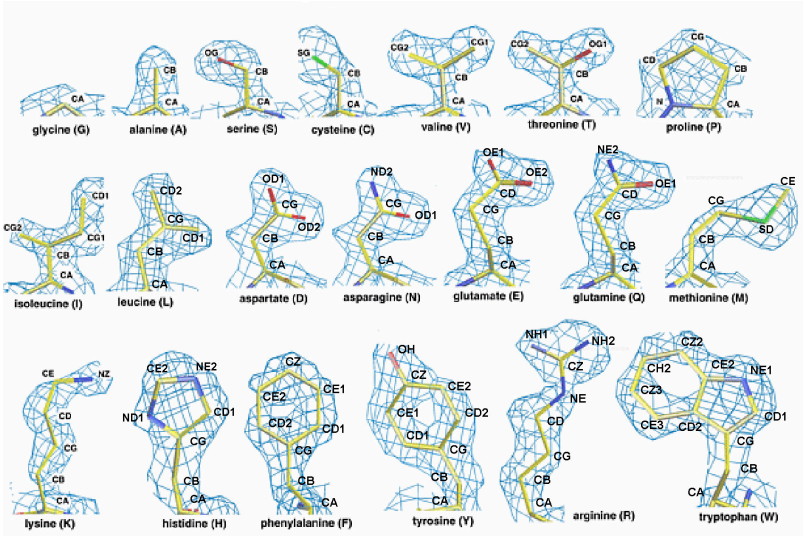
Scipion Tutorial Series

National Center for Biotechnology

Biocomputing Unit

Model Building Basic

September 11, 2019



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

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*, USCF *Chimera* (`https://www.cgl.ucsf.edu/chimera/download.html`), the *CCP4* suite (`http://www.ccp4.ac.uk/download/#os=linux`) including *Refmac* and *Coot*, the *PHENIX suite* (`https://www.phenix-online.org/download/`) and *PowerFit* application (`https://github.com/haddocking/powerfit`). Basic knowledge of chimera 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.

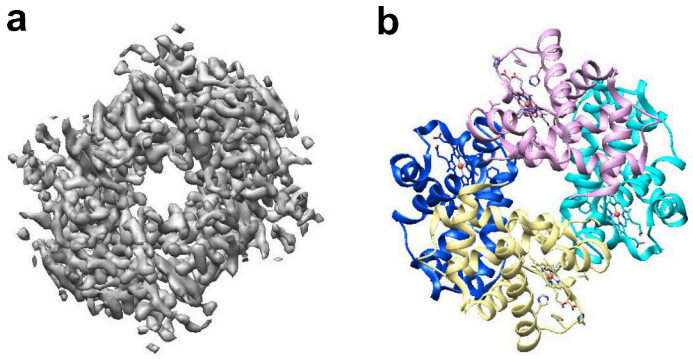


Figure 1: Haemoglobin tetramer (Khoshouei et al., 2017). a) Electron density map at 3.2Å resolution obtained by Cryo-EM single particle analysis with Volta phase plate. b) Atomic structure model inferred from the electron density volume.

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.

The workflow considers as input the lower asymmetrical element (unit cell) of the starting volume and the sequence of each individual structural element (from 1 to n). These sequences are used to get the initial models, *de novo* or by prediction based in homologous structures. Initial model of each structure element has to be fitted to the unit cell volume, and then refined according to its fitted volume. Since the double refinement in real and reciprocal spaces seems to improve protein models (Brown et al., 2015), these two steps of refinement 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 volume to avoid forbidden steric overlaps among them. In the reconstruction of the whole volume, borders between adjacent unit cells will be checked similarly.

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

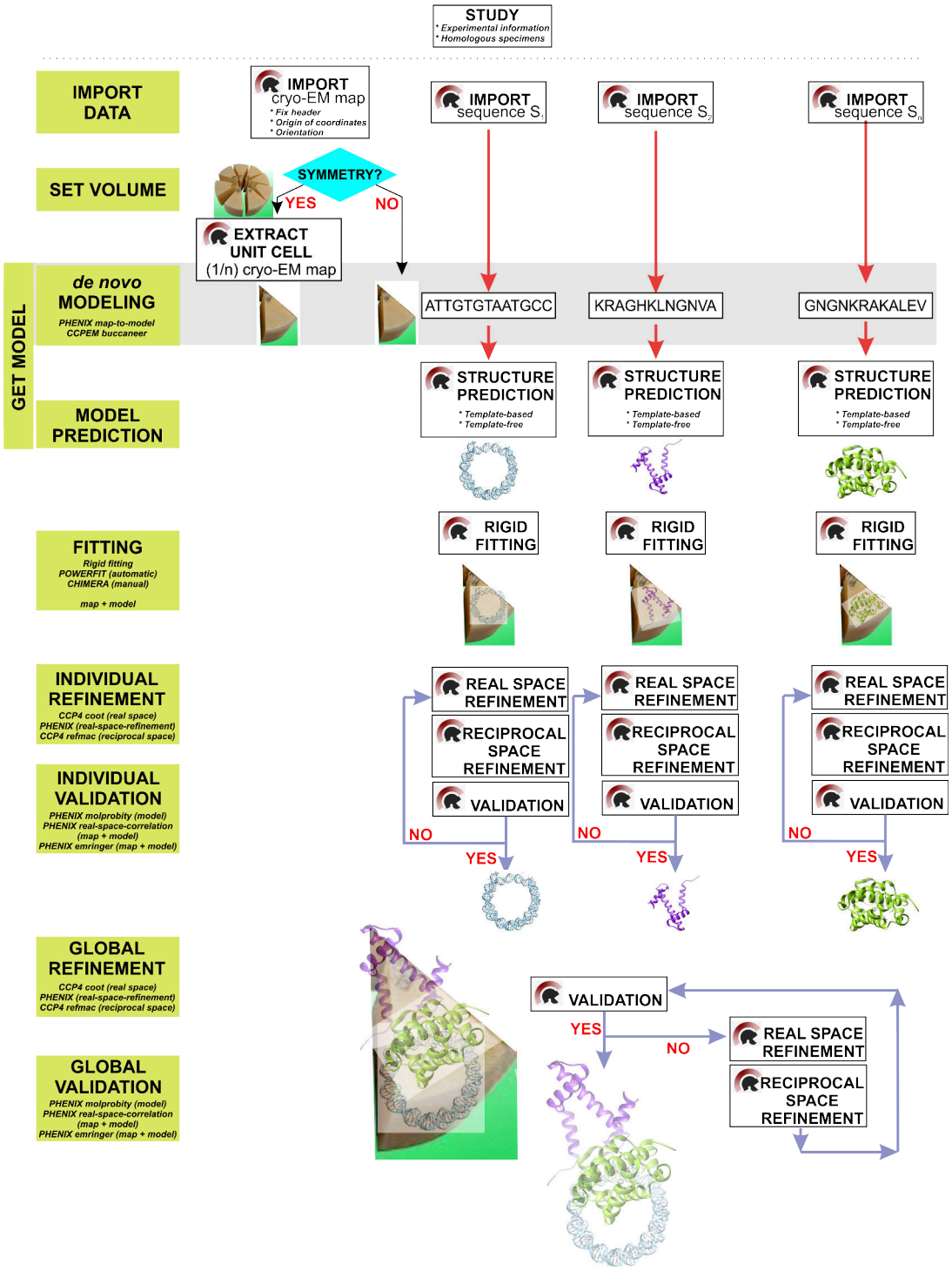


Figure 2: General Model Building Workflow.

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.

3 Input data description

Volume

`EMD-3488`, that can be downloaded from `PDBE` (`http://www.ebi.ac.uk/pdbe/entry/emdb/EMD-3488`).
WARNING: Cryo-EM 3D maps benefit significantly of a "postprocessing" step, normally referred to as "sharpening", that tends to increase signal at medium/high resolution. Therefore, we recommend to sharp the map before tracing the atomic model. Two *Scipion* protocols, xmipp3 - local MonoRes (Vilas et al., 2018) and xmipp3 - localdeblur sharpening (Ramírez-Aportela et al., 2018), consecutively applied, allow map sharpening.

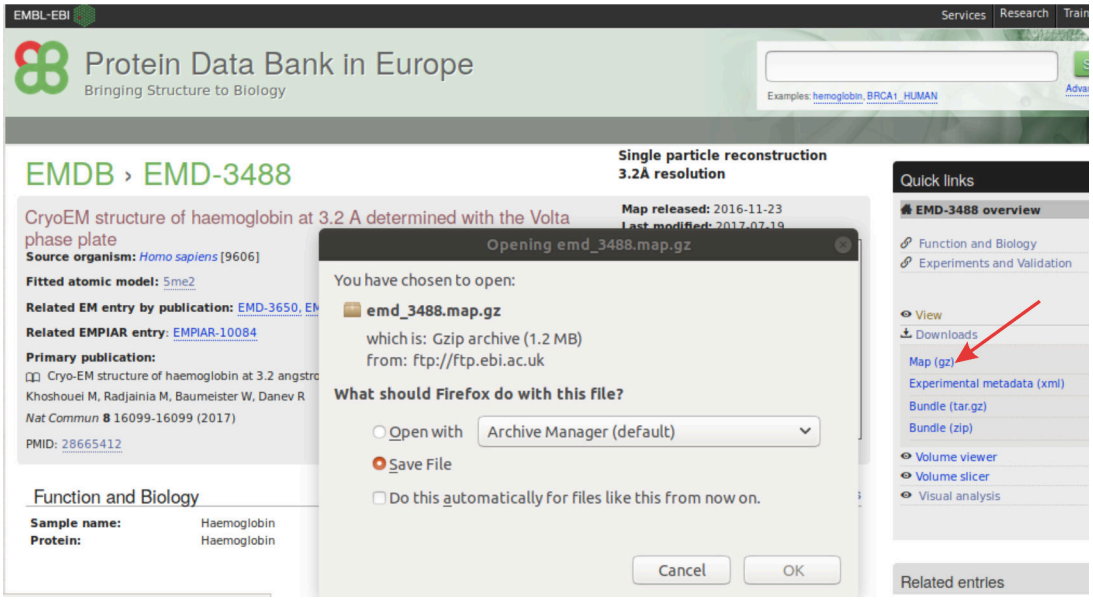


Figure 3: Downloading the volume from `PDBE`.

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 included 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 M and L, respectively.

(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. 4 (1)), fill in the form and execute it (2), and finally you may visualize the volume (3). By default *Chimera* (Pettersen et al., 2004) is used for visualization (Fig. 5). It shows the 3D map and the x (red), y (yellow) and z (blue) axes.

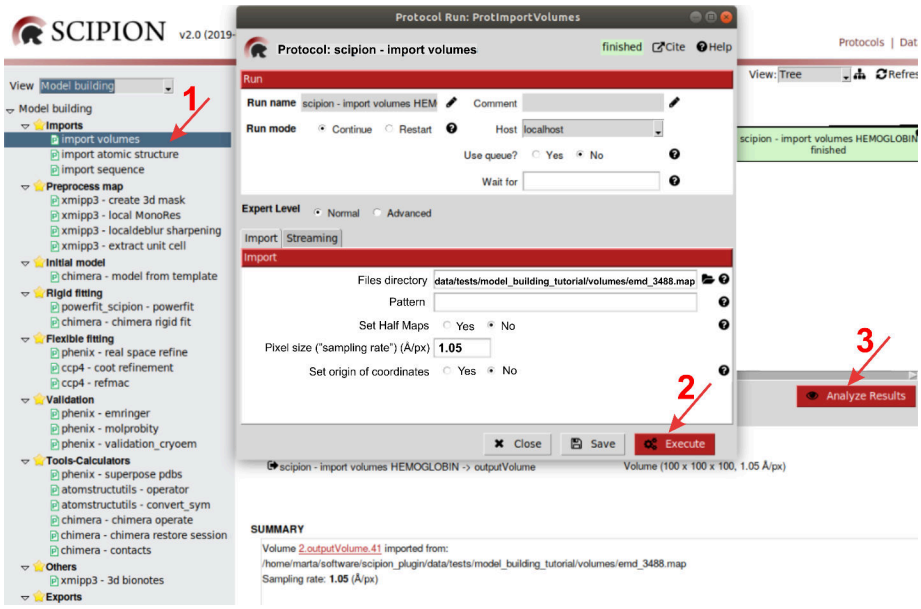


Figure 4: Importing the volume in *Scipion*.

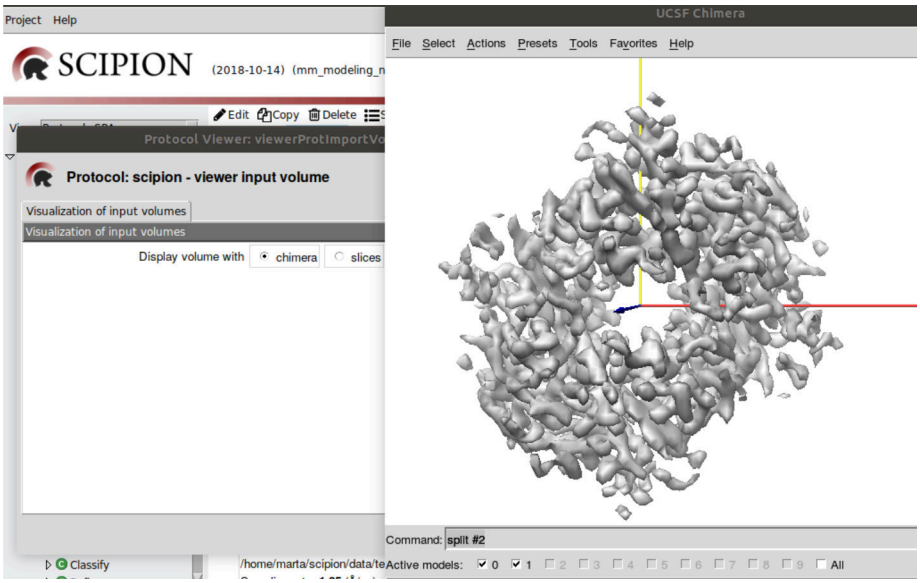


Figure 5: Volume visualized with *Chimera*.

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. 6 (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`.

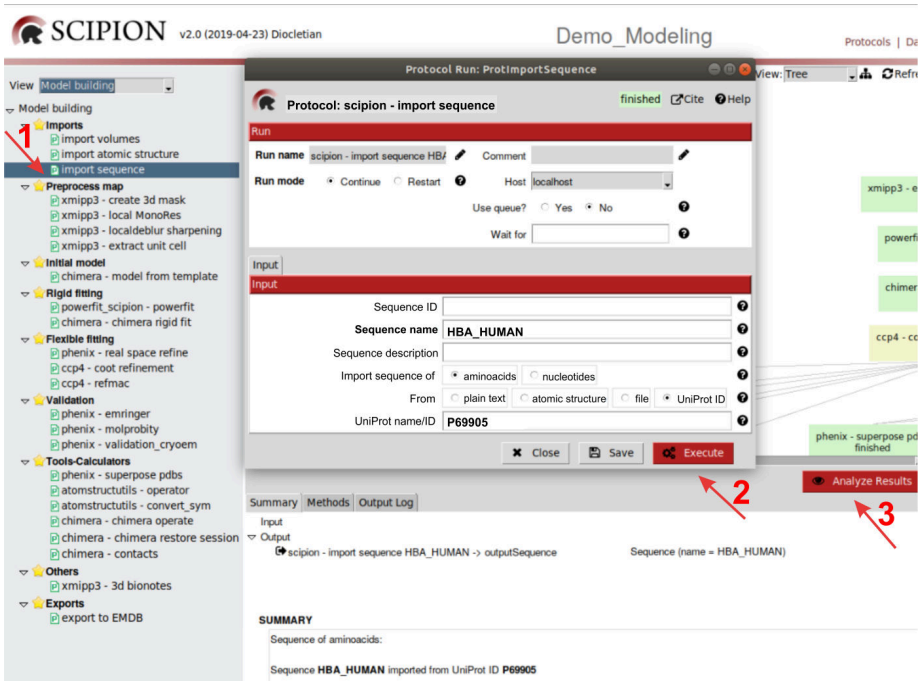


Figure 6: Importing a `UniProtKB` sequence in *Scipion*.

5 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, we are going to use the *Scipion* protocol able to run an automatic method of local sharpening independent of initial model, based on local resolution estimation (xmipp3 - localdeblur sharpening (Ramírez-Aportela et al., 2018) (Appendix N)). Although different algorithms could be used previously to *LocalDeblur* to compute local resolution, we have selected *MonoRes* (Vilas et al., 2018), implemented in *Scipion* in the protocol xmipp3 - local MonoRes (Appendix O). Since a map binary mask has to be included as a parameter in this protocol, the first

step in the local resolution estimation process will be to build the mask by using the *Scipion* protocol xmipp3 - create 3d mask (Appendix I). Open the protocol form (Fig. 7 (1)) and fill in the tap `Map generation` (2) with the input volume (3) and the density threshold (4). By default, the level value observed in *Chimera* main graphics window (Fig. 5) `Tools -> Volume Data -> Volume Viewer -> Level` can be selected as threshold. In the `Postprocessing` tap (Fig. 7 (5)), select `Yes` in `Apply morphological operation` (6) and maintain the rest of options by default.

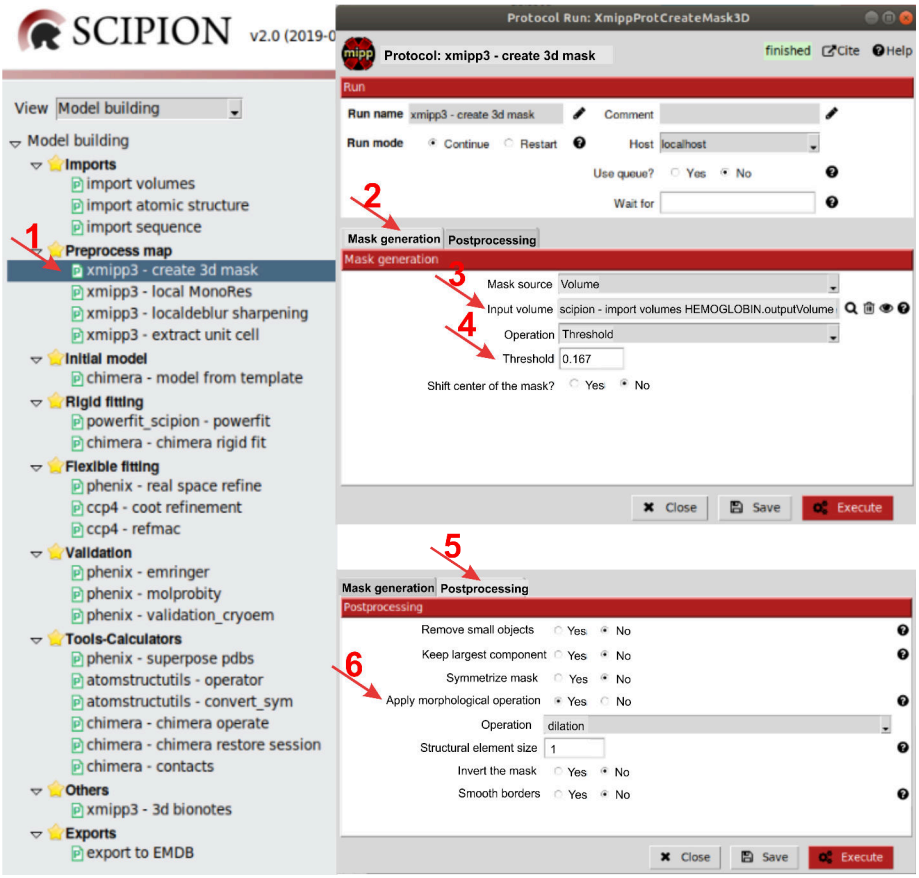


Figure 7: Filling in the protocol to create a mask of the initial volume.

After executing this protocol, the morphology of the mask generated can be checked in slices by clicking `Analyze Results`. *ShowJ*, the default *Scipion* viewer,

allows visualize the mask with shape similar to the starting volume (Fig. 8).

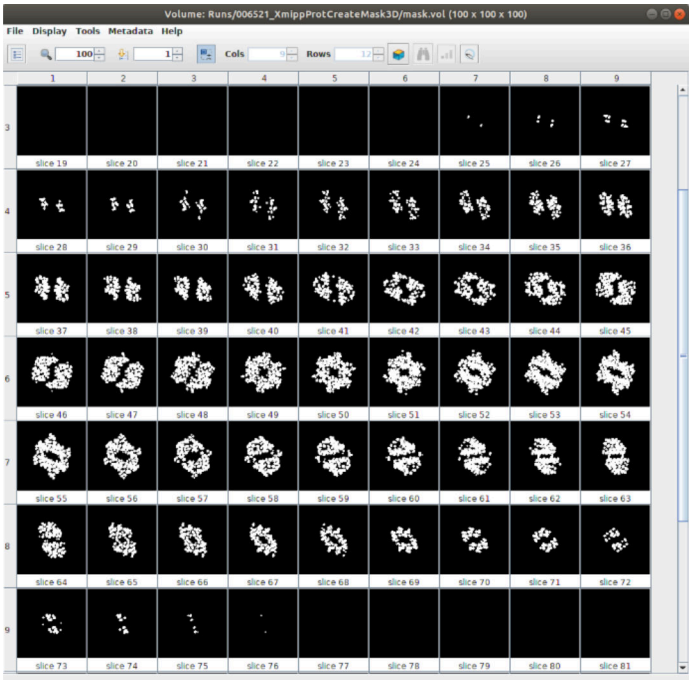


Figure 8: Visualizing the mask of the initial volume.

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. 9 (1)) and include the starting map (2), as well as the binary mask (3). Based on the map resolution (3.2 Å), select a resolution range between `0.0` and `6.0` Å(4). Finally, get the spherical mask radius (px) (5) previously selected by visual inspection (6) through the wizard on the right.

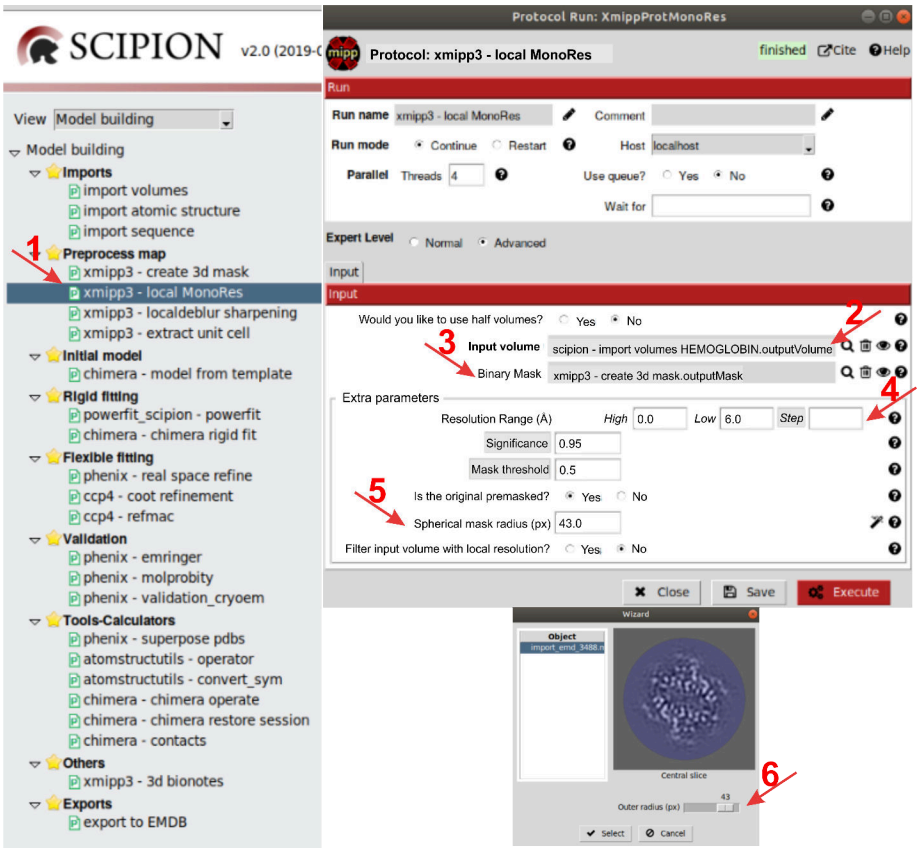


Figure 9: Completing the protocol to estimate the local resolution of the `metHgb` map.

Execute this protocol and analyze the results. The menu of results (Fig. 10 (A)), among other views, shows the histogram of local resolutions (1) and the resolution map in *Chimera* (2). The histogram of resolutions, which displays the number of map voxels showing a certain resolution, allows to conclude that, although the majority of voxels evidence a resolution between 3.25 and 3.5 Å, quite close to the published map resolution (3.2 Å), there are a huge number of voxels displaying higher resolution values, even higher than 6 Å. The resolution map shown by *Chimera* details the resolution of each voxel (Fig. 11). The bar on the left indicates the color code for resolution values.

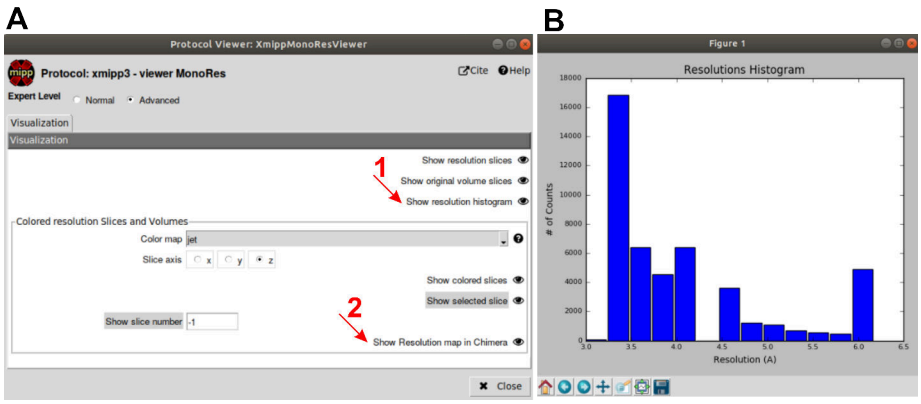


Figure 10: xmipp3 - local MonoRes menu of results (A) and histogram of resolutions (B).

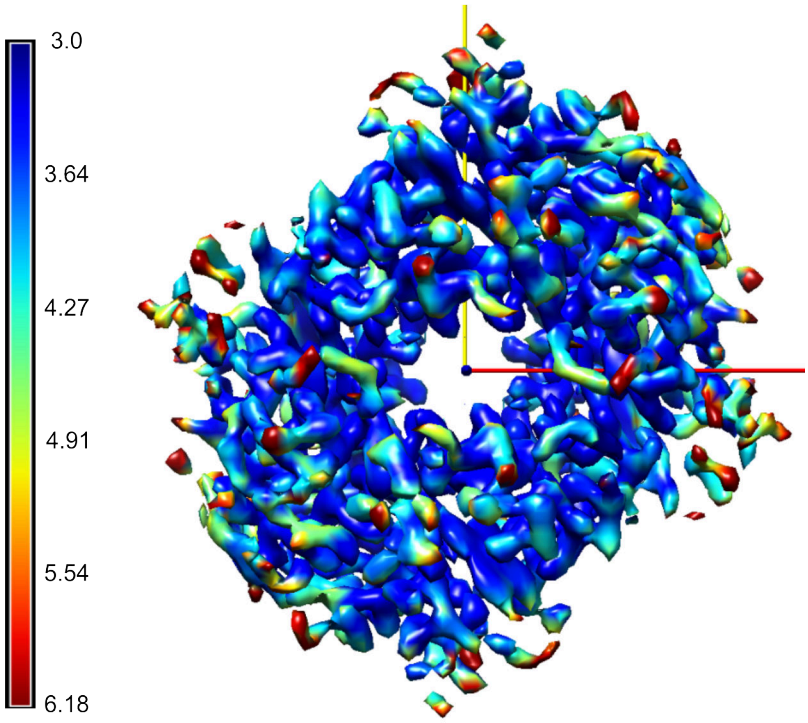


Figure 11: Resolution map in *Chimera*.

Local resolution values of the input map are used to compute the sharpened map

by the xmipp3 - localdeblur sharpening protocol, which implements an iterative steepest descent method that not requires initial model. To accomplish this step, open the protocol (Fig. 12 (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).

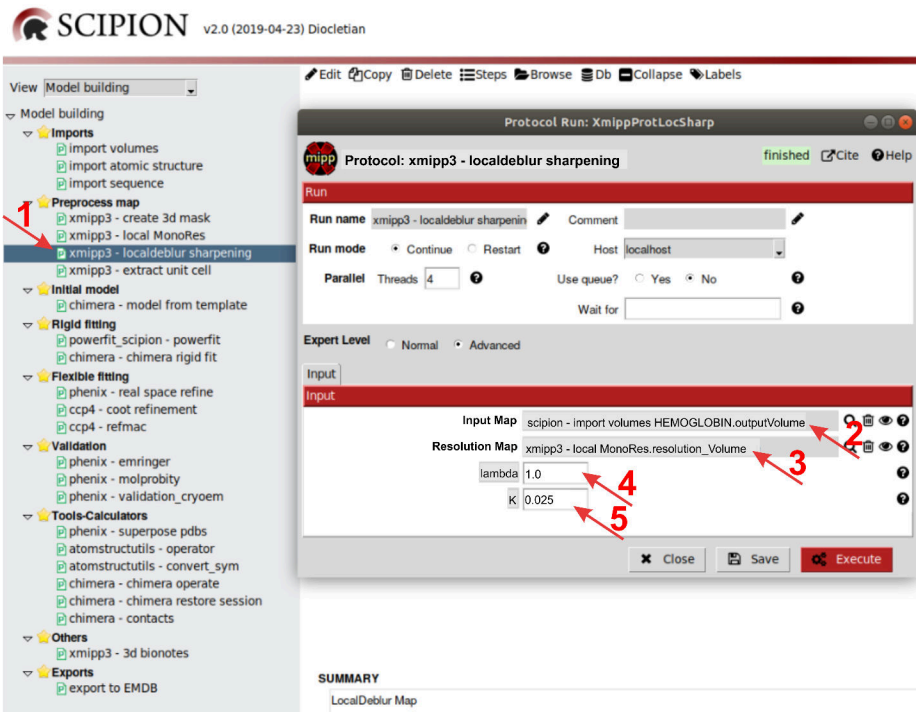


Figure 12: Filling in the protocol to compute the sharpened map.

After two iterations, the sharpening algorithm reaches the convergence criterion, `i.e.` a difference between two successive iterations lower than 1 %, and stops. The two maps obtained in the respective iterations can be observed with *ShowJ* by clicking `Analyze Results` (Fig. 13). Visualization in *Chimera* is also possible selecting "Open with Chimera" in the menu option `File`. The sharpened map obtained after the second iteration will be used in the next step of map preprocessing, the extraction of the unit cell.

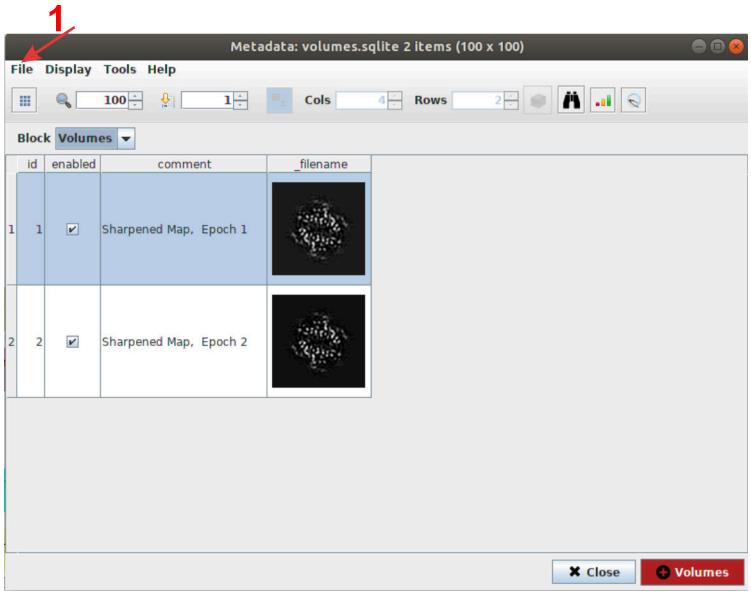


Figure 13: Sharpened maps generated after two iterations.

Extraction of unit cell

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

Fig. 14 shows how to fill in this protocol form (1). Since `metHgb` macromolecule shows symmetry C2, we have selected cyclic symmetry (Cn) as type of symmetry, and 2 as symmetry order. The angle offset selected (-45º) turns around the z axis the mask used to create the unit cell. The extracted fraction of the initial volume will include the volume comprised between the coordinate origin (inner radius 0.0) and the maximum radius (outer radius 50.0), and will be slightly higher than the

unit cell (expand factor 0.2). Again, the respective tutorial appendix J includes a comprehensive explanation of the meaning of parameters.

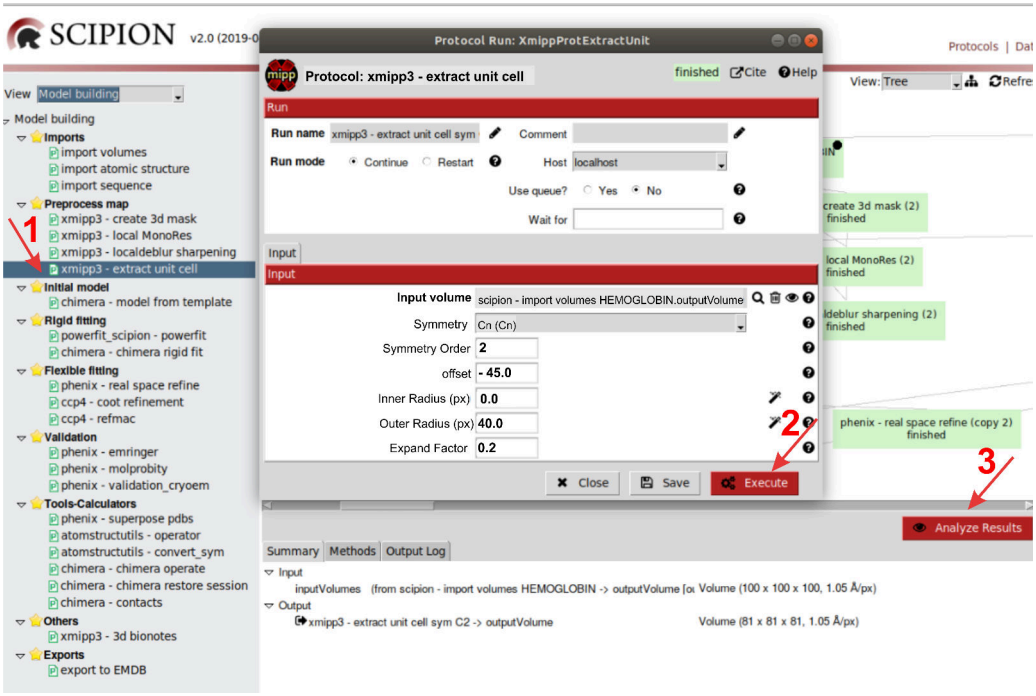


Figure 14: Extracting the unit cell volume.

After executing the protocol (Fig. 14 (2)), the resulting expanded unit cell can be observed (3) with *Chimera* (Fig. 15). Note the additional expanded volume of the unit cell on the left side of the figure. The unit cell itself, on the right side, constitutes the half volume. Since the total volume contains the structure of four proteins, we can predict 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 will be fitted in the unit cell volume in successive modeling workflow steps.

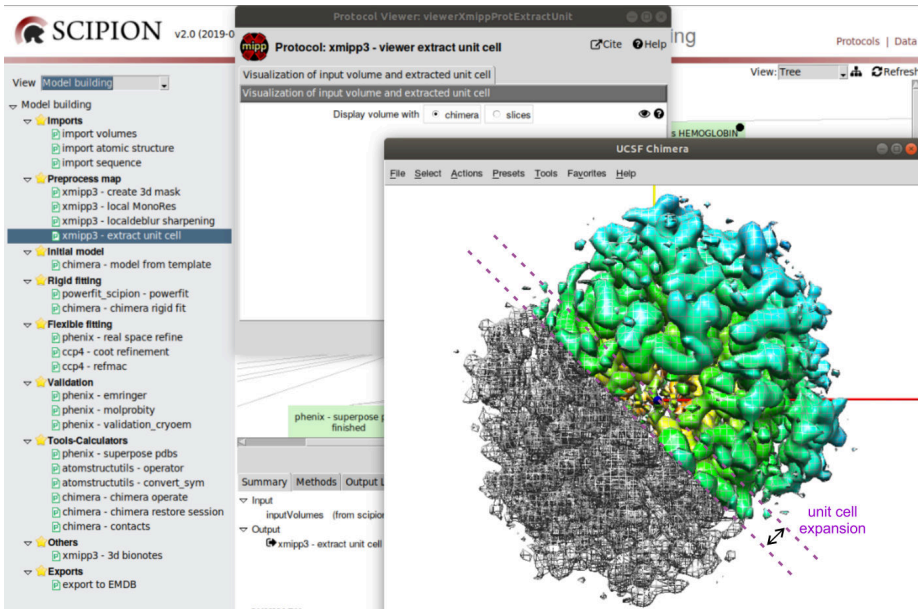


Figure 15: Expanded unit cell (green-blue) and initial volume (gray) visualized with *Chimera*. The purple broken line delimits the unit cell (right) and its expanded volume (left).

6 Structure Prediction by Sequence Homology. Searching for Homologues

As we have mentioned above, from the options indicated in the general workflow (Fig. 2) to get initial models of structural elements, in this tutorial we are going to use tools to predict structure from sequence homology.

Structure prediction by sequence homology only requires the sequence itself, 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 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. 16) (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.

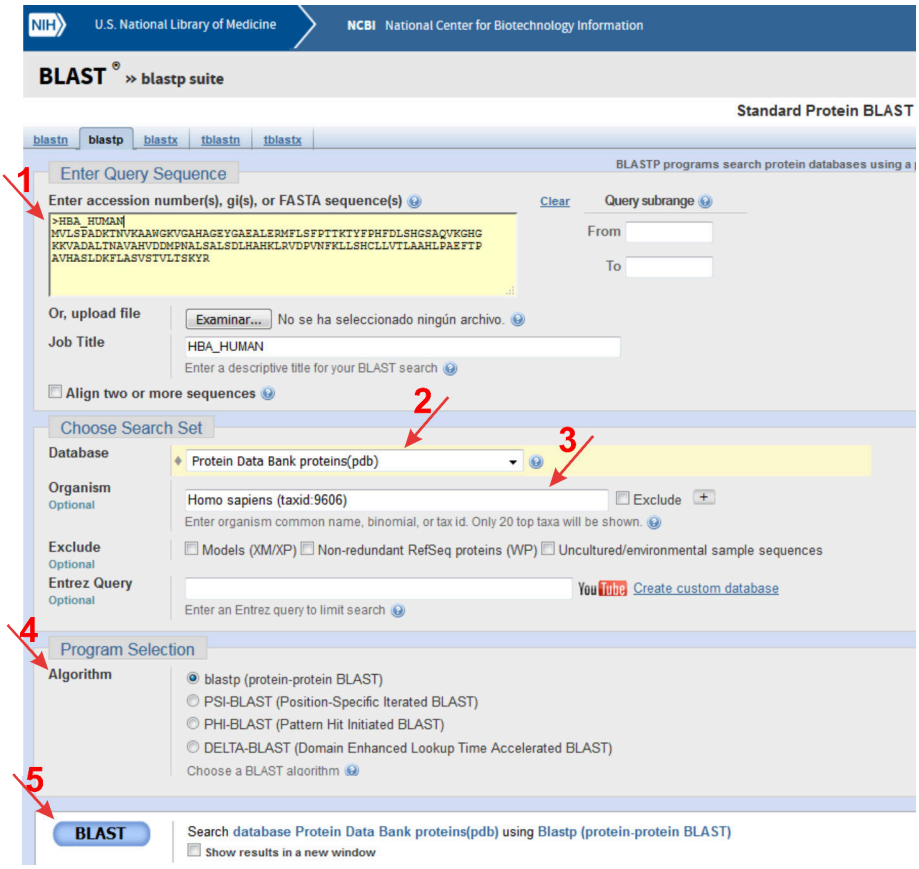


Figure 16: Form of the similarity searching program `BLAST`.

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 A; **Question 6_1**)

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

7 Moving from sequence to atomic structure scenario

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. 17 (1) and Appendix K). Select the option for importing the atomic structure from ID (2), write the `PDB` accession code (3) and execute the protocol (4). You can visualize the imported structure (5) in *Chimera* (Fig. 18). By selecting chain `A` in the *Chimera* upper menu (1) you can distinguish the `Hgb` α subunit (2).

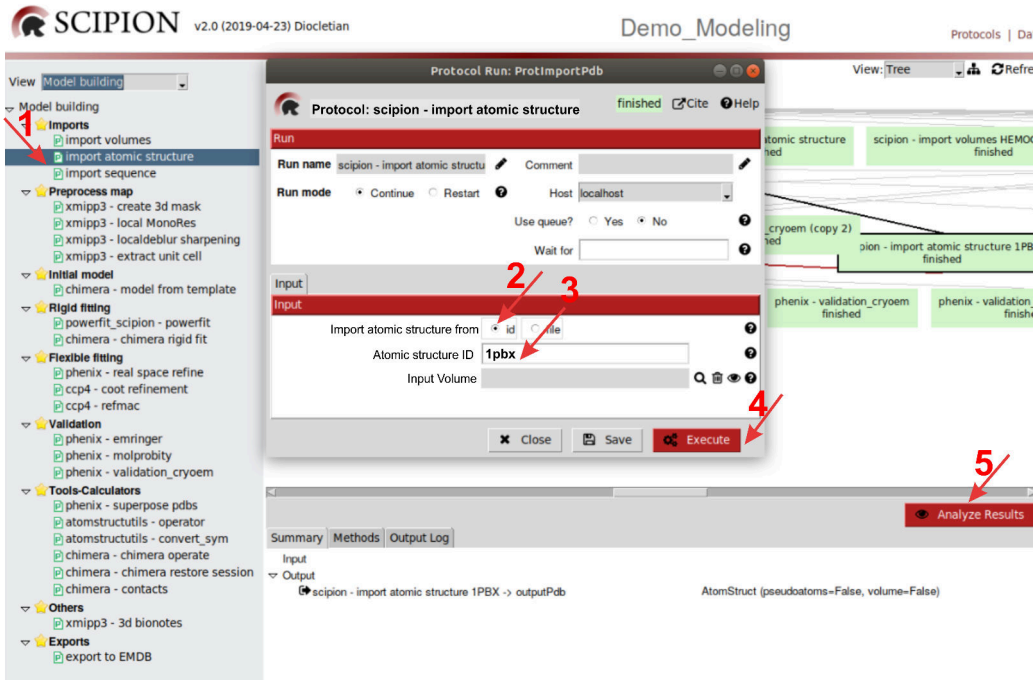


Figure 17: Importing the atomic structure `1PBX`.

