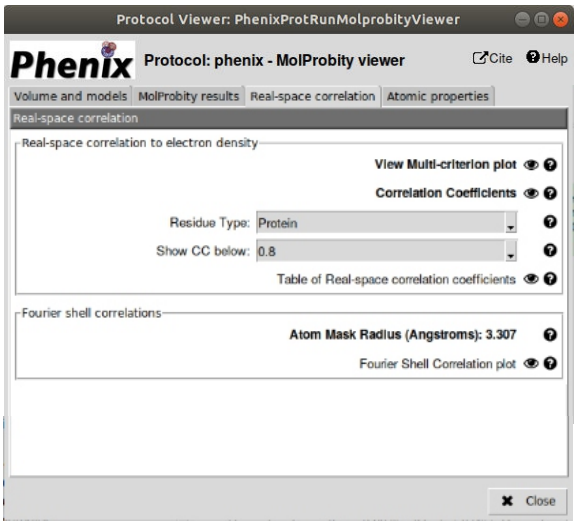


Figure 136: Protocol phenix - molprobity. Real-space correlation results.

* `Real-space correlation to electron density`:
  - `View Multi-criterion plot`: Plot showing cross-correlation and B-factor values for each residue of the macromolecule over 100-residue regions. Additional validation information, such as Ramachandran, Rotamer or Cβ outliers, is also detailed, as well as severe clashes.
  - `Correlations Coefficients`: Three Real-space correlation coefficients are computed (Afonine et al., 2018a):

    `Mask CC`: Correlation coefficient between experimental volume and model-derived map inside the mask region around the model.

    `Volume CC`: Correlation coefficient that considers only map regions with the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case the N points with the highest density, inside the molecular mask, are taken into account.

    `Peak CC`: Correlation coefficient that considers only map regions with the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case

the N points with the highest density, simultaneously present in the model-calculated map and in the experimental map, are taken into account.

- `Residue Type`: Box to select a type of residue: protein residue, other (for example heteroatom), water or everything. Protein residue is selected by default.
- `Show CC below`: Box to select the maximum limiting value of correlation coefficient shown by the residue type selected.
- `Table of Real-space correlation coefficients`: List displaying the selected residues with correlation coefficient value lower than the maximum value selected above. Residues showing the lower correlation might indicate errors in modeling of specific regions of the model.

* `Fourier shell correlations`:
  - `Atom Mask Radius (Angstroms)`: Radius of the "Fourier Shell", a spherical volume mask in Fourier space.
  - `Fourier Shell Correlation plot`: FSC plot regarding the inverse of the spatial frequency.

– `Atomic properties` (Fig. 137): Atom numerical properties:

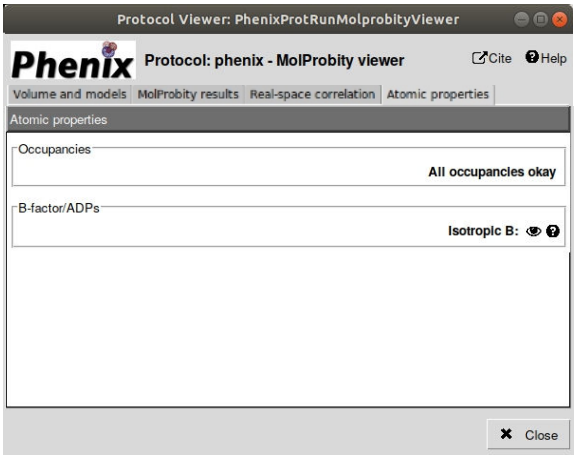


Figure 137: Protocol phenix - molprobity. Atomic properties results.

    * `Occupancies`: Atomic property used in crystallography. It represents the fraction of molecules in which a specific atom is in a given position or conformation at any given time. The sum of occupancies has to be 1 in total. Occupancies of zero indicate that no experimental data support the position of the atom in the model.
    * `B-factor/ADPs`: Temperature factors reflect the vibration status of the atoms in which the observed electron density constitutes an average of all the small motions. Low values (around 10) indicate low vibration of atoms, whereas high values (around 50) show atoms moving so much that locate them properly results difficult. This last is usually the case of atoms located at the protein surface.
        · `Isotropic B`: Temperature factor constrained to be the same in all three directions. By clicking here, a table showing the statistics (`Min`, `Max` and `Mean`) of the isotropic B-factor is displayed.

- Summary content:

  `SUMMARY` box:
  Main statistics included in the above *MolProbity* `Model Final Statistics` table (an example can be seen in Fig. 53 (7)).

22 Phenix Validation CryoEM protocol

Protocol designed to validate through multiple tools the geometry of an atomic structure and the correlation with a model-derived map in *Scipion* by using *phenix.validation_cryoem* program (Afonine et al., 2018a). Integrated in the *Phenix* software suite (versions higher than 1.13; `https://www.phenix-online.org/`), *phenix.validation_cryoem* tool can be applied to assess cryo-EM-derived models in real space. This program computes `Real Space Correlation` coefficients between map and model-derived map and, additionally, it assesses the geometry and dihedral-angle combinations of atomic structures with the aim of following the improvement of models along the refinement process. Validation *MolProbity* scores are shown at the end of the evaluation process.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-phenix`
  - PHENIX software suite (v. higher than 1.13, tested for versions 1.16-3549, 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Validation` (Fig. 138 (A))
- Protocol form parameters (Fig. 138 (B)):

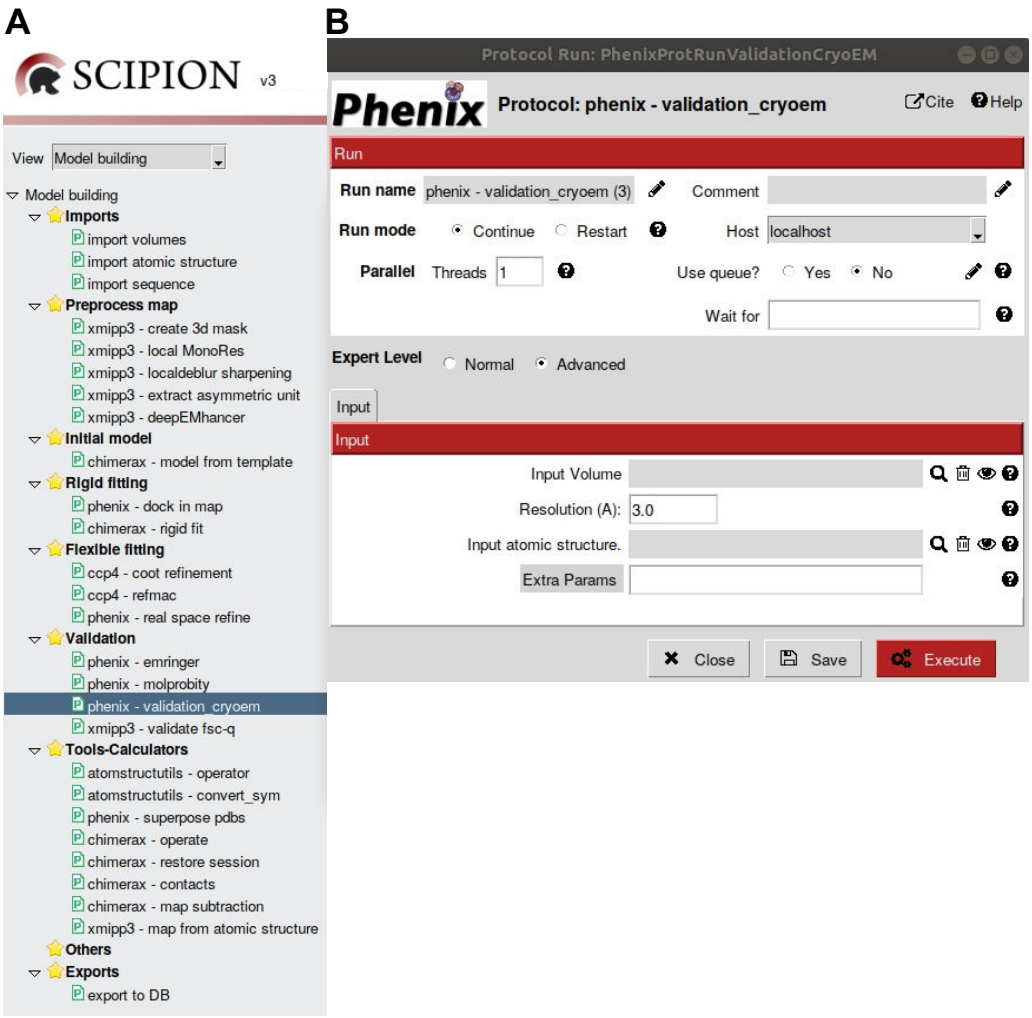


Figure 138: Protocol phenix - validation_cryoem. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input Volume`: Electron density map previously downloaded or generated in *Scipion*.
- `Resolution (Å)`: `Input Volume` resolution.
- `Input atomic structure`: Atomic structure previously downloaded or generated in *Scipion* and fitted to the electron density map.
- `Extra Params`: Advanced param that allows to add a string to the phenix

command including other *phenix.real_space_refine* program params. Syntax to add extra params: `paramName1 = value1 paramName2 = value2`

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and the results window will be opened (Fig. 139).

