

Integration of Cryo-EM Model Building Software in *Scipion*

M. Martínez, A. Jiménez-Moreno, D. Maluenda, E. Ramírez-Aportela, R. Melero, A. Cuervo, P. Conesa, L. del Caño, Y. C. Fonseca, R. Sánchez-García, D. Strelak, J. J. Conesa, E. Fernández-Giménez, F. de Isidro, C. O. S. Sorzano, J. M. Carazo, and R. Marabini*



Cite This: *J. Chem. Inf. Model.* 2020, 60, 2533–2540



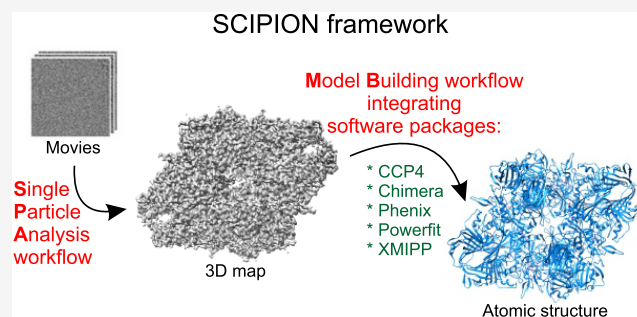
Read Online

ACCESS |

Metrics & More

Article Recommendations

ABSTRACT: Advances in cryo-electron microscopy (cryo-EM) have made it possible to obtain structures of large biological macromolecules at near-atomic resolution. This “resolution revolution” has encouraged the use and development of modeling tools able to produce high-quality atomic models from cryo-EM density maps. Unfortunately, many practical problems appear when combining different packages in the same processing workflow, which make difficult the use of these tools by non-experts and, therefore, reduce their utility. We present here a major extension of the image processing framework *Scipion* that provides inter-package integration in the model building area and full tracking of the complete workflow, from image processing to structure validation.



INTRODUCTION

Recent advances in the field of cryo-EM^{1,2} have increased the number of biomacromolecules that have been solved at near-atomic resolution. In fact, for large macromolecules (greater than 1 MDa) the cryo-EM-derived models now outnumber those produced by X-ray diffraction.³ Not surprisingly, those advances have boosted the interest in algorithms that can derive atomic models from cryo-EM data. Moreover, since many model building tasks require a lot of manual intervention by the user, proficiency programs that are able to automate and accelerate modeling work would be desirable.

The list of software packages related to model building is long, and since the field is growing, this list will surely increase. An exhaustive survey of the state-of-the-art in computational methods for constructing, refining, and validating atomic models can be found in the work by Cassidy et al.⁴ The user interface of these programs is very heterogeneous: while many of them require just a few input parameters, others are interactive and demand manual intervention (e.g., *Chimera*,⁵ *Coot*,⁶ and *Gorgon*⁷). Some of the packages address a specific aspect of modeling (e.g., *Situs*⁸ focuses on rigid fitting, and *EMRinger*⁹ and *MolProbity*¹⁰ on validation), and others are suites that encompass several methods. Specifically, there are two suites with the vocation of including a variety of modeling tools from cryo-EM data: *CCP-EM*¹¹ and *Phenix*.¹²

Because cryo-EM maps present particular challenges³ and there is no algorithm that performs flawlessly in all situations, users may be interested in combining programs implemented in different packages to optimize their processing workflow. This combination is an error-prone process since different

packages follow different formats and conventions, which are not always carefully described. We present in this work a major extension of *Scipion*^{13–18} that provides a unified graphical user interface (GUI) on top of different model building tools and assures package inter-operability, user interface standardization, and monitoring of the workflow.

As many readers may not be familiar with *Scipion*, in the next section we will describe briefly its main features. Second, we will focus on the capabilities added to *Scipion* in order to produce high-quality atomic models. Finally, we will compare *Scipion* with other alternative software tools such as *CCP-EM* and *Phenix*.