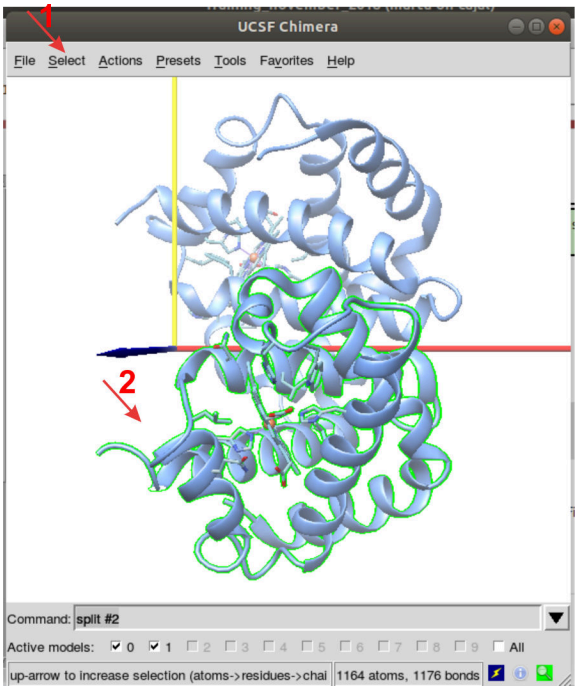


Figure 18: Atomic structure `1PBX` visualized with Chimera. `Hgb` α subunit (chain `A`) is shown green-highlighted.

Structural models of human metHgb subunits from templates

Modeller (Sali and Blundell, 1993) is one of the computational web services used by *Chimera*, 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*.

- Preparing your sequence alignment:
  In addition to the ways to obtain the *target-template* sequence alignment using *Chimera*, this alignment can be also generated in the *Scipion* protocol

model from template (Appendix P). 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* sequence, 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. 19). 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)`.

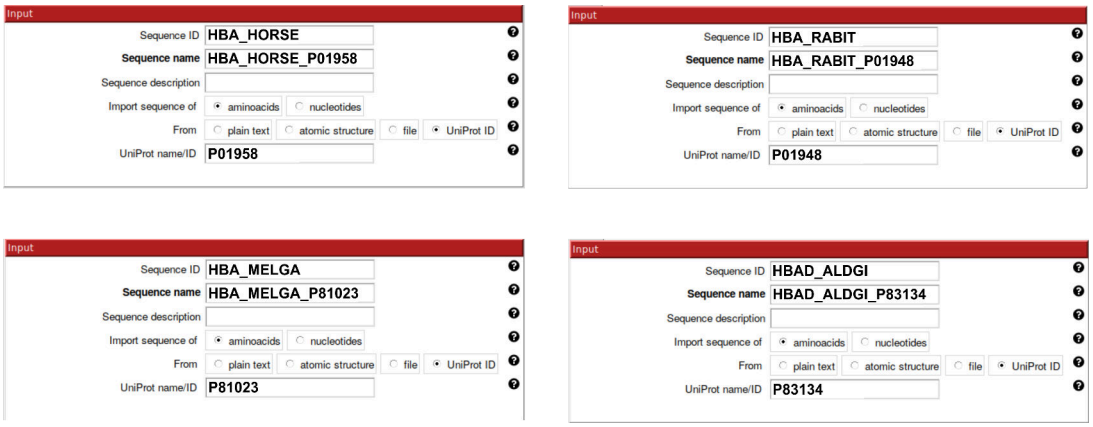


Figure 19: Importing additional sequences to perform the multiple sequence alignment.

- Access to *Modeller* in *Chimera*:
  The protocol model from template allows direct opening of the multiple sequence alignment in *Chimera* and then, access to *Modeller* via web service. Fill in the protocol form (Fig. 20 (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 perform a multiple sequence alignment, we'd like to include additional sequences to align (5), that have to add next (6). Finally, select one of the multiple sequence alignment tools (7).

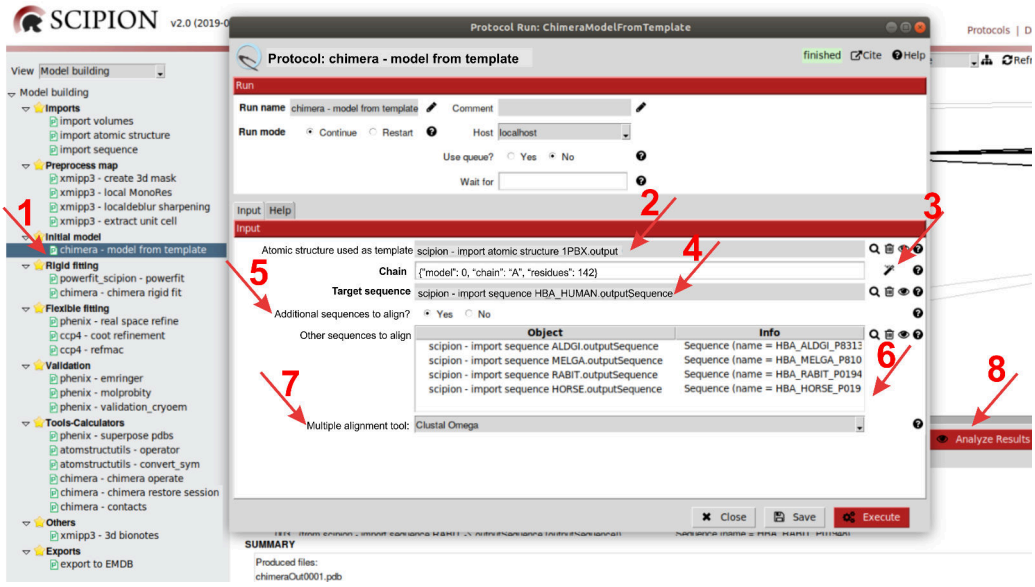


Figure 20: Importing the multiple sequence alignment in *Chimera*.

A couple of windows will be open, the multiple sequence alignment, in the upper part of Fig. 21, and *Chimera* graphics window. 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 has disappeared throughout the evolutionary process. Human `Hgb` α subunit is closer to the protein in mammals (horse, rabbit) than to the protein in unrelated organisms, as we would have anticipated. Corroborate this point by checking the identity percentage %ID between human sequence and the other sequences in Fig. 22 (B).

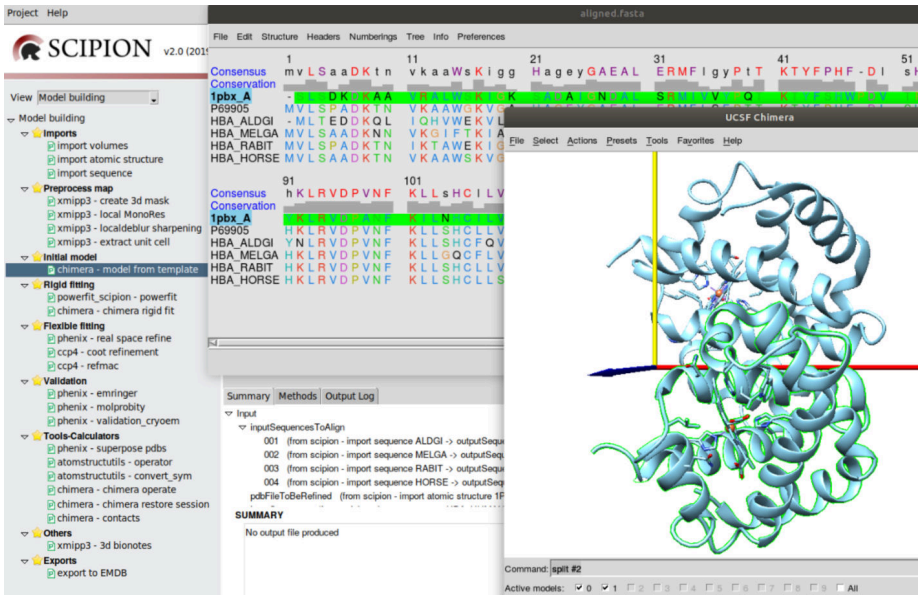


Figure 21: Opening the multiple sequence alignment in *Chimera*.

In case you'd like edit the name or the order of the sequences, add, delete or realign sequences, ..., go to the upper menu of the multiple sequence alignment (`Edit ->`). In particular, we are going to change the name of the *target* sequence from `P69905` to `HBA_HUMAN` in `Edit -> Edit Sequence Name...` (Fig. 22 (A)). To get possible atomic models of the *target* sequence in *Modeller* web service, we have to select `Structure -> Modeller (homology ...)` in the same upper menu. A new window for Comparative Modeling with Modeller will be open (Fig. 22 (B)), that we have to fill in selecting *target* sequence (1) and *template* (2). Modeller license key has to be included here (3). The number of output models can be specified in Advanced Options (5 by default). Finally, press Apply to start the computation without hiding the panel (4), or press OK to start the computation hiding it. In *Chimera* 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. 22 (C)) 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*). Let's select *model* `#2.1`. You can check every model numbers in *Chimera*'s main menu (`Favorites -> Model Panel`).

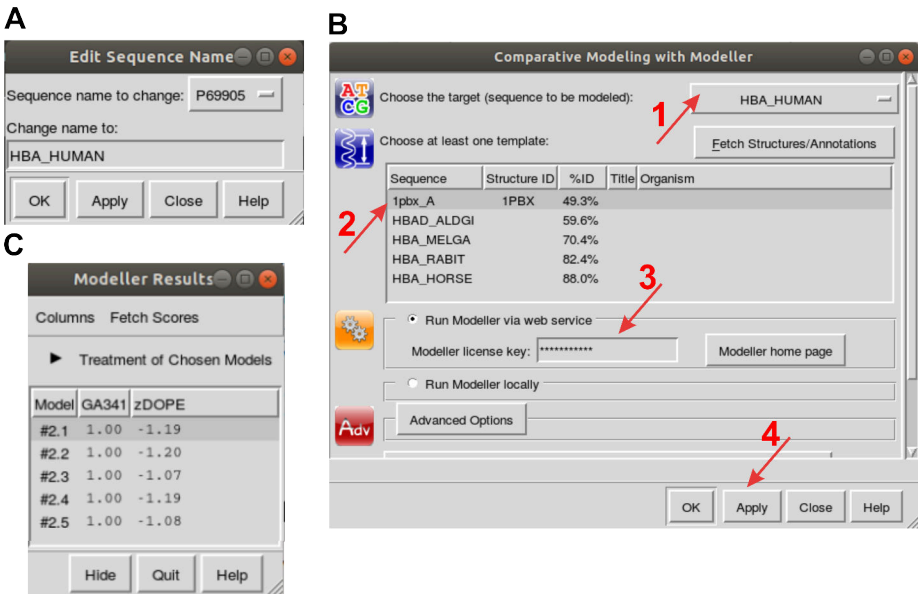


Figure 22: (A) Sequence edition. (B) Completing the form to access to homology modeling with *Modeller*. (C) Resulting *model scores*.

Comparing your selected *model* of human `metHgb` α subunit with `1PBX_A` *template*, we observe that the selected *model* `#2.1` does not contain the `HEME` prosthetic group (Fig. 23 (A)). Before saving the *model*, the *template* `HEME` group will thus be added to your *model* in *Chimera*. With this aim, open *Chimera*'s command line; from *Chimera* main menu (`Favorites -> Command Line`) and delete every atom of `1PBX` *template*, except the `HEME` group associated to chain `A`. To preserve residue 144 (`HEME` group) in chain `A`, write the next command line (Fig. 23 (A)(1)):

```
    delete #1:0-143.A, 145-.A, .B
or
    delete #1:0-143.A
```

```
delete #1:145-.A
delete #1:.B
```

Go to the Model Panel and select models `#1` and `#2.1` simultaneously, then press `Copy/Combine` on the right side column (Fig. 23 (B)(2)). Model Panel shows the new *model* created `#3` (3), that we can save in the command line (4) writing:

```
scipionwrite model #3
```

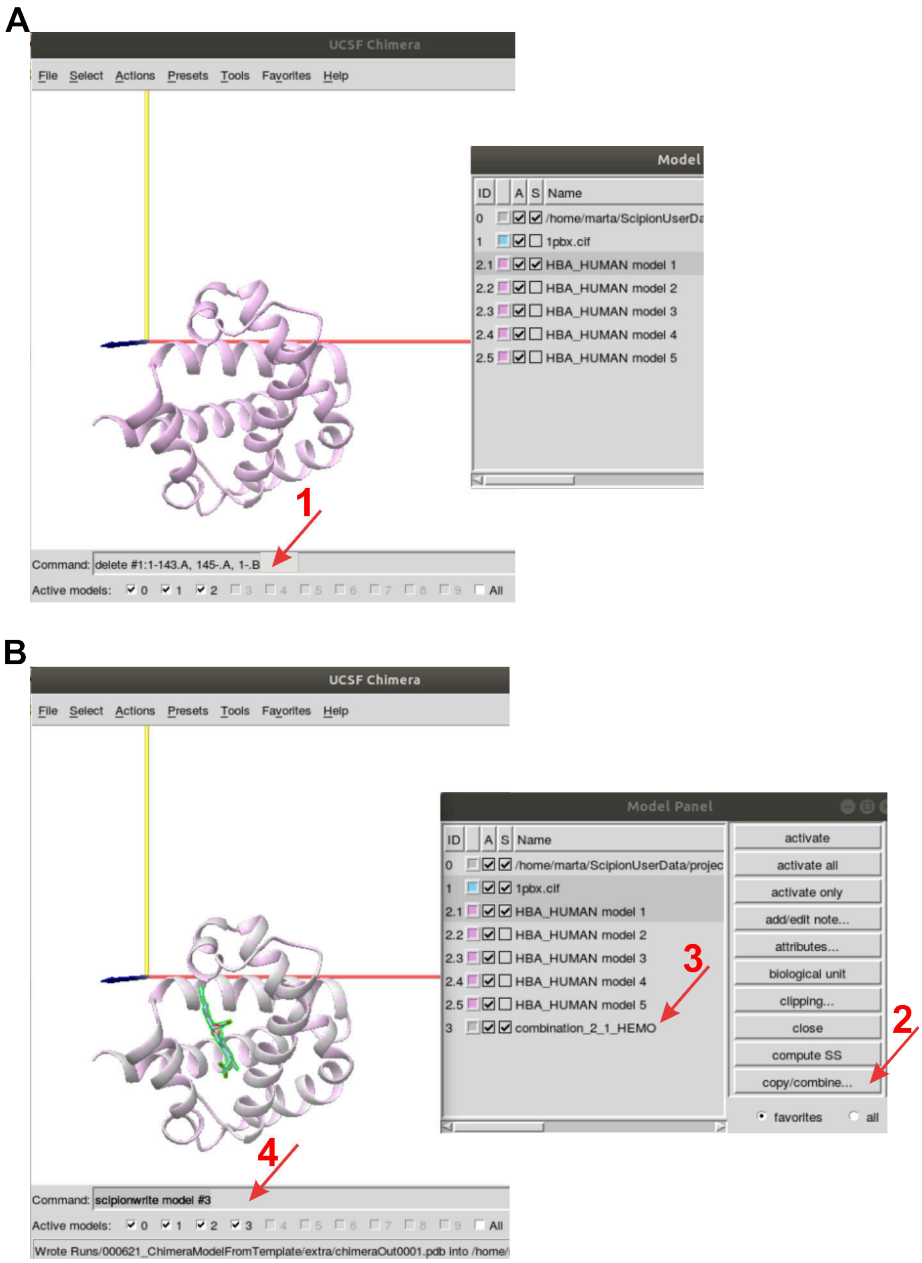


Figure 23: *Model* selection in *Chimera*. (A) Selected *model* `#2.1`. (B) Creation of full final *model* of human `metHgb` α subunit, including `HEME` group.

After closing *Chimera*, you can visualize (Fig. 20 (8)) your full predicted *model*. In a similar process, you can also obtain human `metHgb` β subunit. Use this time a direct way of keeping the `HEME` group in your *model* of `metHgb` β subunit: Select the advanced option `Include non-water HETATM residues from template` included in `Comparative Modeling with Modeller` window (Fig. 22 (B)).

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 chimera restore session protocol (Appendix E). This protocol may be used whenever *Chimera* session had been saved, specifically after using protocols *Chimera* `rigid fit`, *Chimera* `operate`, and *Chimera* `model from template`. In addition to the *Chimera* command line `scipionss`, command line `scipionwrite` also saves *Chimera* session by default. So, if you want to restore a previous session just open the form (Fig. 24, 1), and include the session that you'd like to restore (2).

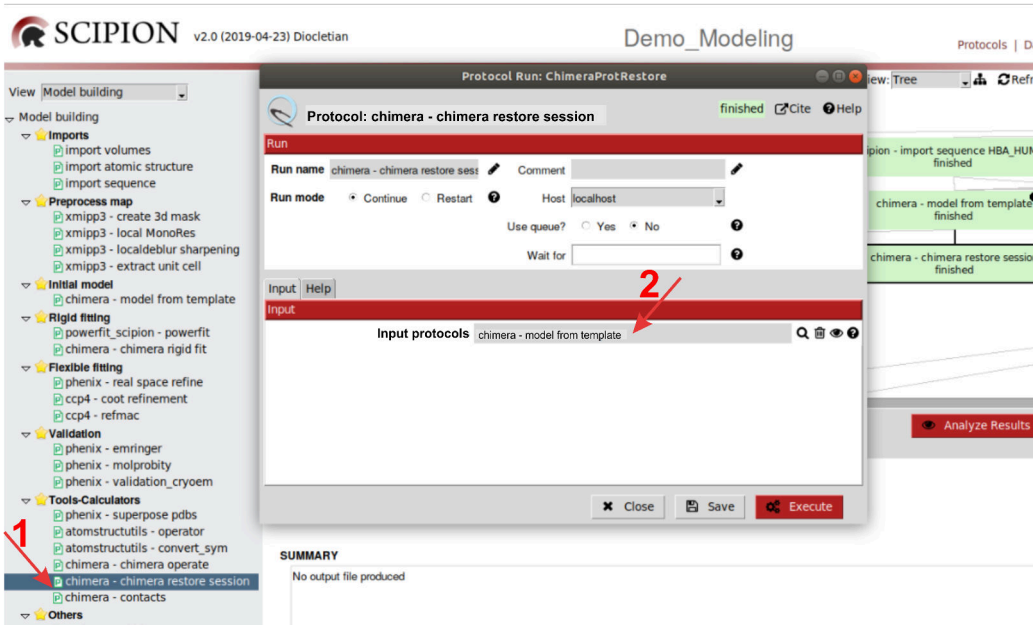


Figure 24: Restoring session in *Chimera*.

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, powerfit (Appendix V, (Van Zundert and Bonvin, 2016)) and chimera rigid fit (Appendix F). 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 *PowerFit* application first, followed by the final `Fit in Map` in *Chimera* `rigid fit`. In our experience *PowerFit* performance decays for large 3D maps or when fitting atomic models that cover a small region of the 3D map.

Initial rigid fit with *PowerFit*

Open powerfit protocol ((Fig. 25 (1)), and complete the form with the *model* of atomic structure previously saved in *Chimera* (2), the extracted unit cell volume (3), and volume resolution (4). Among the advanced parameters, consider carefully the angular step (5) according to the size of your volume and your computing power. Despite getting a more accurate result, fitting of large volumes takes more time as lower values of angular step are requested.

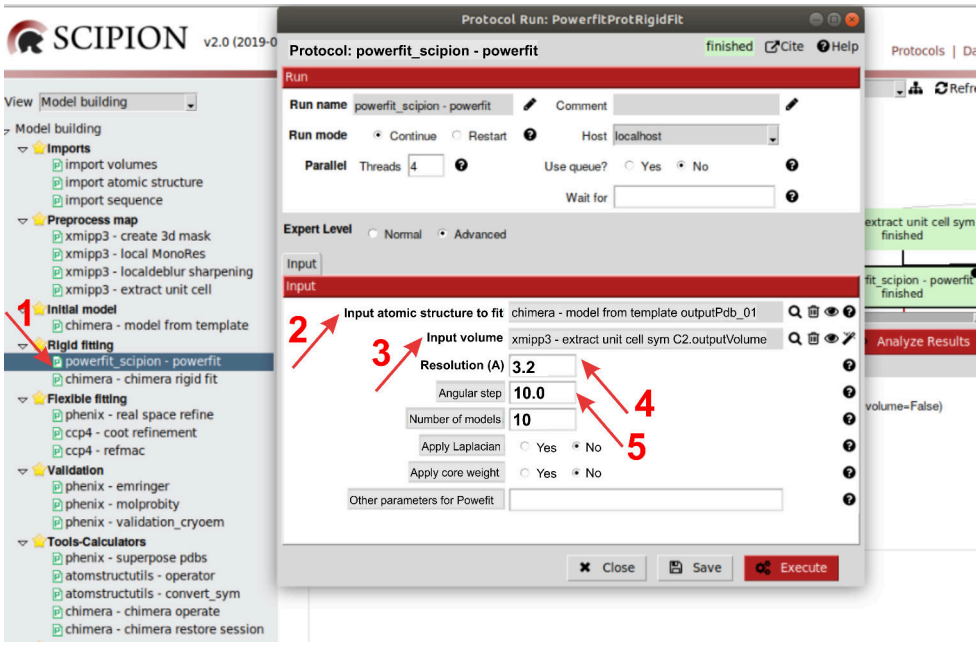


Figure 25: Rigid fit with *PowerFit*: Filling in the protocol form.

After executing the powerfit protocol, you can check results ((Fig. 26 (A)(1)). The first table opened allows you to check the fitting quality (2) of best score-ordered fits in a second table (B). In our example only two fits are proposed. You can check which one fits better to the map by writing the selected fit number in the `Model to visualize` square window, and then displaying the fitting (3).

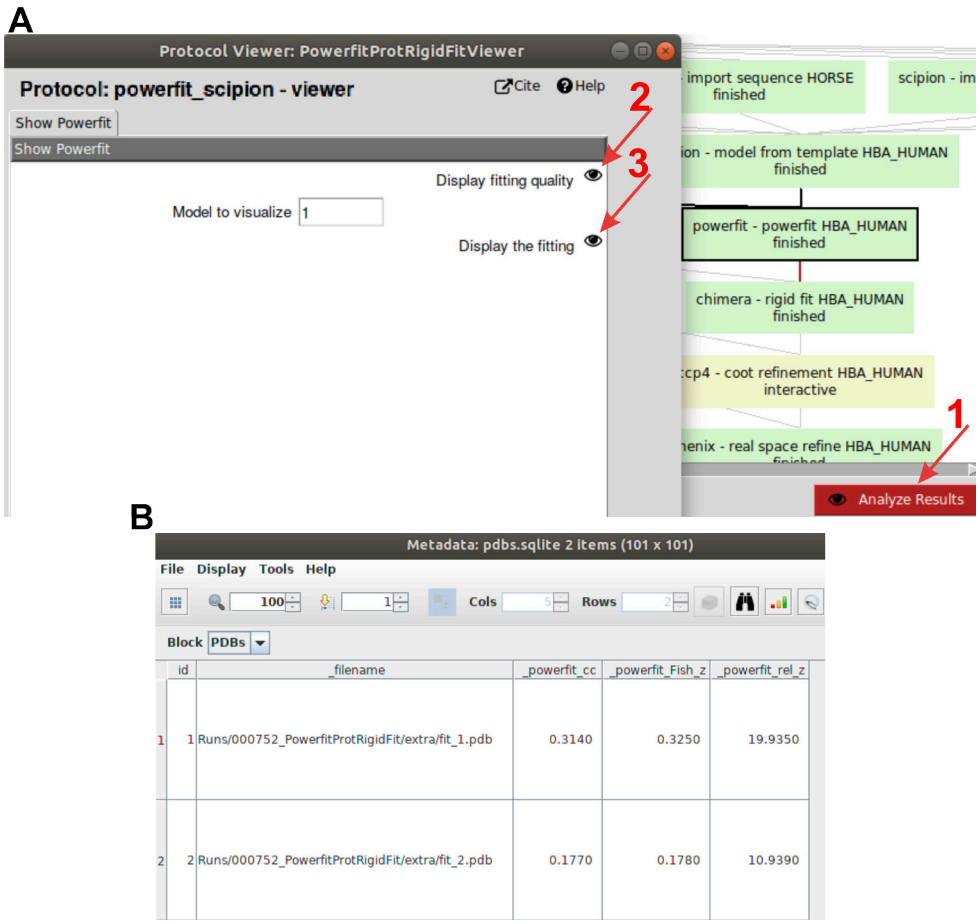


Figure 26: Rigid fit with *PowerFit*: Checking results by score.

Fig. 27 shows the fitting of the two possible fits between map and `models` posed by *PowerFit* (`fit_1.pdb` (A), and `fit_2.pdb` (B), green- and pink-colored, respectively). Although both of them seem to fit quite well the extracted unit cell map, only one of them should be OK. The other *model* one must misfit the volume area that corresponds to `metHgb` β subunit. This anomalous behavior of our *model* is not surprising because `Hgb` α an β subunits are 42.86% sequence identical.

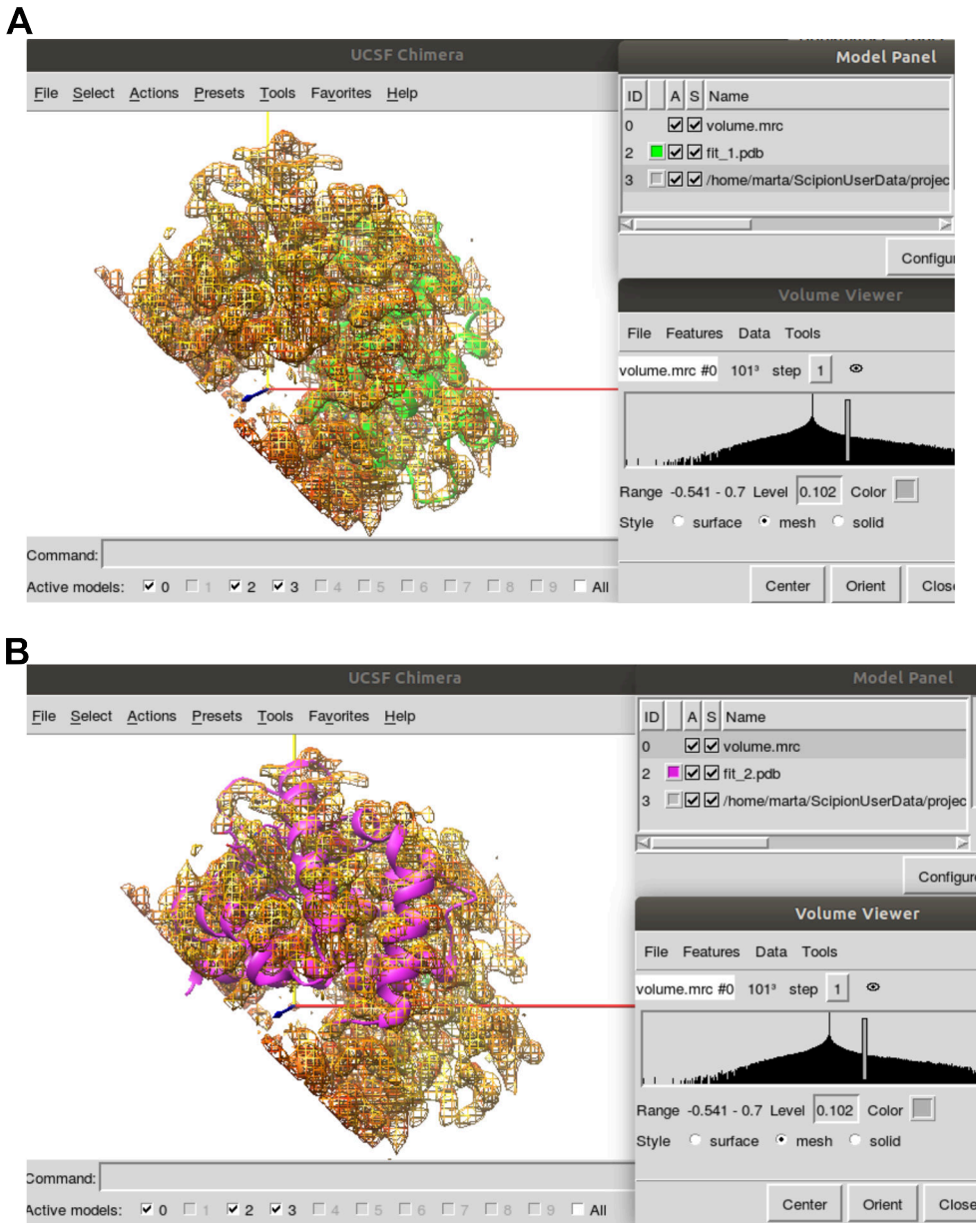


Figure 27: Rigid fit with *PowerFit*: Checking the best fits in *Chimera*.

Completing rigid fit with *Chimera* `rigid fit`

In order to assess which one of the structures fits the map better, the fitting has to be improved with chimera rigid fit protocol. Open this protocol (Fig. 28 (1)), include the extracted unit cell as volume (2), one of the structure fits previously obtained with *PowerFit* (3), the other one (4), and execute the protocol.

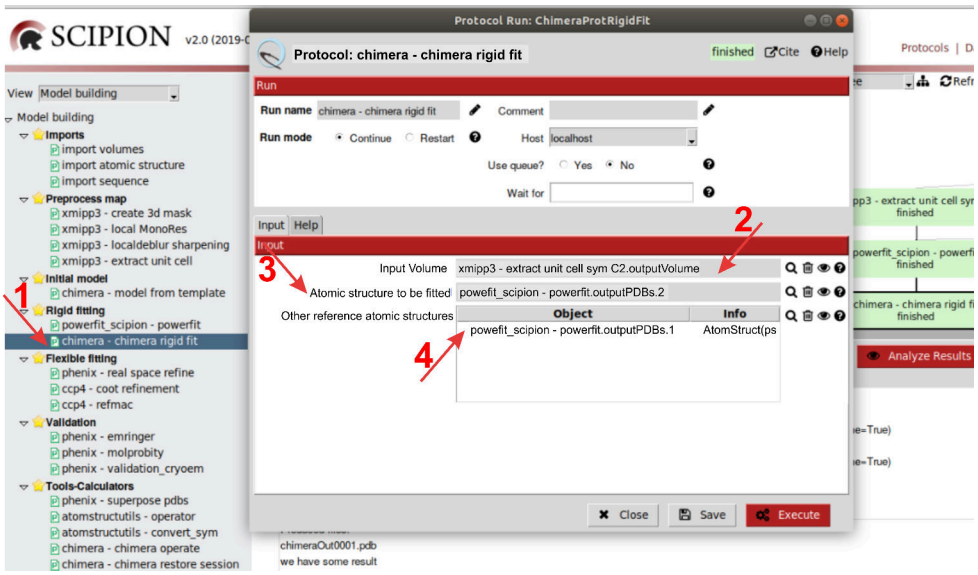


Figure 28: Completing the *Chimera* rigid fit protocol form.

Once opened *Chimera* graphical window, select in the upper main menu `Tools -> Volume Data -> Fit in Map`. A small window will be opened. Select each one of the fits in `Fit` and press `Fit` to allow the automatic rigid fitting. At the lower right side of Fig. 29, respective windows of `Fit in map` for `fit_1.pdb` and `fit_2.pdb` are detailed. In spite that the amount of `atoms outside the contour` is higher in `fit_1.pdb` than in `fit_2.pdb`, we can not conclude that the second fitting is better than the first one, because the `Average map value` is also lower. Visual inspection of both fits should identify the appropriate fitting of *model* in map.

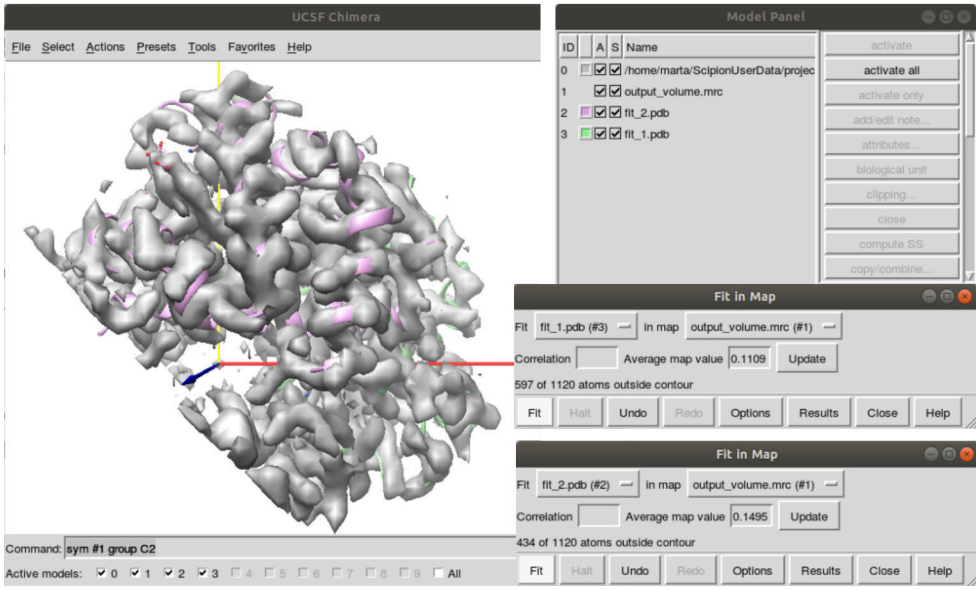


Figure 29: Fitting in map with *Chimera*.

As suggestion, before starting the visual inspection in *Chimera*, check first the parts of the structure that could differ the most between `metHgb` α and β subunits. These divergent parts can be identified by performing an additional alignment including the β subunit sequence. Possible steps to perform this multiple alignment: (1) import the `metHgb` β sequence (UnitProt id=P68871), (2) copy the protocol model from template and (3) add the `metHgb` β in the field `other sequences to align`. Fig. 30 shows the result of incorporating the β subunit sequence (in yellow) to the alignment shown in Fig. 21. Two regions that contain gaps of sequence are remarked in red frames. The first one, involving residues 19 and 20, absent in β subunit, and the second one, after residue 51, could be the most relevant to differentiate between subunits. Look at them and identify the correct fit. You can highlight those residues in *Chimera* writing in the command line:

```
sel :19-20
sel :50-55
```

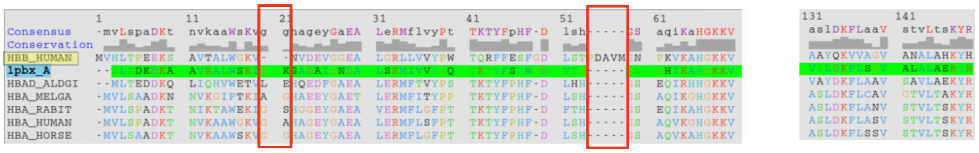


Figure 30: Multiple sequence alignment including `Hgb` β subunit (`HBB_HUMAN`).

Once identified the appropriate fit, you can save it as fitted *model* of the `metHgb` α subunit. Replacing `n` by your *model* number (`1`, `2`), write in the command line of *Chimera* graphical window:

```
scipionwrite model #n refmodel #1 saverefmodel 0
```

In case you are still unable to decide which one is the best fit, don't worry. You will make your mind up with the next step in the workflow, but then save both fits with `scipionwrite` *Chimera* command line. Don't forget to change the *model* number. Your fitted models will be saved by *Chimera* with names `chimeraOut0001.pdb` and `chimeraOut0002.pdb`.

9 Refinement: Flexible fitting

Although the rigid fitting approximates map and atomic *model*, a detailed visual inspection of map and model reveals that part of 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 Fourier space, can be followed. Implemented in *Scipion* are two protocols for real space refinement, ccp4 - coot refinement (Appendix G, (Emsley et al., 2010)) and phenix - real space refine (Appendix T, (Afonine et al., 2018b), and one protocol to refine in reciprocal space, ccp4 - refmac (Appendix H, (Vagin et al., 2004)).

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 ccp4 - coot refinement protocol (Fig. 31 (1)), load the extracted unit cell volume (2), with electron density normalized to 1, and the fitted structure *model* (3). Load also the second fit (4) if you are not still sure about the correct fitted structure. Reading the protocol Help is recommended. After executing the protocol (5), *Coot* graphics window will appear to start working.

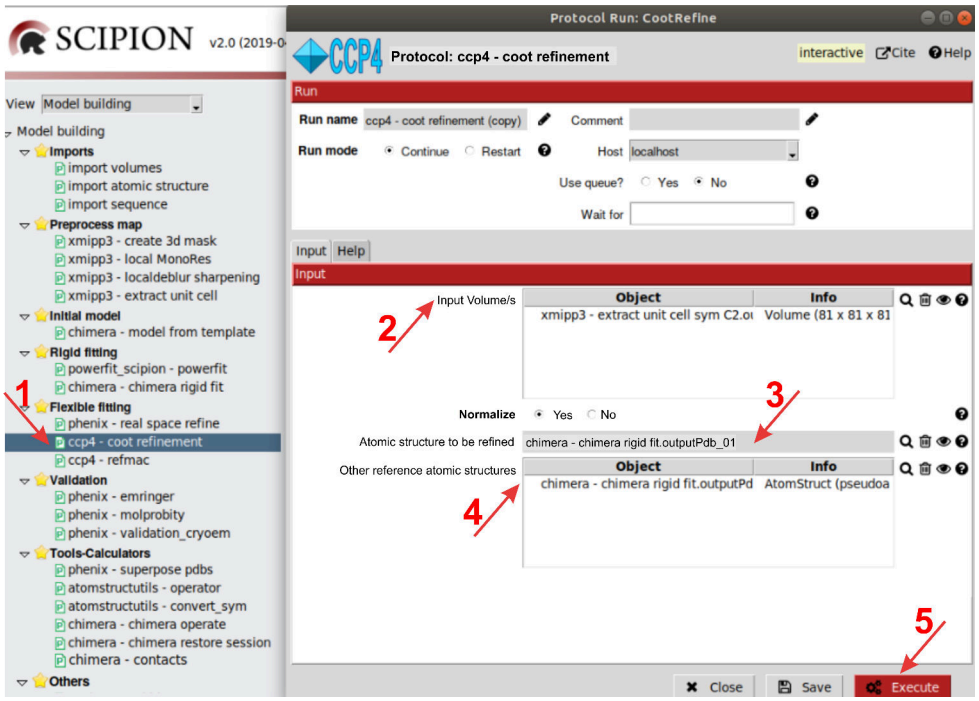


Figure 31: Filling in *Coot* refinement protocol.

To check the objects downloaded in *Coot*, go to the second bar of the main menu and select `Display Manager`. Map `(output_volume.mrc)` (number `#2`) and models `chimeraOut0001.pdb` and `chimeraOut0002.pdb` (numbers `#1` and `#0`, respectively) are displayed. To start, we are going to identify the faired fitted *model* to the density map in order to delete in the `Display Manager` menu the other *model*, which is misfitted. Visual inspection would clarify this point, 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`. This density analysis is compared for the two possible fitted *models* in Fig. 32. As you can see, model `chimeraOut0001.pdb` shows that residues 19, 20 and 22, framed in Fig. 30, do not fit to the density map, as expected from the misfit of the α subunit in the density of the β subunit.

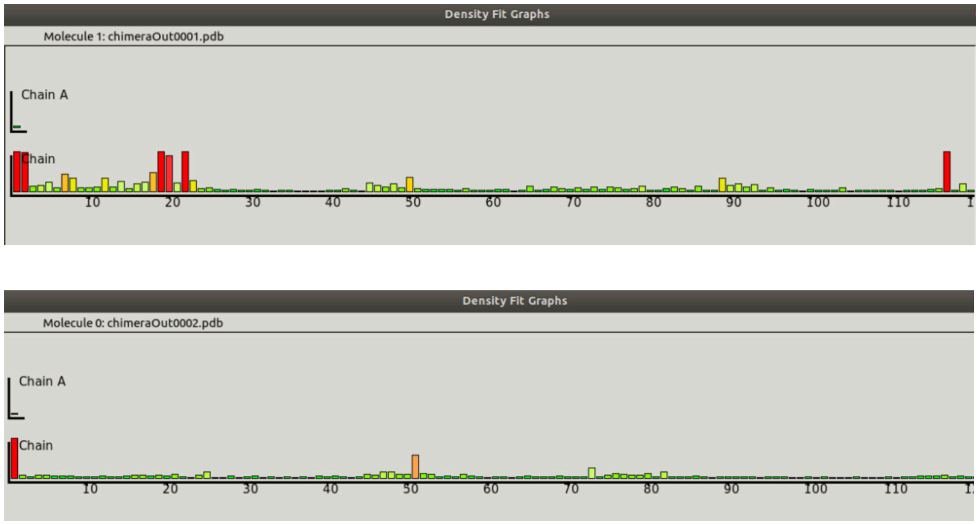


Figure 32: *Coot* comparison of *model* fit in the map density.

Go again to `Display Manager` and delete the *model* `chimeraOut0001.pdb` pressing `Delete Model`. From now ahead, the *model* `chimeraOut0002.pdb` will be refined in the next steps of the modeling workflow.

Before starting the refinement, IDs of chains should be fixed. Current IDs of chains are `Chain A` and `Chain` (see Fig. 32), and will be changed to `Chain HEME`

and `Chain A`, respectively. This can be carried out going to main *Coot* menu and selecting `Edit -> Change Chain IDs`. Verify the identity of your *model* molecule (Fig. 33 (1), select the initial Chain ID (2, 4), and write the new Chain ID (3, 5). Finally, press `Apply New Chain ID`.