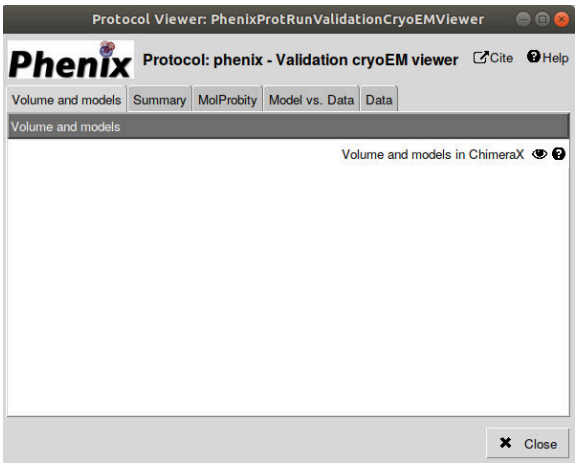


Figure 139: Protocol phenix - validation_cryoem. Taps to visualize *Validation CryoEM* results.

Five taps are shown in the upper part of the results window:

  - `Volume and models`: *ChimeraX* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordi-

nates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85).

– `Summary`: Three different summary tables are shown to describe the results obtained from `Model, Data` and `Model vs. Data` (Fig. 140). Concerning the atomic `Model`, numeric data from chains, residues, atoms and geometry are described, as well as main *MolProbity* statistics. `Data` summarizes experimental map box dimensions and different values of resolution computed with or without a mask. `Model vs. Data` details main real-space correlation coefficients.

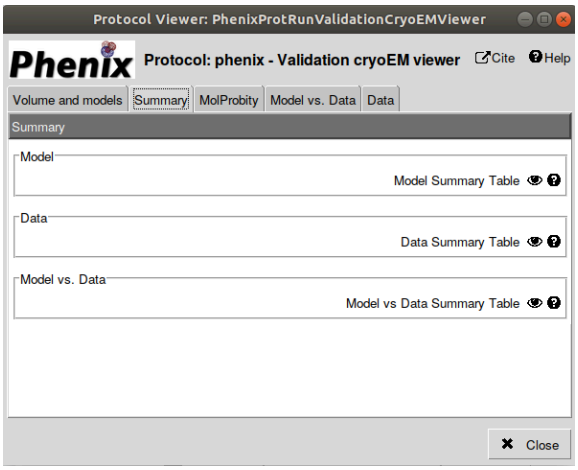


Figure 140: Protocol phenix - validation_cryoem. Summary tables of main*PHENIX* `validation_cryoem` results.

– `MolProbity`: Statistics concerning the atomic model, most of them obtained from *MolProbity* (Fig. 141).

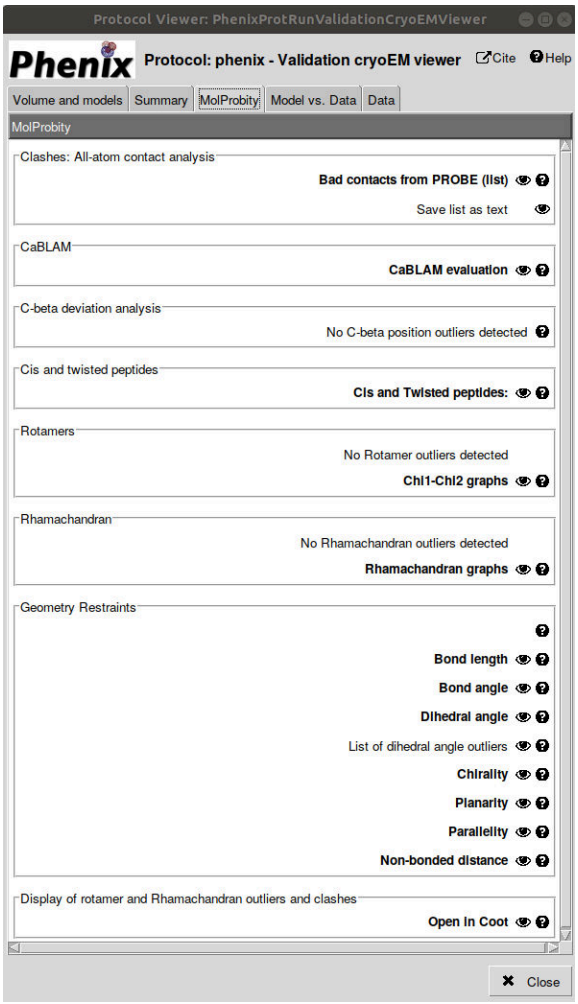


Figure 141: Protocol phenix - validation_cryoem. *MolProbity* and other statistics of the atomic model.

* `Clashes: All-atom contact analysis`: List that contains all severe clashes (non-H atoms overlaping more than 0.4 Å) found by PROBE. All these clashes can be visualized and solved graphically in *Coot*. If no hydrogens were present, REDUCE adds them before running PROBE. The list can be saved in a folder selected by the user.
* `CaBLAM: C-Alpha Based Low-resolution Annotation Method`: Method

designed to assess the mainchain geometry of the atomic model by using protein C_α geometry and to identify areas of probable secondary structure. Residues that fall outside contours of expected protein behaviour based on high-quality datasets are considered outliers.

* `C-beta deviation analysis`: C_β outliers deviate from ideal positions by more than 0.25Å. Ideal C_β position is determined from the average of the ideal C-N-CA-CB and N-C-CA-CB dihedrals. This measure is more sensitive than individual measures to both sidechain and mainchain misfittings. Its deviation is an indicator of incompatibility between sidechain and backbone.
* `Cis and twisted peptides`: Residues showing *cis* or *twisted* conformations that could be modeling errors. *cis* conformations are observed in about 5% of Prolines and 0.03% of general residues. Twisted peptides are almost certainly modeling errors.
* `Rotamers`: Rotamer outlier list contains residues that adopt an unusual conformation of χ dihedral angles. These outliers, commonly used to characterize the conformation of protein sidechains, are detailed in Chi1-Chi2 graph, shown below.
* `Rhamachandran`: Rhamachandran outlier list contains residues that show an unusual combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles. Most of the time, Ramachandran outliers are a consequence of mistakes during the data processing. These outliers are detailed below in Rhamachandran graphs.
* `Geometry Restraints`: Statistics for geometry restraints used in refinement. Although in general a fully refined structure should not have any outliers, exceptionally there are some of them that are obvious in high resolution electron density maps. Types of restraints:
  - `Bond Length`: This table indicates the number of outliers and the number of restraints (in accordance with the bond length restraints library). The list of outliers details the bonded pairs of atoms sorted by deviation (higher than 4 sigmas).

- `Bond Angle`: This table indicates the number of outliers and the number of restraints (in accordance with the bond angle restraints library). The list of outliers details the bonded triplets of atoms sorted by deviation (higher than 4 sigmas).
- `Dihedral Angle`: This table indicates the number of outliers and the number of restraints (in accordance with the side chain dihedral torsion - chi- angle restraints library). The list of outliers details the bonded tetrads of atoms sorted by deviation (higher than 4 sigmas).
- `Chilarity`: This table indicates the number of restraints (in accordance with the volume chilarity restraints library).
- `Planarity`: This table indicates the number of restraints (in accordance with the volume planarity restraints library).
- `Parallelity`: This table indicates the number of restraints (in accordance with the volume parallelity restraints library).
- `Non-bonded distance`: This table indicates the number of restraints (in accordance with the volume non-bonded distance restraints library).

* `Display of rotamer and Rhamachandran outliers and clashes`: Interactive visualization of outliers (Ramachandran, rotamer and C_β) and severe clashes with *Coot*.

– `Model vs. Data`: Real-space correlation coefficients between map and model-derived map (Fig. 142).