SCIPION AS AN IMAGE PROCESSING FRAMEWORK

Scipion was originally developed to address the issues of integration and inter-operability between the different image processing packages available in the field of three-dimensional electron microscopy (3DEM) (see <http://scipion.i2pc.es> for further details). In most cases, image processing packages comprise hundreds of small programs that perform specific tasks and have well-defined inputs and outputs. The *Scipion* building block is the *protocol*, which encapsulates each one of these small programs (or concatenation of a few of them),

Special Issue: Frontiers in Cryo-EM Modeling

Received: November 6, 2019

Published: January 29, 2020



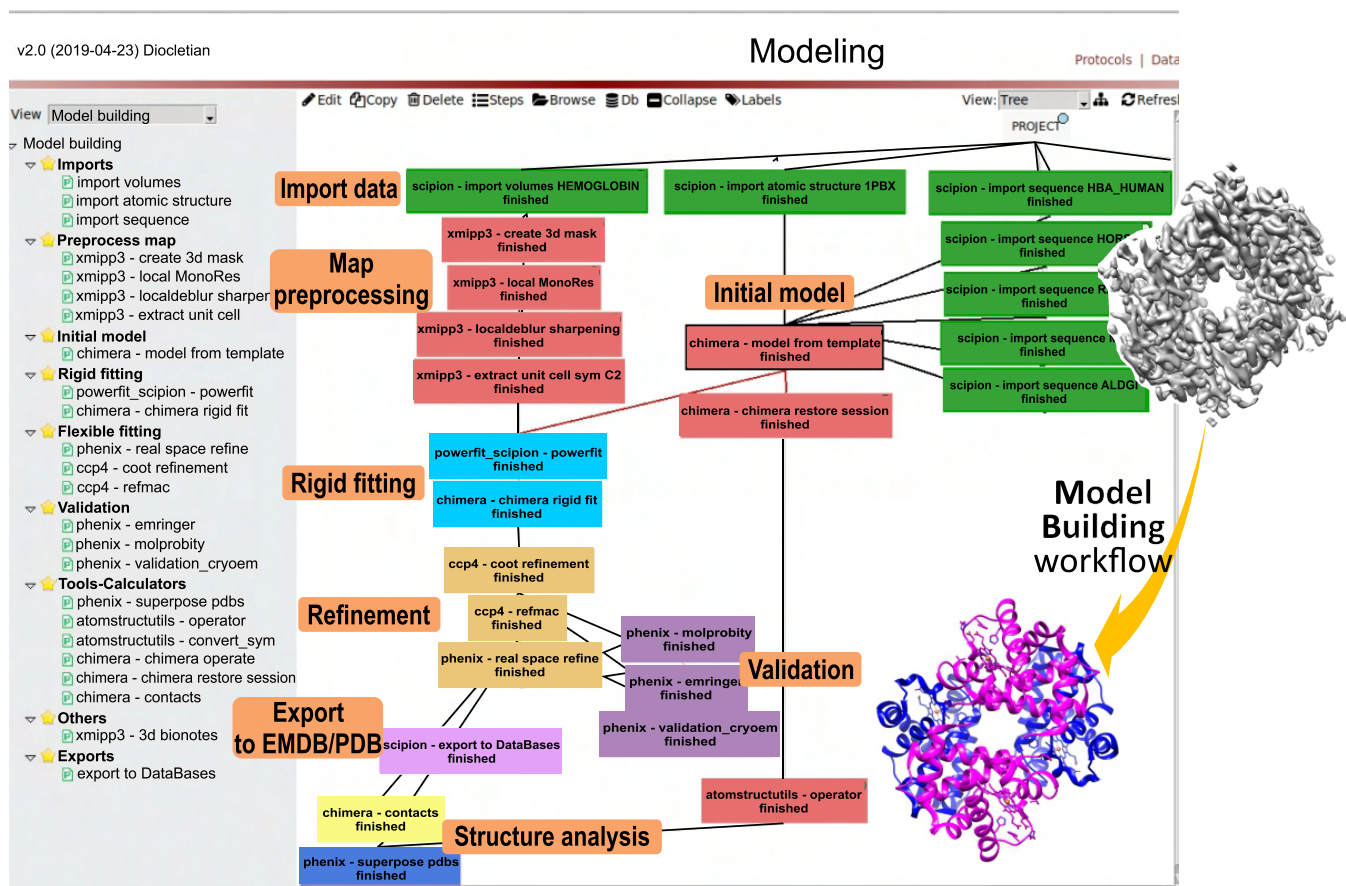


Figure 1. *Scipion* GUI showing the context-sensitive menu (left) and workflow tree (right). Orange labels with bold font identify the set of protocols that implement each step in the model building workflow. This workflow was applied to the human hemoglobin map (EMDB ID: EMD-3488) as described in the tutorial (<https://scipion-em.github.io/docs/docs/user/user-documentation.html#tutorials>). The workflow is available in the *Scipion* workflow repository (<http://workflows.scipion.i2pc.es/>).

handles different formats and conventions, and stores metadata information in a database providing full tracking of the entire image processing workflow.