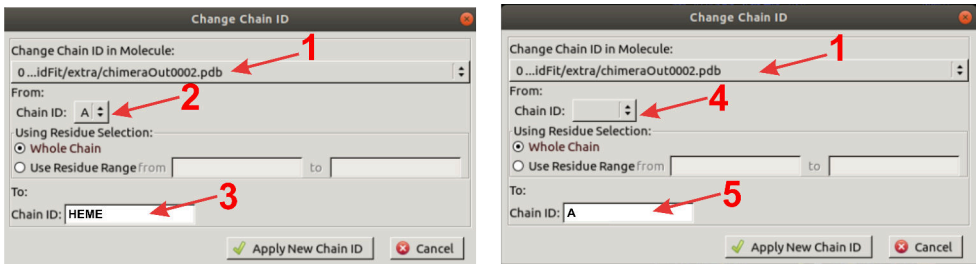


Figure 33: *Coot* change chain ID of *model* `chimeraOut0002.pdb`.

According to Fig. 32, `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. 34 (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. 34 (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 that the red bar shown in `MET` residue (Fig. 32) has disappeared.

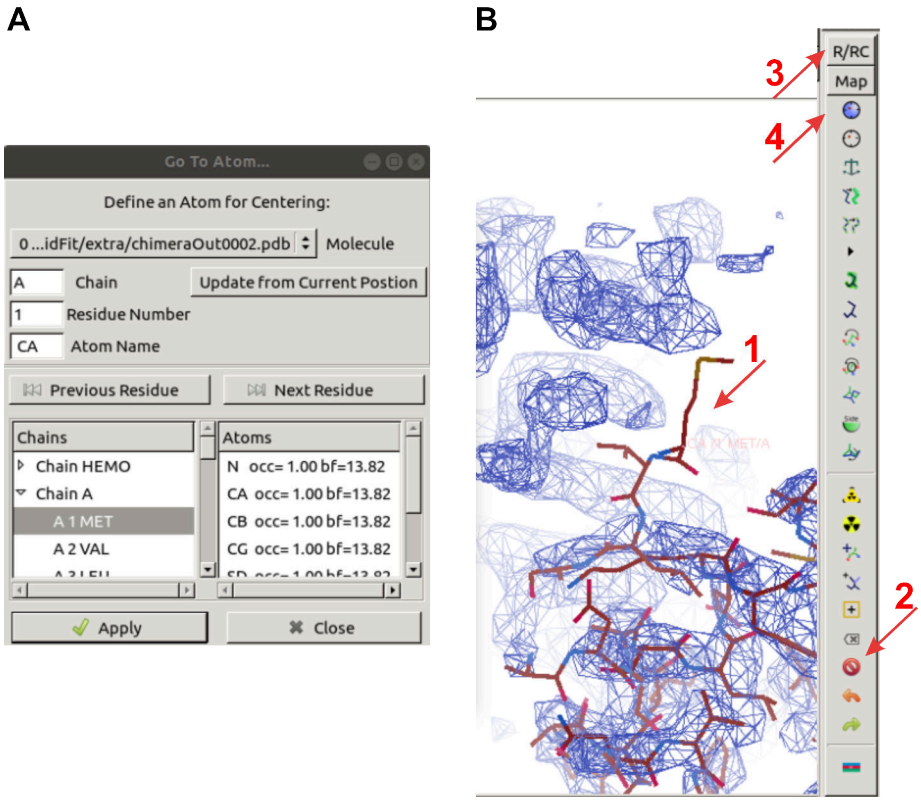


Figure 34: Removing post-translationally processed Methionine residue in *Coot*. Note that the icons shown in the image right side may be partially hidden if the screen is small.

Before a more detailed visual inspection of the *model* fitting, an initial quick refinement may be accomplished. With this purpose, first of all, go to the upper right side menu (Fig. 34 (B)(3)) and select all four restrictions for `Regularization and Refinement` in the respective window of parameters. Secondly, open the `coot.ini` text file, open *Scipion* browser and navigate to the `extra` directory. Modify the file so it matches the information shown below (See Fig. 35).

```
[myvars]
imol:  0
aa_main_chain:  A
```

```
aa_auxiliary_chain:  AA
aaNumber:  4
step:  10
```

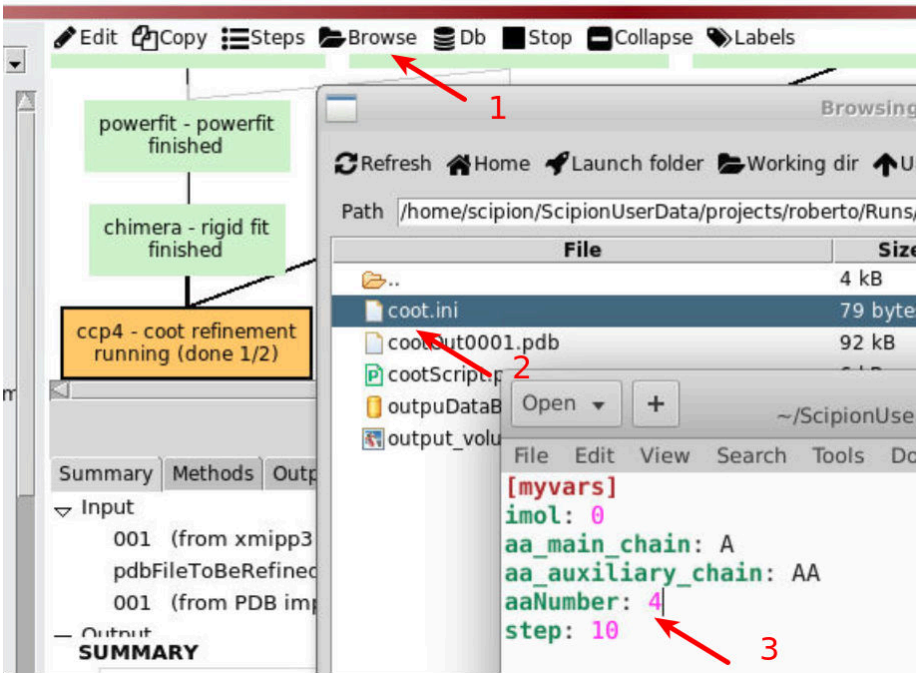


Figure 35: Edit coot.ini file.

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 orange bar of residue number 50 (Fig. 32) 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. 34 (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. 36, 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`.

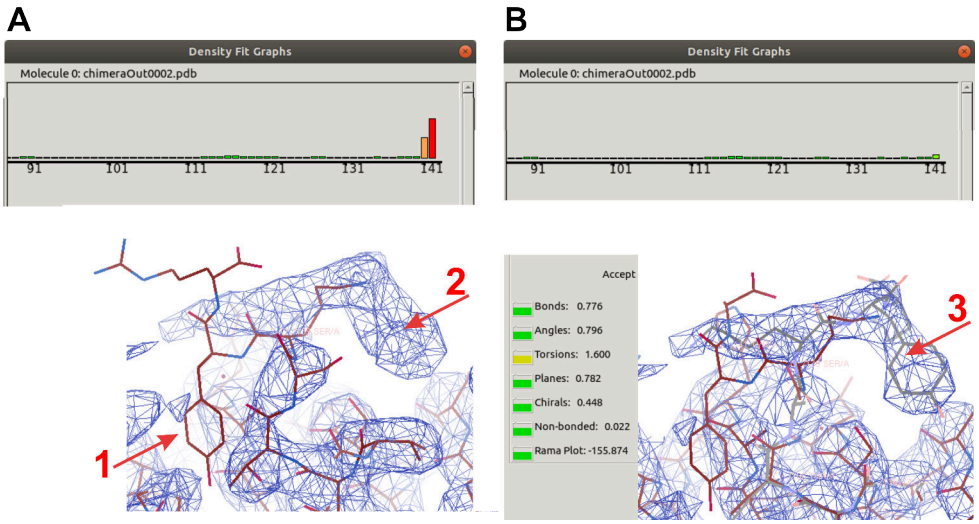


Figure 36: *Coot* fit in the map density of residue `TYR` 141.

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. You can save the atomic structure with its default name by pressing w. If you prefer another name, for instance "HBA_HUMAN.pdb", it can

be saved in *Coot* main menu `Calculate -> Scripting -> Python` and the `Coot Python Scripting` window will be opened. Assuming that `0` is your *model* number, write in Command:
`scipion_write (0, 'HBA_HUMAN')` ⏎

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 restart the process in the last refinement step, maintaining each one of the intermediate refined structures saved in order in *Scipion* tutorial folder `/Runs/000XXX_CootRefine/extra`. In this way, 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 protocol *Coot*.

A similar refinement process to that followed in *Coot* for `metHgb` α subunit chain `A`, has to be carried out for chain `HEME` and for respective chains of `metHgb` β subunit.

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. This protocol 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. 37 (1)), load as input volume the extracted unit cell saved in *Coot* (2), write the volume resolution (3), and load the atomic structure (*model* `chimeraOut0002.pdb`, (4)). After executing the protocol (5), results can be checked (6).

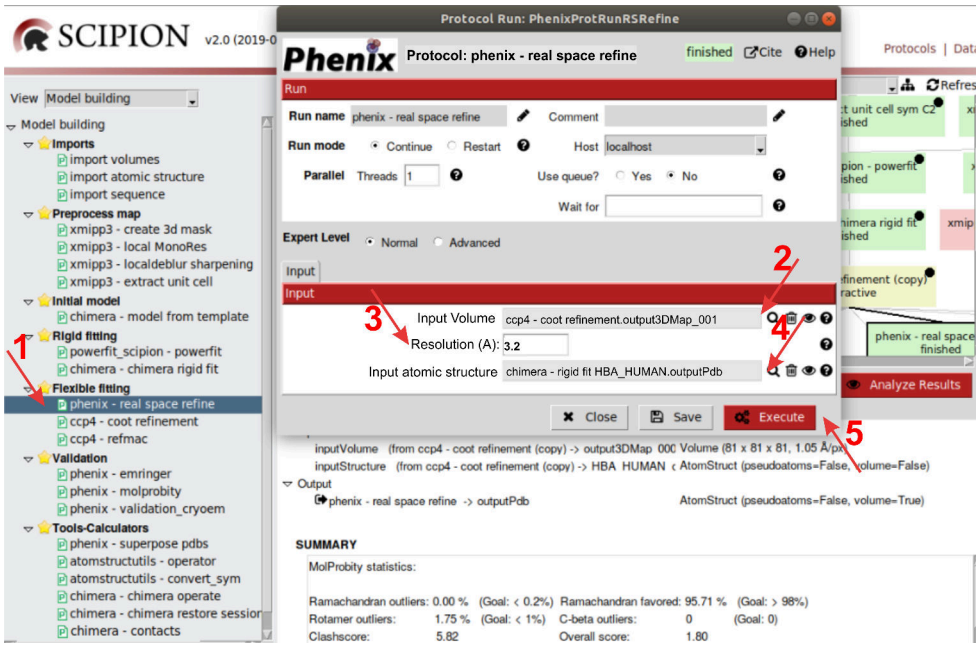


Figure 37: Completing *PHENIX* Real Space Refine protocol.

The first tab of results shows the initial *model* atomic structure as well as the refined one, both fitted to the normalized extract unit cell volume saved in *Coot* (Fig. 38).

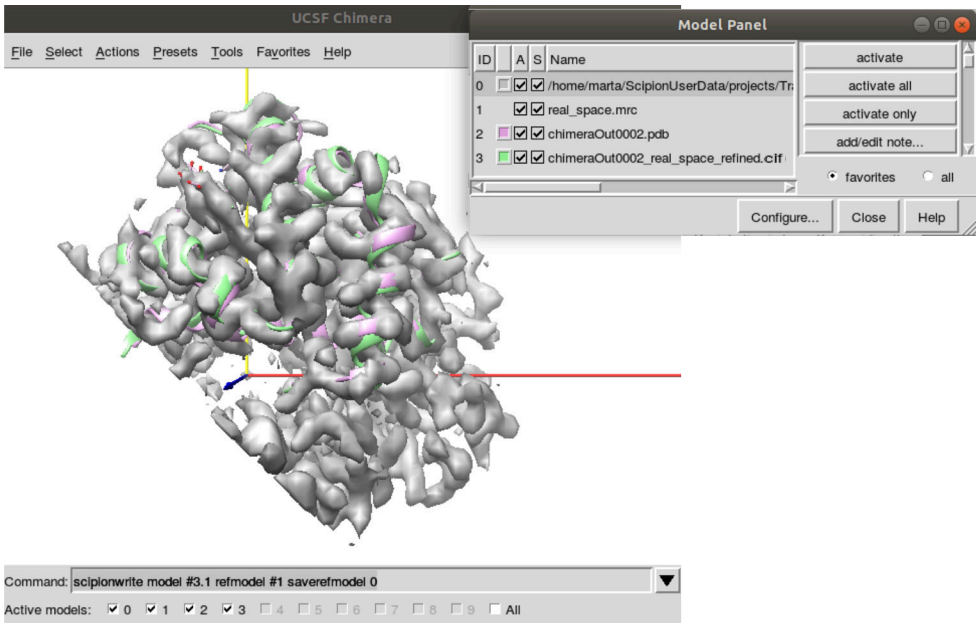


Figure 38: *Chimera* visualization of refined *model* of `metHgb` α subunit by *PHENIX* Real Space Refine protocol.

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}, that 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 correlation values and answer the following questions: (Answers in appendix A; **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?

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 arrangements of molecules are involved. Now, to take advantage of *model* improvements performed with *Coot*, run *PHENIX* `real space refine` after *Coot*. When you finish, check again the above values of correlation. Have they changed? (Answer in appendix A; **Question 9_2**)

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 A; **Question 9_3**)

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 input to perform a second refinement step in the Fourier space with *Refmac* protocol ccp4 - refmac. Firstly, open the *Refmac* protocol form (Fig. 39 (1)), load the volume generated by *Coot* (2), the atomic structure obtained with *PHENIX* `real space refine` (3), and the volume resolution as maximum resolution (4). Execute the protocol (5) and when it finishes, analyze the results (6).

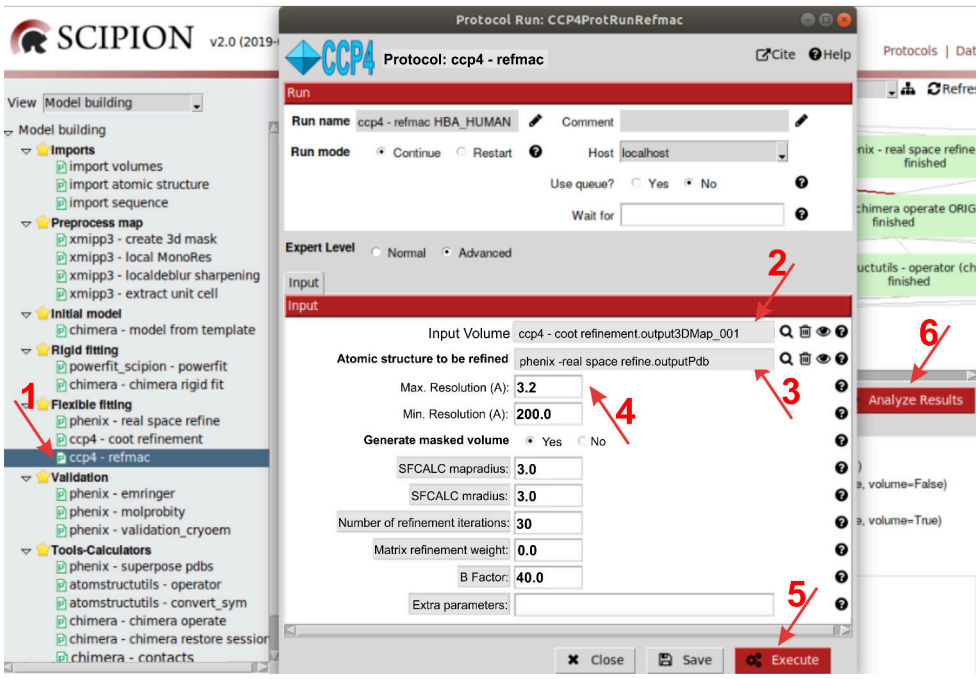


Figure 39: Filling in *Refmac* protocol.

Clicking the first item in the display menu of results (Fig. 40 (1)), *Chimera* graphics window will be opened showing the input volume, the initial *model* (`HBA_HUMAN` obtained with *PHENIX* `real space refine`), and the final *Refmac* refined *model* (Fig. 41). By clicking the third item in the display menu of results (Fig. 40 (2)), a summary of *Refmac* results are shown. Check if values of `R factor` and `Rms BondLength` have improved with this refinement process. Why the improvement seems to be very small? (Answers in appendix A; **Question 9_4**)

Would you have seen a higher improvement running *Refmac* immediately after

Coot, thus ignoring *model* improvements generated by *PHENIX* `real space refine`? (Answers in appendix A; **Question 9_5**)

Regarding the using of mask: Compare *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 unit cell` protocol. Are there any differences? Why? (Answers in appendix A; **Question 9_6**)

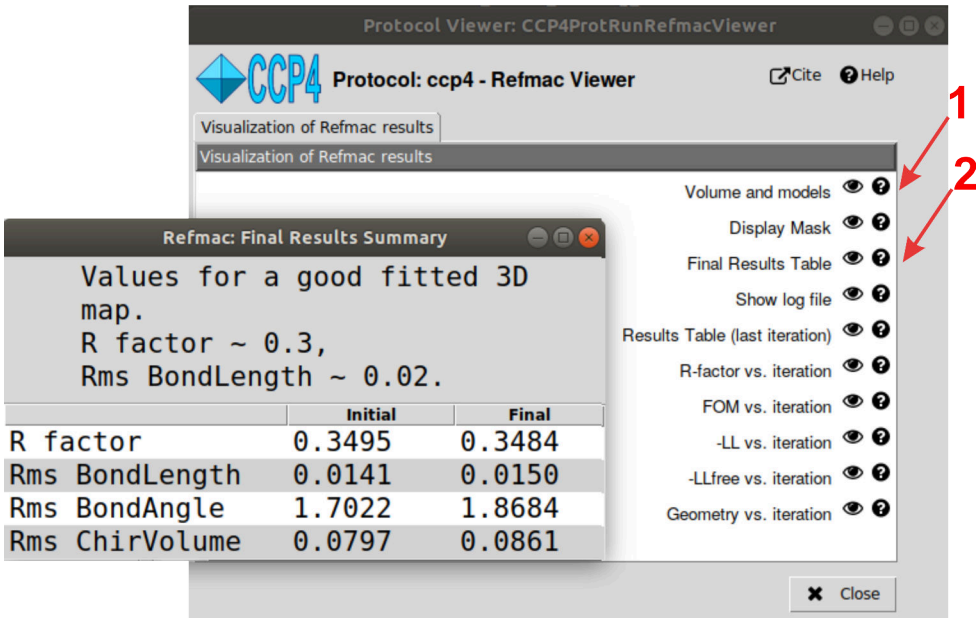


Figure 40: Display menu of *Refmac* results.

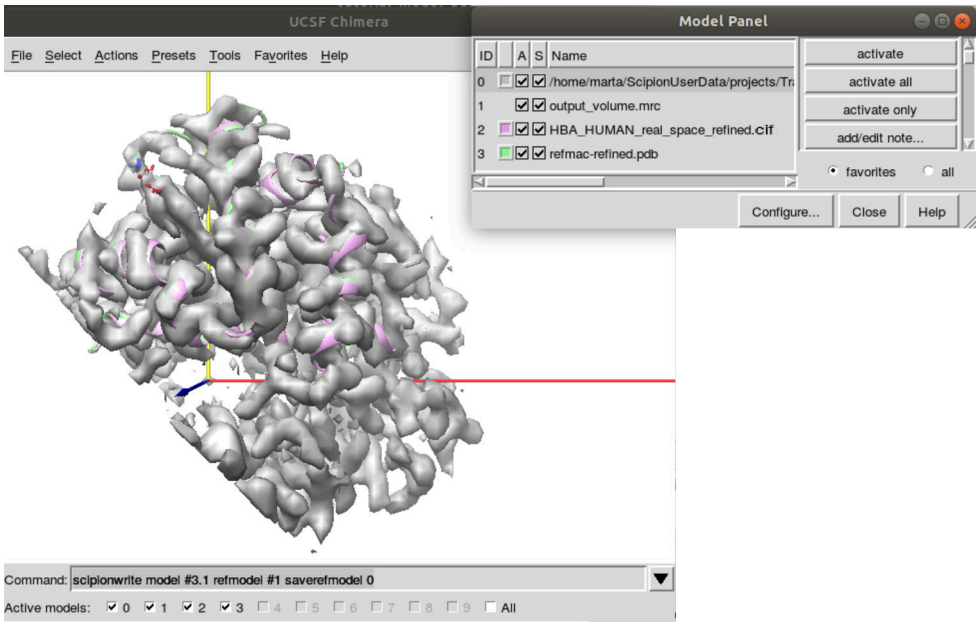


Figure 41: *Chimera* visualization of refined *model* of `metHgb` α subunit by *Refmac*.

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

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 Q, (Barad et al., 2015)), *MolProbity* (phenix - molprobity, Appendix R, (Davis et al., 2004)), and *Validation CryoEM* (phenix - validation_cryoem, Appendix S, (Afonine et al., 2018a)). *Validation CryoEM* protocol will show *MolProbity* validation

values as well as correlation coefficients in real space. Since old versions of *PHENIX* (v. 1.13) do not include this tool, then correlation values in real space will be computed if a volume is provided with *MolProbity* protocol. Additionally, we are going to introduce the protocol phenix - superpose pdbs (Appendix U, (Zwart et al., 2017)) useful to compare visually the geometry of two atomic structures.

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, Cα, Cβ) and (Cα, Cβ, Cγ) (see Fig. 42). 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.

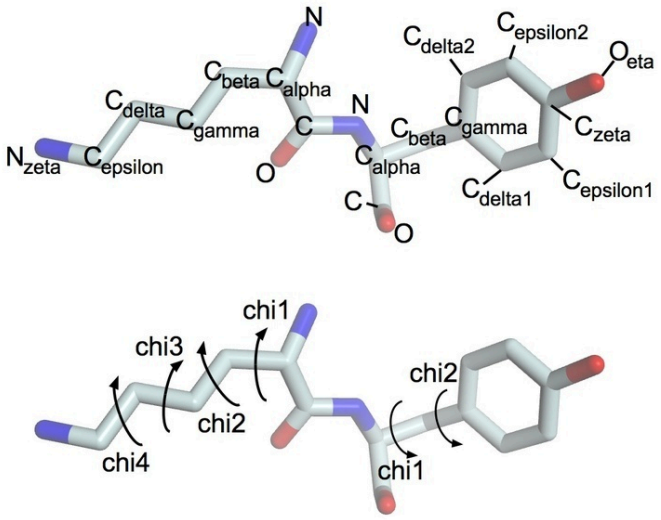


Figure 42: Naming convention in side chains explained in a lysine-tyrosine strand. Note that these two residues are within a protein and thus have no terminal region.

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. 43 (1)), load the extracted unit cell volume (initial or saved with *Coot*) (2) and the atomic structure that you'd like to validate in relation to the volume (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).

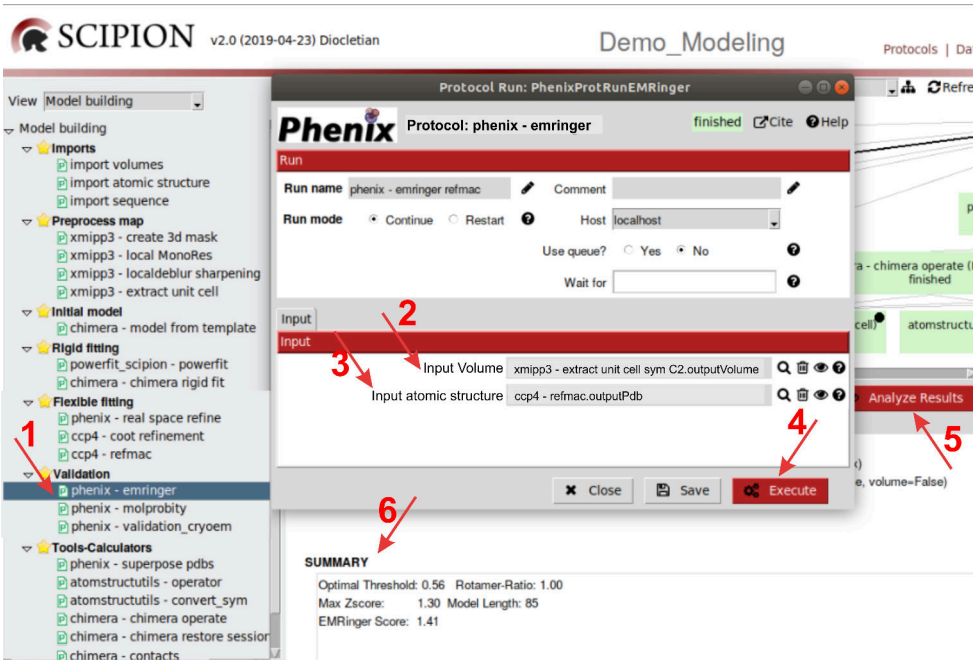


Figure 43: Completing *EMRinger* protocol form.

Run *EMRinger* protocol and determine the respective score after running *PowerFit item2*, *Chimera* `rigid fit` (*model* 2), *Coot* refinement, *PHENIX* `real space refine` after *Coot* (default conditions and last modification of form parameters), 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, have they been improved? (Answers in appendix A; **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. 44 (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).

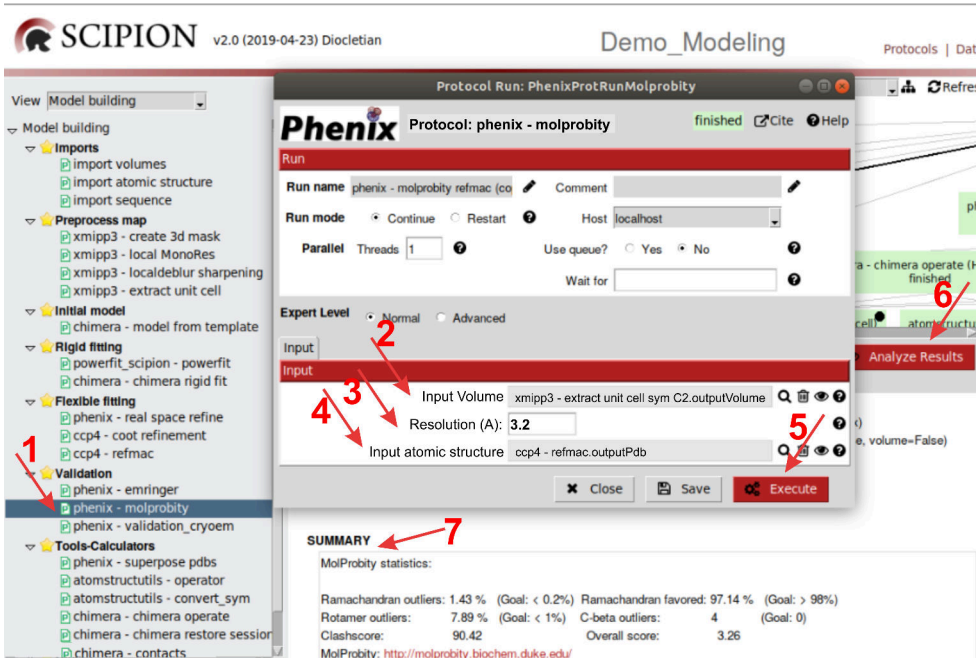


Figure 44: Completing *MolProbity* protocol form.

Run *MolProbity* protocol to obtain its statistics after running *PowerFit item2*, *Chimera* `rigid fit` (*model* 2), *Coot* refinement, *PHENIX* `real space refine` (default conditions and last modification of form parameters) 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. 45 (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` shows the same menu bars available in results section of *PHENIX* `real space refine` protocol. *MolProbity* results bar include validation statistics. Protocol `SUMMARY` emphasizes the most relevant ones.

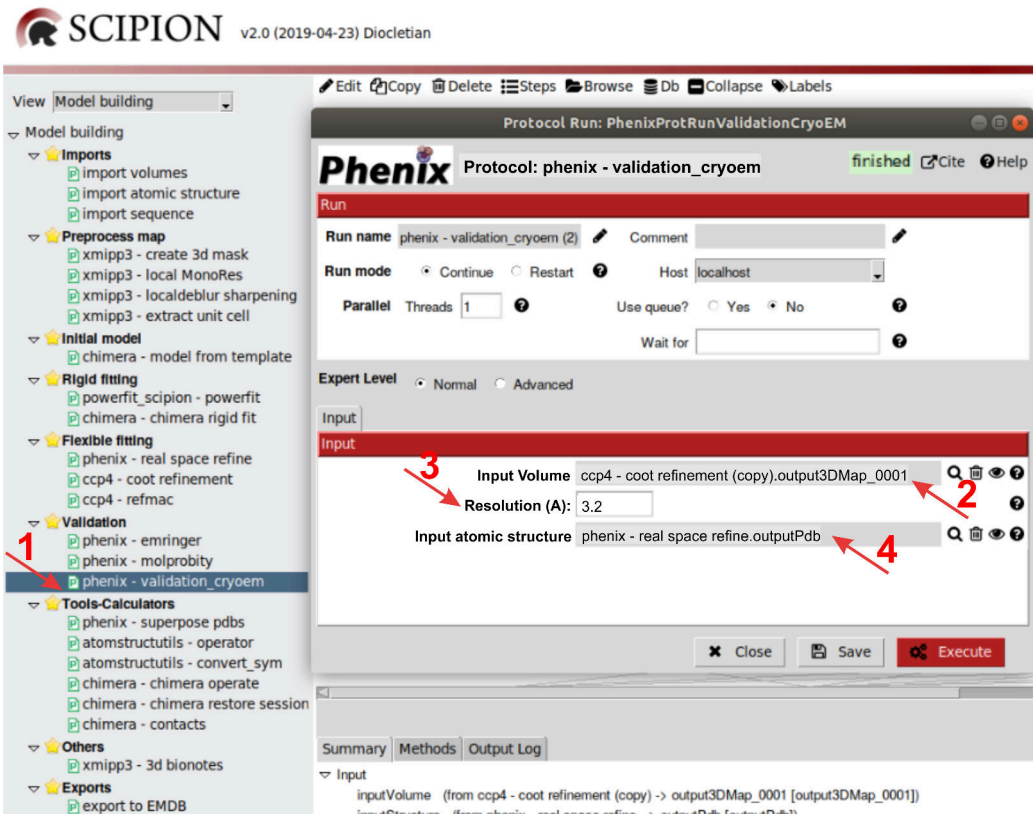


Figure 45: Filling in *PHENIX Validation CryoEM* protocol form.

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 A; **Question 10_2**). The same table (Table 2) can be completed for `metHgb` β subunit (Appendix A; **Question 10_3**)

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` (last modification of form parameters) after *Coot*. Considering *EMRinger* `score`, the best *model* derives from the whole workflow *Coot* `->` *PHENIX* `real space refine` (default conditions). With *MolProbity* `Overall score` as vali-

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

Statistic	*PowerFit* *item* #2	*Chimera* *model* #2	*Coot*	*PHENIX* `RSRAC` (default)	*PHENIX* `RSRAC` (modified)	*Refmac* after *Coot*	*Refmac* after `RSRAC` (modified)	`5NI1`
CC_{MASK}								
EMRinger `score`								
`RMS` (Bonds)								
`RMS` (Angles)								
`Rama favored` (%)								
`Rama allowed` (%)								
`Rama outliers` (%)								
`Rotamer outliers` (%)								
`Clashscore`								
`Overall score`								
Cβ `deviations`								
`RMSD`								

dation 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 next workflow:

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

- Protocol chimera operate (Appendix D):
  Similar to *Chimera* `rigid fit`, *Chimera* `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. 46 (1)), complete the parameter `PDBx/mmCIF` including the atomic structure `5NI1` previously imported (2), and execute the protocol (3).

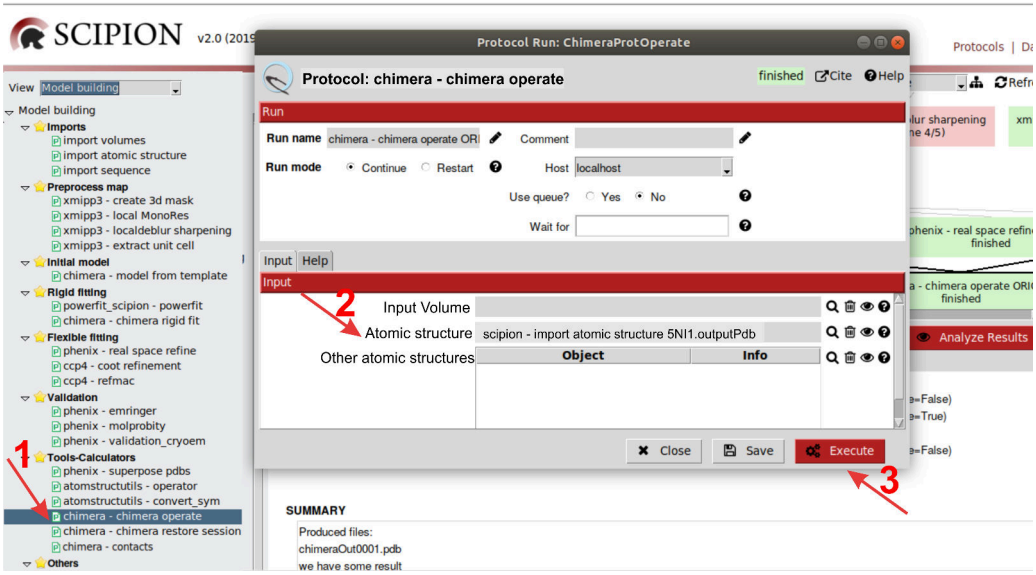


Figure 46: Filling in *Chimera* `operate` protocol form.

The *Chimera* graphics window will be opened with the structure `5NI1` as model number #1. To save independently the structure of human `metHgb` α subunit (chain A), write in *Chimera* command line:

```
split #1
scipionwrite model #1.1
```

Remark that the model saved in *Chimera* 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 B).

- Protocol powerfit:

  Open *PowerFit* protocol and follow the instructions above indicated. The structure saved in *Chimera* operate will replace this time our previous *model* (Fig. 25 (2)). Select `item 2` as best fit.

- Protocol chimera rigid fit: Open again *Chimera* `rigid fit` protocol and, following already indicated instructions, include this time `item 2`, the last fitted structure obtained with *PowerFit* (Fig. 28 (3)). After finishing the rigid fit of the extracted unit cell and `metHgb` α subunit from 5NI1 structure, you can save this fitted structure writing in *Chimera* command line:
  `scipionwrite model #2 refmodel #1 saverefmodel 0`

- Validation protocols phenix - emringer and phenix - validation_cryoem (phenix - molprobity for *PHENIX* version 1.13):

  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 A (**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. 47 (1)), include the published structure of the `metHgb` α subunit as fixed structure (2), each one of the *models* generated along the worflow (3) and execute the protocol (4). Finally, complete the Table 2 with the value of RMSD obtained for each *model*. (Answers in appendix A; **Question 10_2**).

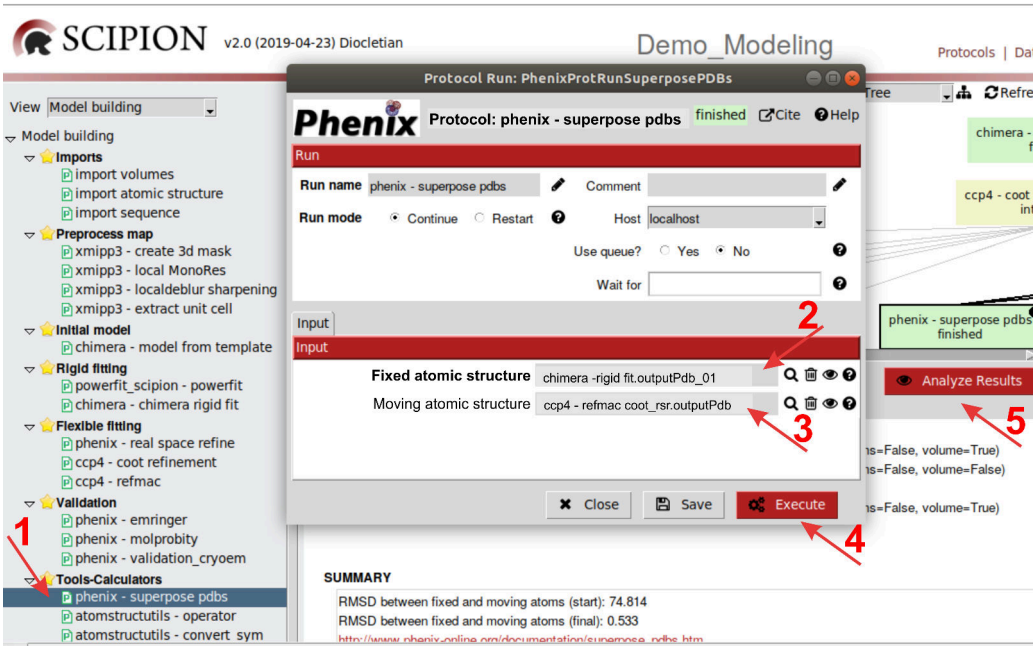


Figure 47: Completing *PHENIX* `superpose pdbs` protocol form.

You can check in *Chimera* the fitted *model* to the published structure by pressing `Analyze results` (Fig. 47 (5)). Arrows of Fig. 48 remark differing parts between both atomic structures. By opening these structures in *Coot* you can see the difference between them.

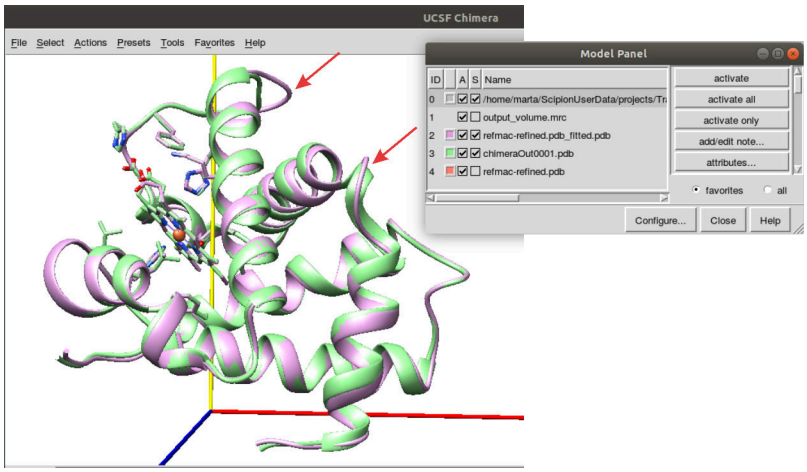


Figure 48: *Model* generated for `metHgb` α subunit superposed to published α chain of `5NI1` structure.

A *model* for `metHgb` α subunit has to be selected at the end of validation process. According to the statistics of Table 6 (Appendix A; **Question 10_2**), *model* obtained in the last step of modeling workflow (*Refmac* after RSRAC (modified)) has been selected due to 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 A **Question 10_3** contains a statistics table for `metHgb` β subunit, similar to that obtained for `metHgb` α subunit.

In the real world 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 unit cell

Once we have selected *models* for `metHgb` α and β subunits, we can regenerate the smallest asymmetrical element (unit cell) of the starting volume. With this aim *Chimera* `rigid fit` and assessment - refinement -assessment protocols will be used.

- Protocol chimera rigid fit to generate the unit cell of human `metHgb`:

  Open again *Chimera* `rigid fit` protocol and following already indicated instructions, include this time *models* for `metHgb` α and β subunits (Fig. 28 (3 and 4)). Firstly, perform the rigid fit of the extracted unit cell (output_volume.mrc) and both subunits #2 and #3, by using `Tools -> Volume Data -> Fit in Map` in *Chimera* (Fig. 49). Next, create a single atomic structure by selecting models #2 and #3 in *Chimera* `Model Panel` and pressing `Copy/-Combine` command in the right column. A new model #4 is shown in *Chimera* `Model Panel`. Finally, save this fitted structure writing in *Chimera* command line:
  `scipionwrite model #4 refmodel #1 saverefmodel 0`

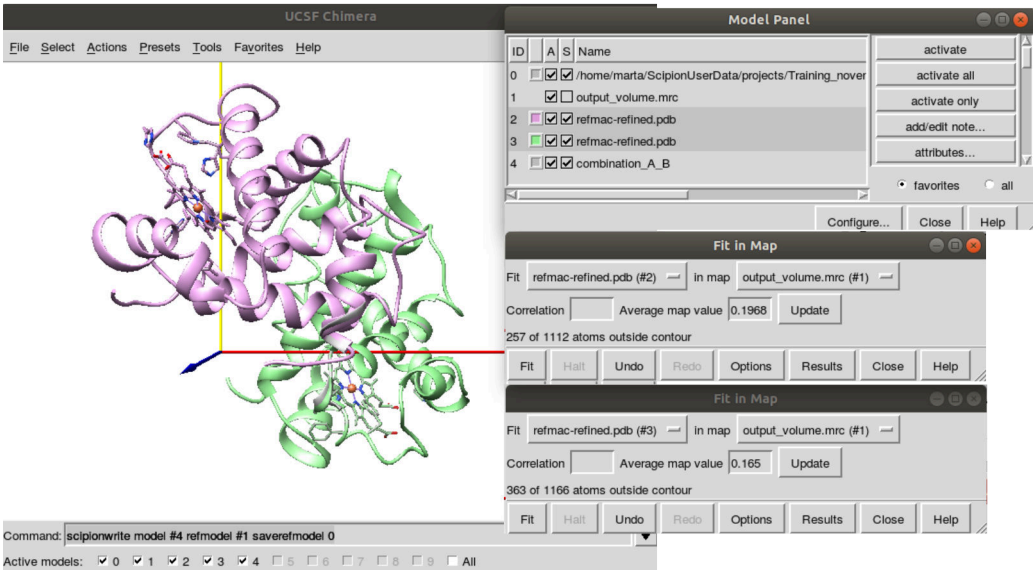


Figure 49: Generation of the human `metHgb` unit cell *model*.