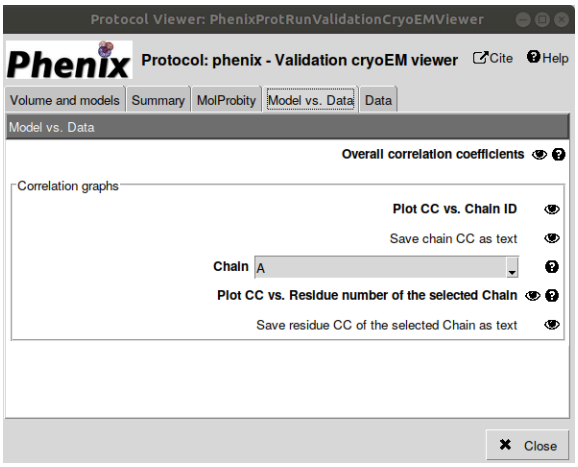


Figure 142: Protocol phenix - validation_cryoem. Real-space correlation results.

* `Overall correlation coefficients` (Afonine et al., 2018a):
  - `Mask CC`: Correlation coefficient between the model-derived map and the experimental map inside the mask region built around the model with a fixed radius. This comparison aims to fit the atomic centers.
  - `Box CC`: Correlation coefficient between the model-derived map and the whole experimental map. This comparison aims to assess the similarity of maps and remark map densities that have not been modeled.
  - `Volume CC`: Correlation coefficient between the model-derived map and the experimental map inside the mask region built around the model considering only model-derived map regions with the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case the N points with the highest density, inside the molecular mask, are taken into account. This comparison aims to fit the molecular envelope defined by the model-derived map.
  - `Peak CC`: Correlation coefficient between the model-derived map and the experimental map that considers only map regions with

the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case the N points with the highest density, simultaneously present in the model-calculated map and in the experimental map, are taken into account. This comparison aims to fit the strongest peaks in model-derived and experimental maps.

- `Main chain CC`
- `Side chain CC`

* `Correlation graphs`:
  - `Plot CC vs. Chain ID`: Plot of correlation coefficients regarding the chain IDs. These correlation coefficient values can be saved in a text file in the folder selected by the user.
  - `Plot CC vs. Residue number of the selected Chain`: Plot of correlation coefficients of each chain residues. The specific chain is selected by the user in the chain option box. These correlation coefficient values for each chain can be saved in a text file in the folder selected by the user.

– `Data` (Fig. 143): Computation of Resolution and FSC.

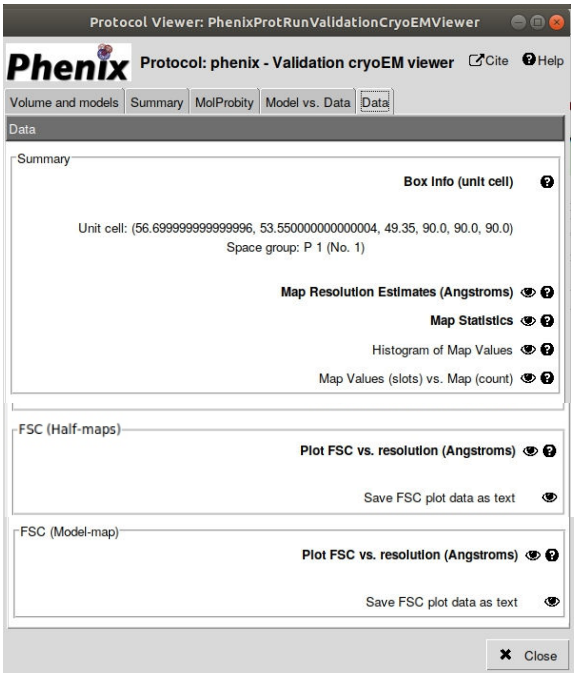


Figure 143: Protocol phenix - validation_cryoem. Experimental data results.

* `Summary`: Basic statistics about the maps and summary of resolution estimates.
  - `Box info (unit cell)`: Map cell dimensions (pixels).
  - `Map Resolution Estimates (Angstroms)`: Resolution estimates computed considering both map experimental data and model-derived information(with and without mask).
    - `Using map alone (d99)`: Resolution cutoff beyond which Fourier map coefficients are negligibly small. Calculated from the full map or from each one of half maps [d99 (half map 1), d99 (half map 2)].
    - `Overall Biso`: Overall isotropic B-value.
    - `d_model`: Resolution cutoff at which the model map is the most similar to the target (experimental) map. Requires map and model. For d_model to be meaningful, model is expected to

fit the map as well as possible.
`- d_model (B factors = 0)`: It tries to avoid the blurring of the map.
`- FSC (model) = 0`: d_FSC_model_0; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.
`- FSC (model) = 0.143`: d_FSC_model_0.143; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.143.
`- FSC (model) = 0.5`: d_FSC_model_0.5; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.5.
`- FSC (half map 1, 2) = 0.143`: d_FSC; Highest resolution at which the experimental data are confident. Obtained from FSC curve calculated using two half-maps and taken at FSC=0.143. The two half maps are required to compute this value.
`- Mask smoothing radius (Angstroms)`: Radius of the default soft mask used since sharp edges resulting from applying a binary mask may introduce Fourier artifacts.

* Fourier shell correlation taps:
  - `FSC(Half-maps)` (Only if two half maps have been added as inputs): FSC plot regarding the resolution (Å) and the spatial frequency (1/Å) based on half maps with and without masking. The intersections of the curves with FSC = 0.143 are shown. FSC plot data can be saved as text file in a folder selected by the user.
  - `FSC (Model-map)`: FSC plot regarding the resolution (Å) and the spatial frequency (1/Å) based on the experimental map and the model-derived map with and without masking. The intersections of the curves with FSC = 0.5 are shown. FSC plot data can be saved as text file in a folder selected by the user.

- Summary content:
  `Protocol output`: Empty.
  `SUMMARY` box:
  Main *MolProbity* statistics computed by the *Phenix* package to assess protein geometry using the same distributions as the MolProbity server:
  - `Ramachandran outliers`: Percentage of residues assessed that show an unusual combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles.
  - `Ramachandran favored`: Percentage of residues assessed that show an normal combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles. Ramachandran outliers and favored residues are detailed in the `Ramachandran plot`. Allowed residues are included in the small region comprised between the favored and the outlier region.
  - `Rotamer outliers`: Percentage of residues assessed that adopt an unusual conformation of χ dihedral angles. Rotamer outliers, commonly used to characterize the conformation of protein sidechains, are detailed in Chi1-Chi2 plot.
  - `C-beta outliers`: Number of residues showing an unusual deviation (higher than 0.25 Å) of the Cβ from its ideal position. This deviation is an indicator of incompatibility between sidechain and backbone.
  - `Clahscore`: Score associated to the number of pairs of non-bonded atoms unsually close to each other, showing probable steric overlaps. Clashscore is calculated as the number of serious clashes per 1000 atoms. This value has to be as low as possible.
  - `Overall score`: *MolProbity* overall score representing the experimental resolution expected for the structure model. This value should be lower than the actual resolution. The lower the value, the better quality of the structure model.

23 Phenix Real Space Refine protocol

Protocol designed to refine in real space an atomic structure into a map in *Scipion* by using *phenix.real_space_refine* program (Afonine et al., 2018b). Integrated in the *Phenix* software suite (`https://www.phenix-online.org/`), *phenix.real_space_refine* tool can be applied to refine cryo-EM-derived models in real space. This program computes `Real Space Correlation` coefficients between map and model-derived map and, additionally, it assesses the geometry and dihedral-angle combinations of atomic structures with the aim of getting the best map-fitted structure by reducing the number of geometry outliers. Validation *MolProbity* scores are shown at the end of the refinement process.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-phenix`
  - PHENIX software suite (v. higher than 1.13, tested for versions 1.16-3549, 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Flexible fitting` (Fig. 144 (A))
- Protocol form parameters (Fig. 144 (B)):