The graphical interface of *Scipion* allows the creation of user-defined workflows by concatenation of protocols. An example of chained protocols can be observed in Figure 1 in the context of model building. More than 300 protocols have been created since the original publication of *Scipion*. These protocols cover the different image processing steps in a typical 3DEM workflow. On top of these protocols, several *metaprotocols* have been built. The first type of *metaprotocols* are called monitors and are used to produce live analysis reports or raise alerts. For example, **scipion - ctf monitor** checks and plots the computed micrograph defocus values and may raise an alert if these defocus values are not within a predefined range. A second set of *metaprotocols* check the consistency of the output datasets generated from the same input data using different algorithms. For example, **xmipp3 - picking consensus** will compare the particles selected by each executed particle-picking protocol and produce as output a set containing those particles selected by most of the algorithms.

Another interesting *Scipion* capability is streaming, i.e., computing on data directly as they are produced or received. A streaming-capable protocol **b** does not need to wait until another protocol **a** finishes before using the output of **a** as an input. That is, if the output is a set of images, as soon as the

first image is produced by **a**, **b** can process it. In this way, the data continuously flow through the workflow of protocols.

Scipion is a widely used tool in the 3DEM image processing community. Since its publication in 2016, the *Scipion* original manuscript¹³ has been cited more than 100 times and the software has been downloaded more than 4000 times (see <http://scipion.i2pc.es/downloadstats> for detailed download statistics). A usage map and a ranking with the more used protocols is available at the *Scipion* Web site (see http://scipion.i2pc.es/report_protocols/scipionUsage and http://scipion.i2pc.es/report_protocols/protocolTable/, respectively). *Scipion* streaming capabilities have attracted interest in several 3DEM centralized facilities. Currently, it has been installed in 11 of these facilities and it is under consideration in many more (see the bottom of page <http://scipion.i2pc.es/facilities> for a map showing all the facilities in which *Scipion* is working or under consideration). Finally, a considerable effort has been made on the dissemination of *Scipion* among the 3DEM community through courses and workshops. Since 2016, 37 *Scipion*-based workshops have been held all around the world (see an exhaustive list at <https://github.com/I2PC/scipion/wiki/News-and-events>).

In this context, it seemed appropriate to extend *Scipion* with the concluding steps, essential to complete the 3DEM pipeline, which include model building, structure validation, and database deposition.

MODEL BUILDING WORKFLOW IN SCIPION

Since the atomic interpretation of biomacromolecules is the final aim of high-resolution cryo-EM, we are implementing in *Scipion* different modeling workflows. A summary of basic tasks in a standard modeling workflow is shown in the schema of Figure 2.

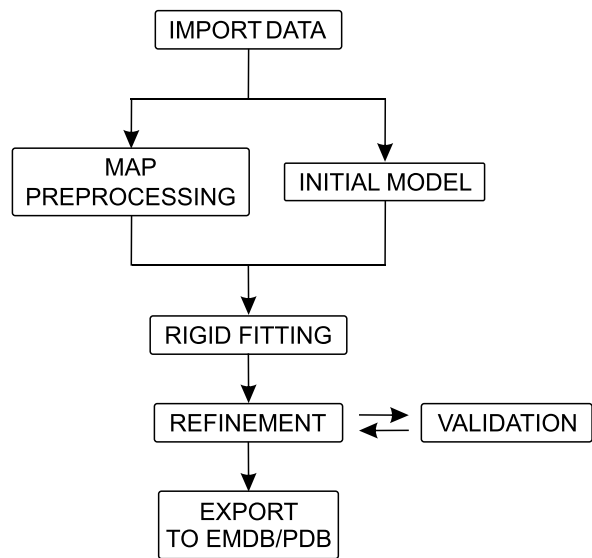


Figure 2. Schematic summary of a typical modeling workflow.

A 3DEM map and the sequence of each one of the structural components of a biomacromolecule are the inputs of the modeling workflow and need to be imported at the beginning of the workflow. The next step splits the workflow in two branches: the first one pre-processes the experimental map, in order to make model building easier, and the second one produces the initial model of each one of the structures, either *de novo* or by a prediction based on structures of homologous sequences. The third step connects again the two branches by fitting the rigid initial atomic models to the 3DEM map. Iterative steps of refinement and validation after the rigid fitting will adapt the skeleton and side chains of the structures to the 3DEM map, until desired scores of validation are obtained. Finally, the so-obtained atomic model is ready to be submitted to the Protein Data Bank (PDB) and the corresponding 3D map to the Electron Microscopy Data Bank (EMDB).