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

  EMRinger and *ValidationCryoEM*(*MolProbity*) validation statistics should be computed for the new *model* of human `metHgb` unit cell, generated by combining `metHgb` α and β subunits. Appendix A (**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* and, additionally, in reciprocal space with *Refmac*, 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. In this tutorial, we have selected the unit cell *model* generated by *Phenix* `real space refine` (modified parameters) 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, *Chimera* `operate` and assessment - refinement -assessment protocols will be used. Starting from the unit cell, *Chimera* `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 accomplish with *Chimera* `operate` protocol to assess volume density occupancy.

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

  Following previous instructions, open *Chimera* `operate` protocol (Fig. 46 (1)), load the selected atomic structure *model* of `metHgb` unit cell (2), and execute the protocol (3). *Chimera* graphics interface will show you the *model* of `metHgb` unit cell. Considering the C2 symmetry of the whole molecule, write in *Chimera* command line to re-generate the whole molecule:

  ```
  sym #2 group C2
  ```

  A symmetric image of the input *model* (Fig. 50; *model* #2) will be generated (*model* #3). *Model* #1 is the initial extracted unit cell volume associated to `metHgb` unit cell *model* loaded. By selecting *models* #2 and #3, and pressing `Copy/Combine` in the right side of `Model Panel`, a new *model* #4 is generated. This new *model* contains the four subunits that integrate the two symmetric unit cells. The whole molecule *model* can be saved by writing in *Chimera* command line:

  ```
  scipionwrite model #4 refmodel #1 saverefmodel 0
  ```

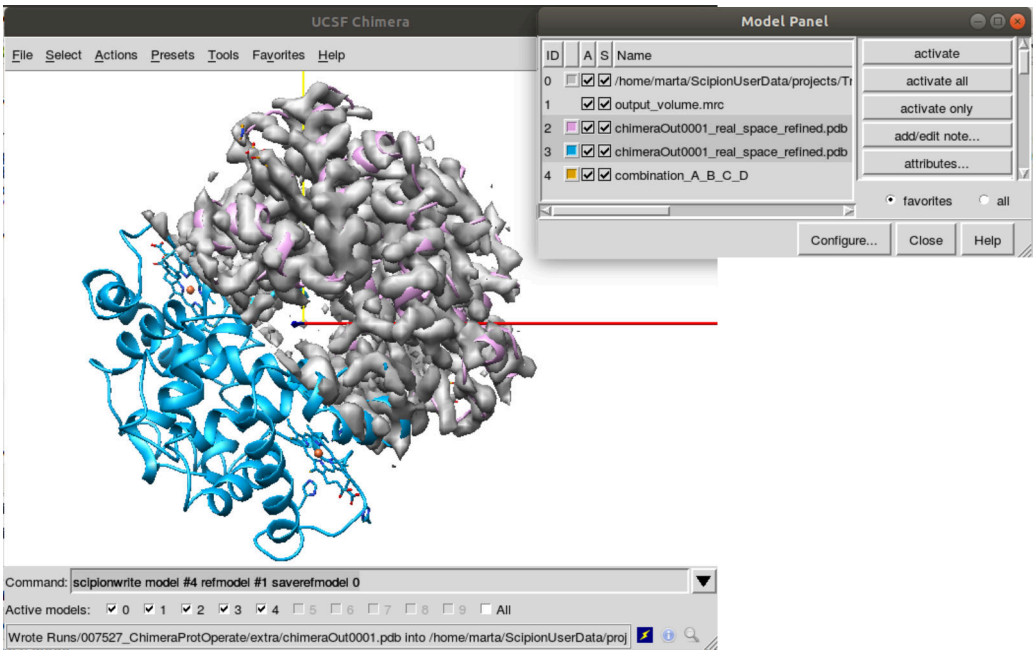


Figure 50: *Model* generated for the whole human `metHgb`.

- 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 *Chimera* `operate` protocol (see results Table 9 in Appendix A; **Question 12_1**). Because of high values of CC_{MASK} and *EMRinger* `score`, as well as acceptable *MolProbity* statistics, *model* generated by *Chimera* `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, RMSD value of the selected atomic structure *model*, regarding the published structure, yields an intermediate value between the best and the worst one.

- Protocol chimera operate to assess volume density occupancy:

As we have seen previously, open *Chimera* `operate` protocol (Fig. 46 (1)), load both the initial volume and the selected atomic structure *model* of the whole human `metHgb` (2), and execute the protocol (3). Fig. 51 shows the initial volume `EMD-3488` (grey) and the selected *model* that we have traced for human `metHgb` (pink):

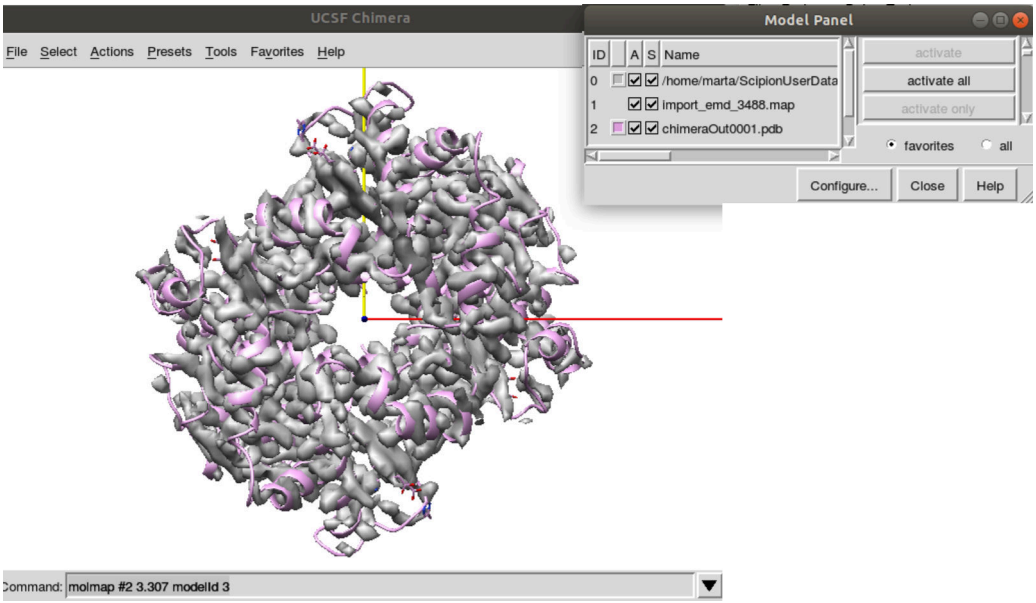


Figure 51: Human `metHgb` *model* opened by *Chimera* `operate` protocol.

To check if the selected model of `metHgb` occupies most part of the starting volume, we have to compare by subtraction the *model*-derived volume and the starting volume `EMD-3488`. To perform this operation, follow the next two steps:

– To generate a volume from the *model* at 3.307Å resolution (resolution shown in *PHENIX-`MolProbity` viewer*; `Real space correlation`; `Atom Mask Radius`), write in *Chimera* command line:

```
molmap #2 3.307 modelId 3
```

Next Fig. 52 shows the volume (*Chimera model* #3) generated in *Chimera* at 3.307Å resolution, starting from the selected atomic structure of human `metHgb` (*Chimera model* #2).

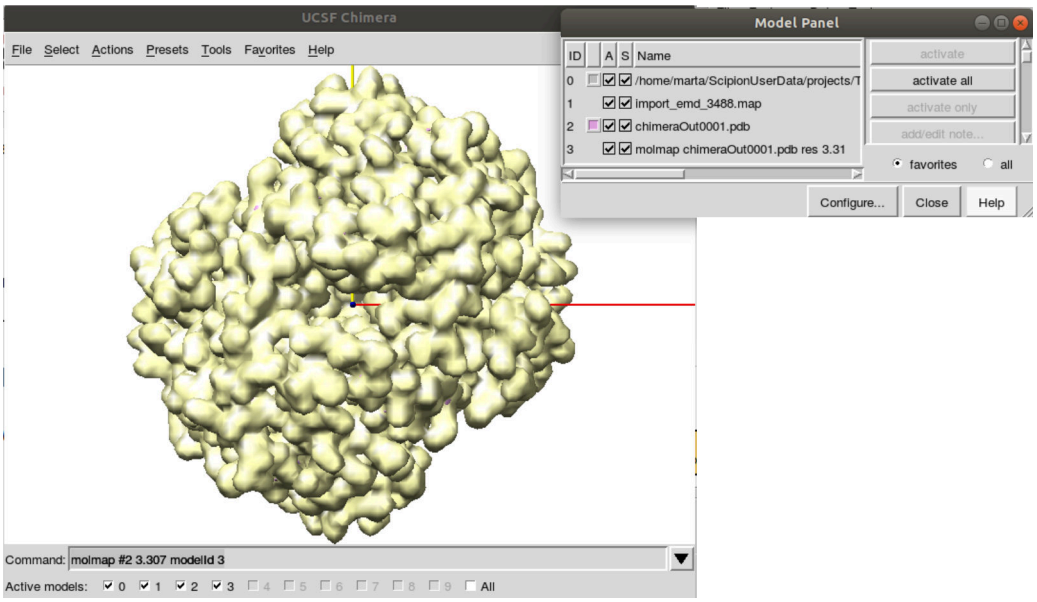


Figure 52: Electron density volume generated from human `metHgb` *model* in *Chimera*.

– To subtract this new model from the starting whole volume `EMD-3488`, write in *Chimera* the command line:

```
vop subtract #1 #3 modelId #4 minRMS true
```

where the minRMS option scales model #3 automatically to minimize the root-mean-square sum of the resulting (subtracted) values at grid points within the lowest contour of model #1.

The resulting volume from this subtraction operation appears in Fig. 53 (*Chimera model* #4). From this result, we can conclude that most part

of the initial density map has been traced, and there are no significant additional densities others than the four monomers and the four prosthetic groups that we have considered so far.

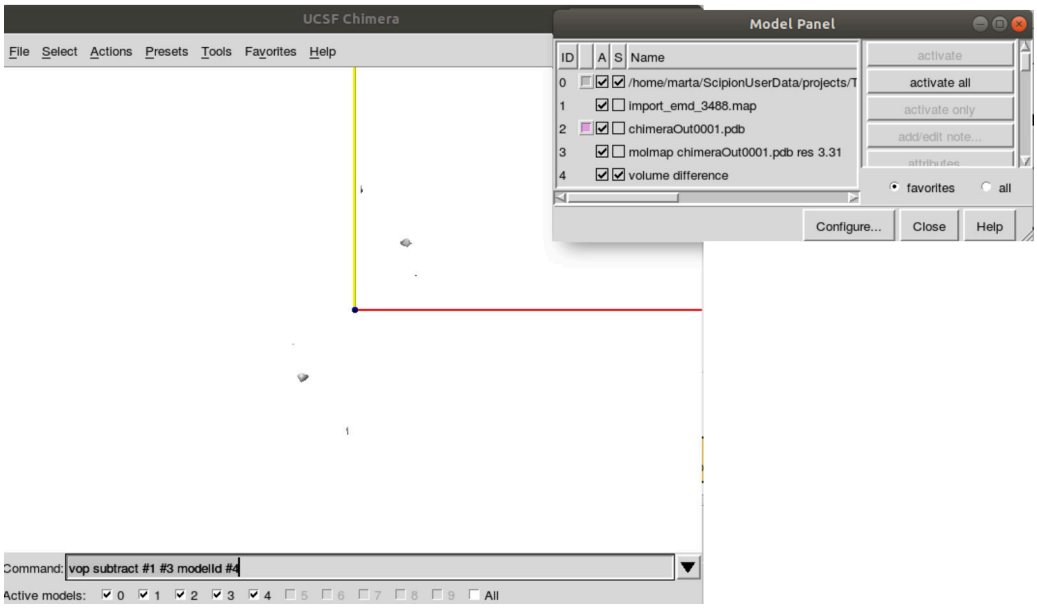


Figure 53: Electron density difference between the starting volume `EMD-3488` and the volume generated from human `metHgb` *model*.

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* and *MolProbity*, 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 W). By using this protocol we can save the basic files that have to be submitted to EMDB in a labelled folder. Basic files are the electron density map, the FSC file and the file of coordinates of the atomic structure. All these files will be saved in an appropriate format for submission.

Open the export to EMDB protocol (Fig. 54 (1)), and complete the form with the *Scipion* elements to export: `Volume` (2), `FSC` (3), `Atomic structure` (4) and `Mask` (5). All submission files will be saved in the `directory` selected (6). A directory name related with the submission (number, date, project,...) is recommended.

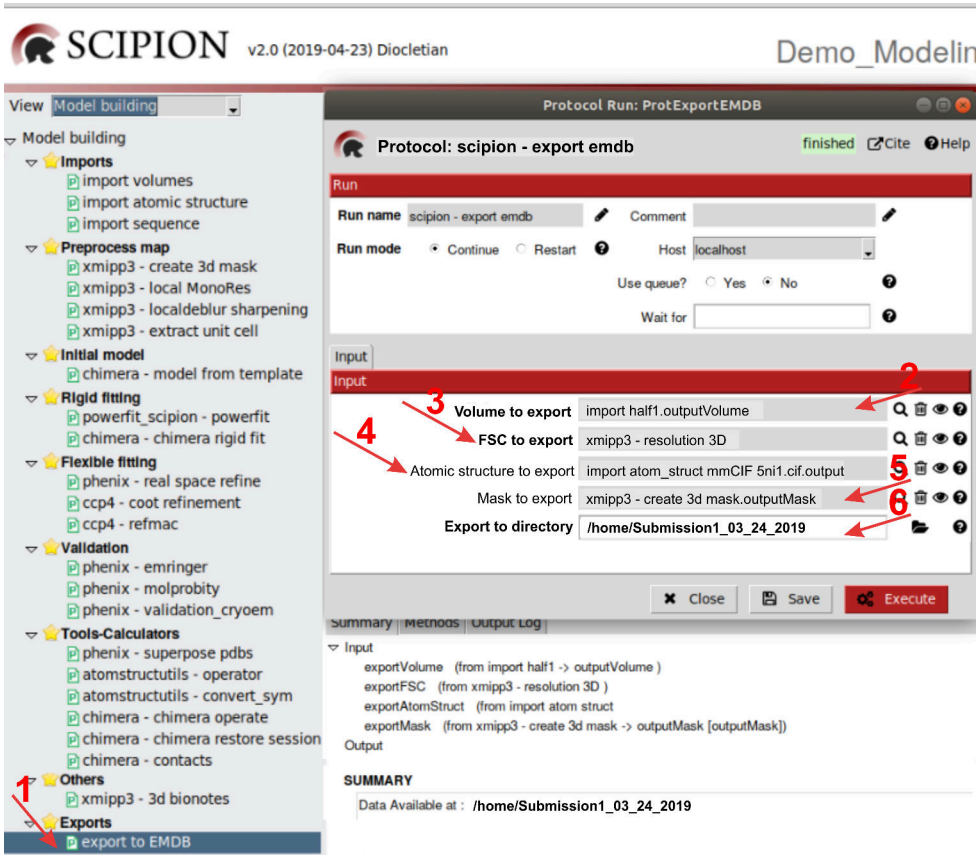


Figure 54: Saving files for submission to EMDB with protocol export to EMDB

After executing the protocol, 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 chimera 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 C.

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. According to the aim of the analysis, two possible scenarios and respective workflows can be considered to compute contacts, a) infering all contacts between any couple of members of the unit cell and between one member of the unit cell and another component from a neighbor unit cell (Fig. 55 (A; 2)), and b) infering all contacts between any couple of members of the whole macromolecule (Fig. 55 (B; 4)).

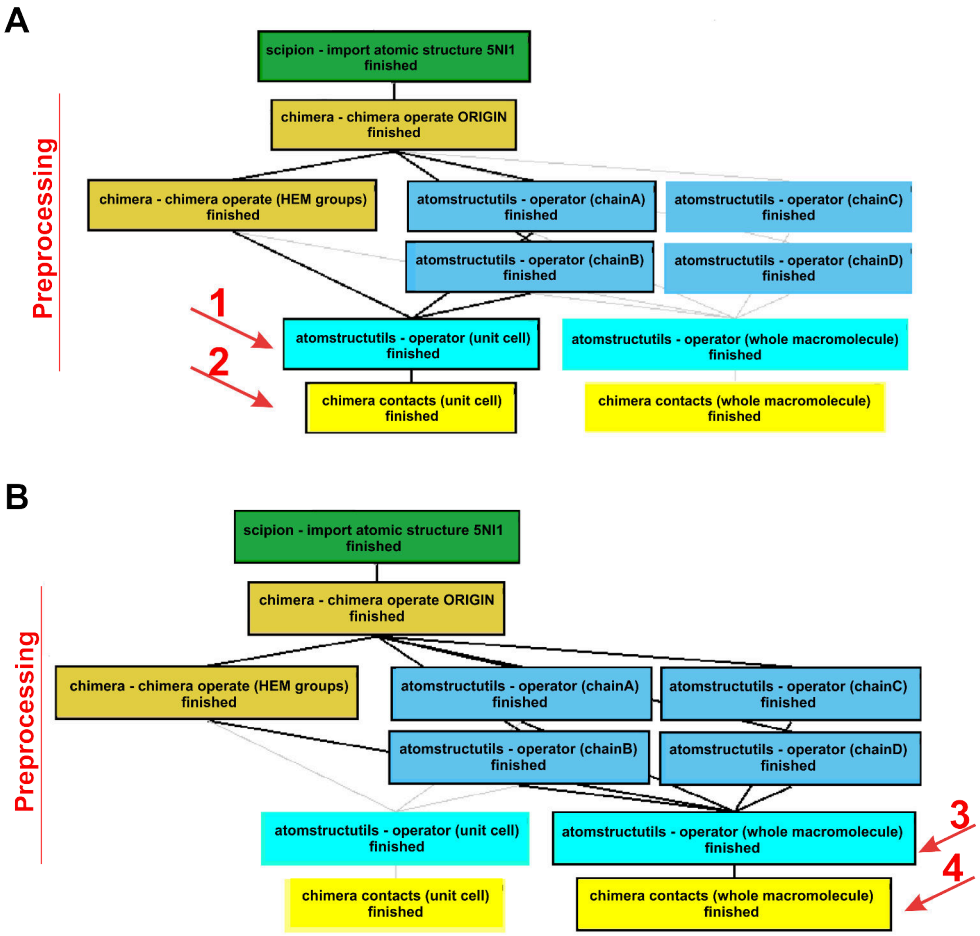


Figure 55: *Scipion* workflows to get contacts between any two chains of the unit cell and between any chain of the unit cell and a chain of a neighbor unit cell (A, bold lines), and to get contacts between any two chains of a macromolecule (B, bold lines).

Since the last step of A workflow (Fig. 55 (2)) requires applying symmetry, the first step involves to move the structure to match its symmetry center to the origin of coordinates. Besides this preprocesing step, and because protocol chimera contacts only computes contacts between independent chains and not whitin the same chain, an additional preprocessing step has to be performed to separate `HEM` groups in

independent chains to get contacts between proteins and ligands (HEM groups).

- Preprocessing: In this step, the macromolecule constituted by four chains, each one containing a HEM group, will be transformed in a molecule of eight independent chains, four proteins and four HEM groups, with its symmetry center in the origin of coordinates. Two protocols already mentioned before are going to be used to move the structure and to extract proteins and ligands of the starting atomic structure, atomstructutils - operator protocol (Appendix B), and the protocol chimera operate (Appendix D).

  - Matching between symmetry center of human haemoglobin metHgb atomic structure 5NI1 and origin of coordinates (upper brown box in A and B workflows of Fig. 55):
    Open the protocol chimera operate and complete the form with 5NI1 as Atomic structure and the structure of the Hgb obtained by phenix - real space refine protocol as Other atomic structures. Follow the next intructions to fit the atomic structure 5NI1 to the refined Hgb α subunit when *Chimera* graphics window opens:

    * *Chimera* main menu: Tools -> Structure Comparison -> MatchMaker

    * MatchMaker window:
      · Reference structure: refined Hgb α subunit
      · Structure(s) to match: atomic structure 5NI1
      · Chain pairing: Specific chain(s) in reference structure with specific chain(s) in match structure
        Select chain A both in Reference structure and Structure(s) to match
      · Press OK.

    * Save the unit cell in the new location:
      Chimera command line:
      scipionwrite model #1 refmodel #2 saverefmodel 0

To visualize the new position of the atomic structure `5NI1`, open the *Chimera* viewer by clicking Analyze Results.

– Independent extraction of aminoacid chains A, B, C and D (dark blue boxes in workflows A and B of Fig. 55):
Open the protocol atomstructutils - operator (Fig. 56 (1)) and fill in the form with the atomic structure (2) and the operation to accomplish, in this case chain extraction (3), to extract chain A (4), that can be selected with the help of the wizard on the right. The default values of starting (5) and ending (6) residues allow to extract the whole chain. Then, execute the protocol. The extracted A chain can be visualized with *Chimera* by clicking Analyze Results. Repeat this process to extract, one by one, chains B, C and D.

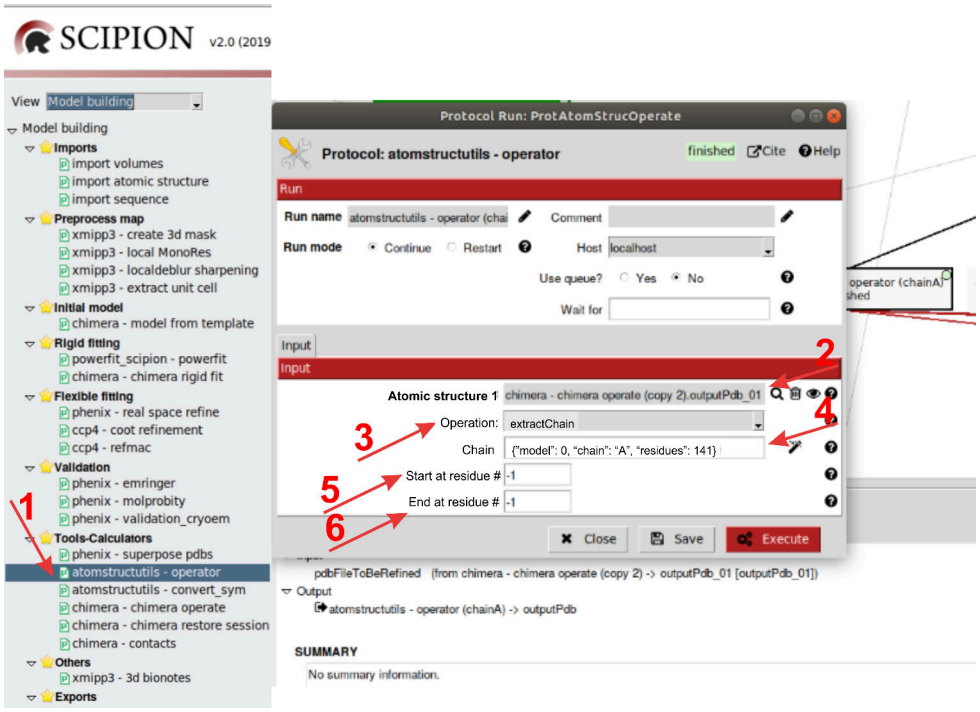


Figure 56: Extraction of chain A of human haemoglobin `metHgb` with protocol atomstructutils - operator.

– Independent extraction of `HEM` groups associated to each aminoacid chain from human haemoglobin `metHgb` atomic structure `5NI1` (lower brown box in workflows A and B of Fig. 55): Open the protocol chimera operate and complete the form with the atomic structure `5NI1` and follow these intructions when *Chimera* graphics window opens:

  * Selection of `HEM` groups:
    Chimera main menu: `Select -> Residue -> HEM`
  * Selection of every element except `HEM` groups:
    Chimera main menu: `Select -> Invert (selected models)`
  * Remove every element except `HEM` groups:
    Chimera command line: `delete sel`
  * Split `HEM` groups in four independent chains:
    Chimera command line: `split #1`
  * Save `HEM` groups one by one as models 1.1 to 1.4:
    Chimera command line: `scipionwrite model #1.1`
    Chimera command line: `scipionwrite model #1.2`
    Chimera command line: `scipionwrite model #1.3`
    Chimera command line: `scipionwrite model #1.4`

  Remark that, although chimera operate protocol might also be used to extract independently aminoacid chains A, B, C and D, we have chosen the protocol atomstructutils - operator to exclude water molecules associated to aminoacid chains.

– Reconstruction of the unit cell of the human haemoglobin `metHgb` atomic structure (Fig. 55 (A; 1)): Protocol atomstructutils - operator will be used to perform this task by selecting, in this case, `addChain` as operation option (Fig. 57 (A; 2)).

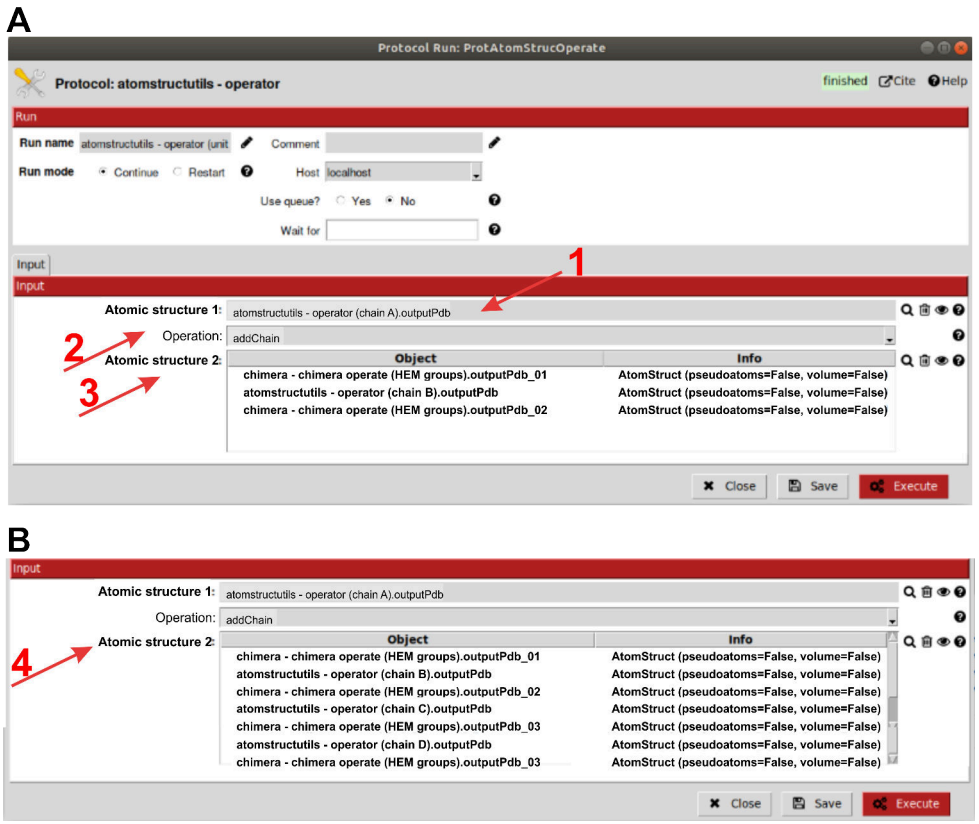


Figure 57: Completing the protocol form to reconstruct the unit cell (A) and the whole macromolecule (B) of the human haemoglobin `metHgb` with protocol atomstructutils - operator.

To the previously extracted chain A (Fig. 57 (A; 1)) we add (2) its respective `HEM` group, as well as chain B and its respective `HEM` group (3). To visualize the reconstructed unit cell, open the *Chimera* viewer by clicking Analyze Results. Remark that the symmetry center is equal to the origin of coordinates and this allows to regenerate the whole molecule by applying symmetry (see Appendix C for a deeper explanation).

– Reconstruction of the whole macromolecule of the human haemoglobin `metHgb` atomic structure (Fig. 55 (B; 3)): Analogously to the unit cell, protocol atomstructutils - operator will be used to build the whole macro-

molecule. In this case, besides adding to chain A (Fig. 57 (B; 4)) the same elements that we added to get the unit cell, we have added chains C and D and their respective `HEM` groups.

Again, the reconstructed whole macromolecule can also be visualized with *Chimera* by clicking Analyze Results. The symmetry center of the macromolecule is set in the center of coordinates. Unlike with the unit cell, no symmetry will be applied to get contacts among chains. The location of the macromolecule is thus irrelevant to analyze the contacts.

- CASE A: Contacts between any couple of members of the unit cell and between one member of the unit cell and another one from a neighbor unit cell (Fig. 55 (A; 2)):
  This option allows to get all contacts between all couples of members of the unit cell and all "non-redundant" interactions between a chain of the unit cell and a chain of a neighbor unit cell by using the protocol chimera contacts. "non-redundant" interaction means any interaction that can not be inferred by symmetry. Fig. 58 (A) shows the total number of interactions of our example including "redundant" interactions. The 33 interactions between the chain `B` of the unit cell (model `#0`) and the chain `A` of the neighbor unit cell (model `#1`) are symmetric to the interactions between chain `A` of the unit cell (model `#0`) and chain `B` of the neighbor unit cell (model `#1`). Since those interactions can thus be inferred by symmetry, they are "redundant" and will be absent of the final list of contacts.

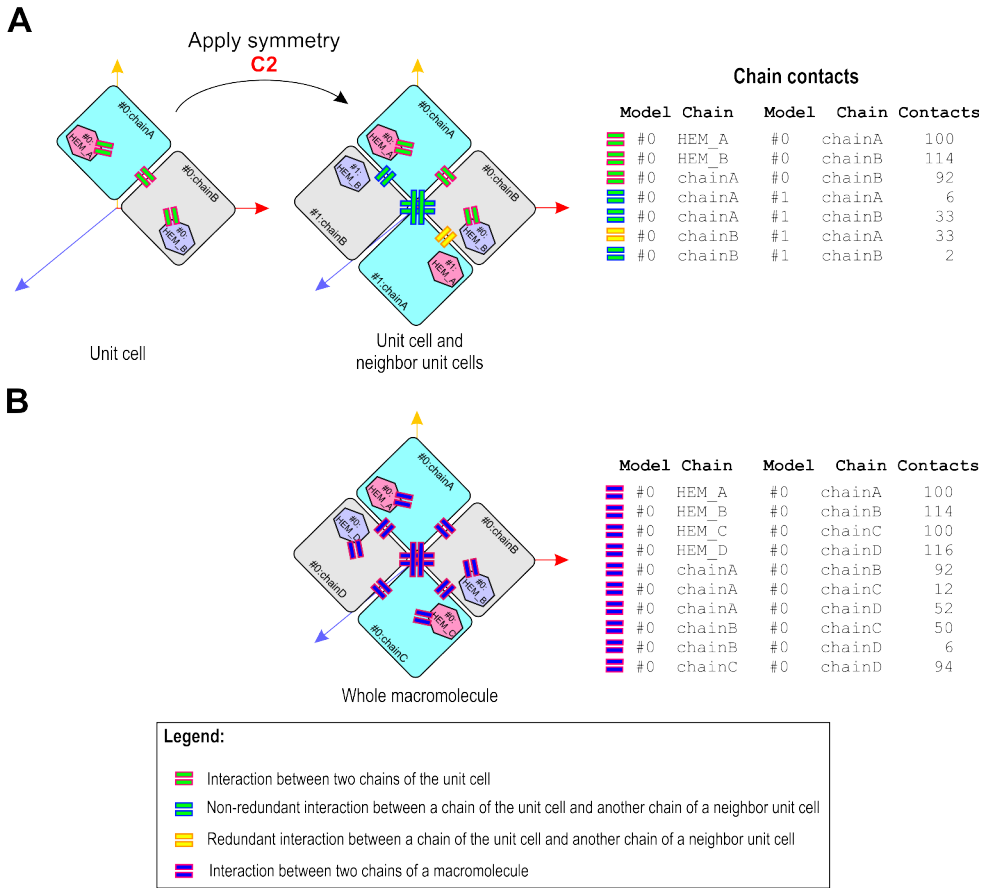


Figure 58: Schema of the human haemoglobin `metHgb` showing protein contacts obtained by applying symmetry to the unit cell (A), and contacts between couples of chains of the whole macromolecule (B).

This list of protein contacts was obtained in *Scipion* by running the protocol chimera contacts. The protocol form opened (Fig. 59 (1)) is completed with the input atomic structure of the human haemoglobin `metHgb` unit cell (2) and labelled each structure chain (3). With this labelling, a specific label is assigned to each chain in order to group chains with the same label (see Appendix C for details). Since in this particular case we do not want to group chains, independent labels will be assigned to each chain ((Fig. 59 (4)) with the help

of the wizard (3). To apply symmetry to the input unit cell, we select `Yes` to the Apply Symmetry option (5). A panel with the different types of symmetry will be displayed to chose our specific Symmetry, cyclic (Cn) in our case, and our Symmetry Order (2).

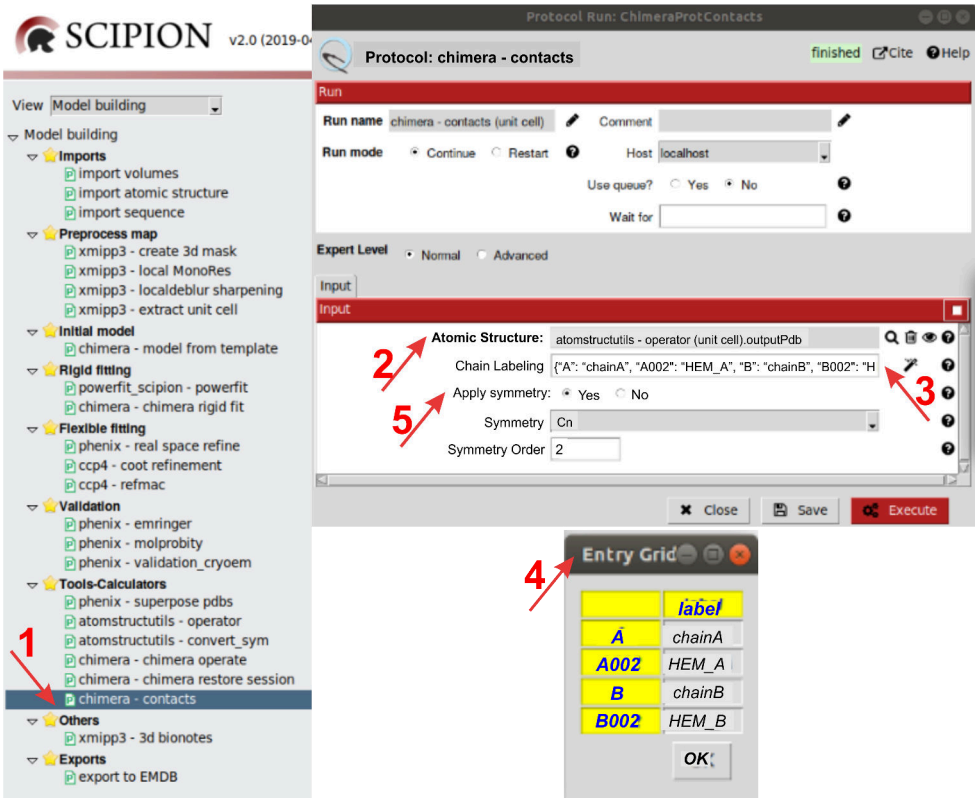


Figure 59: Filling in the chimera contacts protocol form to get atom contacts between couples of chains within the unit cell, and "non-redundant" contacts between a chain of the unit cell and another chain from a neighbor unit cell of the human haemoglobin `metHgb`.

After executing the protocol, non-redundant atom contacts between the couples of proteins indicated in Fig. 58 (A) can be visualized by clicking Analyze Results (Fig. 60 (A)).

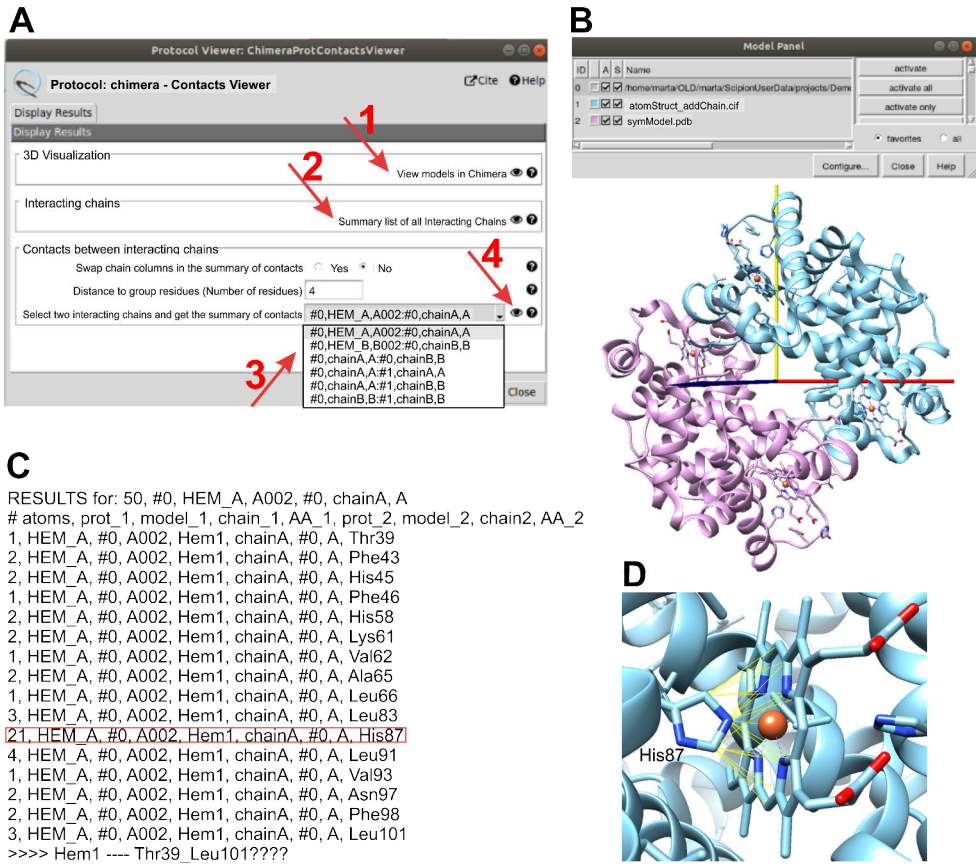


Figure 60: (A) Display of results of atom contacts between couples of chains within the unit cell, and "non-redundant" contacts between a chain of the unit cell and another chain from a neighbor unit cell, of the human haemoglobin `metHgb`. (B) *Chimera* view of input (blue) and symmetrical (pink) unit cells. (C) Summary of atom contacts between residues of input unit cell chain A and its respective `HEM` group. (D) Detail of the 21 atom contacts between the residue `His87` of chain A and its respective `HEM` group (yellow lines).

Models can be visualized with *Chimera* (Fig. 60 (A; 1)). (B) shows the model of the initial unit cell and the new unit cell, generated by symmetry regarding

the origin of coordinates. This structure, combining both models, serves as starting point to compute atom contacts. The list that contains the six couples of interacting chains can be displayed in Fig. 60 (A; 2), and each one of these interactions can be selected in Fig. 60 (A; 3). A text list details all atom contacts between the residues of chain A and its respective `HEM` group (A; 4). The higher number of atom contacts (21) involves the `His87` residue (framed in red). A detail of this interaction is shown in Fig. 60 (D). Analogously, the rest of atom contacts can be observed by selecting each one of the five remaining couples of interacting proteins.

- CASE B: Contacts between any couple of members of the whole macromolecule (Fig. 55 (B; 4)):
  This option allows to get all contacts between all couples of members of the macromolecule (Fig. 58 (B)) by using the protocol chimera contacts. Similarly to case A, the protocol form has to be completed with the Atomic Structure (Fig. 61 (1)), and the Chain Labelling (3) with help of the wizard (2). Unlike in case A, the option `No` in Apply symmetry has to be selected in this case (4).

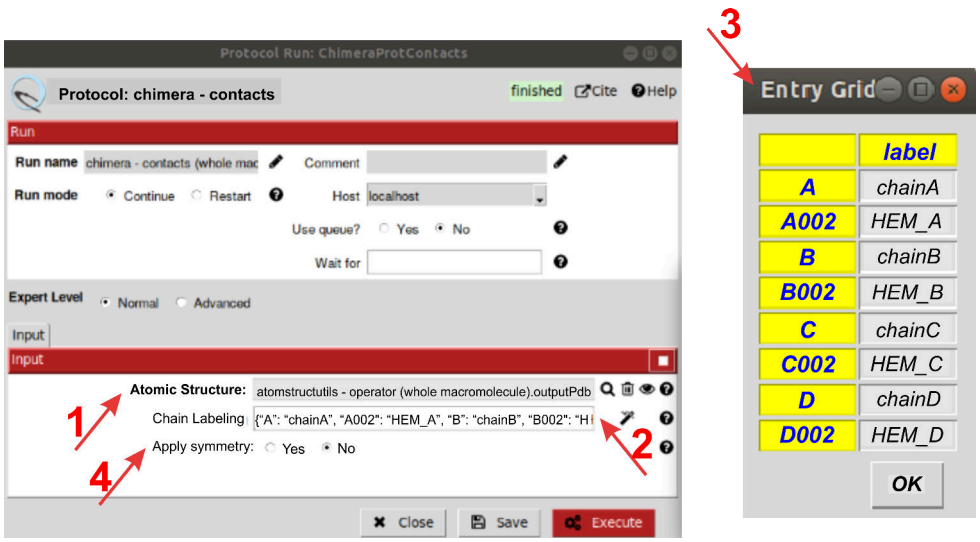


Figure 61: Filling in the chimera contacts protocol form to get all atom contacts between couples of chains within the macromolecule of the human haemoglobin `metHgb`.

After executing the protocol, resulting contacts can be checked similarly to case A.