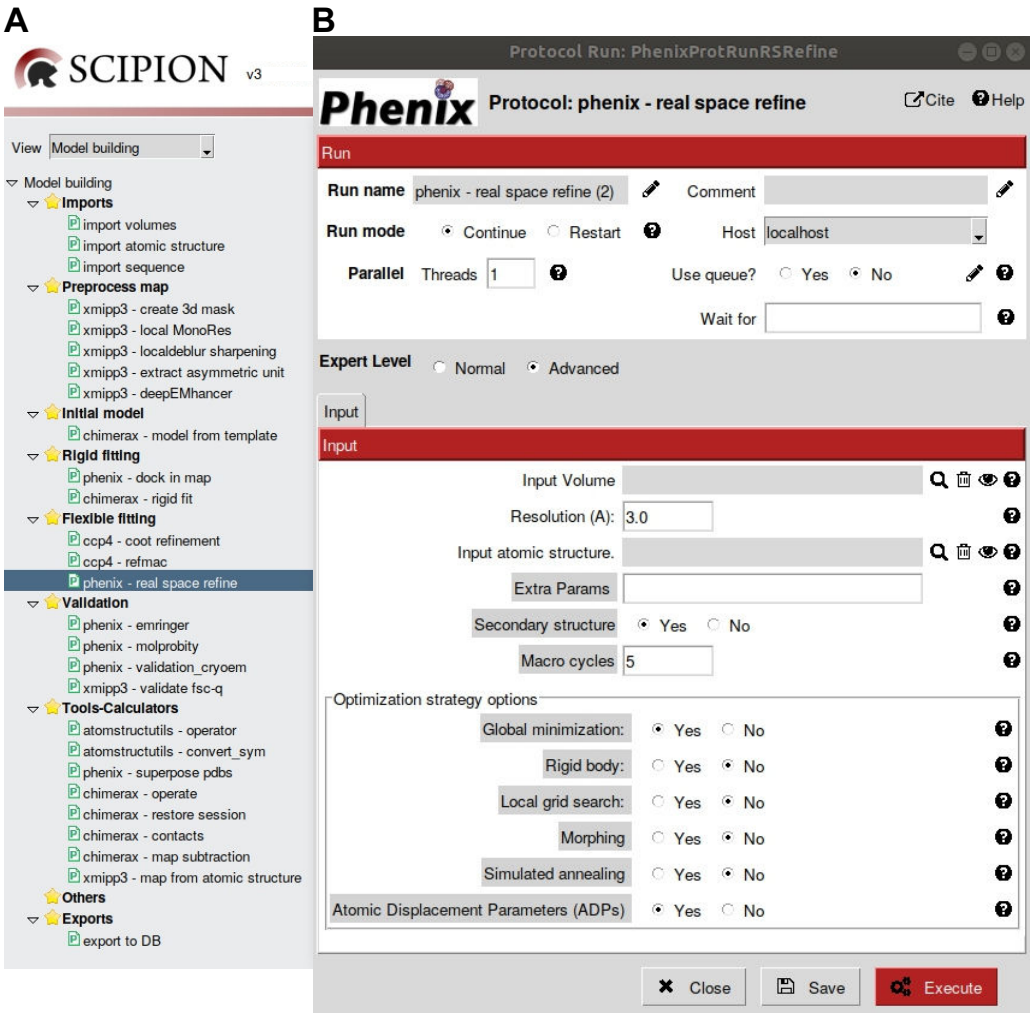


Figure 144: Protocol phenix - real space refine. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input Volume`: Electron density map previously downloaded or generated in *Scipion*.
- `Resolution (Å)`: `Input Volume` resolution.
- `Input atomic structure`: Atomic structure previously downloaded or generated in *Scipion* and fitted to the electron density map.
- `Extra Params`: Advanced param that allows to add a string to the phenix

command including other *phenix.real_space_refine* program params. Syntax to add extra params: `paramName1 = value1 paramName2 = value2`

- `Secondary structure`: Advanced param to choose including secondary structure restraints. It is set to `Yes` by default.
- `Macro cycles`: Advanced param that allows select the number of iterations of refinement. Although 5 macro-cycles, set by default, is usually enough, increasing this value might be helpful when model geometry or/and model-to-map fit is poor. The increase in the number of macro-cycles will also scale the computing times.
- `Optimization strategy options`: Box of advanced params that allow to modify the default refinement optimization strategy:
  - `Global minimization`: Param set to "Yes" by default to look for the global minimum of the model.
  - `Rigid body`: Param set to "No" by default. It considers the movement of groups of atoms as a single body.
  - `Local grid search`: Param set to "No" by default. It is used to fit local rotamers.
  - `Morphing`: Param set to "No" by default. It allows distortions of the model to match the electron density map.
  - `Simulated annealing`: Param set to "No" by default. By molecular dynamics this param minimizes the energy of the model.
  - `Atomic Displacement Parameters (ADPs)`: Param set to "Yes" by default. Model refinement regarding the map param that considers temperature factors. This refinement step is performed only at the last macro-cycle.

- Protocol execution:
  Adding specific map/structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output

summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and the results window will be opened (Fig. 145).

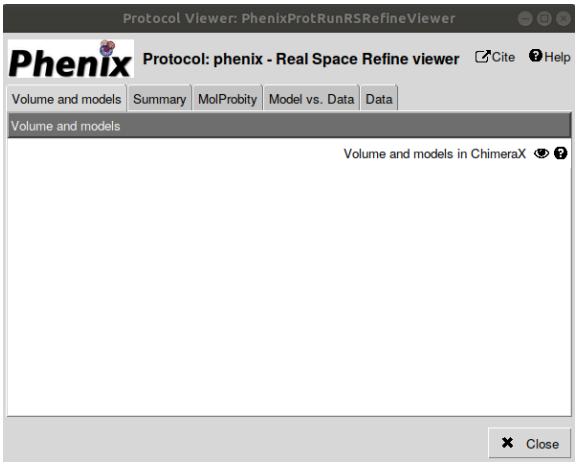


Figure 145: Protocol phenix - real space refine. Taps to visualize *Real Space Refine* results.

Five taps are shown in the upper part of the results window (only four taps with *PHENIX* v. 1.13 identical to those shown in Fig. 134, Fig. 135, Fig. 136 and Fig. 137):

  - `Volume and models`: *ChimeraX* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85).
  - `Summary`: Three different summary tables are shown to describe the results obtained from `Model, Data` and `Model vs. Data` (Fig. 146). Concern-

ing the atomic `Model`, numeric data from chains, residues, atoms and geometry are described, as well as main *MolProbity* statistics. `Data` summarizes experimental map box dimensions and different values of resolution computed with or without a mask. `Model vs. Data` details main real-space correlation coefficients.

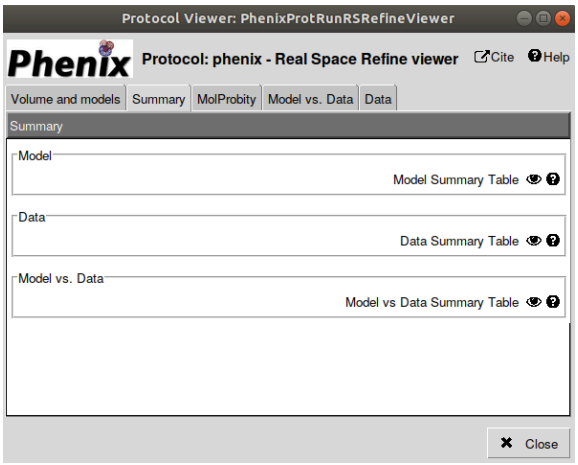


Figure 146: Protocol phenix - real space refine. Summary tables of main*PHENIX* `real space refine` results.

– `MolProbity`: Statistics concerning the atomic model, most of them obtained from *MolProbity* (Fig. 147).

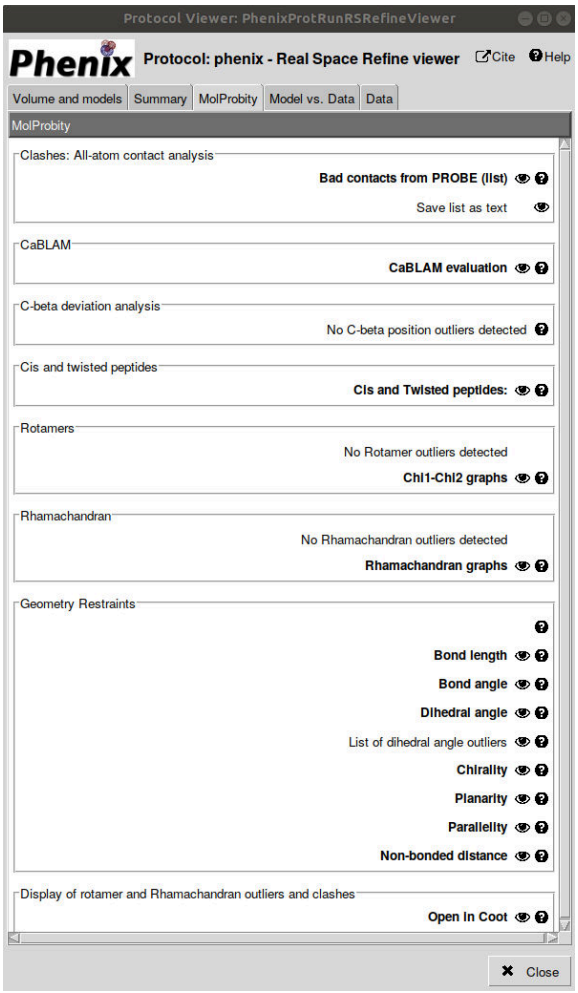


Figure 147: Protocol phenix - real space refine. *MolProbity* and other statistics of the atomic model.