To perform the different tasks of the model building workflow, we have equipped *Scipion* with several new protocols, which internally work with different packages (see Table 1). As mentioned in the previous section, *Scipion* processing workflows are built by selecting algorithms from different packages and assigning the outputs of these algorithms (protocols) as inputs of another one. Each protocol can be selected in the *Scipion* menu, shown on the left in Figure 1. After the protocol is selected, a pop-up window with a form appears in which the different parameters required may be filled in. After the form is completed, the protocol will be executed and a new box will be inserted into the *Scipion* GUI. As an example, the set of protocol boxes generated in the processing of the human metalloprotein hemoglobin in its non-functional ferric status (metHgb) can be observed in Figure 1. To illustrate *Scipion*'s capabilities, in the following we detail the

Table 1. Summary of the Model Building Algorithms Offered by *Scipion*^a

protocol/package	function
extract unit cell/ <i>XMIPP3</i>	extracts an asymmetric unit from a 3D map
model from template/ <i>Chimera</i>	models 3D structures using Modeller
powerfit/ <i>Powerfit</i>	automatic rigid body fitting
rigid fit/ <i>Chimera</i>	semiautomatic rigid body fitting
coot/ <i>CCP4-Coot</i>	interactive refinement
refmac/ <i>CCP4-Refmac</i>	automatic refinement
real-space refine/ <i>Phenix</i>	automatic refinement
emringer/ <i>Phenix</i>	validation
molprobtity/ <i>Phenix</i>	validation
validation cryoem/ <i>Phenix</i>	validation
contacts/ <i>Chimera</i>	identifies interatomic contacts
superposePDBs/ <i>Phenix</i>	matches two atomic structures

^aThe first column lists the name given in *Scipion* to each offered algorithm followed by the package used to implement it. The second column contains a brief description of the implemented functionality. Auxiliary functions used for atomic structure file manipulation as well as standard image processing tools in single particles are omitted.

metHgb modeling workflow, which yields an atomic model based on sequence homology.

Hemoglobin metHgb is composed by two copies of two different subunits (called α and β) and presents 2-fold symmetry (cyclic group of order 2, also known as C₂). In this example we model the α subunit of the human metHgb (UnitProt ID: P69905) and assume that we have (1) generated by cryo-EM a 3D map and (2) identified as a suitable reference the atomic structure of the hemoglobin chain α of the antarctic fish *Pseudotrematomus bernacchii* (PDB ID: 1PBX). We start the modeling workflow by importing the cryo-EM map, the reference atomic structure, and the human sequence, using the protocols **scipion - import volume**, **scipion - import atomic structure**, and **scipion - import sequence**, respectively (upper green part of Figure 1). In general, importing data is a non-trivial step because (a) most 3D reconstruction algorithms produce 3D maps with inconsistent headers, i.e., they assume that the center of the coordinates is in the middle of the map but neglect to annotate this information in the file header, and (b) PDB/MMCIF formats are complex and some programs are not able to handle the atomic structure files produced by others. In order to minimize these problems, (a) **scipion - import volume** stores explicitly the default origin of coordinates if the user does not provide an alternative value, and (b) whenever **scipion - import atomic structure** reads an atomic structure from a PDB format file, the corresponding structure is curated using the *MAXIT* suite¹⁹ and saved using MMCIF format. See [File Format Compatibility](#) for a discussion regarding how we achieve compatibility among the atomic structure files created by different packages.

At this point the workflow is divided in two branches, one of which pre-processes the 3D map and the other one gets the initial atomic structure, as was indicated in Figure 2. Protocols involved in these two branches are shown in red in Figure 1.

In the first branch we perform the pre-processing of the 3D map in order to facilitate the next model building tasks. Two consecutive actions can be distinguished in this branch. The first one, directed to increase the detectability of the different features, involves (a) creation of a 3D mask (e.g., using **xmipp3 - create 3d mask**²⁰ or **relion - create**

3d mask²¹) to differentiate signal from noise, (b) estimation of the local resolution (e.g., using **xmipp3 - local MonoRes**,²² **resmap - local resolution**,²³ **bsoft - blocres**,²⁴ or **xmipp3 - deepres**²⁵), and finally, (c) restoration of the 3D map by local deblurring using a set of kernels derived from the previous protocol (**xmipp3 - localdeblur sharpening**²⁶). The second action of pre-processing takes advantage of the molecule's symmetry by modeling the smallest asymmetric unit. From this asymmetric cell, the whole macromolecule's atomic structure can be eventually restored by applying the corresponding symmetry operators. It should be noted that *Scipion* keeps track of the origin of coordinates of any 3D map created by the different protocols. Therefore, if we are able to fit the atomic structure in the asymmetric unit map, the atomic structure will also fit in the original 3D map.