14 A Note on Software Installation

All the protocols shown in this document are available in the stable *Scipion* release `2.0.0` (code name *Diocletian*). 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): 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 and 1.16-3549.
- **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 TODO

List of protocols in the process to be incorporated:

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

- **buccaneer:** (ccp4) *de novo* model building.
- **chimera** *de novo* model building; it does not use the 3D map.

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 Pagothenia 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.

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.

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

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.

Pettersen, E. F., Goddard, T. D., Huang, C. C., Couch, G. S., Greenblatt, D. M., Meng, E. C., Ferrin, T. E., Oct 2004. UCSF Chimera–a visualization system for exploratory research and analysis. J Comput Chem 25 (13), 1605–1612.

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.

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

Van Zundert, G. C. P., Bonvin, A. M. J. J., 08 2016. Defining the limits and reliability of rigid-body fitting in cryo-EM maps using multi-scale image pyramids. J. Struct. Biol. 195 (2), 252–258.

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

A 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:**

Table 3: *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_5:**

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

Table 4: *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 seems to be higher because the starting position was worse. Anyway, *Refmac* seems to partially compensate the effect of the additional refinement in the real space accomplished with *Phenix*.

- **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 3) and after *Coot* (compare with Table 4). 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 *PowerFit item2*: -0.57; This value makes sense to the highest score of *item*1 model in *PowerFit*.
  EMRinger `score` after *Chimera* `rigid fit` *item*2: 3.86; Things clearly change after *Chimera* `rigid fit`.

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` (default) after *Coot*: 5.38
EMRinger `score` after *Phenix* `real space refine` (last modification of form parameters) after *Coot*: 5.23
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` (last modification of form parameters) 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

- **Question 10_3:**

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

  Table 7

- **Question 11_1:**

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

Statistic	*Powerfit* *item* #2	*Chimera* *model* #2	*Coot*	*Phenix* `RSRAC` (default)	*Phenix* `RSRAC` (modified)	*Refmac* after *Coot*	*Refmac* after `RSRAC` (modified)	`5NI1`
CC_{MASK}	0.194	0.569	0.725	0.787	0.805	0.803	0.801	0.843
EMRinger `score`	-0.57	3.86	2.37	5.38	5.23	2.87	5.34	3.98
`RMS` (Bonds)	0.0187	0.0188	0.0183	0.0090	0.0066	0.020	0.0191	0.0126
`RMS` (Angles)	2.41	2.41	2.02	1.30	1.16	1.94	1.84	1.43
`Rama favored` (%)	97.14	97.14	97.12	95.68	95.68	97.12	95.68	94.24
`Rama allowed` (%)	2.15	2.15	2.88	4.32	4.32	2.88	4.32	5.76
`Rama outliers` (%)	0.71	0.71	0.00	0.00	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	1.75	1.75	24.78	0.00	0.00	23.01	1.77	0.88
`Clashscore`	71.68	70.34	26.24	1.81	1.36	21.73	1.36	2.26
`Overall score`	2.67	2.66	3.12	1.24	1.16	3.02	1.35	1.39
`C`β `deviations`	1	1	7	0	0	2	0	0
`RMSD`	0.841	0.841	0.447	0.456	0.407	0.414	0.384	0.0

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

Statistic	*Powerfit* *item* #2	*Chimera* *model* #2	*Coot*	*Phenix* `RSRAC` (default)	*Phenix* `RSRAC` (modified)	*Refmac* after *Coot*	*Refmac* after `RSRAC` (modified)	`5NI1`
CC_{MASK}	0.340	0.524	0.690	0.776	0.793	0.765	0.767	0.830
EMRinger `score`	1.41	1.13	3.93	4.76	4.94	3.70	5.32	4.87
`RMS` (Bonds)	0.0313	0.0313	0.0169	0.0078	0.0098	0.0191	0.0183	0.0117
`RMS` (Angles)	2.17	2.17	1.97	1.33	1.39	1.95	1.87	1.40
`Rama favored` (%)	96.55	96.55	97.92	96.53	96.53	97.22	95.14	95.83
`Rama allowed` (%)	3.45	3.45	2.08	3.47	3.47	2.78	4.86	4.17
`Rama outliers` (%)	0.0	0.0	0.00	0.00	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	1.68	1.68	29.66	0.85	0.85	27.97	5.93	0.00
`Clashscore`	75.50	75.93	34.24	3.89	3.03	25.57	2.16	4.32
`Overall score`	2.74	2.75	3.16	1.40	1.32	3.14	1.92	1.50
`C`β `deviations`	0	0	8	0	0	1	0	0
`RMSD`	0.935	0.935	0.495	0.470	0.535	0.441	0.494	0.0

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

Statistic	*Chimera* `rigid fit`	*Phenix* `RSR` (default)	*Phenix* `RSR` (modified)	*Refmac* after `RSR` (modified)	`5NI1`
CC_{MASK}	0.787	0.808	0.807	0.789	0.840
EMRinger `score`	4.64	4.58	4.91	4.35	4.11
`RMS` (Bonds)	0.0187	0.0093	0.0093	0.0182	0.0122
`RMS` (Angles)	1.860	1.380	1.390	1.840	1.410
`Rama favored` (%)	95.41	95.41	95.41	94.70	95.05
`Rama allowed` (%)	4.59	4.59	4.59	5.30	4.95
`Rama outliers` (%)	0.0	0.0	0.00	0.00	0.00
`Rotamer outliers` (%)	3.90	0.43	0.00	3.90	0.43
`Clashscore`	5.31	3.54	2.87	3.32	3.53
`Overall score`	2.05	1.46	1.40	1.94	1.49
`C`β `deviations`	0	0	0	0	0
`RMSD`	0.494	0.509	0.579	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	*Chimera* `operate`	*Phenix* `RSR` (default)	*Phenix* `RSR` (modified)	*Refmac* after `RSR` (modified)	`5NI1`
CC_{MASK}	0.810	0.803	0.796	0.792	0.842
EMRinger `score`	4.95	4.70	3.86	4.05	4.18
`RMS` (Bonds)	0.0093	0.0076	0.0067	0.0181	0.0122
`RMS` (Angles)	1.390	1.350	1.350	1.860	1.410
`Rama favored` (%)	95.41	95.41	96.82	95.41	95.23
`Rama allowed` (%)	4.59	4.59	3.18	4.59	4.77
`Rama outliers` (%)	0.00	0.00	0.00	0.00	0.00
`Rotamer outliers` (%)	0.00	0.00	1.30	5.41	0.43
`Clashscore`	4.97	3.21	3.98	2.54	3.53
`Overall score`	1.58	1.43	1.46	1.94	1.48
`C`β `deviations`	0	0	0	0	0
`RMSD`	0.579	0.642	0.517	0.454	0.00

B 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. 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-atomstructutils`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Tools-Calculators` (Fig. 62 (A))

- Protocol form parameters (Fig. 62 (B and C)):

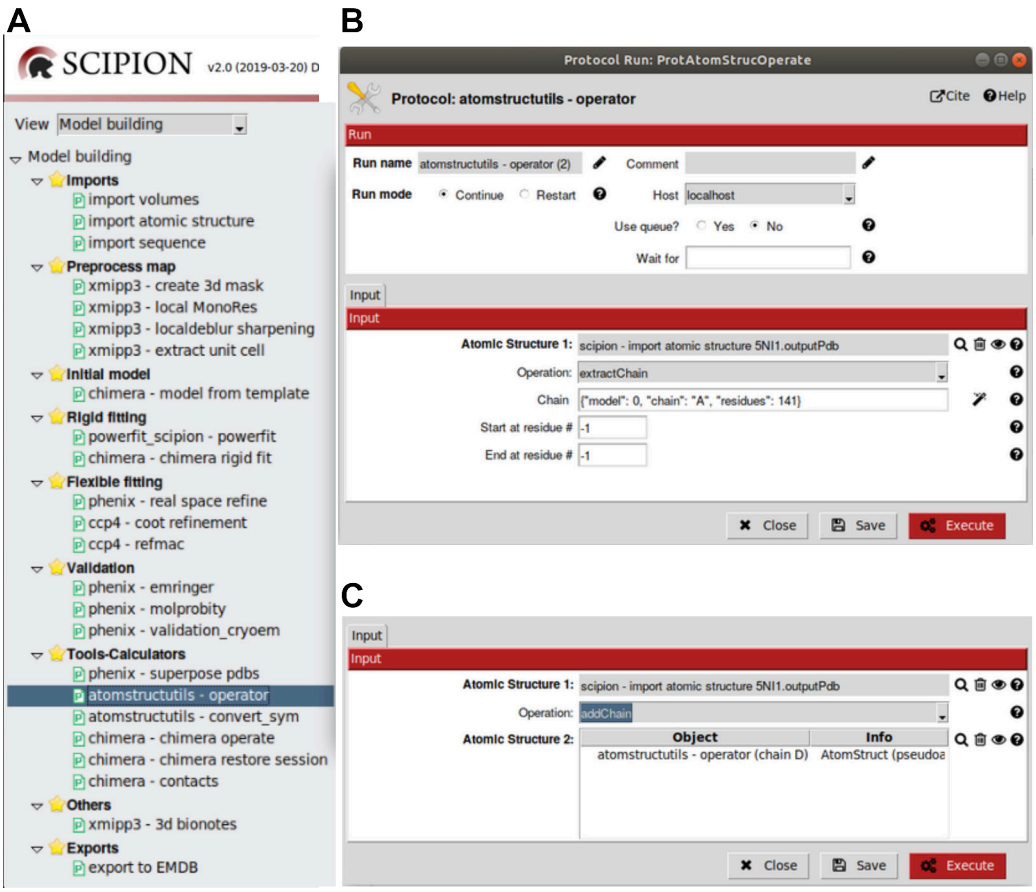


Figure 62: Protocol atomstructutils - operator. A: Protocol location in *Scipion* menu. B: Protocol form to extract a chain from an atomic structure. C: Protocol form to add one or several chains to an atomic structure.

  - `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. 62 (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. 62 (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 *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes and the new atomic structure generated are model numbers `#0` and `#1`, respectively, in *Chimera* `Model Panel`. Write in *Chimera* command line:
`split #1`
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.

C Chimera 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 *Chimera*.

- Requirements to run this protocol and visualize results:

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

- *Scipion* menu: `Model building -> Tools-Calculators` (Fig. 63 (A))

- Protocol form parameters (Fig. 63 (B)):

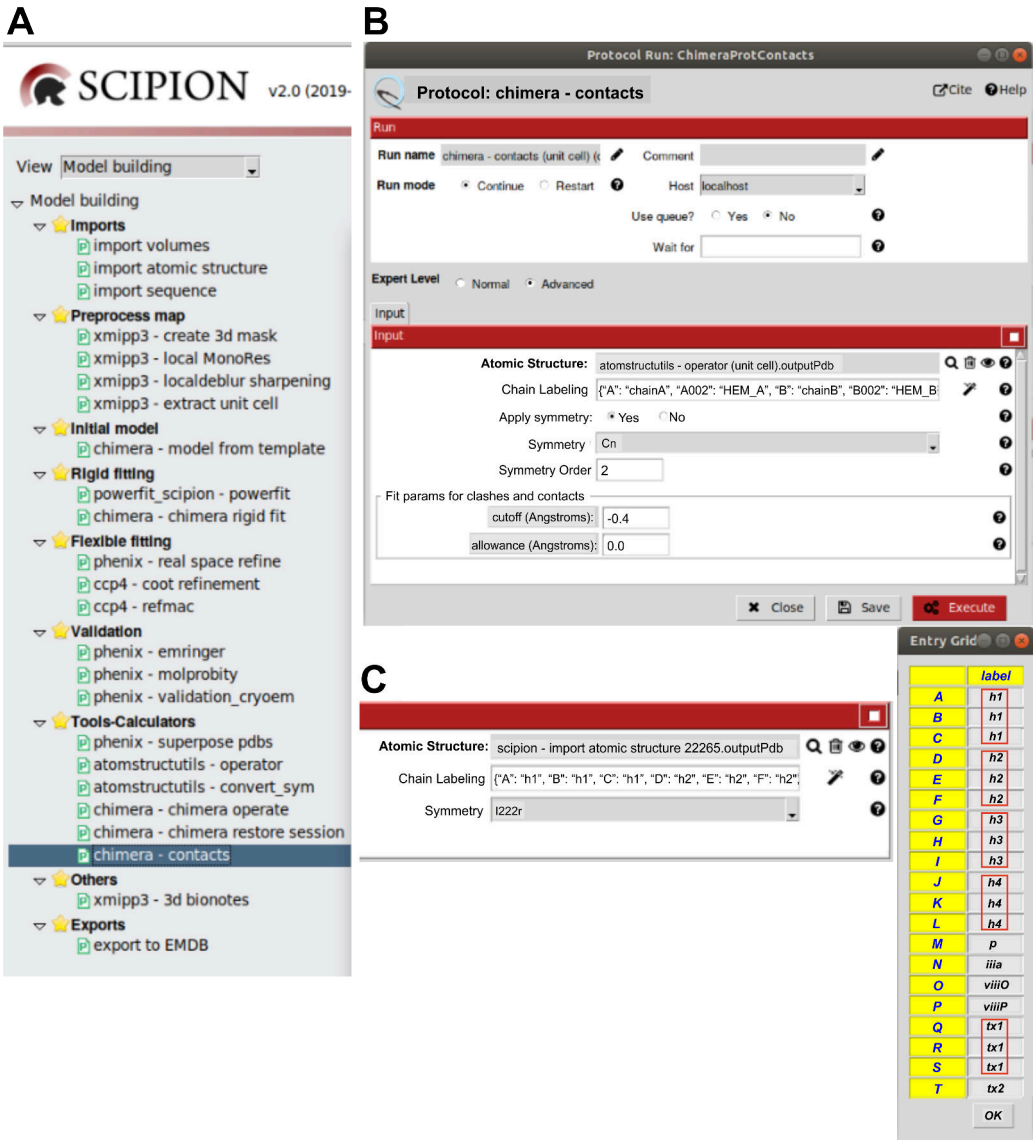


Figure 63: Protocol chimera contacts. A: Protocol location in *Scipion* menu. B: Protocol form. C: Protocol form detailing `Chain Labelling` for `I222r` symmetry.

– `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. 63 (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 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 unit cell of a macromolecule and you'd like to know the contacts between any two chains within the unit cell and the contacts between any chain of the unit cell and a chain from a neighbor unit cell. Consider, in this case, that only neigbor 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 unit cell, in order to generate neighbor unit cells able to interact with the input unit cell.
  - 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 macromolecule 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 *Chimera* orientation

(https://www.cgl.ucsf.edu/chimera/current/docs/UsersGuide/midas/sym.html):

  * I222: *Chimera* orientation 222; two-fold symmetry axes along the X, Y, and Z axes.
  * I222r: *Chimera* orientation 222r; *Chimera* orientation 222 rotated 90°about Z.
  * In25: *Chimera* orientation n25; two-fold symmetry along Y and 5-fold along Z.
  * In25r: *Chimera* orientation n25r; *Chimera* orientation n25 rotated 180°about X.
  * I2n3: *Chimera* orientation 2n3; two-fold symmetry along X and 3-fold along Z.
  * I2n3r: *Chimera* orientation 2n3r; *Chimera* orientation 2n3 rotated 180°about Y.
  * I2n5: *Chimera* orientation 2n5; two-fold symmetry along X and 5-fold along Z.
  * I2n5r: *Chimera* orientation 2n5r; *Chimera* 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 *Chimera* orientation have been included (https://www.cgl.ucsf.edu/chimera/current/docs/UsersGuide/midas/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/chimera/current/docs/UsersGuide/midas/sym.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, chimera contacts viewer window will be opened. This window includes three boxes (Fig. 60 (A)):
  - `3D Visualization`: *Chimera* 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. 58 (A). In addition, input atomic structure is model `#0`, whereas models generated by symmetry will be `#1`, if only one is generated, and `#1.1, #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. 58 (B)). There is only one model in this case, model `#0`.

– `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.

D Chimera Operate protocol

Protocol designed to perform operations with atomic structures in *Scipion* by using *Chimera*. 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 *Chimera* commands. *Chimera* `rigid fit` protocol constitutes a particular case of this protocol to perform rigid fitting in *Scipion* by using *Chimera* (Appendix F).

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Tools-Calculators` (Fig. 64 (A))
- Protocol form parameters (Fig. 64 (B)):

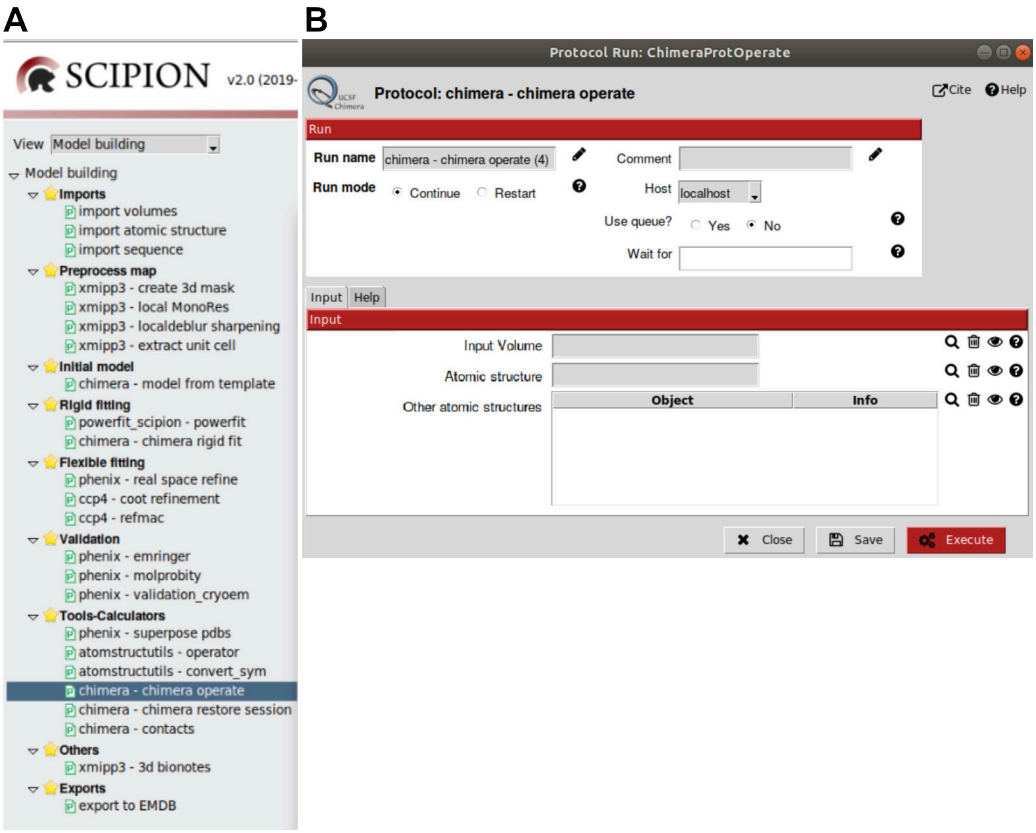


Figure 64: Protocol chimera operate. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Input Volume`: Optional parameter to be completed with the electron density map previously downloaded or generated in *Scipion*.
  - `Atomic structure`: Atomic structure previously downloaded or generated in *Scipion*.
  - `Other atomic structures`: Additional atomic structures.
- `Help` section

  This section contains *Chimera* commands required to save *models* according to their reference volumes, which can also be saved if required. Remark that using `scipionwrite` command, *Chimera* session will be saved by de-

fault, without prejudice that it may be saved with `scipionss` command. *Chimera* sessions can be restored by using chimera 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.

Chimera graphics window will be opened after executing the protocol. Electron density map, 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, and they have to be saved in *Scipion*.

- To save an atomic structure generated with this *Chimera* protocol: Write in *Chimera* command line:
  `scipionwrite model #n`
  If you want to save the model regarding any electron density map, write in *Chimera* command line:
  `scipionwrite model #n refmodel #n saverefmodel 0/1`.
  Replace `#n` by model numbers shown in *Chimera* `Model Panel`. After `saverefmodel`, select `1` if you want to save the reference map or `0` otherwise, to avoid duplicate data unnecessarily.
- To save a volume generated with this *Chimera* protocol: New volume data sets have to be saved first in a separate volume file. Otherwise, those volumes will not be saved in the *Chimera* session since session files only record file paths to volumes. Assuming that the name chosen for the new volume generated is `volume_name.mrc`, and `abspath` the absolute path in

which you want to save the volume, write in *Chimera* command line:
`volume #n save abspath/volume_name.mrc`
`scipionwrite model #n refmodel #n saverefmodel 1`.
Replace `#n` by model numbers shown in *Chimera* `Model Panel`. Remark that you have to save an atomic structure if you want to save a volume in *Scipion*.

- Close *Chimera* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, volume, and atomic structure are model numbers `#0`, `#1`, and `#2`, respectively, in *Chimera* `Model Panel`. If no volumes have been included, coordinate axes and each atomic structure are model numbers `#0` and `#1`, respectively.

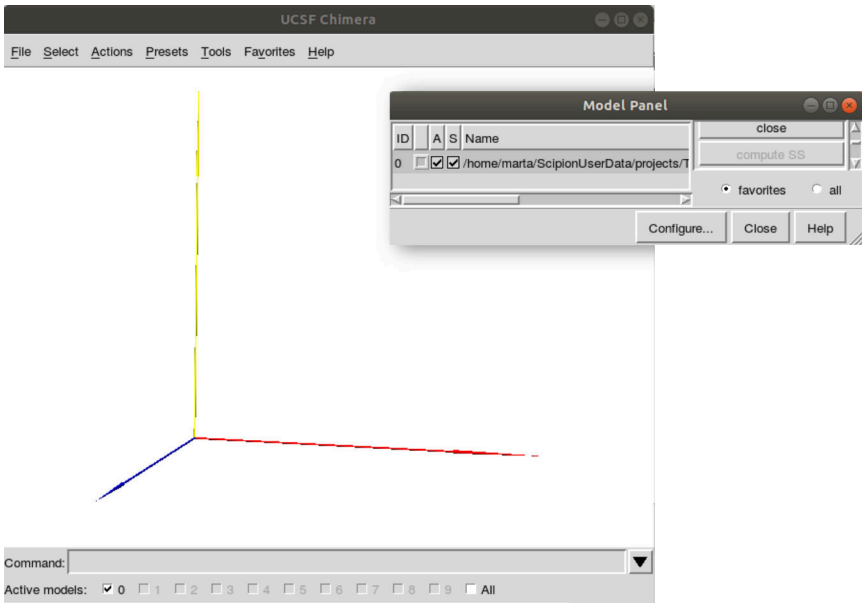


Figure 65: Default *Chimera* graphics window with coordinate axes.

- Summary content:

  - If an atomic structure is generated:

    * Protocol output (below *Scipion* framework):
      `chimera - chimera operate -> ouputPdb_01`; `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:
      chimeraOut0001.pdb
      we have some result

  - If a volume is generated:

* Protocol output (below *Scipion* framework):
  `chimera - chimera operate -> ouput3Dmap`; `Volume (x, y, and z dimensions, sampling rate)`.
  `chimera - chimera operate -> ouputPdb_01`; `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:
  chimeraOut0001.pdb
  chimeraOut0001.mrc
  we have some result

E Chimera Restore Session protocol

Protocol designed to restore *Chimera* session, provided that this session has been saved previously in *Scipion*. Currently, three protocols save *Chimera* sessions when *Chimera* commands `scipionwrite` or `scipionss` are used, chimera rigid fit, chimera operate and model from template (Appendices F, D and P, respectively). Restored sessions allow inspect any element contained in a previously saved *Chimera* session, perform *Chimera* operations, and finally save electron density volumes or atomic structures.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Tools-Calculators` (Fig. 66 (A))
- Protocol form parameters (Fig. 66 (B)):

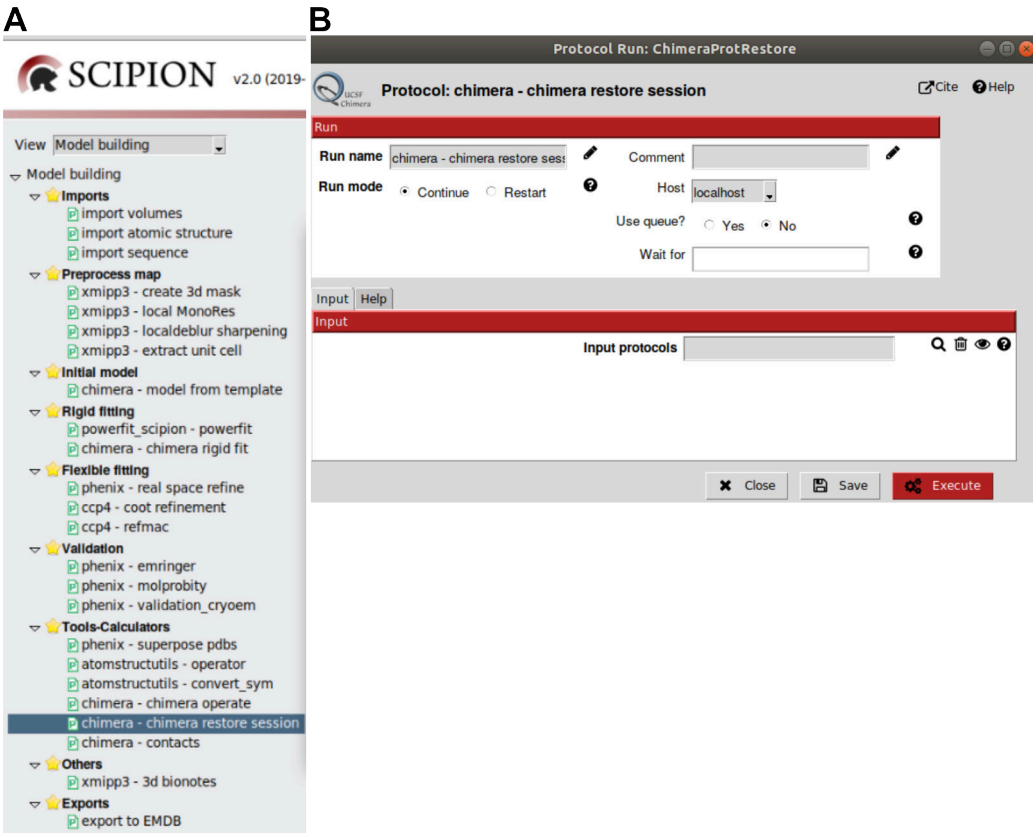


Figure 66: Protocol chimera restore session. A: Protocol location in *Scipion* menu. B: Protocol form.

– `Input` section

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

– `Help` section

  This section contains *Chimera* commands required to save *models* according to their reference volumes, which can also be saved if required. Remark that using `scipionwrite` command, *Chimera* session will be saved by default, without prejudice that it may be saved with `scipi-`

`onss` command. *Chimera* sessions can be restored again by using this same chimera 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.

  Chimera graphics window will be opened after executing the protocol showing the complete list of elements that appeared in *Chimera* 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 *Chimera*, and they can be saved in *Scipion* in the common way.

  – To save an atomic structure generated with this protocol: Write in *Chimera* command line:
    `scipionwrite model #n`
    If you want to save the model regarding any electron density map, write in *Chimera* command line:
    `scipionwrite model #n refmodel #n saverefmodel 0/1`.
    Replace `#n` by model numbers shown in *Chimera* `Model Panel`. Select `1` if you want to save the reference map or `0` otherwise, to avoid duplicate data unnecessarily.

  – To save a volume generated with this protocol:
    New volume data sets have to be saved first in a separate volume file. Otherwise those volumes will not be saved in the *Chimera* session since session files only record file paths to volumes. Assuming that the name

chosen for the new volume generated is `volume_name.mrc`, and `abspath` the absolute path in which you want to save the volume, write in *Chimera* command line:
`volume #n save abspath/volume_name.mrc`
`scipionwrite model #n refmodel #n saverefmodel 1`.
Replace `#n` by model numbers shown in *Chimera* `Model Panel`. Remark that you have to save an atomic structure if you want to save a volume in *Scipion*.

- Close *Chimera* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, volume, and atomic structure are model numbers `#0`, `#1`, and `#2`, respectively, in *Chimera* `Model Panel`. If no volumes have been included, coordinate axes and each atomic structure are model numbers `#0` and `#1`, respectively.

- Summary content:

  - If an atomic structure is generated:

    * Protocol output (below *Scipion* framework): `chimera - chimera restore session -> ouputPdb_01`; `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:

chimeraOut0001.pdb
we have some result

- If a volume is generated:
  - Protocol output (below *Scipion* framework): `chimera - chimera restore session -> ouput3Dmap`; `Volume (x, y, and z dimensions, sampling rate)`.
    `chimera - chimera restore session -> ouputPdb_01`; `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:
    chimeraOut0001.pdb
    chimeraOut0001.mrc
    we have some result

F Chimera Rigid Fit protocol

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

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Rigid fitting` (Fig. 67 (A))

- Protocol form parameters (Fig. 67 (B)):

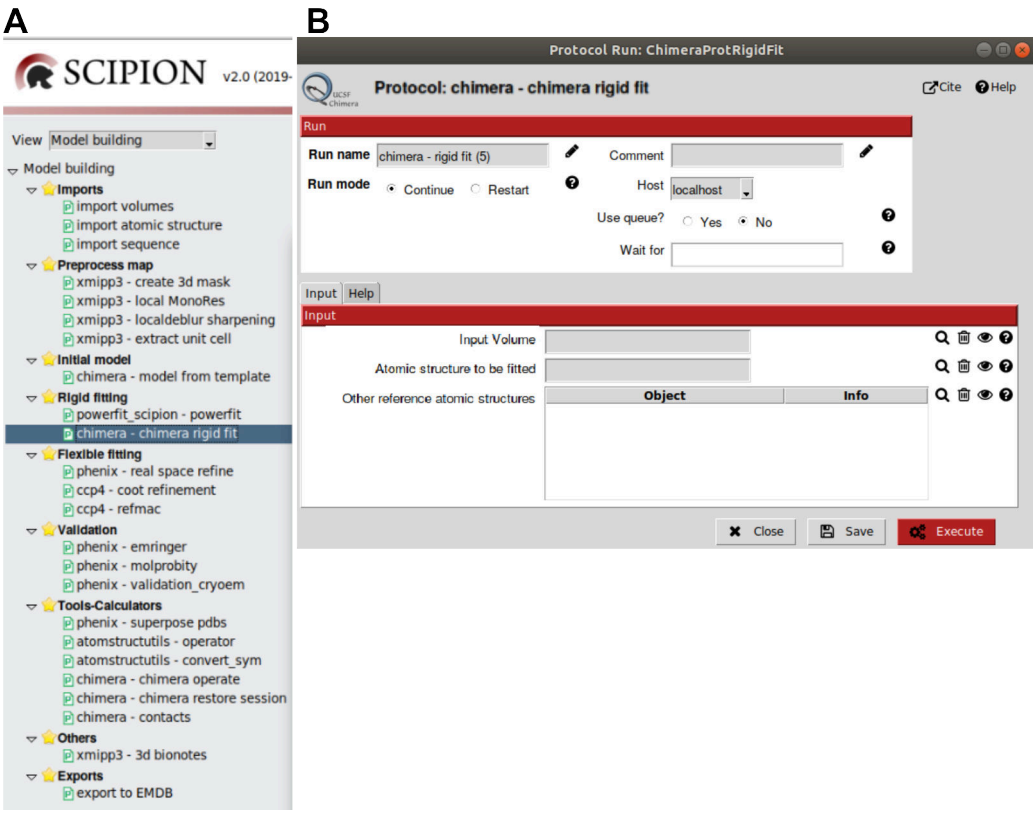


Figure 67: Protocol chimera rigid fit. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input` section
    - `Input Volume`: Electron density map previously downloaded or generated in *Scipion* to fit the atomic structure.
    - `Atomic structure to be fitted`: 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 *Chimera* commands required to save *models* according to their reference volumes, which can also be saved if required. Remark that using `scipionwrite` command, *Chimera* session will be saved by default, without prejudice that it may be saved with `scipionss` command. *Chimera* sessions can be restored by using chimera restore session protocol.

- 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.

  Chimera 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 *Chimera* main menu and select `Tools -> Volume -> Fit in Map`. A small `Fit in Map` window will be opened. Once atomic structure and electron density volume have been selected, fit them by clicking `Fit`. Press `Help` to consider different fitting options.

  – If map and *model* are far to each other, start the fitting process interactively activating and inactivating *Chimera* objects alternatively to finally get map and *model* close enough to go to the previous step.

  – Save fitted *model* regarding the electron density map by writing in *Chimera* command line:
  `scipionwrite model #n refmodel #n saverefmodel 0/1`.
  Replace `#n` by model numbers shown in *Chimera* `Model Panel`. Select `1`

if you want to save the reference map or `0` otherwise, to avoid duplicate data unnecessarily.

- Close *Chimera* graphics window.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, volume, and atomic structure are model numbers `#0`, `#1`, and `#2`, respectively, in *Chimera* `Model Panel`.

- Summary content:

  - If only the atomic structure has been saved by `scipionwrite` command:

    * Protocol output (below *Scipion* framework):
      `chimera - rigid fit -> ouputPdb_01; 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:
      chimeraOut0001.pdb
      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):
      `chimera - rigid fit -> ouput3Dmap; Volume (x, y, and z dimensions, sampling rate)`.
      `chimera - rigid fit -> ouputPdb_01; 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:
chimeraOut0001.pdb
chimeraOut0001.mrc
we have some result

G 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-ccp4`
  - CCP4 software suite (version 7.0.056 or higher)
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu:
  `Model building -> Flexible fitting` (Fig. 68 (A))
- Protocol form parameters (Fig. 68 (B)):

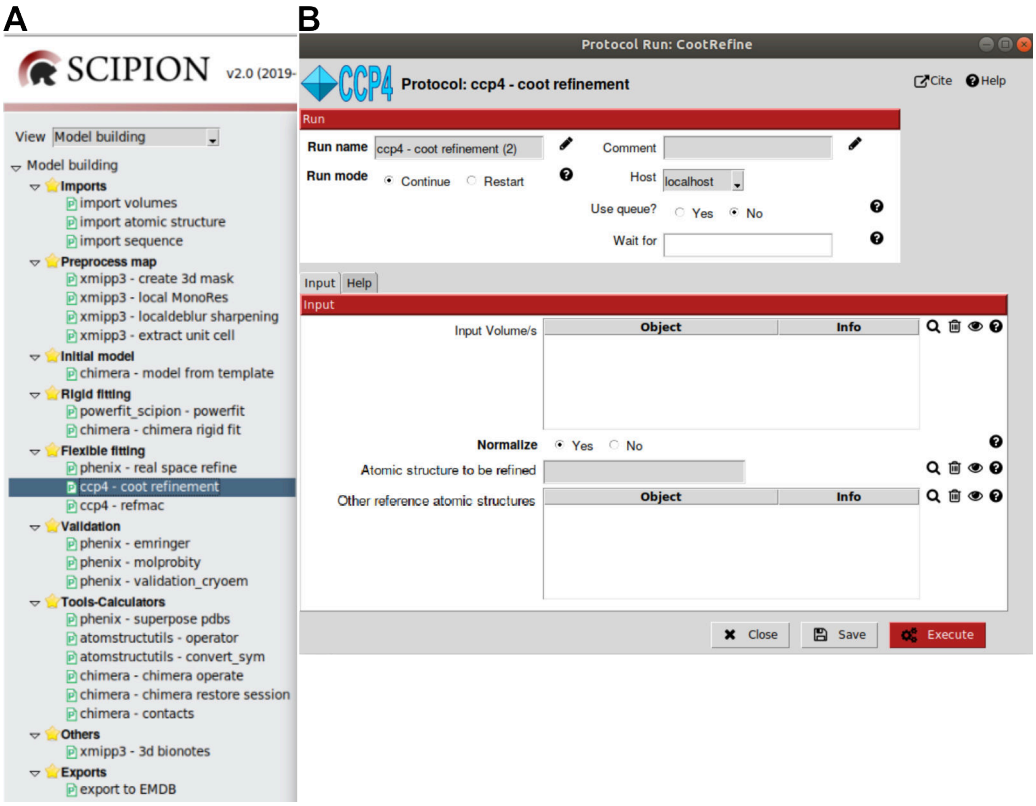


Figure 68: Protocol ccp4 - coot refinement. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input` section
    - `Input Volume/s`: 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`: 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. 73 (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. 69 (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. 69 (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. 69 (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 `cootOut0001.pdb` by default, 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. 69 (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.

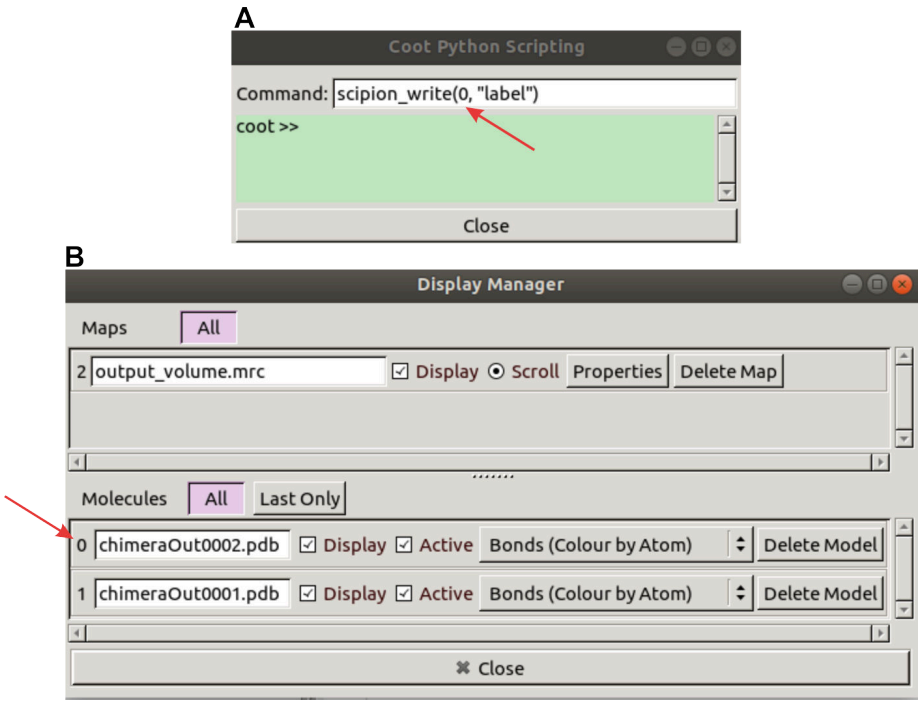


Figure 69: Protocol ccp4 - coot refinement. A: Saving labeled atomic structure with `Coot Python Scripting` window. B: `Display Manager` window.

- 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. 69 (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. 73 (2)), edit it and save it if needed.
- Set refinement conditions:
  Click `Refine/Regularize control` button (upper right side of *Coot* graphics window) (Fig. 70 (1)) and select the four restriction types in `Refinement and regularization Parameters` window (2).

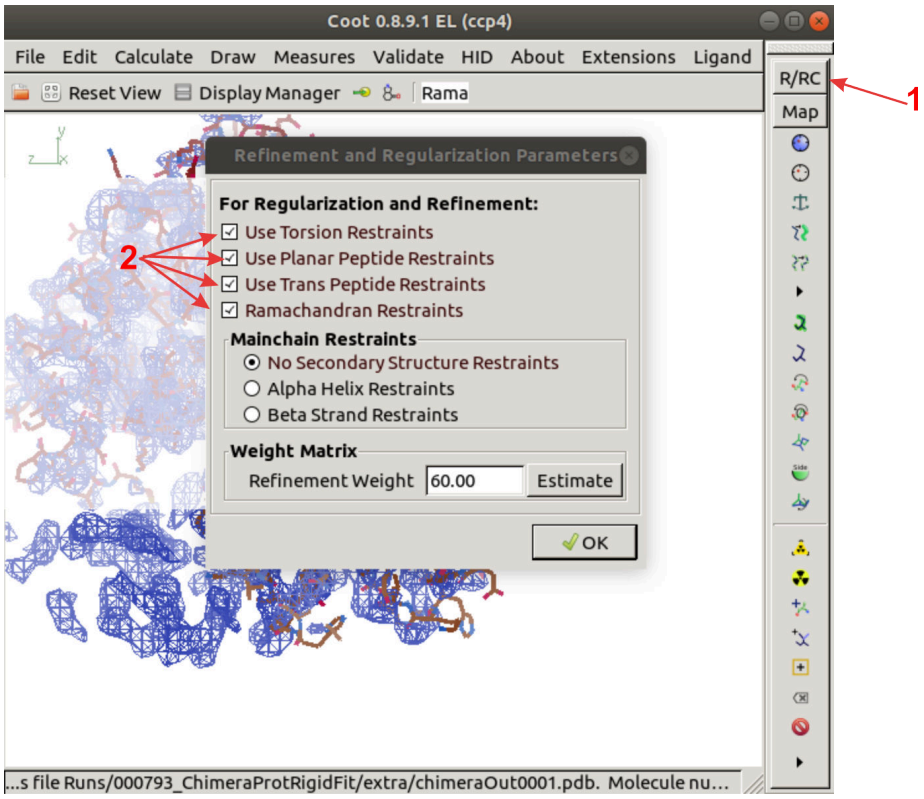


Figure 70: Protocol ccp4 - coot refinement. `Refinement and regularization Parameters` window.

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 `Validation -> 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. 71 (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. 71 (A)).
* If refinement parameters get acceptable values, press `Accept` in `Accept Refinement?` window (Fig. 71 (B)).