* `Clashes: All-atom contact analysis`: List that contains all severe clashes (non-H atoms overlaping more than 0.4 Å) found by PROBE. All these clashes can be visualized and solved graphically in *Coot*. If no hydrogens were present, REDUCE adds them before running PROBE. The list can be saved in a folder selected by the user.
* `CaBLAM: C-Alpha Based Low-resolution Annotation Method`: Method

designed to assess the mainchain geometry of the atomic model by using protein C_α geometry and to identify areas of probable secondary structure. Residues that fall outside contours of expected protein behaviour based on high-quality datasets are considered outliers.

* `C-beta deviation analysis`: C_β outliers deviate from ideal positions by more than 0.25Å. Ideal C_β position is determined from the average of the ideal C-N-CA-CB and N-C-CA-CB dihedrals. This measure is more sensitive than individual measures to both sidechain and mainchain misfittings. Its deviation is an indicator of incompatibility between sidechain and backbone.
* `Cis and twisted peptides`: Residues showing *cis* or *twisted* conformations that could be modeling errors. *cis* conformations are observed in about 5% of Prolines and 0.03% of general residues. Twisted peptides are almost certainly modeling errors.
* `Rotamers`: Rotamer outlier list contains residues that adopt an unusual conformation of χ dihedral angles. These outliers, commonly used to characterize the conformation of protein sidechains, are detailed in Chi1-Chi2 graph, shown below.
* `Rhamachandran`: Rhamachandran outlier list contains residues that show an unusual combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles. Most of the time, Ramachandran outliers are a consequence of mistakes during the data processing. These outliers are detailed below in Rhamachandran graphs.
* `Geometry Restraints`: Statistics for geometry restraints used in refinement. Although in general a fully refined structure should not have any outliers, exceptionally there are some of them that are obvious in high resolution electron density maps. Types of restraints:
    - `Bond Length`: This table indicates the number of outliers and the number of restraints (in accordance with the bond length restraints library). The list of outliers details the bonded pairs of atoms sorted by deviation (higher than 4 sigmas).

- `Bond Angle`: This table indicates the number of outliers and the number of restraints (in accordance with the bond angle restraints library). The list of outliers details the bonded triplets of atoms sorted by deviation (higher than 4 sigmas).
- `Dihedral Angle`: This table indicates the number of outliers and the number of restraints (in accordance with the side chain dihedral torsion - chi- angle restraints library). The list of outliers details the bonded tetrads of atoms sorted by deviation (higher than 4 sigmas).
- `Chilarity`: This table indicates the number of restraints (in accordance with the volume chilarity restraints library).
- `Planarity`: This table indicates the number of restraints (in accordance with the volume planarity restraints library).
- `Parallelity`: This table indicates the number of restraints (in accordance with the volume parallelity restraints library).
- `Non-bonded distance`: This table indicates the number of restraints (in accordance with the volume non-bonded distance restraints library).

* `Display of rotamer and Rhamachandran outliers and clashes`: Interactive visualization of outliers (Ramachandran, rotamer and C_β) and severe clashes with *Coot*.

– `Model vs. Data`: Real-space correlation coefficients between map and model-derived map (Fig. 148).

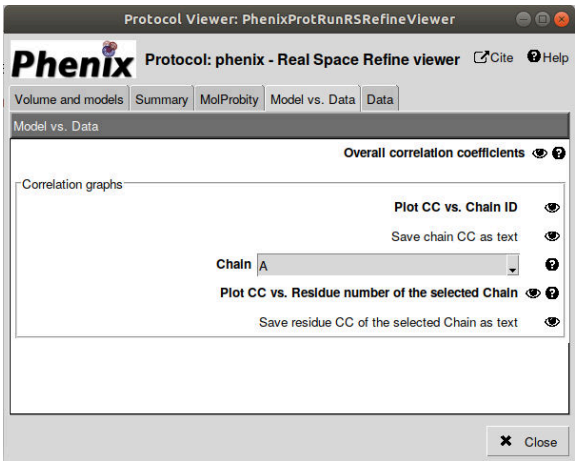


Figure 148: Protocol phenix - real space refine. Real-space correlation results.

* `Overall correlation coefficients` (Afonine et al., 2018a):
  - `Mask CC`: Correlation coefficient between the model-derived map and the experimental map inside the mask region built around the model with a fixed radius. This comparison aims to fit the atomic centers.
  - `Box CC`: Correlation coefficient between the model-derived map and the whole experimental map. This comparison aims to assess the similarity of maps and remark map densities that have not been modeled.
  - `Volume CC`: Correlation coefficient between the model-derived map and the experimental map inside the mask region built around the model considering only model-derived map regions with the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case the N points with the highest density, inside the molecular mask, are taken into account. This comparison aims to fit the molecular envelope defined by the model-derived map.
  - `Peak CC`: Correlation coefficient between the model-derived map and the experimental map that considers only map regions with

the highest density values, ignoring regions below a certain contouring density threshold. Particularly, in this case the N points with the highest density, simultaneously present in the model-calculated map and in the experimental map, are taken into account. This comparison aims to fit the strongest peaks in model-derived and experimental maps.

- `Main chain CC`
- `Side chain CC`

* `Correlation graphs`:
  - `Plot CC vs. Chain ID`: Plot of correlation coefficients regarding the chain IDs. These correlation coefficient values can be saved in a text file in the folder selected by the user.
  - `Plot CC vs. Residue number of the selected Chain`: Plot of correlation coefficients of each chain residues. The specific chain is selected by the user in the chain option box. These correlation coefficient values for each chain can be saved in a text file in the folder selected by the user.

– `Data` (Fig. 149): Computation of Resolution and FSC.

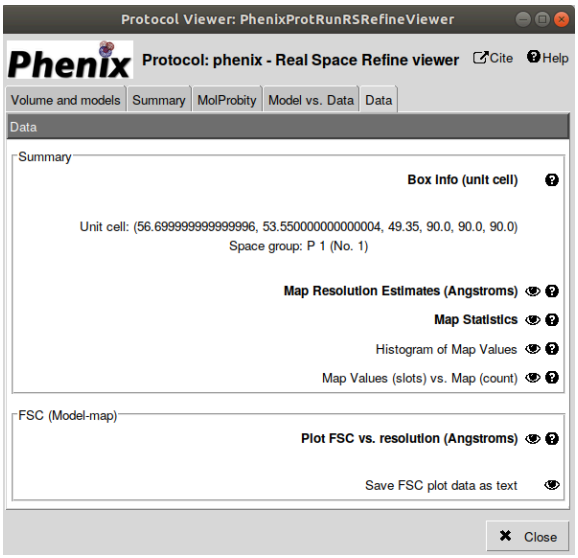


Figure 149: Protocol phenix - real space refine. Experimental data results.

* `Summary`: Basic statistics about the maps and summary of resolution estimates.
  - `Box info (unit cell)`: Map cell dimensions (pixels).
  - `Map Resolution Estimates (Angstroms)`: Resolution estimates computed considering both map experimental data and model-derived information(with and without mask).
    `- Using map alone (d99)`: Resolution cutoff beyond which Fourier map coefficients are negligibly small. Calculated from the full map or from each one of half maps [d99 (half map 1), d99 (half map 2)].
    `- Overall Biso`: Overall isotropic B-value.
    `- d_model`: Resolution cutoff at which the model map is the most similar to the target (experimental) map. Requires map and model. For d_model to be meaningful, model is expected to fit the map as well as possible.
    `- d_model (B factors = 0)`: It tries to avoid the blurring of the map.

`- FSC (model) = 0`: d_FSC_model_0; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.
`- FSC (model) = 0.143`: d_FSC_model_0.143; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.143.
`- FSC (model) = 0.5`: d_FSC_model_0.5; Resolution cutoff up to which the model and map Fourier coefficients are similar at FSC value 0.5.
`- FSC (half map 1, 2) = 0.143`: d_FSC; Highest resolution at which the experimental data are confident. Obtained from FSC curve calculated using two half-maps and taken at FSC=0.143. The two half maps are required to compute this value.
`- Mask smoothing radius (Angstroms)`: Radius of the default soft mask used since sharp edges resulting from applying a binary map may introduce Fourier artifacts.