With the aim of extracting the non-redundant part of the 3D map, we designed the protocol **xmipp3 - extract unit cell**. This protocol makes it possible to reduce the map size, optimizing computational resources and facilitating the modeling process, which makes it especially powerful when processing large macromolecules with a high order of symmetry, such as icosahedral viruses. Among the different parameters that have to be set in the protocol **xmipp3 - extract unit cell**, two of them, relevant to select the part of the macromolecule in which the whole structure of a specific chain can be completely fitted, are illustrated in **Figure 3**: offset and expand factor. The offset makes it

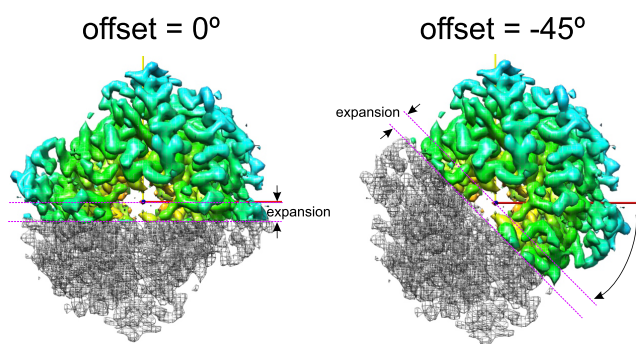


Figure 3. Results of applying **xmipp3 - extract unit cell** to the human metHgb for two different values of the parameter offset (0° left and -45° right). The asymmetric units are shown using shades of green, and the whole human metHgb is shown with a gray mesh centered on the origin of the coordinate axes (*X*, *Y*, and *Z* are drawn in red, yellow, and blue, respectively, with *Z* pointing toward the reader). In both cases the expand factor is 0.2.

possible to rotate the asymmetric unit a certain angle around the *Z* axis, and the **expand** factor supports the enlargement or expansion of the asymmetric unit. For specimens with high-order symmetry, as icosahedral viruses, several pre-defined orientations are available that place the symmetry axis in four different standard orientations: I222, I222r, I_n25, and I₂Sr (see <https://scipion-em.github.io/docs/docs/developer/symmetries> for details). In addition to the constraint imposed by the symmetry, users may apply a spherical mask defined by a minimum and a maximum radius. In this way, some areas, such as the virus genome, may be filtered out, reducing the size of the output box. We note here that **extract_unit_cell** computes asymmetric unit cells that are mathematically correct but may not be

meaningful from a biological point of view. Therefore, users may need to try several “expand factors” until a region that contains a proper asymmetric unit is produced.

In the second branch, a initial guess of the atomic structure that we wish to trace is created with the protocol **chimera - model from template**. This protocol uses as input an alignment of the human metHgb α subunit and the sequence of the reference atomic structure (*P. bernacchii* hemoglobin). This protocol provides an interface for running the program *Modeller* via a web service hosted by the UCSF-RBVI²⁷ and produces as result several initial structure models of the target sequence. Additional sequences might be imported to improve the quality of the alignment, as we show in **Figure 1** (green boxes on the right).

The selected model, as well as the 3D map, can be made available to *Scipion* using two extra commands that we have added to *Chimera*: **scipionwrite** and **scipionss** (“ss” is the abbreviation for “save session”). **scipionwrite** saves a model to disk and informs *Scipion* that a new model has been created. When *Scipion* receives this information, it registers the new model in a centralized database and makes it available to other protocols as input. **scipionss** saves a *Chimera* session in a way that can be recovered with the protocol **chimera - restore session** for further use. Remark that the use of these extra *Chimera* commands allows the tracking in *Scipion* of results obtained from an interactive program.

In the next step we combine the results obtained from the two previously described branches of the workflow by rigid fitting of the first estimation of the atomic model into the asymmetric unit map (**Figure 2**). This task can be achieved automatically with **scipion_powerfit - powerfit**²⁸ and/or manually with **chimera - rigid fit** (blue color in **Figure 1**).