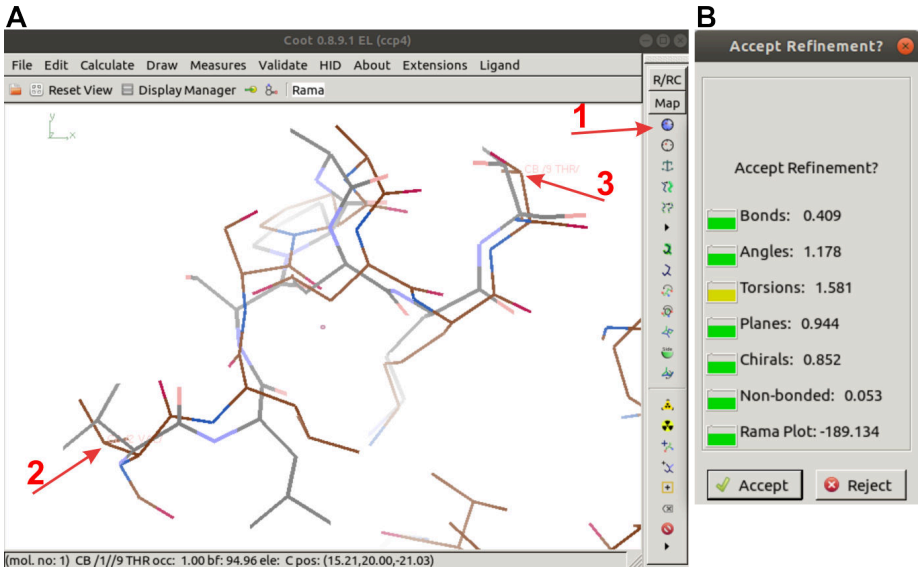


Figure 71: Protocol ccp4 - coot refinement. (A) Interactive refinement of the chain fragment between residues 2 and 9. (B) Accepting refinement window.

– 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 `Validation -> 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 *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, volume, and first atomic structure are model numbers `#0`, `#1`, `#2`, respectively, in *Chimera* `Model Panel`. Every atomic structure saved during *Coot* refinement process will appear in `Model Panel` (Fig. 72). 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.

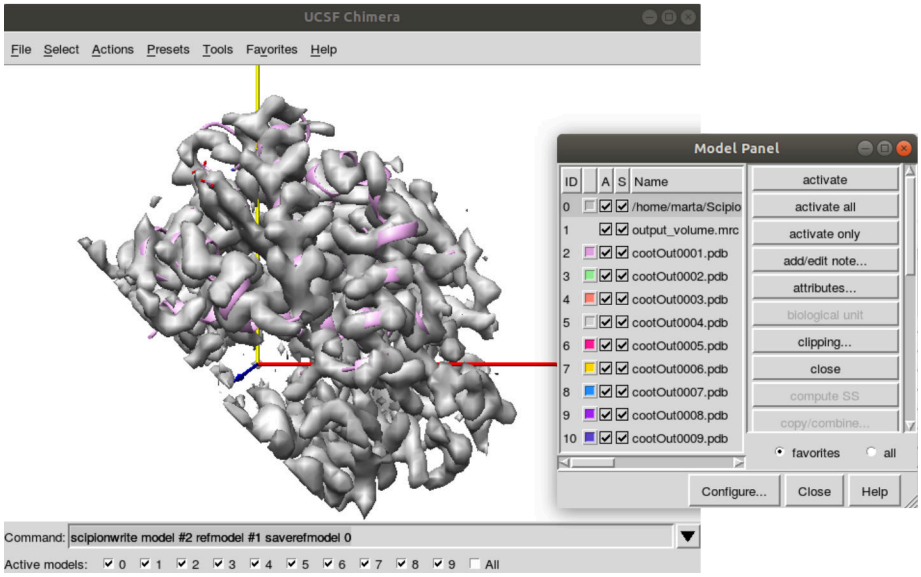


Figure 72: Protocol ccp4 - coot refinement. *Coot* results visualized in *Chimera*.

Since *Scipion* projects keep every intermediate atomic structure partially refined (Fig. 73 (1, 3), users can include any of them in successive following modeling workflow steps performed in *Scipion* (Fig. 74).

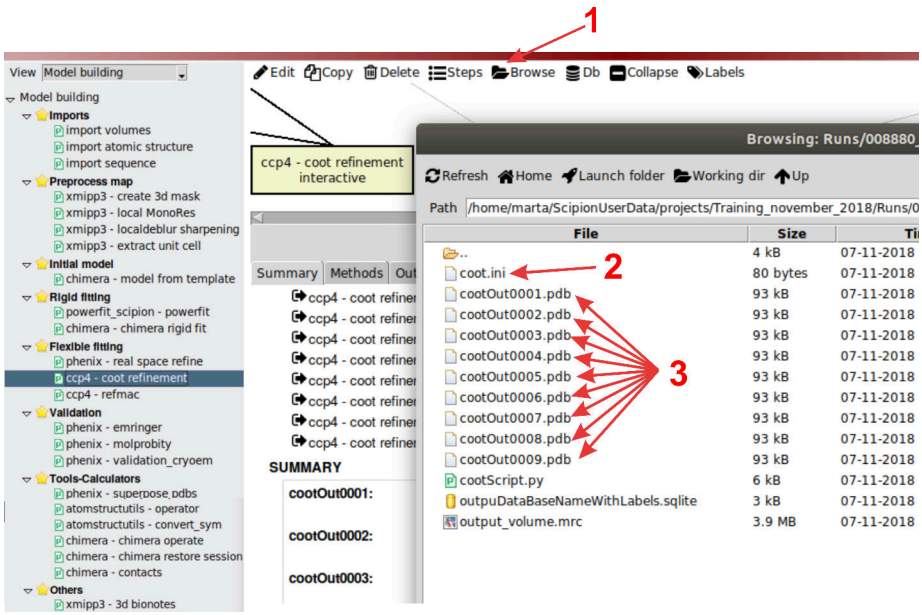


Figure 73: Protocol ccp4 - coot refinement. Browse content after several runs of interactive *Coot* protocol.

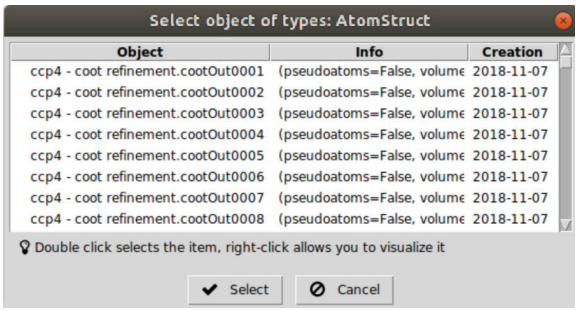


Figure 74: Protocol ccp4 - coot refinement. *Scipion* window that allows to select any of *Coot* partially refined structures.

- Summary content:
  - Protocol output (below *Scipion* framework) for each *Coot* intermediate atomic structure partially refined (`#n`):
    `ccp4 - coot refinement -> cootOut000#n`; `PdbFile (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.

– `SUMMARY` box for each *Coot* intermediate atomic structure partially refined (`#n`):
`cootOut000#n`:

H 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 scatttering 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-ccp4`

  - CCP4 software suite (version 7.0.056 or higher)
  - *Scipion* plugin: `scipion-em-chimera`

- *Scipion* menu: `Model building -> Flexible fitting` (Fig. 75 (A))
- Protocol form parameters (Fig. 75 (B)):

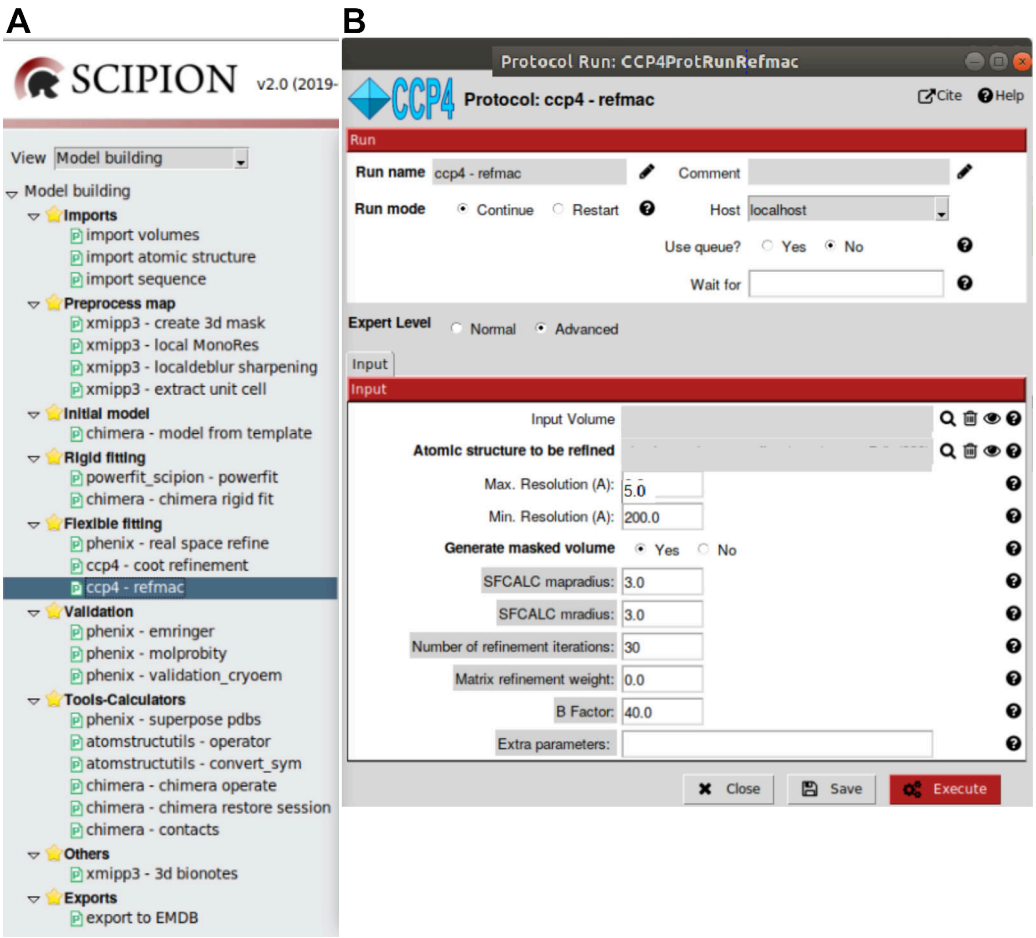


Figure 75: Protocol ccp4 - refmac. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input Volume/s`: One 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. 76). Results can be visualized by selecting each menu element.

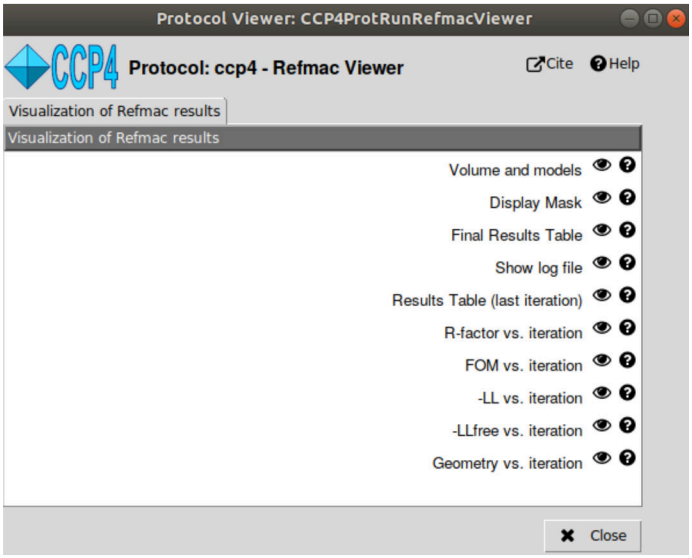


Figure 76: Protocol ccp4 - refmac. Menu to visualize *Refmac* results.

Options to visualize *Refmac* results:

  – Volume and models: *Chimera* graphics window displays coordinate axes, selected input volume, starting atomic structure generated by *Coot*, and final *Refmac* refined structure (Fig. 77).

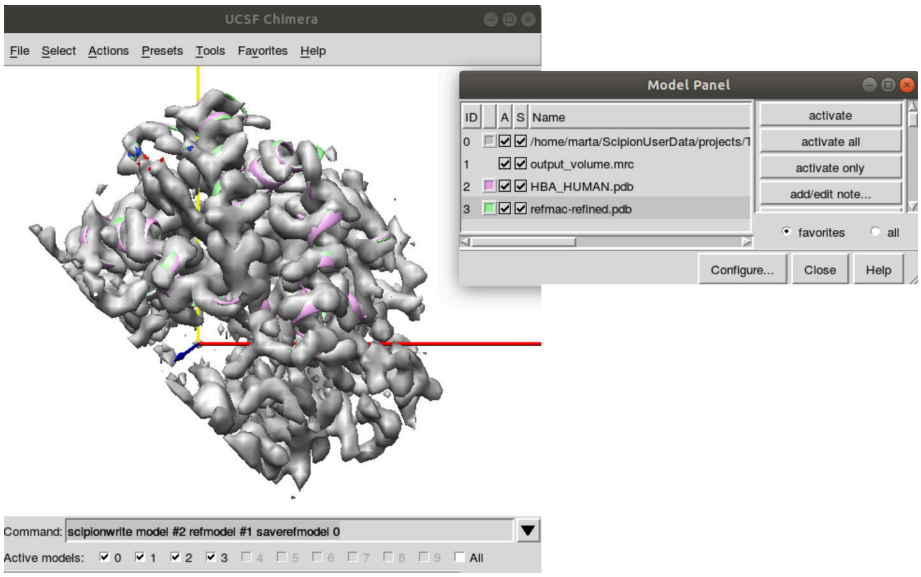


Figure 77: Protocol ccp4 - refmac. Map and models visualized with *Chimera*.

– Display Mask: *Chimera* graphics window displays the mask generated around the model atomic structure that has to be refined (Fig. 78).

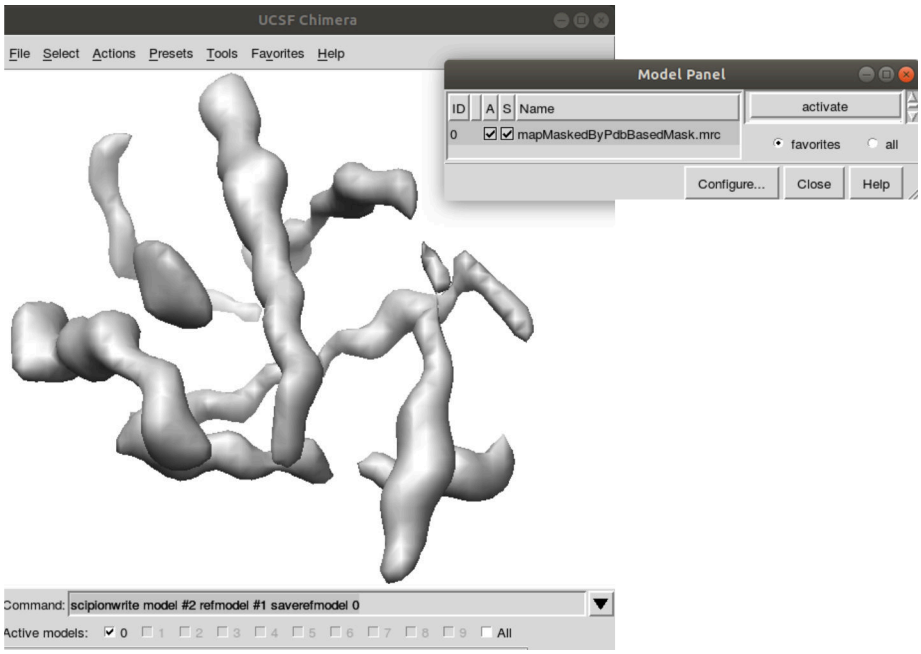


Figure 78: Protocol ccp4 - refmac. Mask visualized with *Chimera*.

– 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. 79).

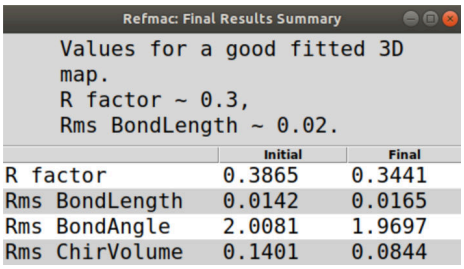


Figure 79: Protocol ccp4 - refmac. *Refmac* final results table.

– Show log file: *Refmac*-generated text file containing statistics of every *Refmac* running cycle (Fig. 80).