* Fourier shell correlation taps:
  - `FSC(Half-maps)` (Only if two half maps have been added as inputs): FSC plot regarding the resolution (Å) and the spatial frequency (1/Å) based on half maps with and without masking. The intersections of the curves with FSC = 0.143 are shown. FSC plot data can be saved as text file in a folder selected by the user.
  - `FSC (Model-map)`: FSC plot regarding the resolution (Å) and the spatial frequency (1/Å) based on the experimental map and the model-derived map with and without masking. The intersections of the curves with FSC = 0.5 are shown. FSC plot data can be saved as text file in a folder selected by the user.

- Summary content:

  `SUMMARY` box:
  Main *MolProbity* statistics computed by the *Phenix* package to assess protein

geometry using the same distributions as the MolProbity server:

- `Ramachandran outliers`: Percentage of residues assessed that show an unusual combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles.
- `Ramachandran favored`: Percentage of residues assessed that show an normal combination of their ϕ (C-N-CA-C) and ψ (N-CA-C-N) dihedral angles. Ramachandran outliers and favored residues are detailed in the `Ramachandran plot`. Allowed residues are included in the small region comprised between favored and outlier regions of that plot.
- `Rotamer outliers`: Percentage of residues assessed that adopt an unusual conformation of χ dihedral angles. Rotamer outliers, commonly used to characterize the conformation of protein sidechains, are detailed in the Chi1-Chi2 plot.
- `C-beta outliers`: Number of residues showing an unusual deviation (higher than 0.25 Å) of the Cβ from its ideal position. This deviation is an indicator of incompatibility between sidechain and backbone.
- `Clashscore`: Score associated to the number of pairs of non-bonded atoms unsually close to each other, showing probable steric overlaps. Clashscore is calculated as the number of serious clashes per 1000 atoms. This value has to be as low as possible.
- `Overall score`: *MolProbity* overall score representing the experimental resolution expected for the structure model. This value should be lower than the actual resolution. The lower the value, the better quality of the structure model.

24 Phenix Superpose PDBs protocol

Protocol designed to superpose two atomic structures in *Scipion* by using *phenix.superpose_pdbs* program (Zwart et al., 2017). Integrated in the *Phenix* software suite (`https://`

`www.phenix-online.org/`), *PHENIX* protocol phenix - superpose pdbs allows to compare visually the geometry of two atomic structures by overlapping them. Root mean square deviation (RMSD) between fixed and moving structures is computed before and after the superposition.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-phenix`
  - PHENIX software suite (tested for versions 1.16-3549, 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Tools-Calculators` (Fig. 150 (A))
- Protocol form parameters (Fig. 150 (B)):

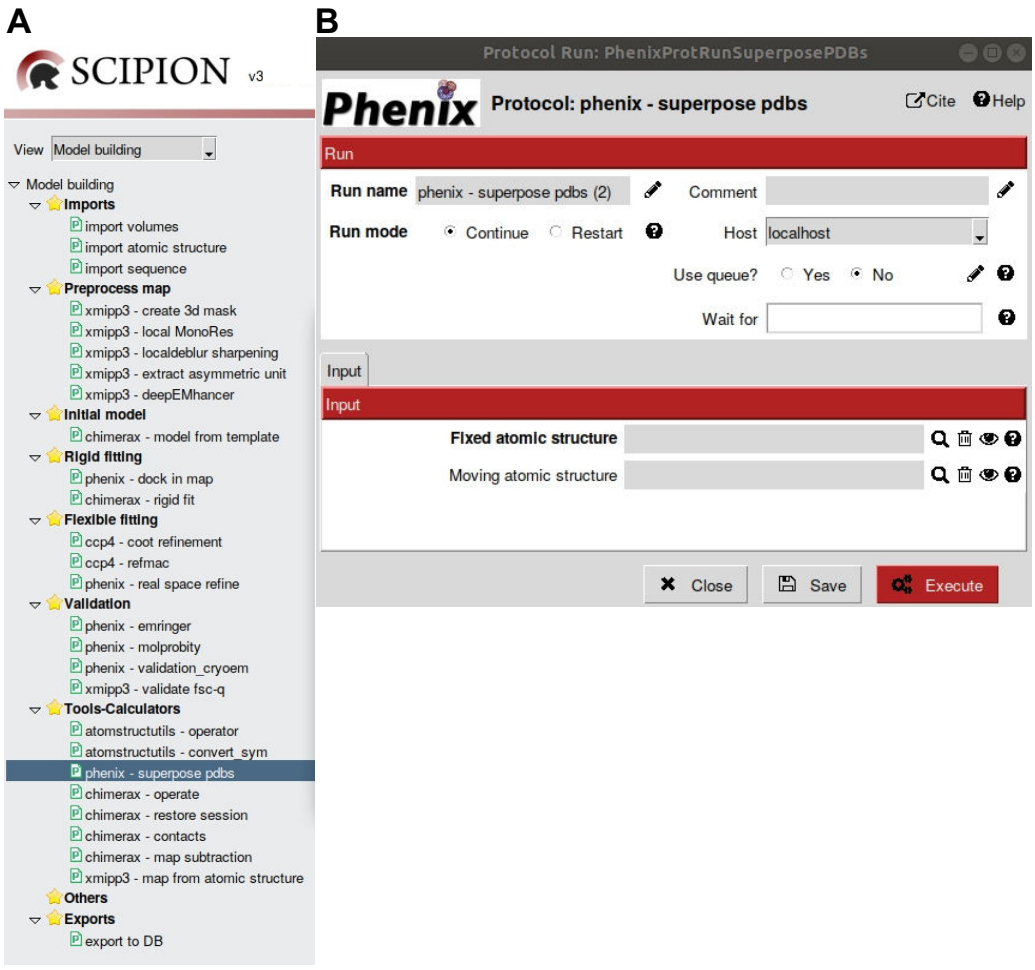


Figure 150: Protocol phenix - superpose pdbs. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Fixed atomic structure`: Fixed PDBx/mmCIF, previously downloaded or generated in *Scipion*, to which the moving one will be aligned.
  - `Moving atomic structure`: PDBx/mmCIF, previously downloaded or generated in *Scipion*, that will be aligned to the fixed one.

- Protocol execution:
  Adding specific moving_structure/fixed_structure label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press

the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85).

- Summary content:

  `SUMMARY` box:
  RMSD between fixed and moving atoms (start and final values).

25 Phenix Dock in Map protocol

Protocol designed to automatically fit atomic structures to electron density maps in *Scipion* by using *PHENIX* `dock in map` ((Liebschner et al., 2019)), application that uses a convolution-based shape search with which it finds the parts of the *map* that are similar to the *model*. Additional information can be found in `http://www.phenix-online.org/documentation/reference/dock_in_map.html`.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
  - *Scipion* plugin: `scipion-em-phenix`
  - *PHENIX* package (tested for versions 1.17.1-3660 and 1.18.2-3874)
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu:
  `Model building -> Rigid fitting` (Fig. 151 (A))

- Protocol form parameters (Fig. 151 (B)):