At this point, the initial atomic model should be refined using either an automatic refinement protocol, such as **phenix - real space refine** or **ccp4 - reffmac**, or an interactive one, like **ccp4 - coot** (light brown color in **Figure 1**). This step is usually iterative (several refinement steps are needed) and involves validation protocols that help to detect those regions of the model that require a more careful revision. At present, three validation protocols are available: **phenix - molprobtity**, **phenix - emringer**, and **phenix - validation cryoEM** (purple color in **Figure 1**). As in the case of *Chimera*-derived protocols, we have added the **scipion_write** command to *Coot* so that users may save structures modeled in this interactive program and incorporate them into the *Scipion* tracking system. Unlike most of the *Scipion* protocols, *Coot* runs by default in the interactive mode. This means that we can open the protocol, save a partially refined structure, and exit without finishing the execution of the protocol. If we re-launch the protocol, the last saved atomic structure will be automatically opened, and we can continue the refinement.

The implementation of each protocol in *Scipion* includes an **Analyze Results** function that opens an analyzer GUI from which a collection of images, tables, plots, etc., can be displayed and used to explore the protocol result. Considerable effort has been invested in developing the offered analysis to make it comprehensive but not overwhelming. An example of the analyzer GUI, for the program *Reffmac*, can be seen in **Figure 4**. Pressing the button **Analyze Results** (**Figure 4A-b**) opens up the GUI (**Figure 4B**). This panel allows users