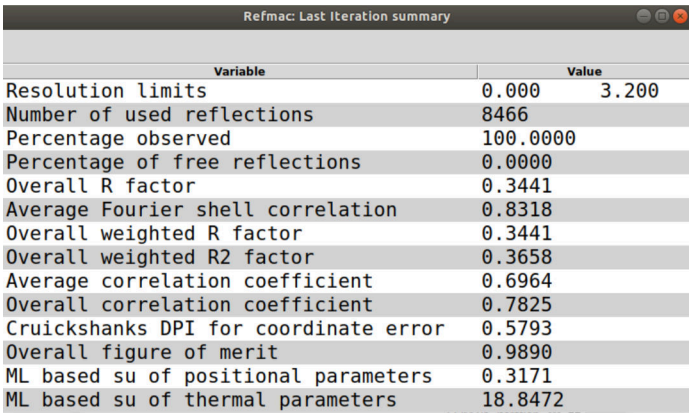
```
<B><FONT COLOR="#FF0000"><!--SUMMARY_BEGIN-->
<html> <!-- CCP4 HTML LOGFILE -->
<hr>
<!--SUMMARY_END--></FONT></B>
<B><FONT COLOR="#FF0000"><!--SUMMARY_BEGIN-->
<pre>

 ##################################################################
 ##################################################################
 ##################################################################
 ### CCP4 7.0.056: Refmac          version 5.8.0222 : 03/14/18##
 ##################################################################
 User: marta  Run date: 27/10/2018 Run time: 16:28:14


 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 80: Protocol ccp4 - refmac. *Refmac* raw log file.

– Results Table (last iteration) (Fig. 81):



Variable	Value
Resolution limits	0.000 3.200
Number of used reflections	8466
Percentage observed	100.0000
Percentage of free reflections	0.0000
Overall R factor	0.3441
Average Fourier shell correlation	0.8318
Overall weighted R factor	0.3441
Overall weighted R2 factor	0.3658
Average correlation coefficient	0.6964
Overall correlation coefficient	0.7825
Cruickshanks DPI for coordinate error	0.5793
Overall figure of merit	0.9890
ML based su of positional parameters	0.3171
ML based su of thermal parameters	18.8472

Figure 81: 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. 82):

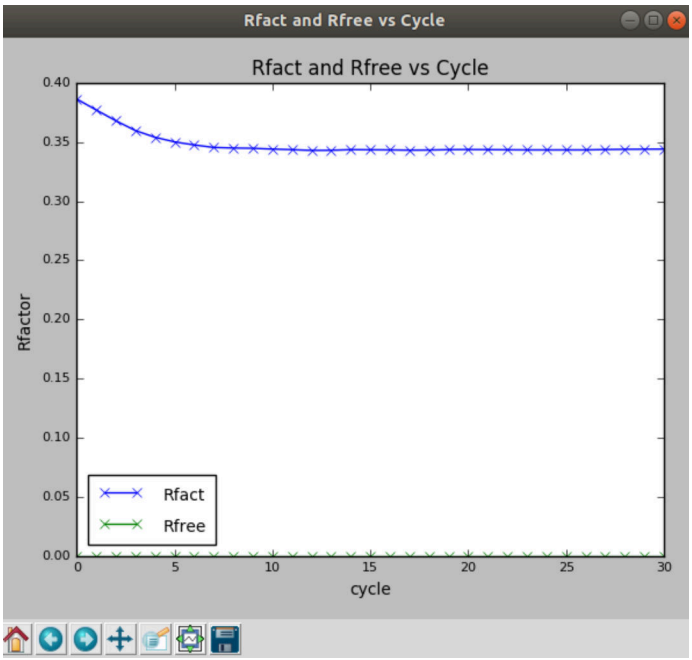


Figure 82: Protocol ccp4 - refmac. `R factor` vs. cycle plot.

– `FOM` vs. iteration: Plot to visualize Figure Of Merit regarding iterations

(Fig. 83):

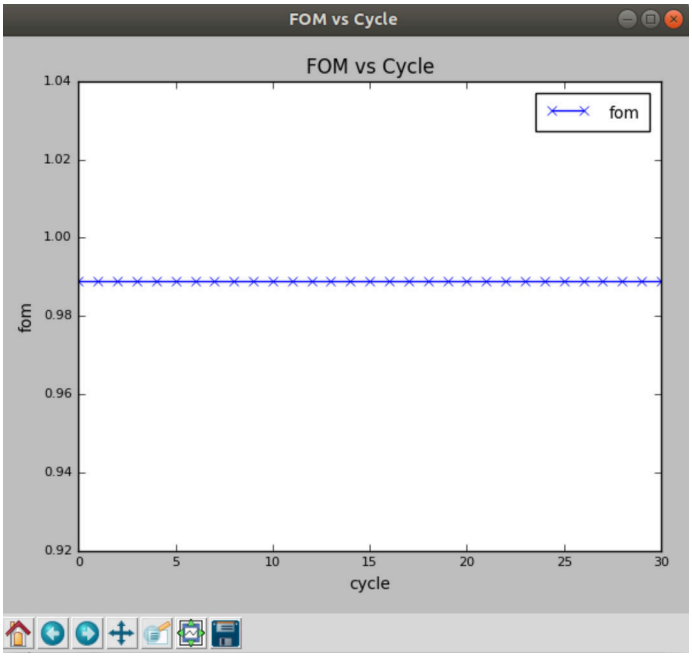


Figure 83: Protocol ccp4 - refmac. `Figure Of Merit` vs. cycle plot.

– `-LL` vs. iteration: Plot to visualize the log(Likelihood) regarding iterations. Likelihood indicates the probability of a refined model, given the specific observed data (Fig. 84):

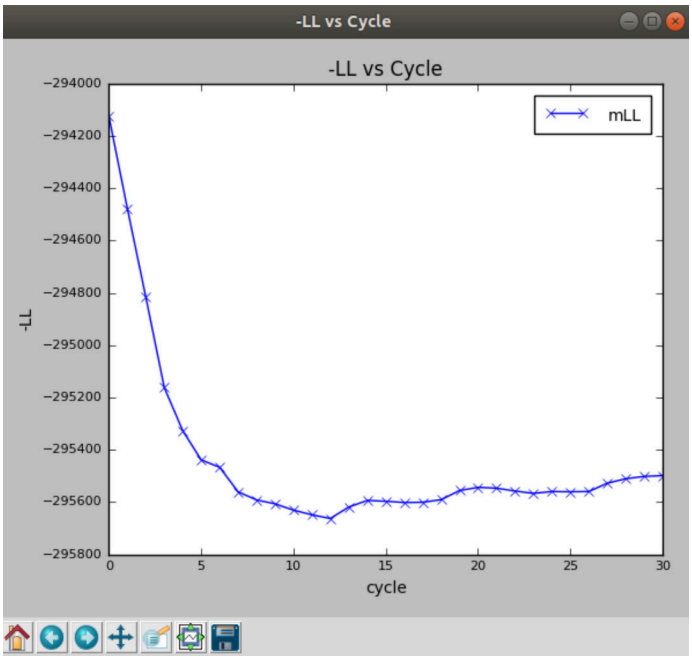


Figure 84: Protocol ccp4 - refmac. log(Likelihood) vs. cycle plot.

– `-LLfree` vs. iteration: Same definition as -LL vs. iteration, although considering only "free" reflections not included in refinement (Fig. 85):

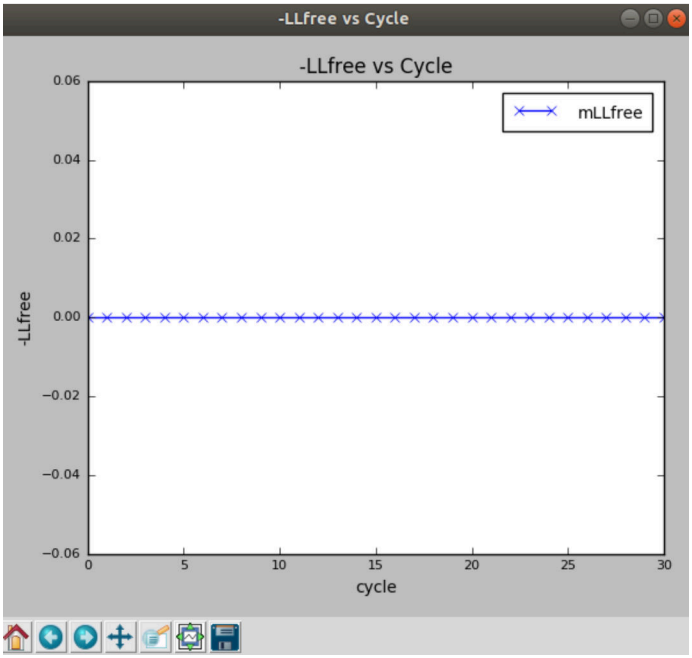


Figure 85: Protocol ccp4 - refmac. log(Likelihood) for "free" reflections vs. cycle plot.

– Geometry vs. iteration: Plot to visualize geometry parameter statistics regarding iterations (Fig. 86):

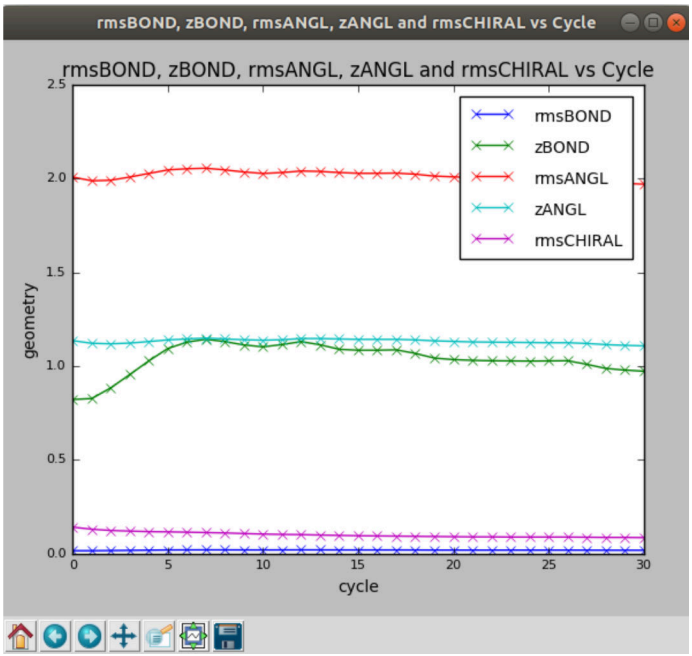


Figure 86: Protocol ccp4 - refmac. Geometry parameter statistics vs. cycle plot.

* `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. 87):

Figure 87: Protocol ccp4 - refmac. Summary.

.

I 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-xmipp`

- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 88 (A))

- Protocol form parameters (Fig. 88 (B: Mask generation; C: Postprocessing)):

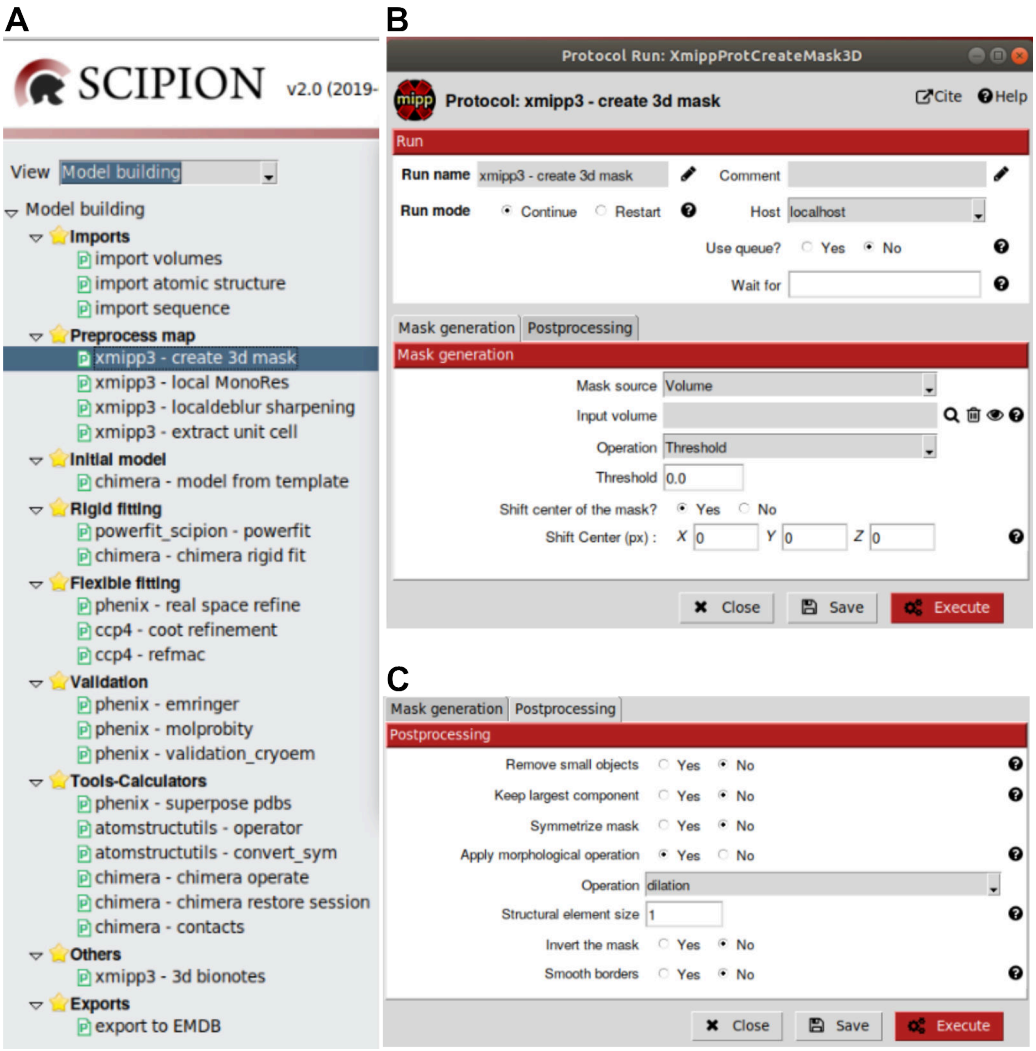


Figure 88: Protocol `xmipp3 - create 3d mask`. A: Protocol location in *Scipion* menu. B, C: Protocol form.

  – `Mask generation`

* `Mask source`: Selection of one of the two possible types of sources for the mask, the map volume provided by the user or a specific geometrical design.

  If a volume is selected:

  · `Input volume`: Electron density map previously downloaded or generated in *Scipion*.

  · `Operation`: Approach applied to generate the mask, besides methods that can be selected in Postprocessing, such as establishing a particular density `threshold`, a segmentation process according to the number of voxels, number of aminoacids, atomic mass (Daltons) or automatic.

  If a geometric shape is selected:

  · `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`.

– `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 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 *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`), the default *Scipion* viewer, will open the mask by slices. The *ShowJ* window menu (`File -> Open with Chimera`) allows to open the mask volume in *Chimera* 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`.

J Extract unit cell 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-xmipp`
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 89 (A))
- Protocol form parameters (Fig. 89 (B)):

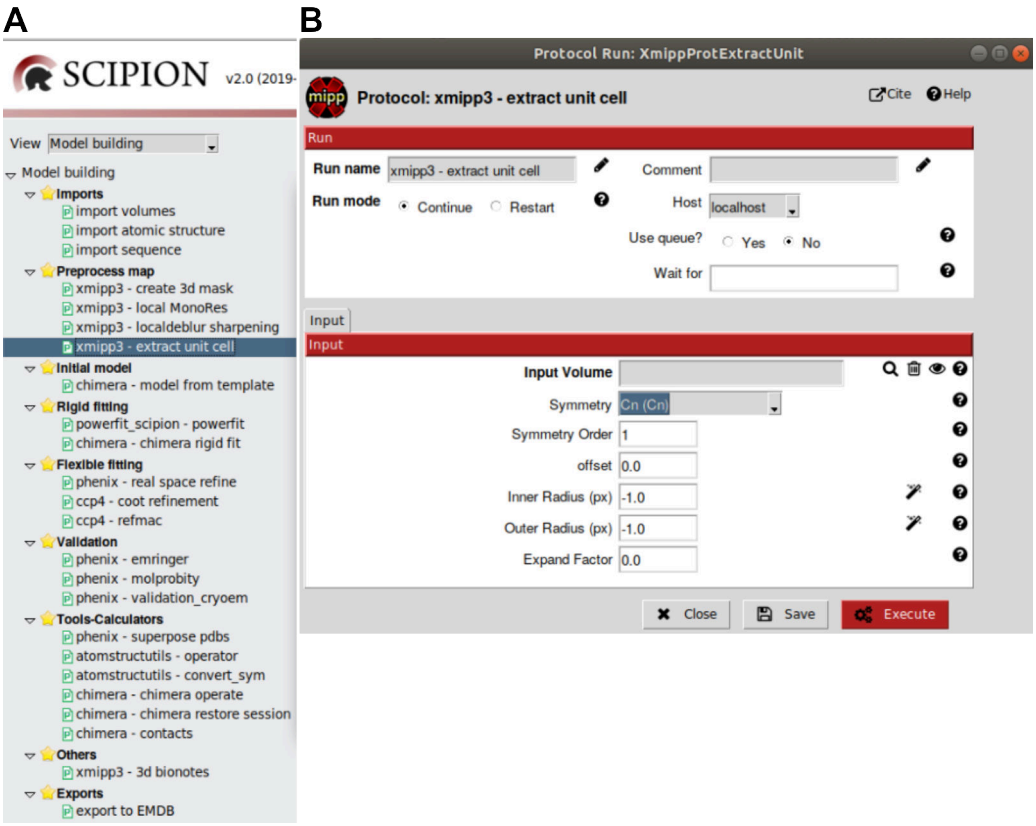


Figure 89: Protocol xmipp3 - extract unit cell. A: Protocol location in *Scipion* menu. B: Protocol form.

– `Input Volume`: Volume already downloaded in *Scipion* from which the unit cell 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. 90. 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 *Chimera*'s nomenclature, which is, in turn,

the same symmetry nomenclature of the International Union of Cristallography.

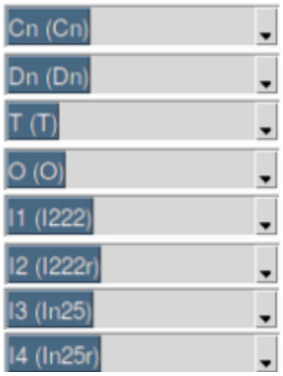


Figure 90: Protocol xmipp3 - extract unit cell. Types of rotational symmetry.

* 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 (Dn)`: 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, `360°/n` determines the rotation angle.

· `offset`: Starting angle around Z axis.

* Tetrahedral symmetry `T (T)`: 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 icosaedral symmetry depends on its initial orientation. Check in *Chimera* each one of them from the main graphics menu: `Tools -> Higher-Order Structure -> Icosahedron Surface -> Orientation` (`I222`: order 2 axes follow XYZ coordinate axes; `I222r`: idem rotated 90º around Z axis; `In25`: an order 2 axis and an order 5 axis follow Y and Z axes, respectively, `In25r`: idem rotated 90º around Z axis.

– `Inner Radius (px)`: Minimal distance from the geometric center that delimits inwards the part of the map electron density 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 map electron density 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. 91). This window allows you to select between `chimera` (*Chimera* graphics window) and `slices` (*ShowJ*, the default *Scipion* viewer), to visualize the volume.

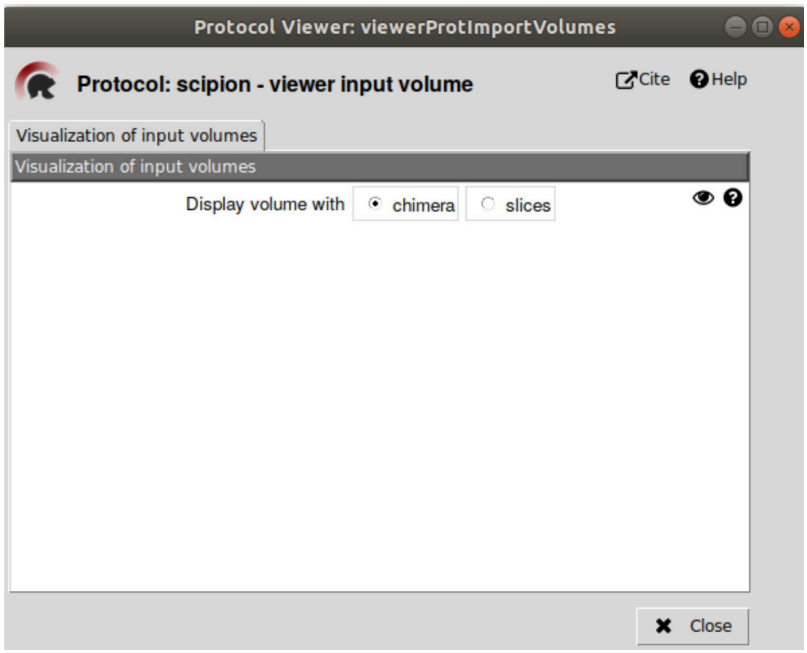


Figure 91: Menu to select a visualization tool.

  – `chimera`: *Chimera* graphics window

Initial whole volume and extracted volume appear referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, initial volume, and extracted unit cell volume are model numbers `#0`, `#1` and `#2`, respectively, in *Chimera* `Model Panel`. Volume coordinates and pixel size can be checked in *Chimera* main menu `Tools -> Volume Data -> Volume Viewer -> Features -> 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. 92.

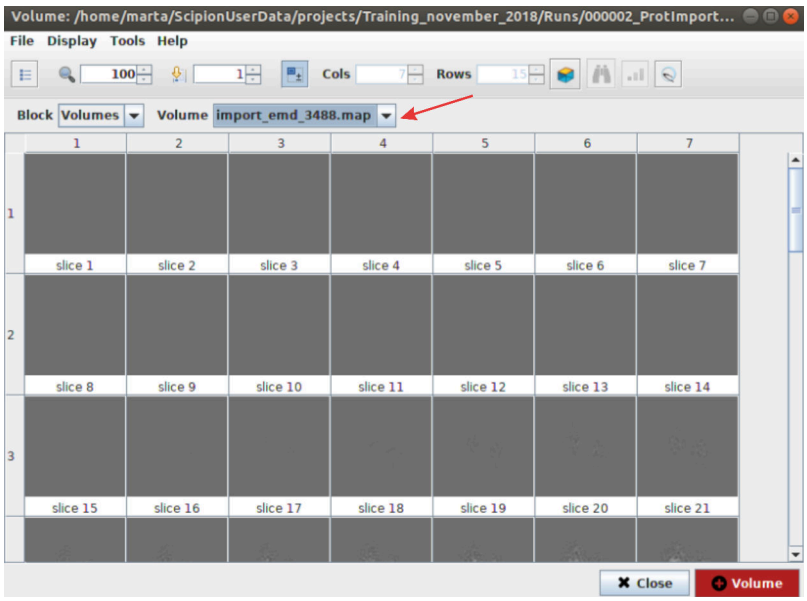


Figure 92: Protocol extract unit cell. Volume selection with ShowJ.

- Summary content:

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

K 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-chimera`
- *Scipion* menu: `Model building -> Imports` (Fig. 93 (A))
- Protocol form parameters (Fig. 93 (B)):

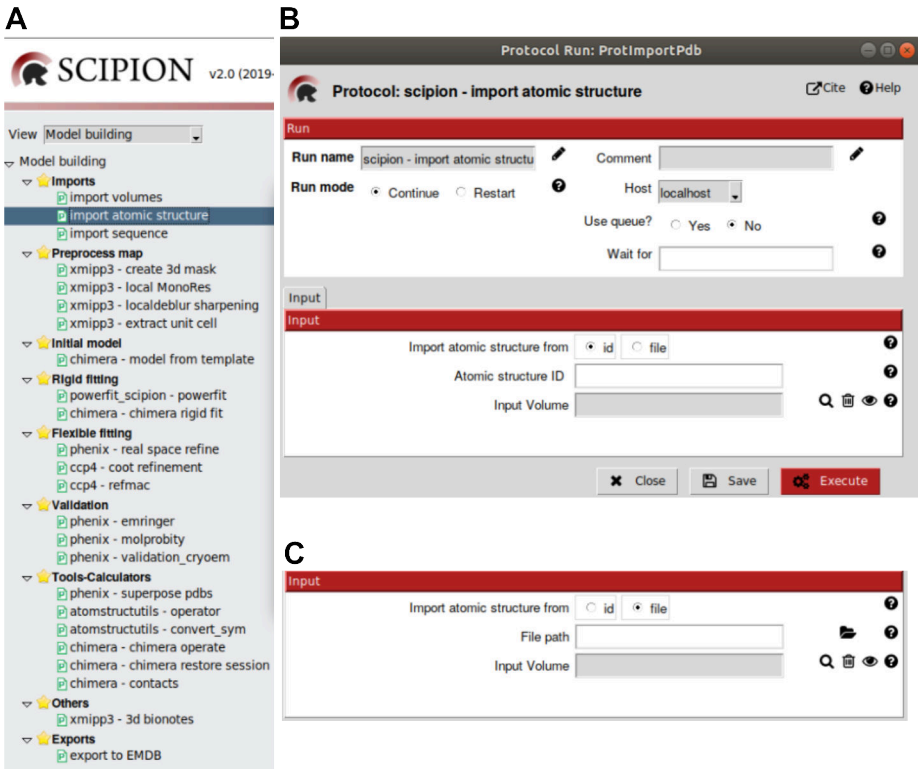


Figure 93: Protocol import atomic structure. A: Protocol location in *Scipion* menu. B: Protocol form to import the atomic structure from PDB. C: Protocol form to import the atomic structure from a file.

– `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. 93 (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 *Chimera* graphics window will be opened by default (Fig. 65). Atomic structures are referred to the origin of coordinates in *Chimera*. 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 `#0` and `#1`, respectively, in *Chimera* `Model Panel`. If a volume has been associated to the atomic structure, coordinate axes and imported atomic structure are model numbers `#1` and `#2`, respectively, in *Chimera* `Model Panel`, whereas structure-associated volume has model number `#0`. Volume coordinates and pixel size can be checked in *Chimera* main menu `Tools -> Volume Data -> Volume Viewer -> Features -> Coordinates: Origin index/ Voxel size`. WARNING: Take into account that coordinates appear in pixels while they have been introduced in Å.

- Summary content:

  - Protocol output (below *Scipion* framework):
    `scipion - import 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`

L 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 file in your computer, text file of the user's computer, and `UniProtKB/ GeneBank` databases).

- *Scipion* menu: `Model building -> Imports` (Fig. 94 (A))
- Protocol form parameters (Fig. 94 (B)):

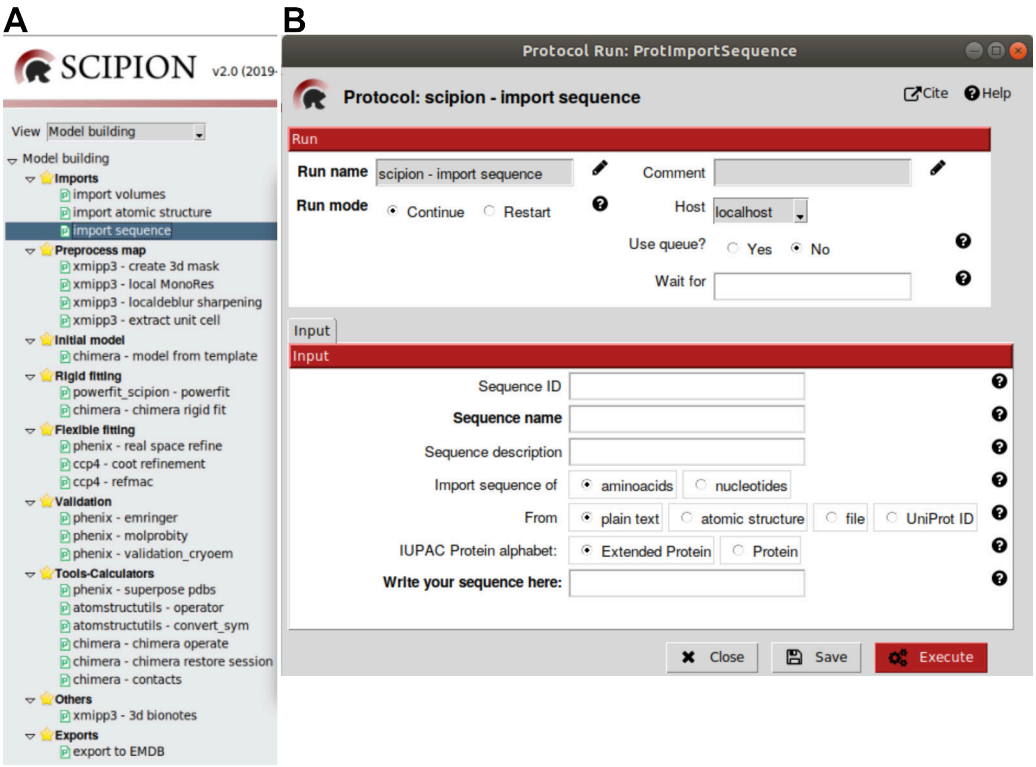


Figure 94: Protocol import sequence. A: Protocol location in *Scipion* menu. B: Protocol form.

- `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`.
- `Sequence name`: Compulsory short name to identify your sequence (`PGH2_MOUSE`). Names with certain meaning are recommended.
- `Sequence description`: Optional description of your sequence. It can include functionality, organism, size, etc... (e.g. `Prostaglandin G/H synthase 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. 95 (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. 95 (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. 95 (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.

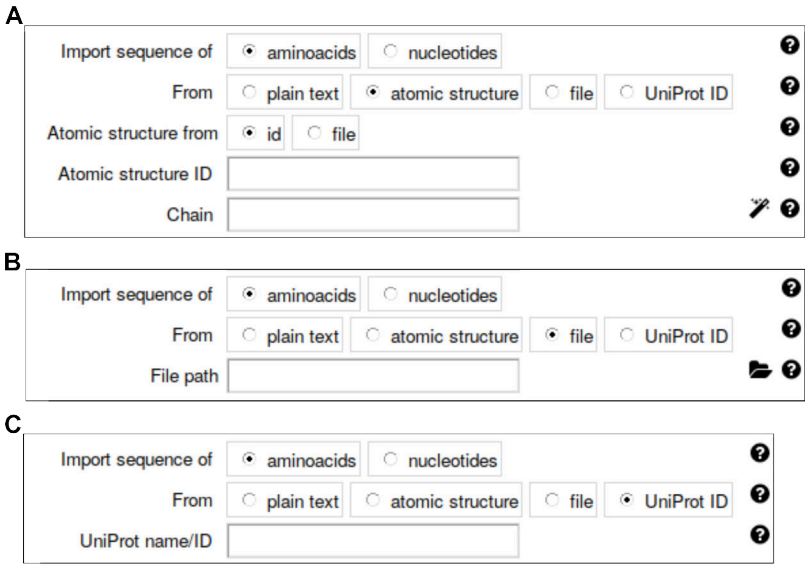


Figure 95: Protocol import sequence. Protocol form to import aminoacid sequences from the `PDB` database by indicating its respective `ID` (A), from a file (B), or from `UniProtKB` by writing the database `ID/name` (C).

* `nucleotides`: Analogously to `aminoacids` parameter, select one of these four options:
  - `plain text`: Parameter to introduce your own single letter nucleotide sequence (Fig. 96 (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 `Unam-`

`biguous 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. 96 (B)).
- `file`: Information described for aminoacids is valid for nucleotides (Fig. 96 (C)).
- `GeneBank`: Information described for aminoacids is valid for nucleotides, this time replacing `UniProtKB` by `GeneBank` (Fig. 96 (D)).

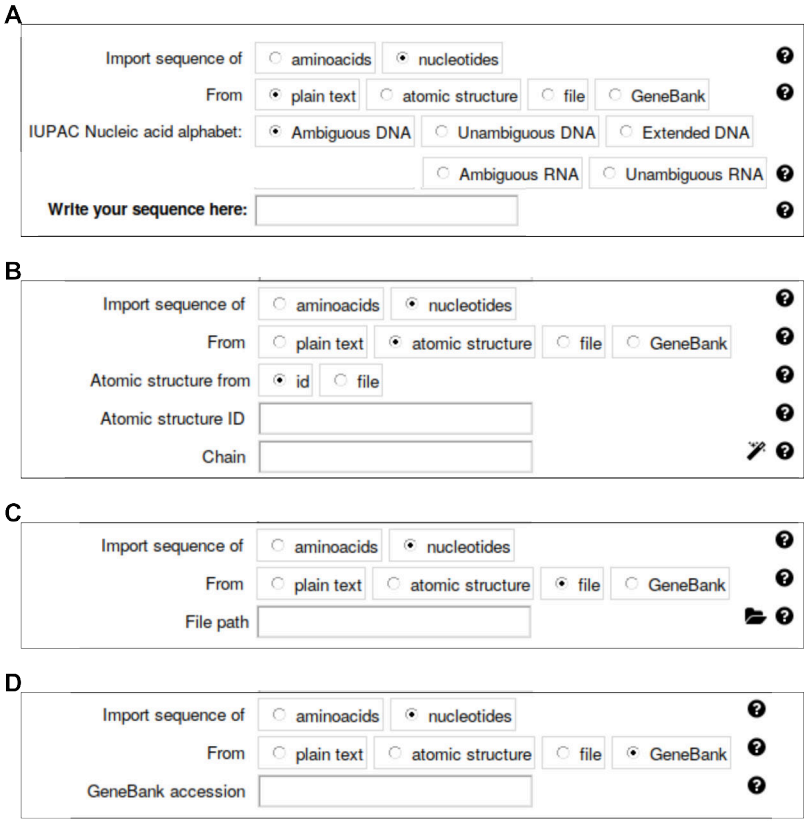


Figure 96: Protocol import sequence. Protocol form to write (A) or import nucleotide sequences from PDB database by indicating its respective ID (B), from a file (C), or from GeneBank by writing the database accession number (D).

- 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):
    `scipion - import sequence -> ouputSequence`; `Sequence name`
  - `SUMMARY` box:
    Sequence of aminoacids/ nucleotides:
    Sequence `Sequence name` imported from plain text/ atomic structure/ file/ UniProt ID.

M 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-chimera`

- *Scipion* menu: `Model building -> Imports` (Fig. 97 (A))

- Protocol form parameters (Fig. 97 (B)):

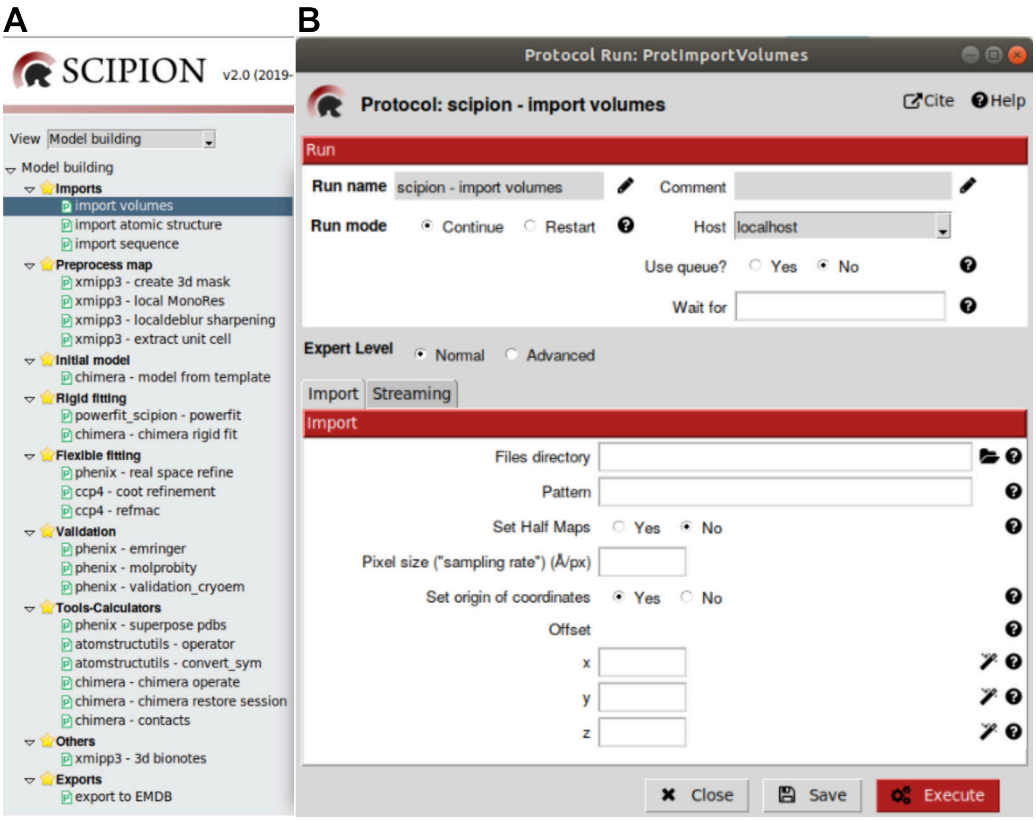


Figure 97: Protocol import volumes. A: Protocol location in *Scipion* menu. B: Protocol form.

– `Input` section

  * `Files directory`: Folder that contain one or several volumes (a set of volumes) that you'd like to import. By clicking the folder symbol right to the `Files directory` box, a browser will be opened to allow you 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.

* `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.
* `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 unit cell protocol. 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.

– `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. 91). This window allows you to select between `chimera` (*Chimera* graphics window) and `slices` (*ShowJ*, the default *Scipion* viewer), to visualize the volume.

  - `chimera`: *Chimera* graphics window

    Volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes and imported volume are model numbers `#0` and `#1`, respectively, in *Chimera* `Model Panel`. Volume coordinates and pixel size can be checked in *Chimera* main menu `Tools -> Volume Data -> Volume Viewer -> Features -> 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`

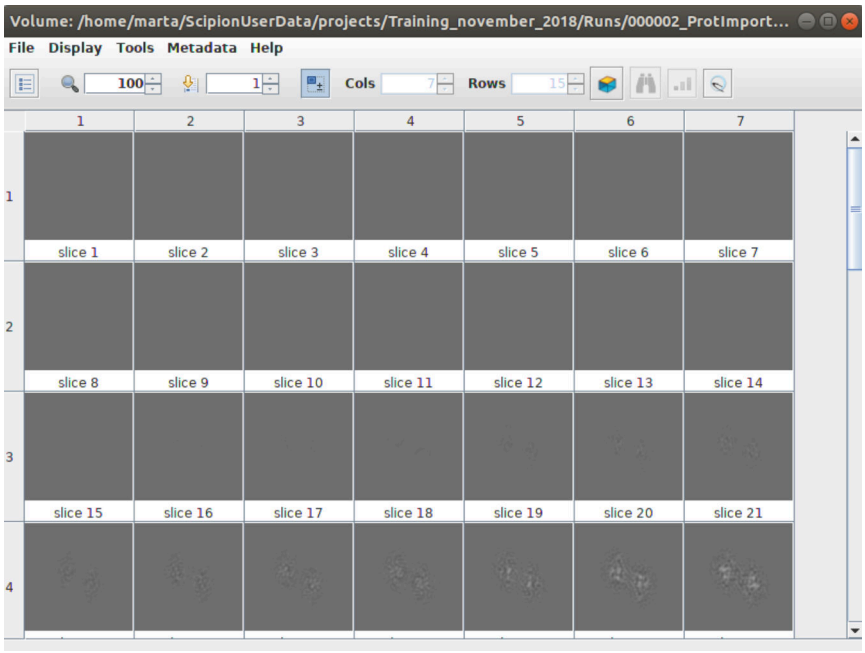


Figure 98: Protocol import volumes. Gallery model of *ShowJ* to visualize volume slices.

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

N 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.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em-xmipp`
- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 99 (A))
- Protocol form parameters (Fig. 99 (B)):

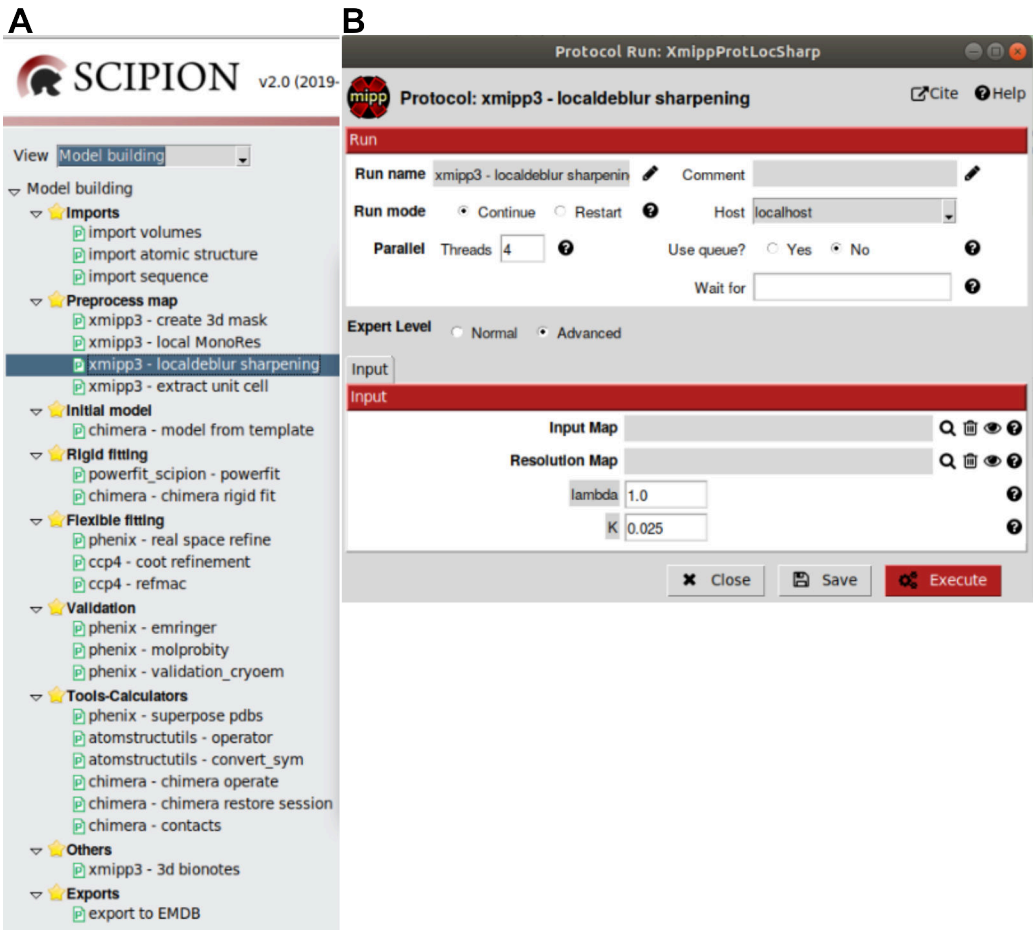


Figure 99: Protocol xmipp3 - localdeblur sharpening. A: Protocol location in *Scipion* menu. B: Protocol form.

- `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.

- 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 with *ShowJ* (`https://github.com/I2PC/scipion/wiki/ShowJ`), the default *Scipion* viewer, including the maps generated in each independent iteration before getting convergence. The sharpening algorithm stops when the difference between two successive iterations is lower than 1%, thus generating variable number of maps before stopping. The *ShowJ* window menu (`File -> Open with Chimera`) allows to open the selected map in *Chimera* graphics window.

- 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`.

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

- *Scipion* menu:
  `Model building -> Preprocess map` (Fig. 100 (A))

- Protocol form parameters (Fig. 100 (B)):

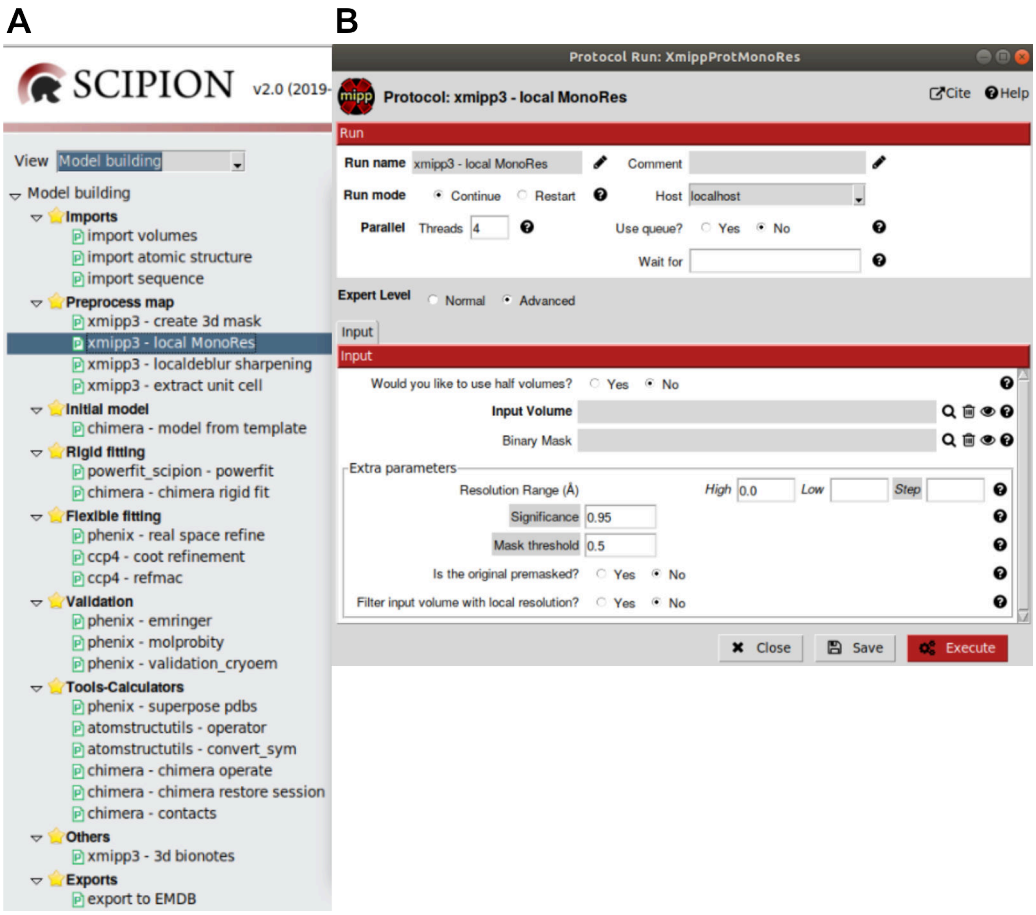


Figure 100: Protocol xmipp3 - local MonoRes. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `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 performed based on half volumes, select "Yes" when half volumes are available. A couple of boxes will thus be opened to select both half volumes, `Volume Half 1` and `Volume Half 2`.
  - `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.

- `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 maximum and minimal 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 it is not (0.5 by default). Density values below the threshold will be changed to 0 and values above the threshold will be changed to 1.
  - `Is the original premasked?`: "No" by default has to be changed to "Yes" if the original volume is already masked inside a spherical mask. The respective wizard will allow to specify the spherical mask radius (in pixels). By default, the half of the map volume size will be considered as radius. If the original is premasked, the noise will only be estimated in the volume contained between this premask and the binary mask provided before.
  - `Filter input volume with local resolution?`: This parameter allows filtering the input volume using the local resolution ("No" by default).

- 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. 101):

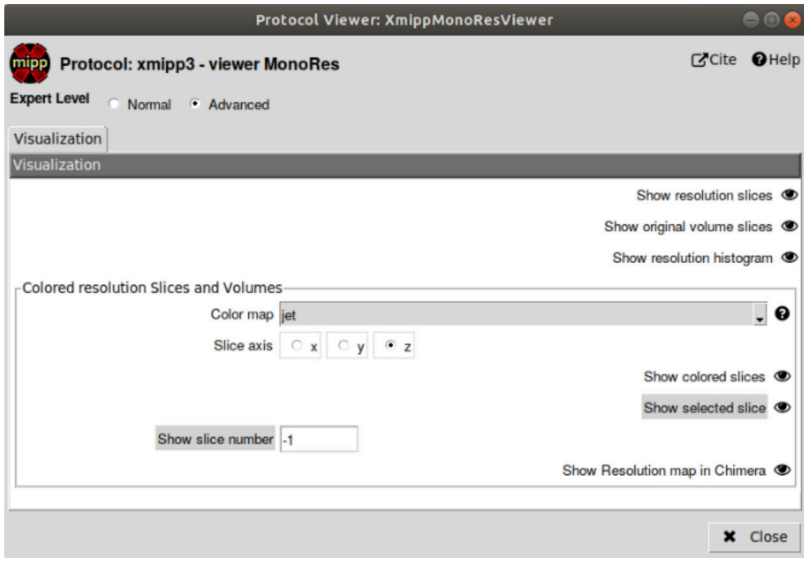


Figure 101: Protocol xmipp3 - local MonoRes. Menu to visualize results.

  - `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.

    * `Color map`: Color to apply to the local resolution map (`http://matplotlib.org/1.3.0/examples/color/colormaps_reference.html`).
    * `Slice axis`: Axis 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`: Slice number to be shown by `Show selected slice`.
    * `Show Resolution map in Chimera`: The resolution map is shown using *Chimera*. Left hand bar indicates resolution colour code.

- 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`.

P 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 *Chimera*.
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-chimera`

  - Multiple sequence alignment tools: `Clustal Omega`, `MUSCLE`

- *Scipion* menu: `Model building -> Initial model` (Fig. 102 (A))
- Protocol form parameters (Fig. 102 (B)):

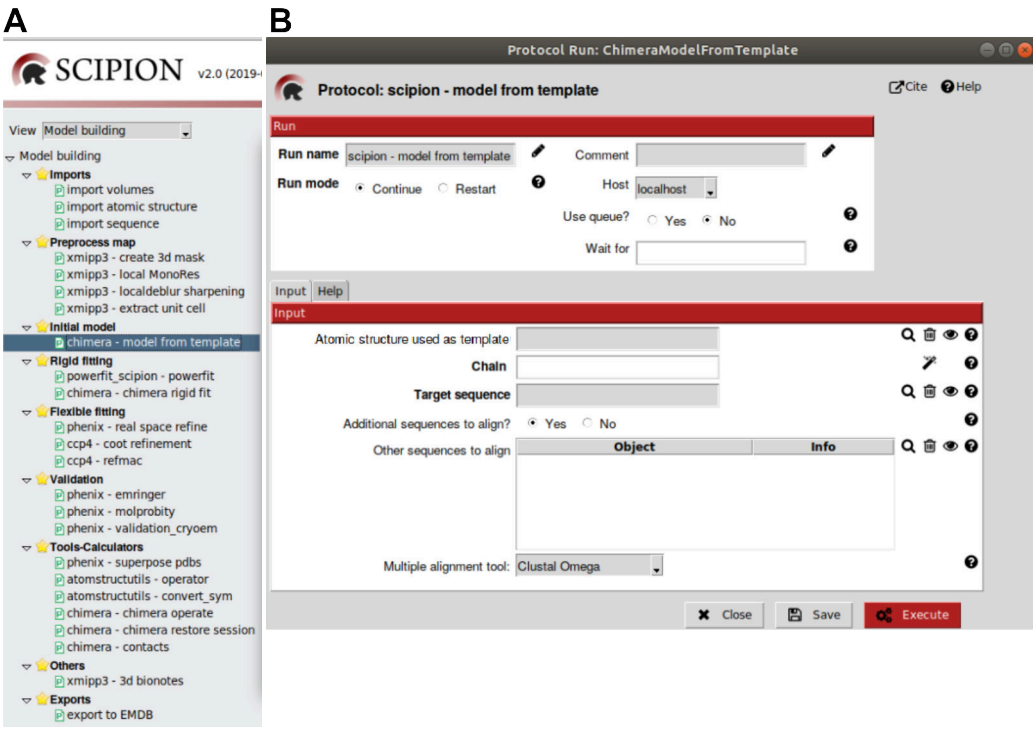


Figure 102: Protocol model from template. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input` section
    - `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`.
    * `Additional sequences to align?`: *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. Set to "No" this form parameter if no more sequences are going to be included in the alignment. Nevertheless, set the parameter to "Yes" if you want to perform a multiple sequence alignment. Additional sequences, others than `template` and `target` sequences, are required to accomplish this multiple alignment. That's why a new form parameter appear with the option "Yes":
        · `Other sequences to align`: Box to complete with the additional sequences used to perform the multiple sequence alignment. All of them were previously downloaded in *Scipion*.
    * `Multiple alignment tool`: Box to select an alignment program. Three possible options are given for pairwise alignments (`Bio.pairwise2, Clustal Omega, MUSCLE`) whereas only the two last ones are allowed for multiple sequence alignments.