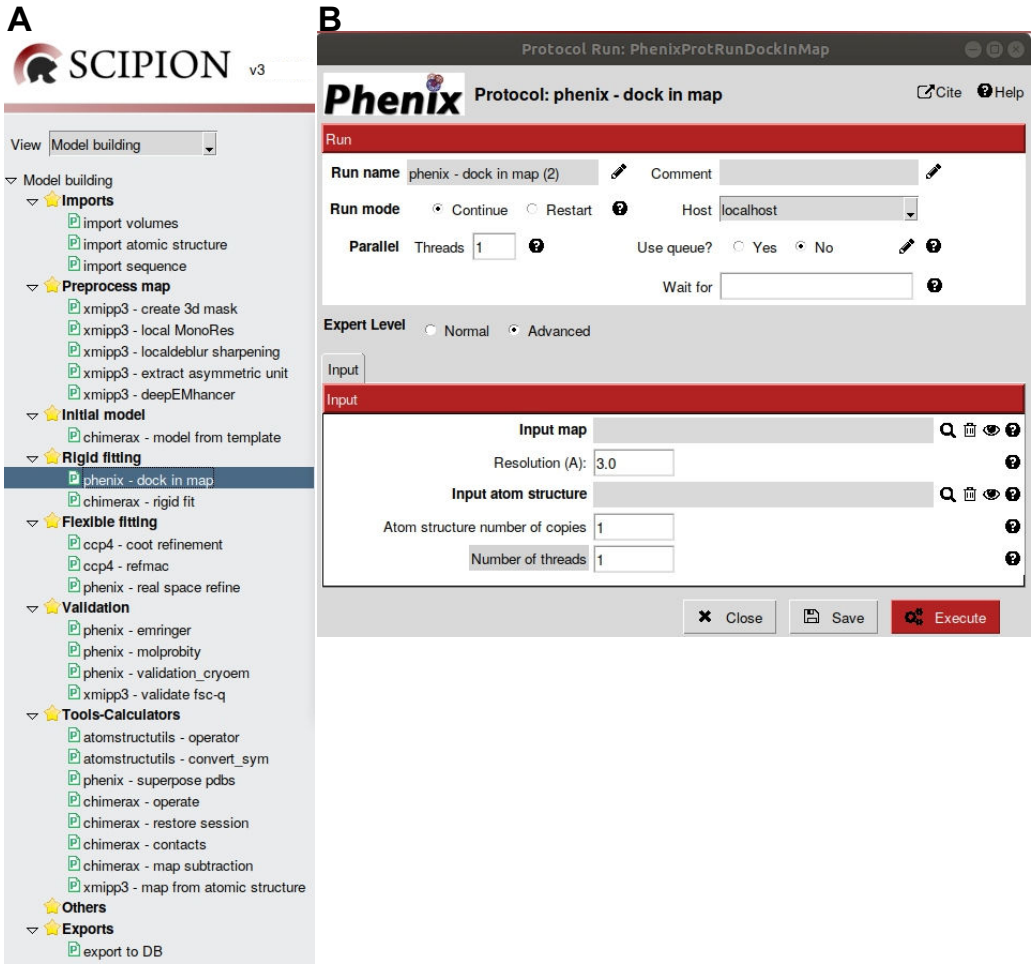


Figure 151: Protocol phenix - dock in map. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input map`: Electron density map previously downloaded or generated in *Scipion* to fit the atomic structure.
  - `Resolution (Å)`: Electron density map resolution.

  - `Input atom structure`: Atomic structure previously downloaded or generated in *Scipion* to be fitted to an electron density map.
  - `Atom structure number of copies`: Number of *models* that have to be simultaneously fitted to an electron density map.
  - `Number of threads`: Advanced param. Depending on the size of *map* and *model*, and the number of *models* to fit the process could be quite slow and you can accelerate it by increasing the number of threads.

- Protocol execution:

  Adding specific protocol label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and the *ChimeraX* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *ChimeraX*. To show the relative position of atomic structure and electron density volume, the three coordinate axes are represented; X axis (red), Y axis (yellow), and Z axis (blue) (Fig. 85). Coordinate axes, map, initial unfitted *model* and final fitted atomic structure are model numbers `#1`, `#2`, `#3` and `#4`, respectively, in *ChimeraX* `Models` panel.

- Summary content:

  - Protocol output (below *Scipion* framework):
    `phenix - dock in map -> ouputPdb`;
    `AtomStruct (pseudoatoms=True/ False, volume=True/ False)`.
    Pseudoatoms is set to `True` when the structure is made of pseudoatoms

instead of atoms. Volume is set to `True` when an electron density map is associated to the atomic structure.

– `SUMMARY` box:
No summary information

26 Protocol to assign map sampling rate and origin of coordinates

Protocol designed to modify the values of sampling rate and origin of coordinates of electron density maps that are already in the *Scipion* workflow. Remark that the new map generated in the *Scipion* workflow will associate these two attributes, samplig rate and origin of coordinates, WITHOUT modifying the map header. If you want to modify the map header inside *Scipion*, for example to deposit it to EMDB, you should use the protocol export to DB.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em`
- *Scipion* menu: It does not appear in `Model building` view. Press Ctrl + f and the pop up window to search a protocol will be opened ((Fig. 152 (A)). Write any word related with the title of the protocol that you are looking for in the `Search` box. In this particular case we have written `origin`. Several protocols have been found related with this searching word. Select the third one dessigned for the purpose that we are interested in (`pwem - assign orig & sampling`).

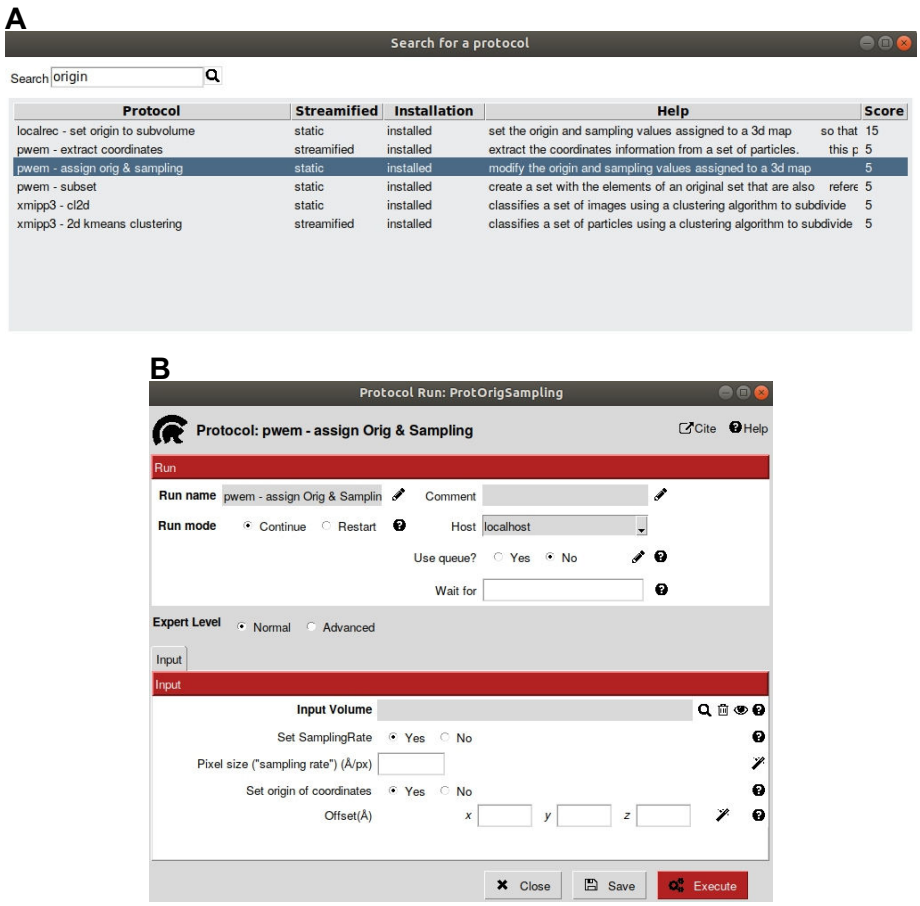


Figure 152: Protocol assign Orig & Sampling. A: Window to search the protocol. B: Protocol form.

- Protocol form parameters (Fig. 152 (B)):
  - `Input` section
    - `Input Volume`: Include here any map previously downloaded or generated in *Scipion* that you would like to assing a new sampling rate and/or origin of coordinates.
    - `Set SamplingRate)`: Select `Yes` if you want to give the map a new value of sampling rate. Then, a new form box will appear:

· `Pixel size (‘‘sampling rate’’)` (Å/px): Write the new sampling rate rate value in the box. Remark that you have a wizard on the right to check the current value.

∗ `Set origin of coordinates`: You have to choose between setting the previously origin of coordinates assigned in *Scipion* (option "No") or another origin of coordinates ("Yes"). If you decide to set your own origin of coordinates (option "Yes"), a new form parameter (`Offset`) will appear below.

· `Offset`: Write here x, y, and z coordinates of your preference (in Å). As in the case of the `Pixel size`, remark that you have a wizard on the right side of the parameter to check the header current coordinates of the origin.

- Protocol execution:

  Adding specific volume label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK, and finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *ShowJ*, the default *Scipion* viewer, will allow you to visualize the `slices` window of the map (Fig. 153). The *ShowJ* window menu (`File -> Open with ChimeraX`) allows to open the selected map in *ChimeraX* graphics window.