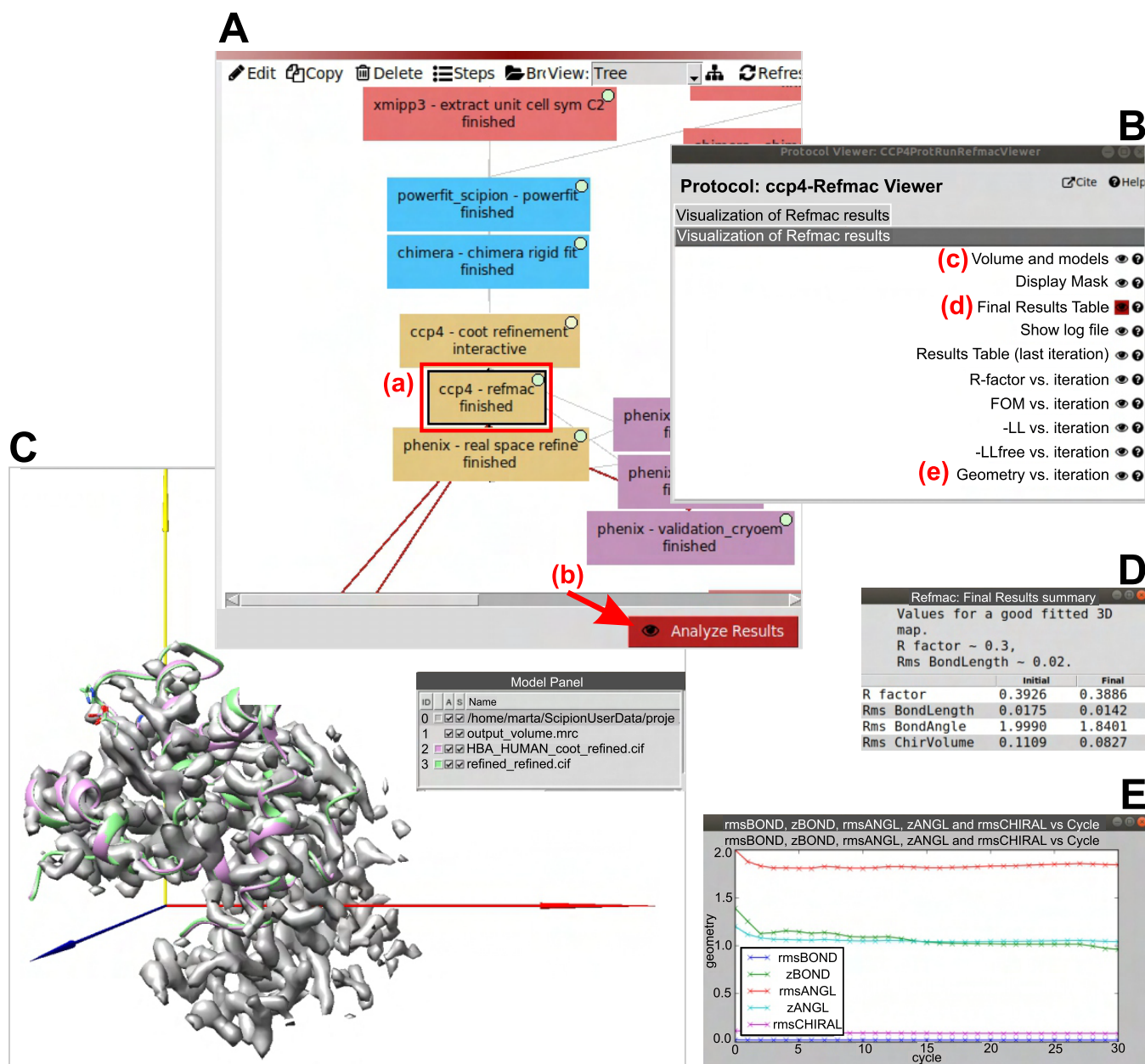


Figure 4. Analyze Results function detailed for **ccp4 - reffmac** protocol. (A) *Scipion* GUI showing part of the model building workflow: (a) red box highlighting the selected protocol **ccp4 - reffmac**, and (b) button used to launch the results panel. (B) Results panel. By selecting (c), (d) and (e), the panels showing the *Chimera* GUI (C), the final result summary (D), and the plot of geometry estimates (E), respectively, will open.

to select among several options. The first option, Volume and models (Figure 4B-c), initiates a *Chimera* GUI displaying four elements: coordinate axes, 3D map, input atom structure, and output refined structure (Figure 4C). On selecting Final Results Table (Figure 4B-d), a summary of results is shown (Figure 4D). Finally, the selection of Geometry vs. iteration (Figure 4B-e) displays a plot showing several statistics of the structure geometry (Figure 4E).

Finally, the protocol **scipion - export to Data-Bases** (pink color in Figure 1) will save in a directory, using the appropriate format, all the files needed to upload the atomic structure to PDB/EMDB: the 3D map, the atomic

model, a Fourier shell correlation (FSC) curve, a 3D mask, and the two “half” reconstructions used for the FSC estimation.

Once the atomic model is refined, the interpretation begins. Many different questions can be posed based on the atomic structure, e.g., what are the atomic interactions among the different chains that compose a macromolecule? In order to answer this question, the user can run **chimera - contacts** to identify and group the residues involved in contacts between any pair of interacting molecules. **chimera - contacts** will find the possible interatomic contacts and group them by regions.

Other protocols involved in analysis and comparison of structures, as well as the information summarized in this

subsection, are covered by our model building tutorial, a document that describes the process in detail (see URL https://github.com/I2PC/scipion/wiki/tutorials/tutorial_model_building_basic.pdf). The tutorial starts with a short introduction to modeling, followed by a detailed description of each one of the workflow steps. Finally, at the end of the document there is an appendix dedicated to each one of the protocols, where the parameters are described.

FILE FORMAT COMPATIBILITY

Some PDB/MMCIF files produced by different packages present peculiarities involving format and nomenclature that make them unsuitable for inter-package use. Our approach to deal with this problem is as follows. If the protocol **a** creates an atomic structure file that should be read by protocol **b**, the first option will be to let **b** try to read the input file as it has been produced by **a**. If the program fails, then a cascade of conversions is done, and the program is restarted until the execution finishes successfully. Three different conversions are attempted with the package *MAXIT*: to translate a PDB format file into MMCIF and vice versa, and to update a MMCIF format file. The fourth conversion is implemented with *Biopython*,²⁹ which removes most of the MMCIF auxiliary tables and keeps only the atomic coordinates. So far, at least one of these four possibilities, besides the starting one, has provided a suitable input structure for all the protocols.

Regarding the 3D map file format, most of the model building programs assume that data are stored in CCP4/MRC format.³⁰ *Scipion*, if needed, will convert the maps from Spider³¹ or raw format to CCP4/MRC.

DISCUSSION AND CONCLUSIONS

As in many other areas in structural biology, there is a general trend among atomic model building packages to improve software inter-operability. Some packages (e.g., *Phenix*) have incorporated others (e.g., *MolProbity*) or interfaced them (e.g., *Coot*). A special mention has to be made for the “Collaborative Computational Project for Electron cryo-Microscopy” (*CCP-EM*), which is currently the only other framework (together with *Scipion*) whose main objective is to integrate other software packages. Concerning cryo-EM, both *Scipion* and *CCP-EM*¹¹ intend to integrate tools addressed to both image processing and model building. *Phenix*, although it integrates programs for data quality assessment and map improvement, mainly focuses on modeling programs.³

One of the main differences between *Scipion* and the above-mentioned packages is that *Scipion*'s developers have defined abstraction layers which simplify extension and maintenance. For example, adding a new protocol to *Scipion* requires the creation of a single Python file. From this file, *Scipion* detects the new protocol, automatically builds the corresponding input form, and stores the protocol-related information, such as parameters or input/output data, in the framework database. This plug-in approach allows the developers to focus on the algorithm to be included, while *Scipion* takes care of transparently handling data persistence and provenance, as well as managing the workflow execution and the user interface.