  – `Help` section

    Follow this section steps to run *Modeller* via web service in *Chimera* 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 *Chimera* windows will be opened after executing the protocol. The two more relevant ones are *Chimera* graphics window and the multiple alignment window. In both windows the template chain is shown highlighted (see an example of these windows in Fig. 21). Main steps to follow ahead are:

- Edit the alignment if needed. Sequences can be renamed, added, deleted, etc.. in the upper menu of the multiple sequence alignment window (`Edit ->`).
- Send this alignment to *Modeller* by selecting `Structure -> Modeller (homology ...)` in the upper menu of the multiple sequence alignment window.
- Complete the new window opened for `Comparative Modeling with Modeller` (Fig. 103 (A)) with *target* sequence (1), *template* (2), Modeller license key (3) and Advanced options like the number of models retrieved by *Modeller* (4), as well as *model* inclusion of heteroatoms (5) or water molecules (6). By pressing `Apply` or `OK` (5) the computation starts without hiding or hiding this panel window, respectively. The status of the job can be checked in the lower left corner of *Chimera* graphics window.
- After a while a new panel window will show retrieved models of the `target` sequence (Fig. 103 (B)). Two statistics assess these models: `GA341`, statistical potentials derived-score, (1) and `zDOPE`, normalized Discrete Optimized Protein Energy, atomic distance depending-score (2). Reliable models show `GA341` values higher than 0.7, and negative `zDOPE` values correspond to better models. One of the retrieved models has to be selected. Selected model and the rest of models can be checked in *Chimera* `Favorites -> Model Panel`.
- Save the retrieved model selected according to the model number (`#n`) shown in *Chimera* `Favorites -> Model Panel` by writing in *Chimera* `Favorites -> Command Line`:
  `scipionwrite model #n`.

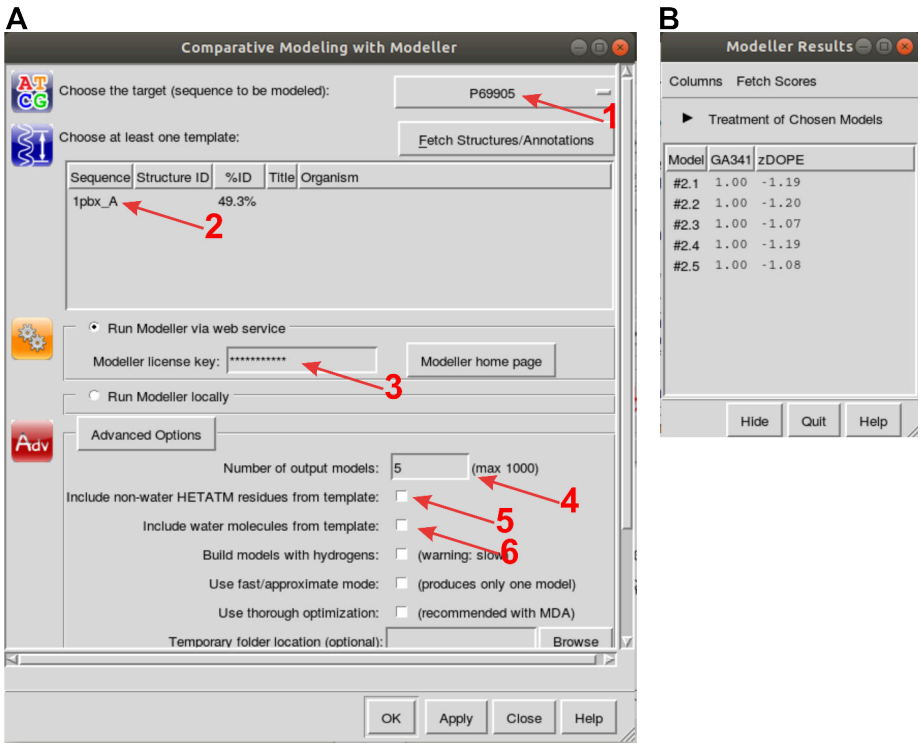


Figure 103: (A) Form to access to homology modeling with *Modeller*. (B) Panel window of *Modeller* retrieved models.

- Visualization of protocol results:

  After executing the protocol, press `Analyze Results` and *Chimera* graphics window will be opened by default. Atomic structures are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes and selected atomic structure `model` are model numbers `#0` and `#1`, respectively, in *Chimera* `Model Panel`.

- Summary content:

  – Protocol output (below *Scipion* framework):
  `scipion - model from template -> ouputPdb_01; 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:
chimeraOut0001.pdb
we have some result

Q 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-phenix`
  - PHENIX software suite (version 1.13-2998)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Validation` (Fig. 104 (A))
- Protocol form parameters (Fig. 104 (B)):

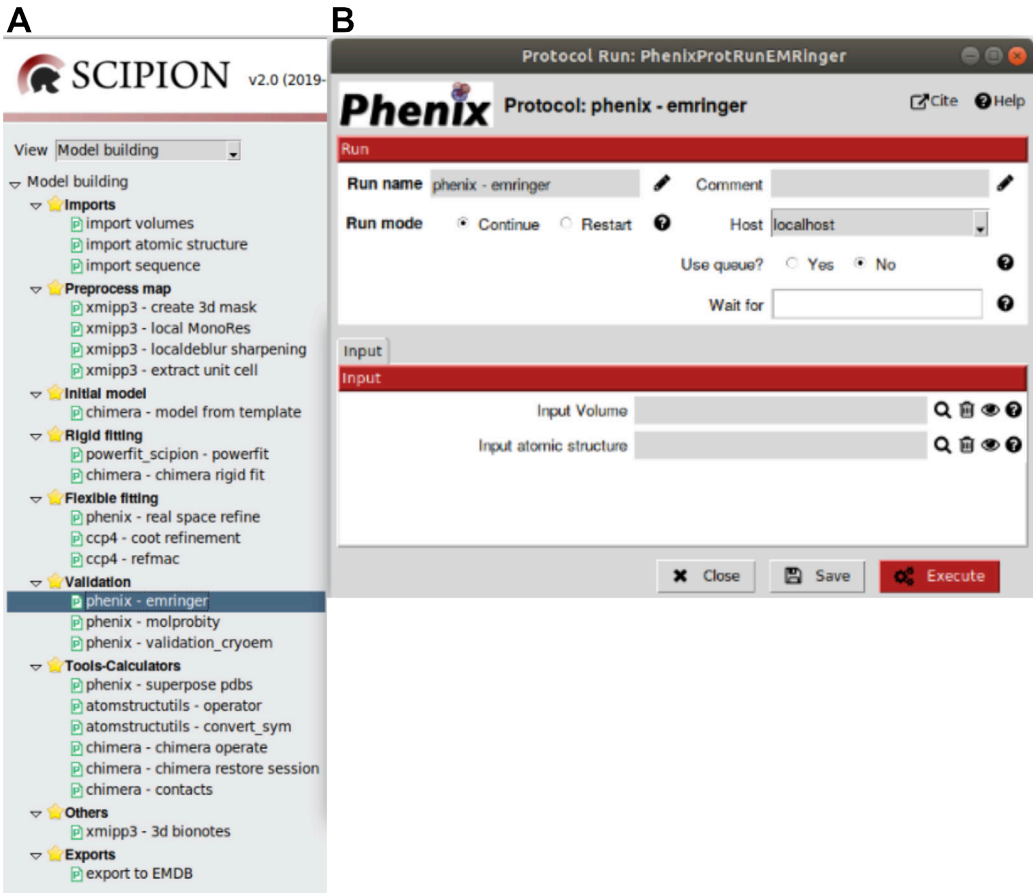


Figure 104: Protocol phenix - emringer. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `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. 105).

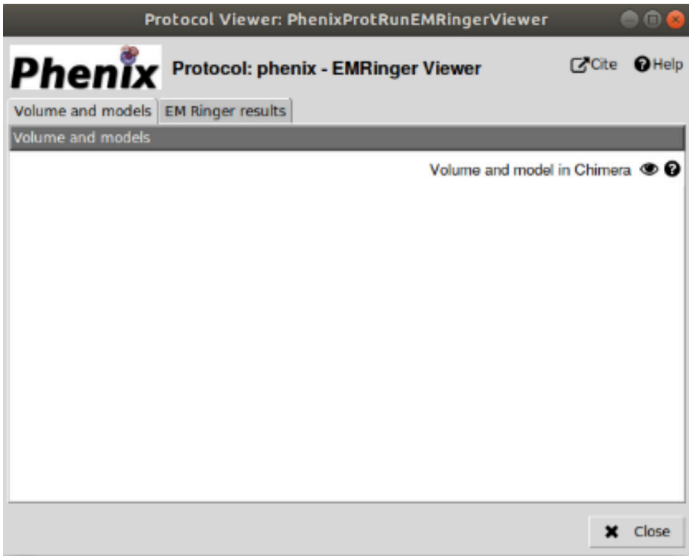


Figure 105: Protocol phenix - emringer. Taps to visualize Volume and models and *EMRinger* results.

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

  - `Volume and models`: *Chimera* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordinates in *Chimera*. 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. 65).
  - `EMRinger Results` (Fig. 106):

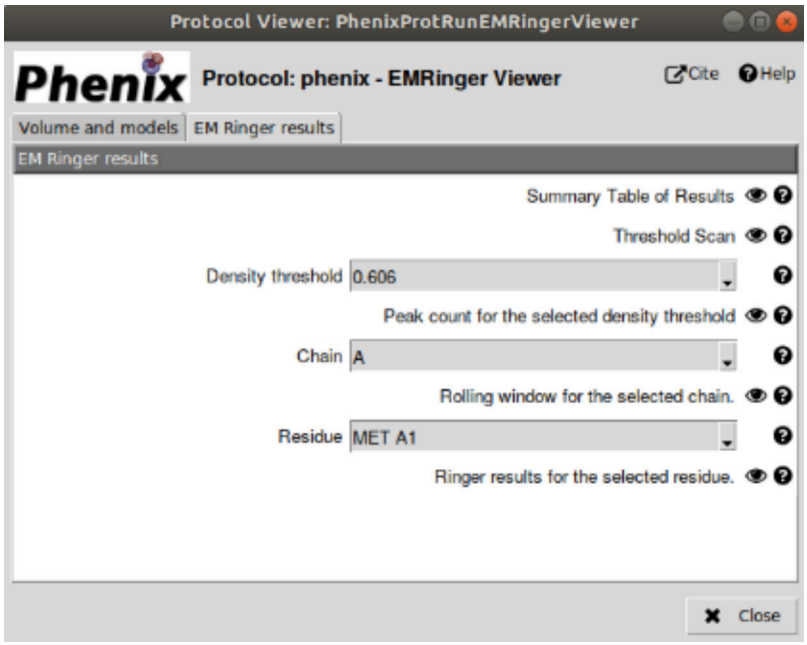


Figure 106: Protocol phenix - emringer. Menu to visualize *EMRinger* results.

∗ `Summary Table of Results` (Fig. 107):

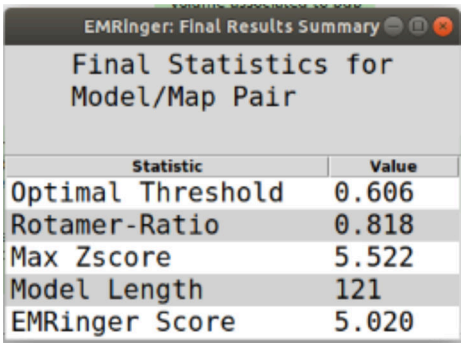


Figure 107: Protocol phenix - emringer. Final *EMRinger* results.

· `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 3.2-3.5Åresolution 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. 43 (6).

R 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-phenix`
  - PHENIX software suite
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu: `Model building -> Validation` (Fig. 108 (A))
- Protocol form parameters (Fig. 108 (B)):

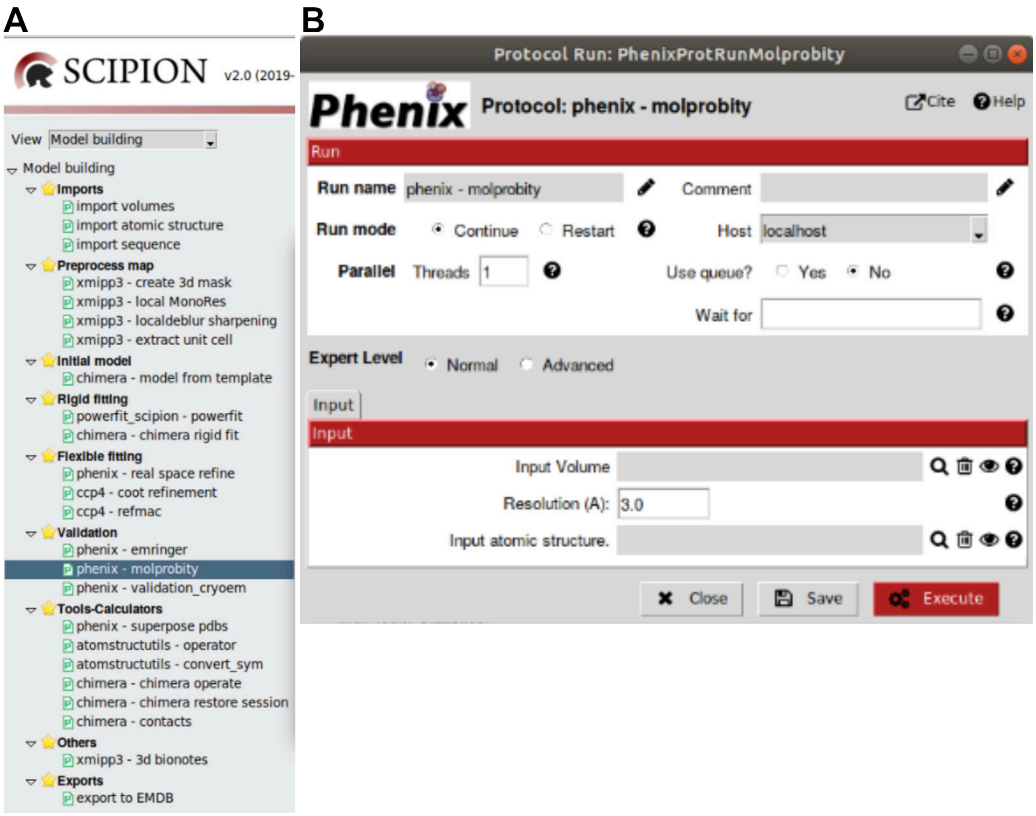


Figure 108: Protocol phenix - molprobity. A: Protocol location in *Scipion* menu. B: Protocol form.

- `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. 109).

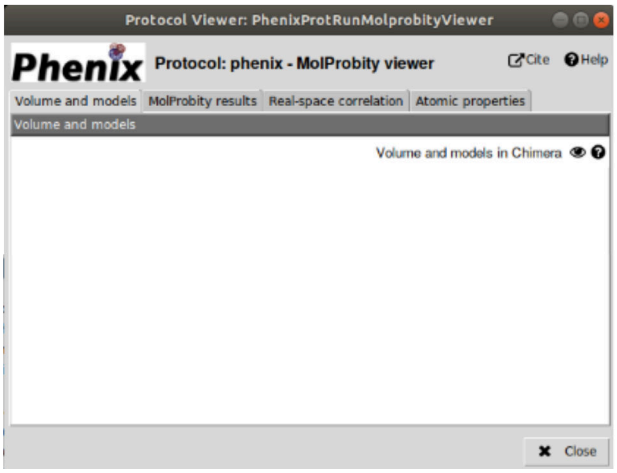


Figure 109: Protocol phenix - molprobity. Taps to visualize *MolProbity* and Real space correlation results. `Real-space correlation` tap only appears with *PHENIX* v. 1.13.

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

  - `Volume and models`: *Chimera* graphics window will be opened by default. Volume and atomic structure, if it is present, are referred to the origin of coordinates in *Chimera*. 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. 65).

– `MolProbity results` (Fig. 110):

Figure 110: 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 G). 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. 111):

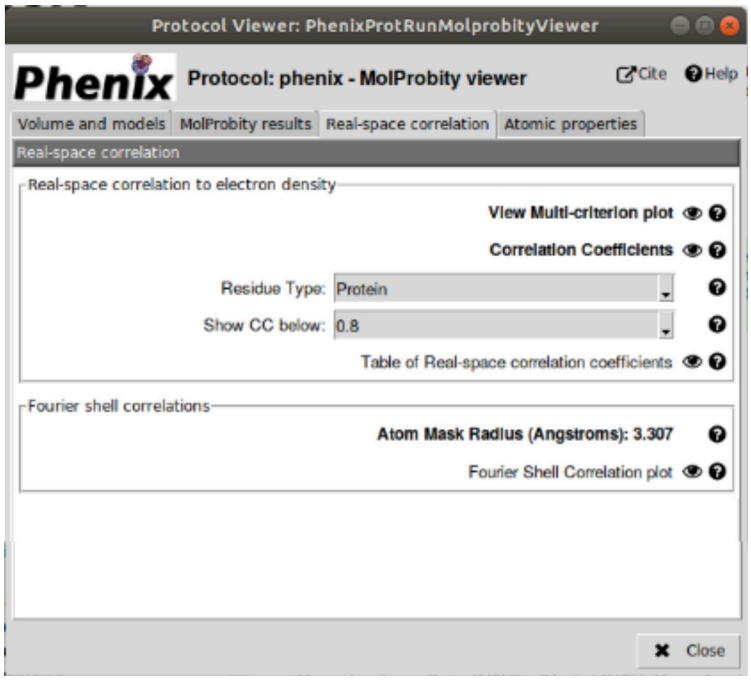


Figure 111: Protocol phenix - molprobity. Real-space correlation results.

* `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. 112): Atom numerical properties:

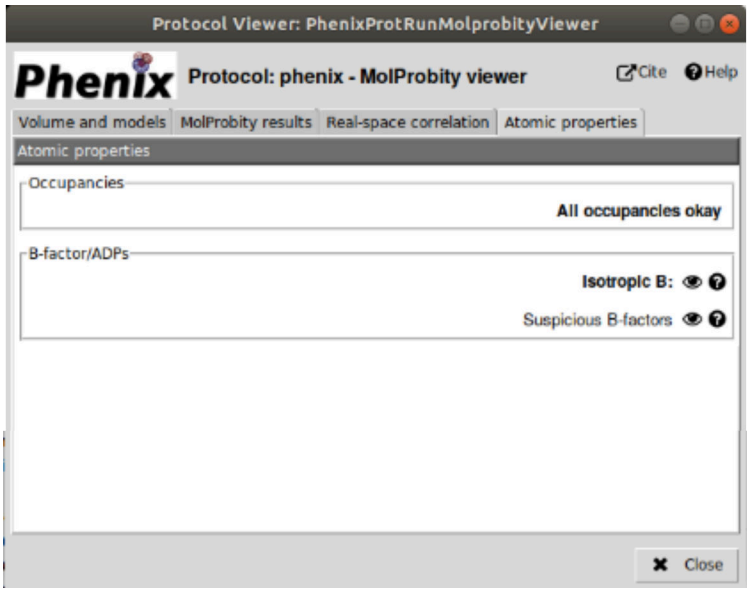


Figure 112: Protocol phenix - molprobity. Atomic properties results.

* `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. 44 (7)).

S 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-phenix`
  - PHENIX software suite (v. higher than 1.13)
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Validation` (Fig. 113 (A))
- Protocol form parameters (Fig. 113 (B)):

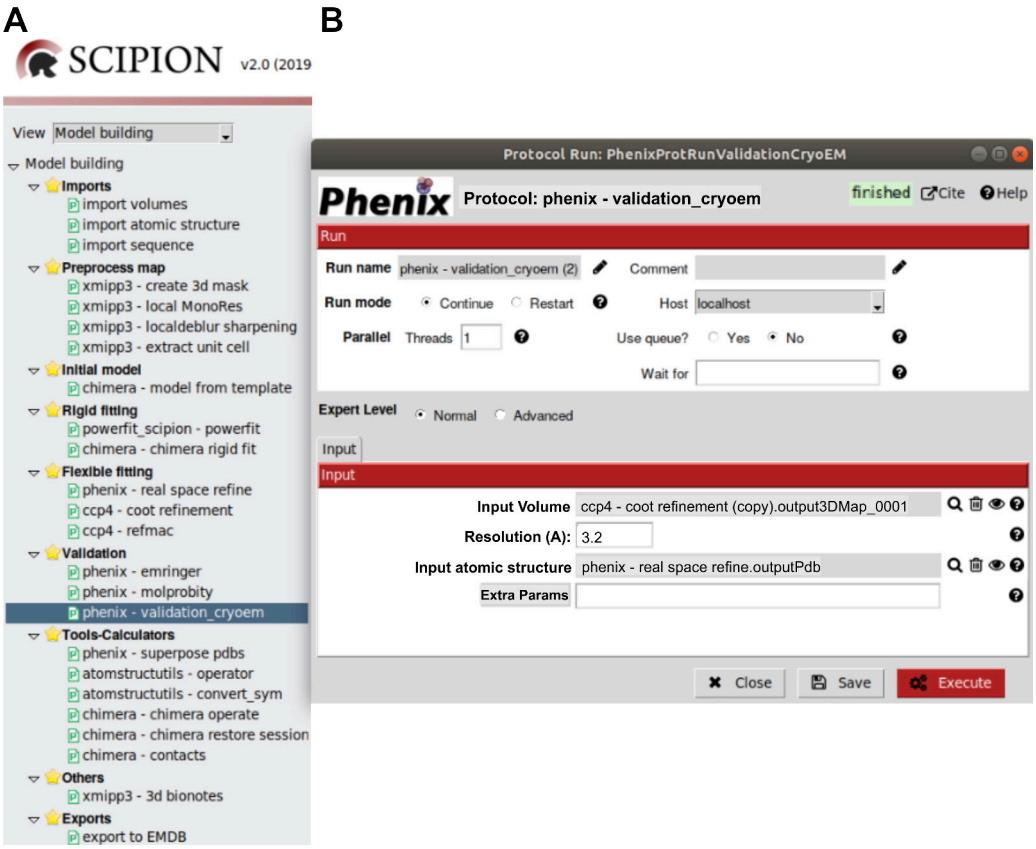


Figure 113: Protocol phenix - validation_cryoem. A: Protocol location in *Scipion* menu. B: Protocol form.

- `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. 114).

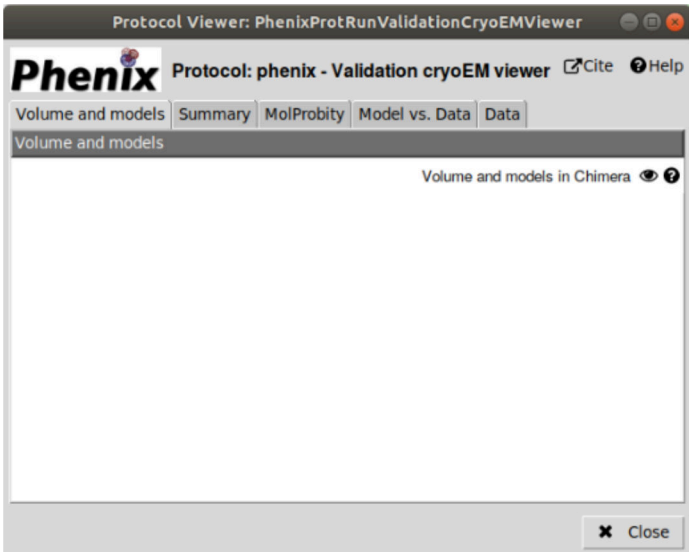


Figure 114: Protocol phenix - validation_cryoem. Taps to visualize *Validation CryoEM* results.

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

  - `Volume and models`: *Chimera* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordinates in *Chimera*. 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. 65).

– `Summary`: Three different summary tables are shown to describe the results obtained from `Model, Data` and `Model vs. Data` (Fig. 115). 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.

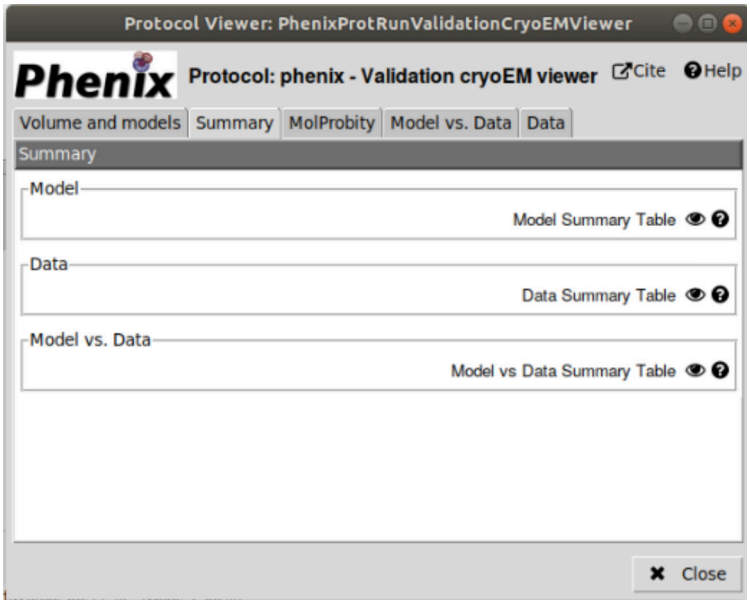


Figure 115: Protocol phenix - validation_cryoem. Summary tables of main*PHENIX* `validation_cryoem` results.

– `MolProbity`: Statistics concerning the atomic model, most of them obtained from *MolProbity* (Fig. 116).

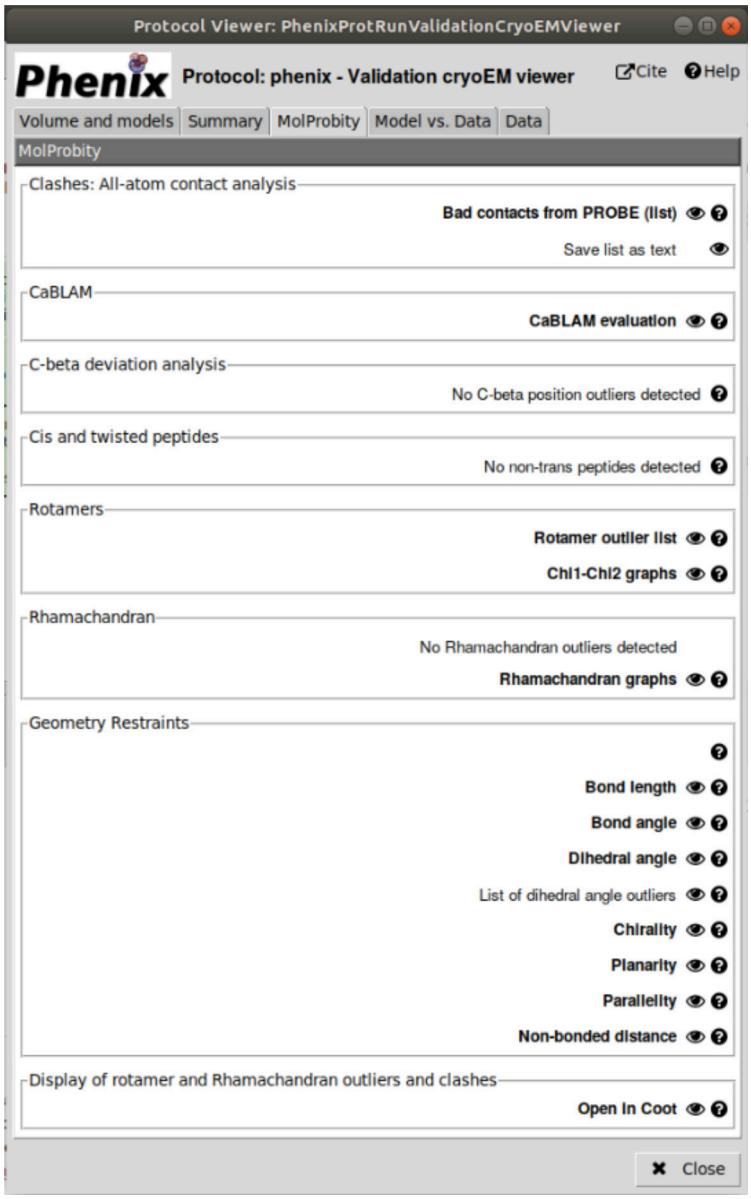


Figure 116: Protocol phenix - validation_cryoem. *MolProbity* and other statistics of the atomic model.

* `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. 117).

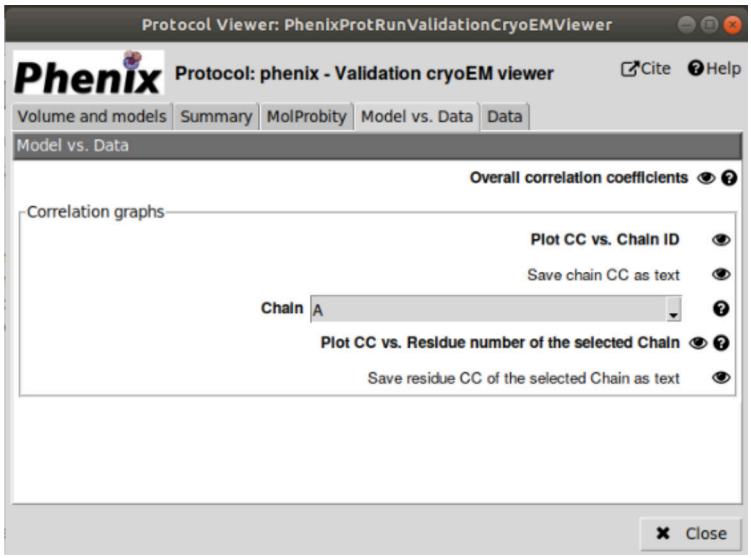


Figure 117: Protocol phenix - validation_cryoem. Real-space correlation results.

* `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 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 save 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 save in a text file in the folder selected by the user.

– `Data` (Fig. 118): Computation of Resolution and FSC.

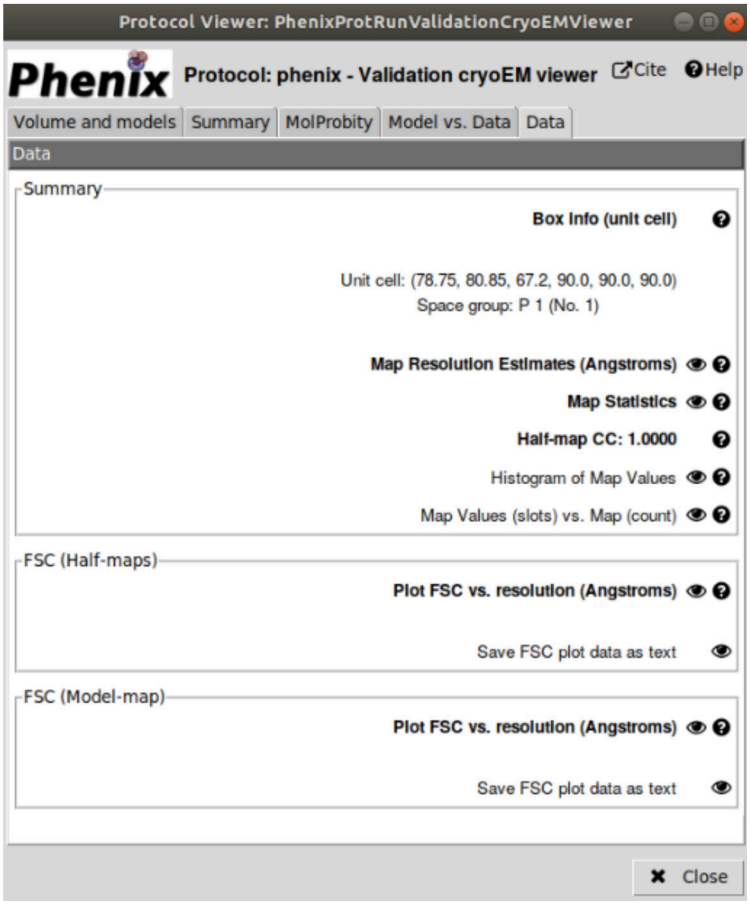


Figure 118: Protocol phenix - validation_cryoem. Experimental data results.

* `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 good 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 add 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`, 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.
  - `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.

T 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-phenix`
  - PHENIX software suite
  - *Scipion* plugin: `scipion-em-ccp4`
  - CCP4 software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Flexible fitting` (Fig. 119 (A))
- Protocol form parameters (Fig. 119 (B)):

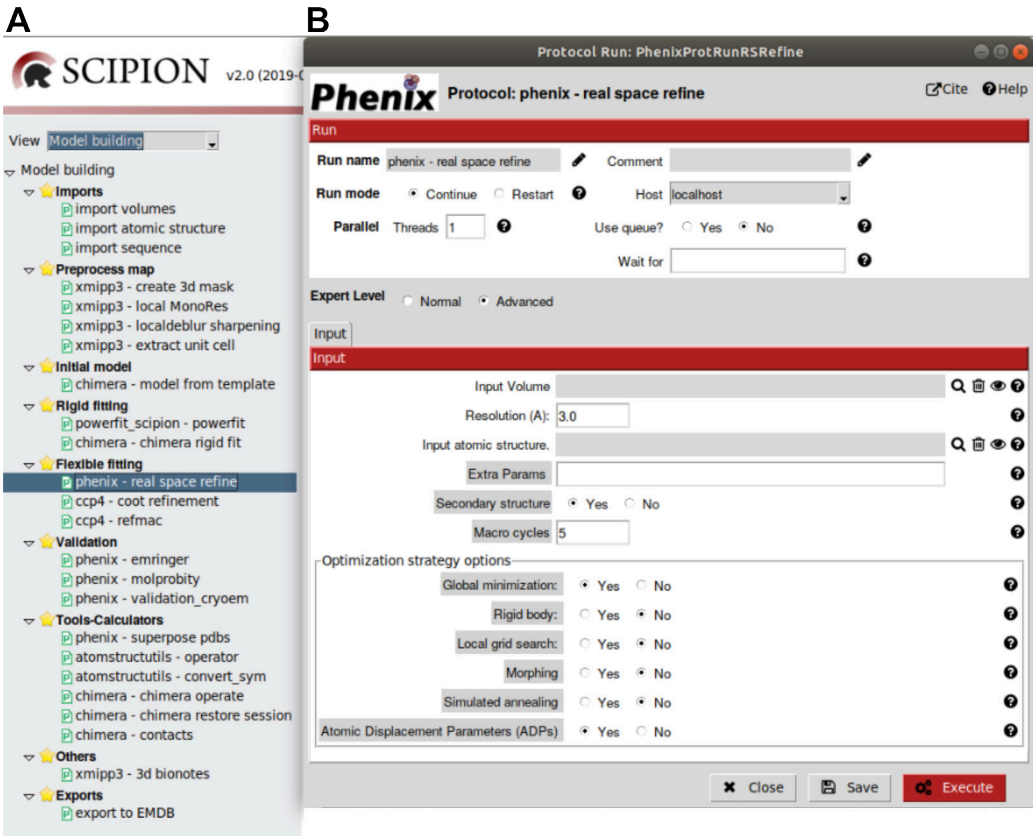


Figure 119: Protocol phenix - real space refine. A: Protocol location in *Scipion* menu. B: Protocol form.

- `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 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. 120).

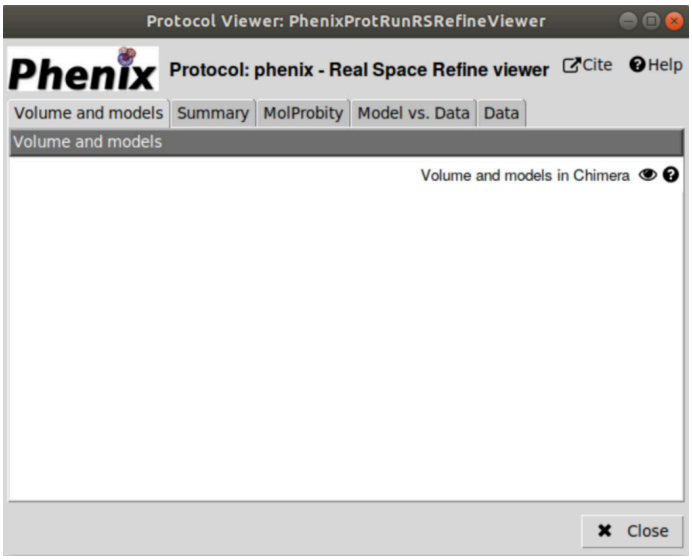


Figure 120: Protocol phenix - real space refine. Taps to visualize *Real Space Refine* results.

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. 109, Fig. 110, Fig. 111 and Fig. 112):

- `Volume and models`: *Chimera* graphics window will be opened by default. Atomic structure and volume are referred to the origin of coordinates in *Chimera*. 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. 65).
- `Summary`: Three different summary tables are shown to describe the results obtained from `Model`, `Data` and `Model vs. Data` (Fig. 121). 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 resolu-

tion computed with or without a mask. `Model vs. Data` details main real-space correlation coefficients.

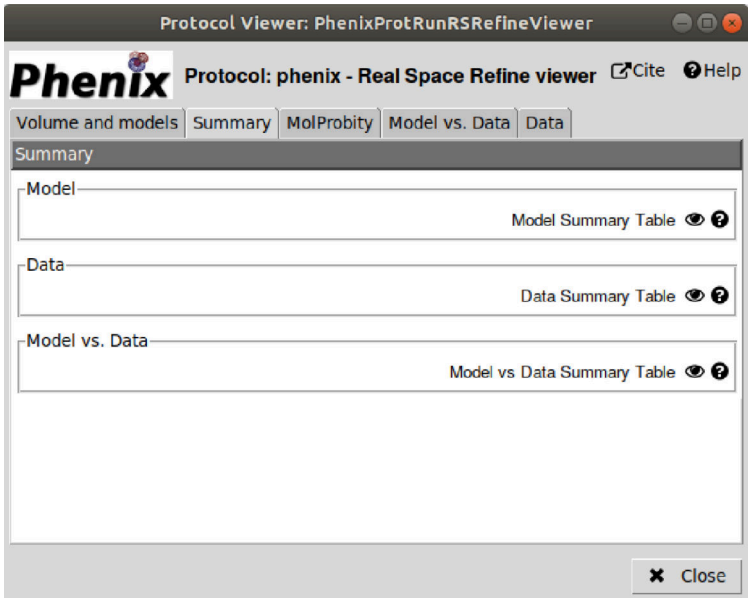


Figure 121: Protocol phenix - real space refine. Summary tables of main*PHENIX* `real space refine` results.

– `MolProbity`: Statistics concerning the atomic model, most of them obtained from *MolProbity* (Fig. 122).

Figure 122: 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. 123).

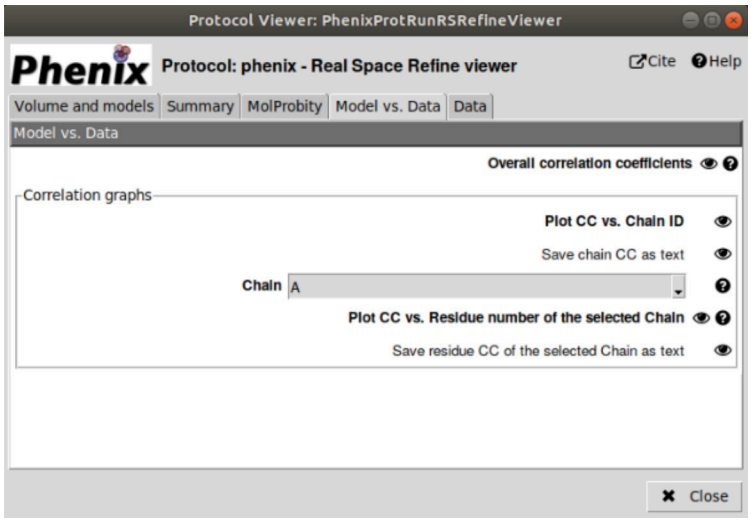


Figure 123: Protocol phenix - real space refine. Real-space correlation results.

* `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 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 save 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 save in a text file in the folder selected by the user.

– `Data` (Fig. 124): Computation of Resolution and FSC.

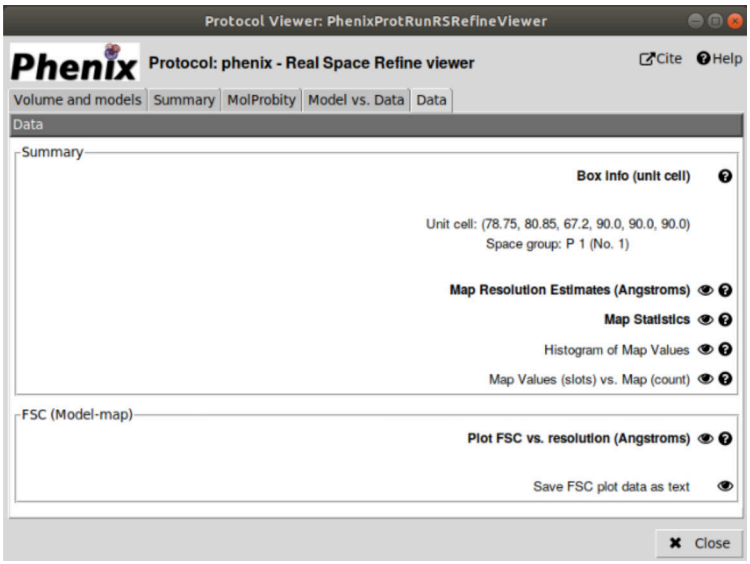


Figure 124: Protocol phenix - real space refine. Experimental data results.

* `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 good 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 add 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`, 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 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.

U 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-phenix`
  - PHENIX software suite
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Tools-Calculators` (Fig. 125 (A))
- Protocol form parameters (Fig. 125 (B)):

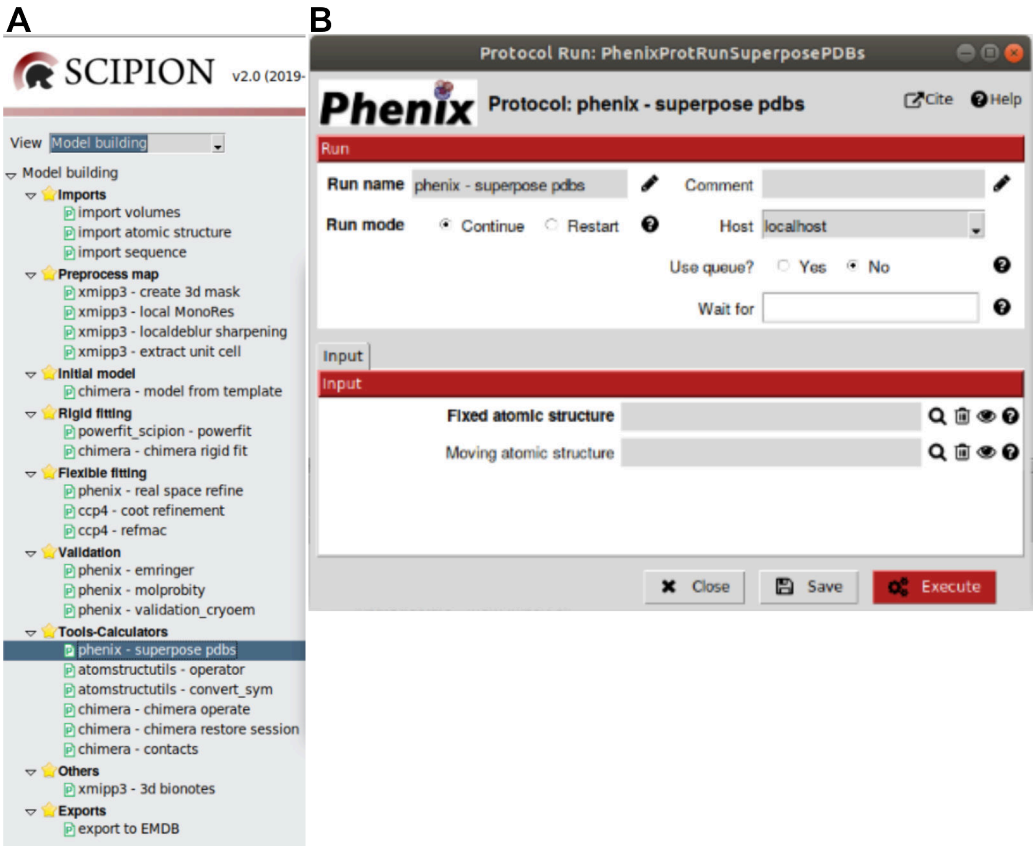


Figure 125: Protocol phenix - superpose pdbs. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `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 *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65).

- Summary content:

  `SUMMARY` box:
  RMSD between fixed and moving atoms (start and final values).

V Powerfit protocol

Protocol designed to automatically fit atomic structures to electron density maps in *Scipion* by using *PowerFit* ((Van Zundert and Bonvin, 2016)), application that performs a rigid body search based on cross-correlation between atomic structure and electron density map. You can follow additional instructions to run *PowerFit* in `http://www.bonvinlab.org/education/powerfit/`.

- Requirements to run this protocol and visualize results:
  - *Scipion* plugin: `scipion-em-powerfit`
  - *PowerFit* program (version 2.0.0)
  - *Scipion* plugin: `scipion-em-chimera`
- *Scipion* menu:
  `Model building -> Rigid fitting` (Fig. 126 (A))

- Protocol form parameters (Fig. 126 (B)):

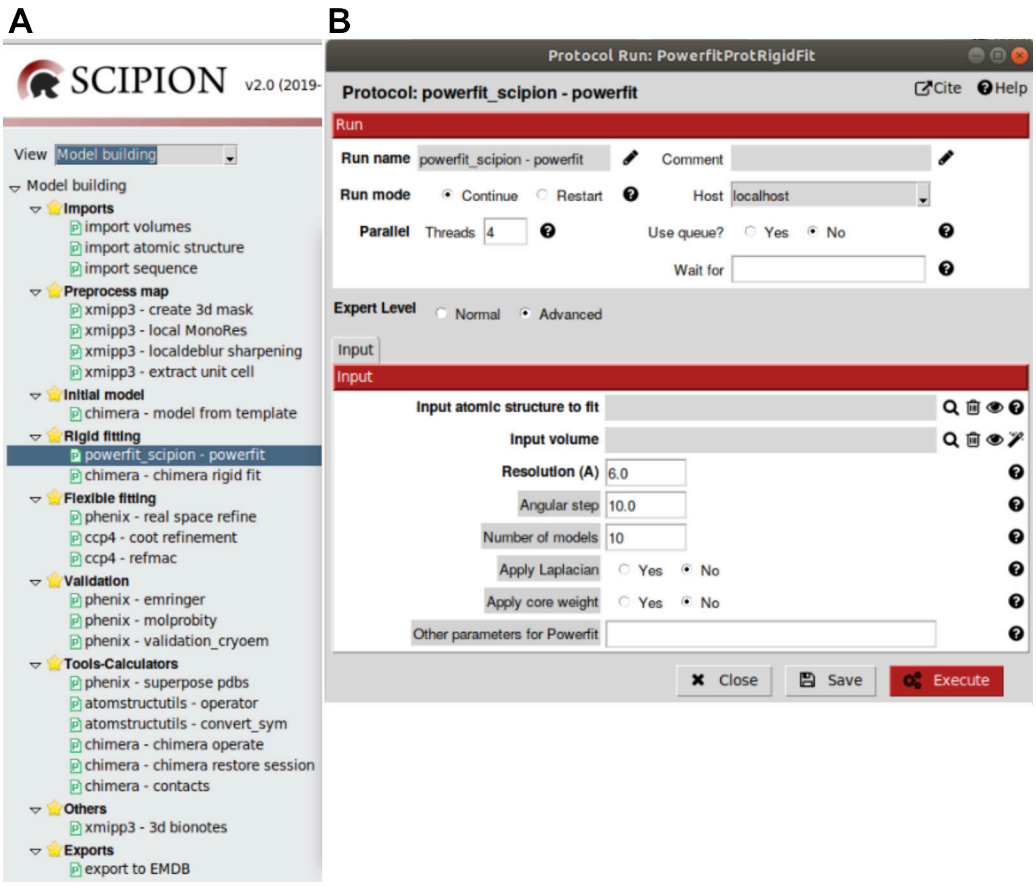


Figure 126: Protocol powerfit. A: Protocol location in *Scipion* menu. B: Protocol form.

  - `Input atomic structure to fit`: Atomic structure previously downloaded or generated in *Scipion* to be fitted to an electron density map.
  - `Input volume`: Electron density map previously downloaded or generated in *Scipion* to fit the atomic structure.
  - `Resolution (Å)`: Electron density map resolution.
  - `Angular step`: Advanced parameter to indicate rotational sampling interval (degrees). 10º is the default value. Lower values usually gener-

ate better fits because they allow more subtle searches in the 3D space. However, the process gets computationally more expensive. So, this parameter value should be carefully selected considering the whole size of the molecule. An estimation of *PowerFit* runtime can be checked here: `http://milou.science.uu.nl/cgi/services/POWERFIT/powerfit/`.

- `Number of models`: Advanced parameter to select the maximum number of fits generated, 10 by default. To avoid structure redundancy, a lower number of best fit structures than indicated might be shown by *PowerFit*.
- `Apply Laplacian`: Advanced parameter to apply a Laplacian pre-filter to the electron density map to enhance its edges and increase cross-correlation scores between atomic structure and electron density map. Parameter set to "No" by default.
- `Apply core weight`: Advanced parameter of local cross-correlation score, designed to bias the weight of electron density toward the core of the map, thus minimizing the effect of overlapping among neighboring subunits. Parameter set to "No" by default.
- `Other parameters for` *Powerfit*: Advanced parameter to include, for instance, the number of CPU available or if GPU is going to be used for computation.

- 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 a small window will be opened (Fig. 127). This window allows selecting between visualize fitting

quality scores and the fitting itself in *Chimera* for each non-redundant fit generated by *PowerFit*. The number inside `Model to visualize` box allows to select one specific fit.

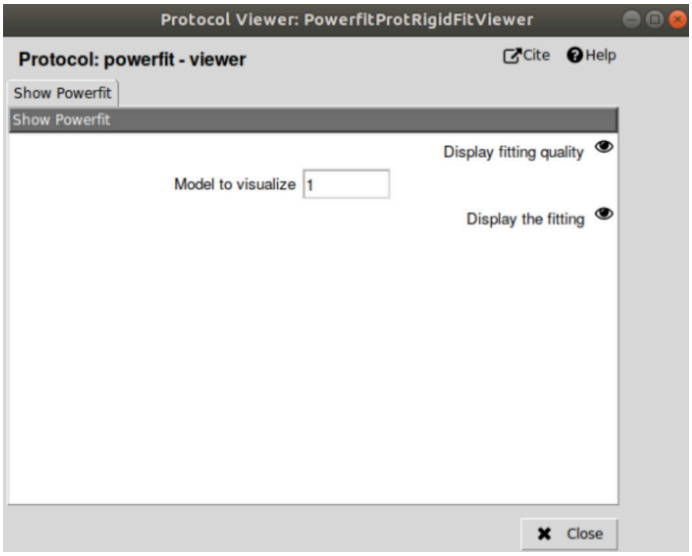


Figure 127: Protocol powerfit. Menu to visualize *PowerFit* results.

– `Display fitting quality`:

When this option is selected, a five-column table opens (Fig. 128). First and second columns of this table contains rank and file name of best fits generated by *PowerFit*. Cross correlation score between atomic structure and electron density map, Fisher Z-score, and the number of standard deviations are values included in `_powerfit_cc`, `_powerfit_Fish_z`, and `_powerfit_rel_z` columns, respectively.

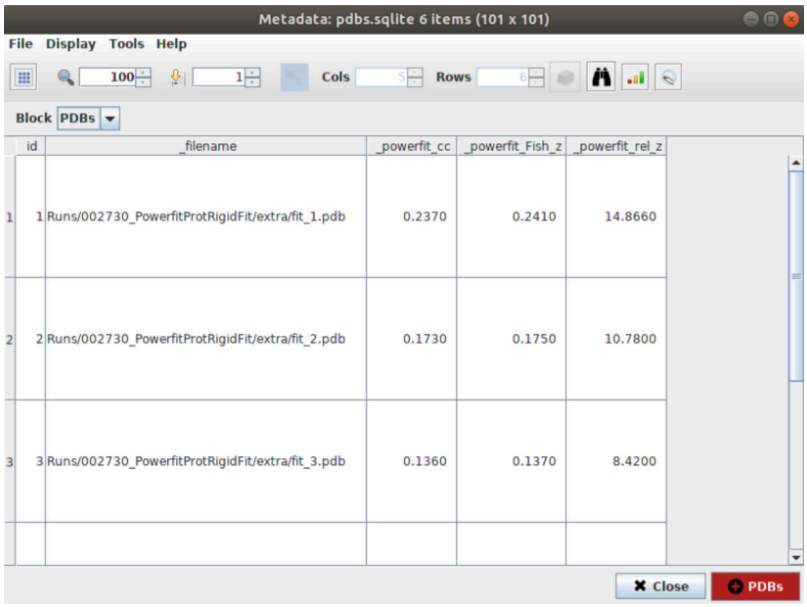


Figure 128: Protocol powerfit. Fitting quality scores of *PowerFit* results.

- `Display the fitting`:

  After executing the protocol, press `Analyze Results` and *Chimera* graphics window will be opened by default. Atomic structures and volumes are referred to the origin of coordinates in *Chimera*. 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. 65). Coordinate axes, volume, and each atomic structure are model numbers `#0`, `#1`, and `#3`, respectively, in *Chimera* `Model Panel` (Fig. 128). Model number `#2` corresponds to `lcc.mrc` volume, i.e., a cross-correlation density map that shows the highest cross-correlation found in each grid position, indicating the most likely location of the center of mass of the atomic structure.

- Summary content:

  - Protocol output (below *Scipion* framework):

`powerfit - powerfit -> ouputPDBs`; `SetOfPDBs (#n items)`.
n items indicates the number of non-redundant structures best fitted to the electron density map, found by *PowerFit*.

– `SUMMARY` box:
Angular step: 10.000000

W Submission to EMDB protocol

Protocol designed to save in a specified folder the three main files required to submit cryo-EM derived electron density maps and derived atomic structures to EMDB (`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. 129 (A))

- Protocol form parameters (Fig. 129 (B)):

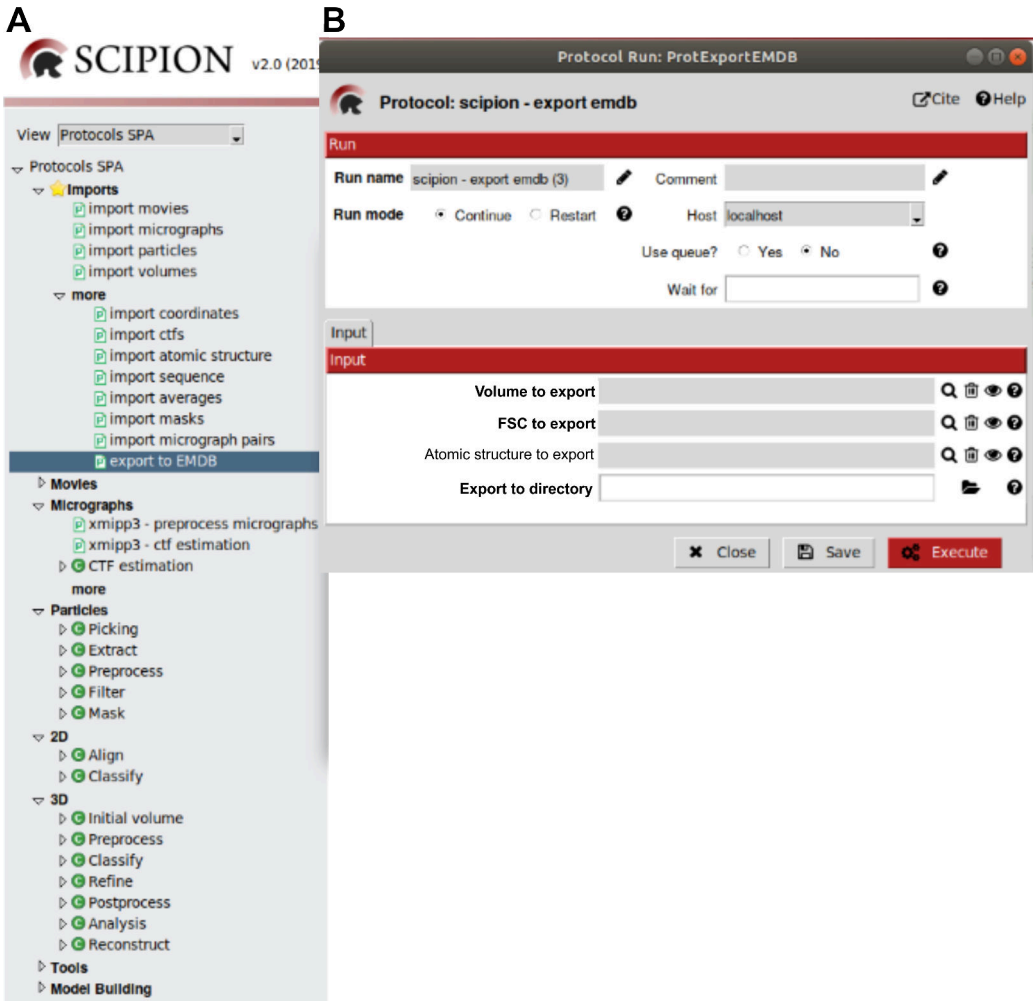


Figure 129: Protocol export to EMDB. A: Protocol location in *Scipion* menu. B: Protocol form.

- `Input` section
  - `Volume to export`: Param to select the electron density map previously downloaded or generated in *Scipion* that we would like to submit to EMDB. The map file will be saved with `.mrc` format.
  - `FSC to export`: Param to select the FSC file previously downloaded or generated in *Scipion* that we would like to submit to EMDB. This file will be saved with `.xml` 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.
  - `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.

  The three previously selected files will be saved in the chosen directory after executing the protocol and this can be checked by opening that folder. 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 three files:
  `Data available at:` *path*