  – `slices`: *ShowJ*

    `https://github.com/I2PC/scipion/wiki/ShowJ`

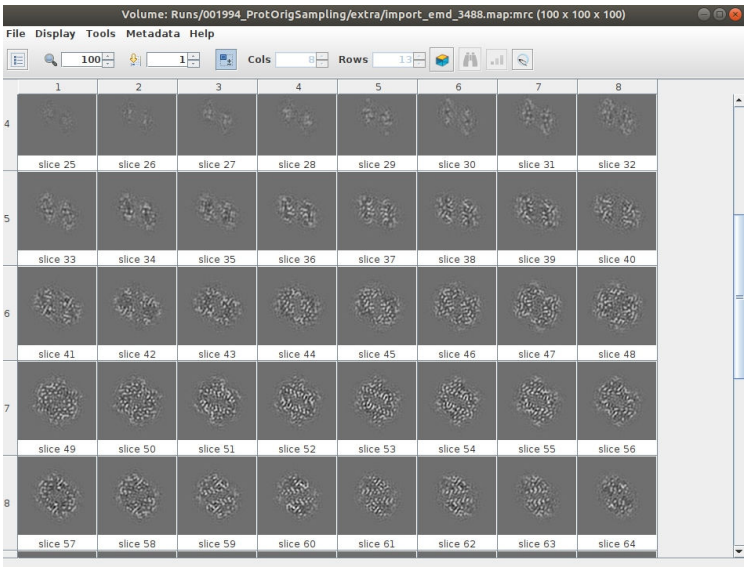


Figure 153: Protocol assign Orig & Sampling. Gallery model of *ShowJ* to visualize the map slices.

- Summary content:
  - Protocol output (below *Scipion* framework):
    `pwem - assign Orig & Sampling -> ouputVolume`;
    `Volume (x, y, and z dimensions, NEW sampling rate)`.
  - `SUMMARY` box:
    `New Sampling`: New assigned value of sampling rate.
    `New Origin`: Coordinates `x, y, z` of the new assigned origin of coordinates.

27 Submission to EMDB protocol

Protocol designed to save in a specified folder main files required to submit cryo-EM derived electron density maps and derived atomic structures to EMDB, as well as other additional files that EMDB encourages to submit (`https://deposit-pdbe.wwpdb.org/deposition//`). Although the submission has to be performed online,

this protocol tries to help the user to organize their results in different folders according to each particular submission date, project, and so on.

- *Scipion* menu:

  `Model building -> Exports` (Fig. 154 (A))

- Protocol form parameters (Fig. 154 (B)):

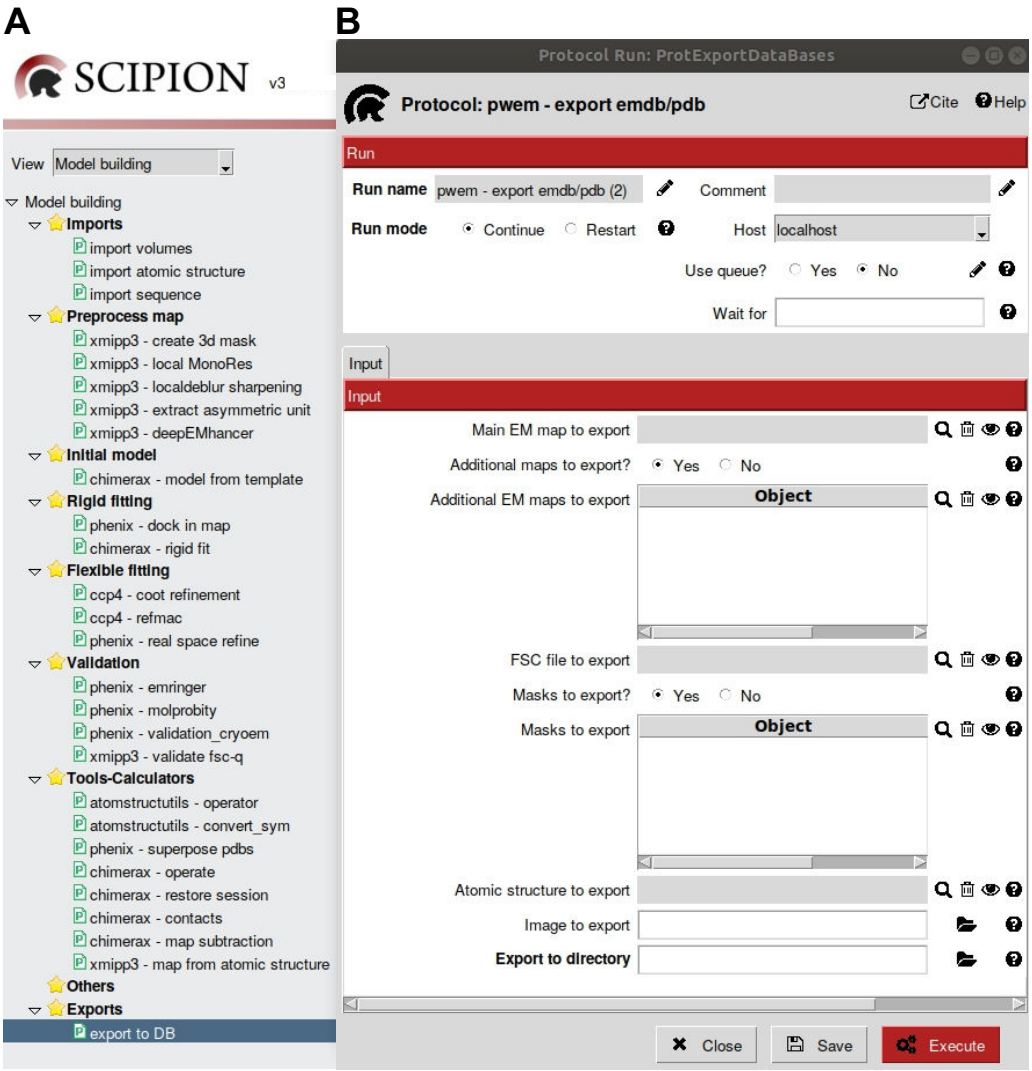


Figure 154: Protocol export to EMDB. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Main EM map to export`: Param to select the electron density map previously downloaded or generated in *Scipion* that you would like to submit to EMDB as `main map`. The map file will be saved with `.mrc` format. If the `main map` has the two half maps associated, they will

be also saved at the same time in the same directory.

  * `Additional maps to export?`: In case you would like to submit other types of maps, specially those generated during the postprocessing like sharpening maps, select `Yes` and a new form param (`Additional EM maps to export`) will be opened to interrogate about the additional files (.mrc format). Take into account that all of them should be previously generated or imported in *Scipion*.
  * `FSC to export`: Param to select the FSC file previously generated in *Scipion* that we would like to submit to EMDB. This file will be saved with `.xml` format.
  * `Masks to export?`: EMDB also encourages to submit masks relevant in reconstruction or postprocessing steps. Select `Yes` if you want to include one or several masks and a new form param (`Masks to export`) will open to interrogate about the masks (.mrc format).
  * `Atomic structure to export`: Param to select the file of coordinates from the volume-associated atomic structure previously downloaded or generated in *Scipion* that we would like to submit to EMDB. This file will be saved with `.cif` format.
  * `Image to export`: Map image to represent the map in the database.
  * `Export to directory`: Directory specified by the user to save the three above selected files. In order to get appropriate data organization, a name related with the submission is recommended (date, project, number, ...).

- Protocol execution:

  Adding specific protocol label is recommended in `Run name` section, at the form top. To add the label, open the protocol form, press the pencil symbol at the right side of `Run name` box, complete the label in the new opened window, press OK and, finally, close the protocol. This label will be shown in the output summary content (see below). If you want to run again this protocol, do not forget to set to `Restart` the `Run mode`.
  Press the `Execute` red button at the form bottom.

All the previously selected files will be saved in the chosen directory after executing the protocol and this can be checked by opening that folder. Content of the selected directory:

- `main_map.mrc`
- `half_map_1.mrc`
- `half_map_2.mrc`
- Folder of additional maps: `addMaps`, which contains `map_01.mrc, map_02.mrc`, etc.
- `FSC_file_name.xml`
- Mask folder: `masks`, which contains `mask_01, mask_02`, etc.
- Input atomic structure: `atomic_structure_file_name.cif/pdb`
- Atomic structure complete: `coordinates.cif`
- Atomic structure symplified: `symplified_atom_structure.cif`
- `image.png`

As you can see, two coordinate files have been created, complete and symplified, to try to satisfy different format demands.
No additional specific visualization tools have been added to this protocol.

- Summary content:

  The summary specifies the path to the directory selected to save the files:
  `Data available at:` *path*