Another important difference between *Scipion* and other alternatives is how *projects* have been implemented. Most processing frameworks include the notion of *project* as the collection of steps (and outputs) performed to analyze a given

dataset. Usually these steps are listed together by the package GUI, and the files created are saved in a collection of subdirectories that share a common root directory. As far as we know, *Scipion* is the only framework that not only lists the steps but also shows them forming a graph (see Figure 1) in a way that clearly identifies how the data flow from one step to another and which “branches” did not produce a satisfactory result and were no longer pursued. *Scipion* follows a data-driven approach in which the transformations applied to the dataset are carefully recorded while the alternative frameworks (such as *Phenix* or *CCP-EM*) are more algorithm-driven. Algorithm-driven approaches usually record which algorithms have been executed, but no explicit register on the created output data is kept. In this approach, users need to check carefully the input and output file names of each step if they wish to understand the pipeline followed by the data.

Concerning interactive programs, another relevant difference should be highlighted between *Scipion* and other frameworks used to perform modeling. *Scipion* is the only one able to track the results obtained after using interactive tools such as *Chimera* and *Coot*. The extra commands added to these two programs allow the results generated by them to be saved in the same project diagram (and database) in which automatic protocols appear. In this way, all those manual processes (and their outputs) are carefully registered.

Scipion not only keeps track of the objects created as output of the different protocols but also includes a data model that helps to categorize them and establish how they are related. Some of these categories are images, sets of images, masks, movies, micrographs, Fourier shell correlations, volumes, geometric transformations, etc. Relevant to this molecular modeling extension, these categories include atomic structures, sequences, and origin of coordinates. In this way, when a user starts a protocol which requires as input an element belonging to a given category, *Scipion* is able to supply a list with the valid inputs and checks if a selected input satisfies the protocol requirements. That is, users do not need to navigate through the tree of directories searching for a file but rather receive a short list with all registered objects that can be used as input.

Scipion workflow handling offers some unique advantages. For example, it is possible to select a branch of a workflow and re-execute it with a few mouse clicks, or even to save the workflow on the hard drive so that it can be used as a template with other datasets (or sets of parameters). We have even created a repository web site (<http://workflows.scipion.i2pc.es/>) that can be used to either save or retrieve predefined workflows.

Since *Scipion* covers image processing of data acquired via electron microscopy, it is the only current framework that implements every step of the image processing pipeline going from the microscope images to the atomic model. Therefore, *Scipion* may be especially interesting for those users with expertise in the 3DEM field who wish to work in atomic model building (or the other way around) by reducing the learning curve. It not only provides a standard interface for both fields, as already commented in this Article, but also hides the complexity of adapting the maps produced by the 3DEM software to the conventions and formats used in the atomic model building field. Furthermore, *Scipion* is able to reduce the computational demands imposed by the large macromolecules produced in single-particle 3DEM by extracting their unit cells.

One of the major advantages of *Scipion* is the coexistence of algorithms, coming from different packages, that perform

similar or identical functions. In this way, users not only can compare the performance of different algorithms for their particular problem but also can use the consensus protocols that compare the output of two of these algorithms.

Scipion is a public domain image processing framework focused in electron microscopy. Online documentation describing how to download and install it is available at <http://scipion.i2pc.es>. For more information about how to install Scipion, suggestions on how to improve it, or how to contribute by adding new plugins, contact the authors at scipion@cnb.csic.es.

AUTHOR INFORMATION

Corresponding Author

R. Marabini – *Escuela Politécnica, Universidad Autónoma de Madrid, 28049 Madrid, Spain*; Email: roberto.marabini@uam.es

Authors

M. Martínez – *CNB-CSIC, 28049 Madrid, Spain*
A. Jiménez-Moreno – *CNB-CSIC, 28049 Madrid, Spain*
D. Maluenda – *CNB-CSIC, 28049 Madrid, Spain*
E. Ramírez-Aportela – *CNB-CSIC, 28049 Madrid, Spain*
R. Melero – *CNB-CSIC, 28049 Madrid, Spain*
A. Cuervo – *CNB-CSIC, 28049 Madrid, Spain*
P. Conesa – *CNB-CSIC, 28049 Madrid, Spain*
L. del Caño – *CNB-CSIC, 28049 Madrid, Spain*
Y. C. Fonseca – *CNB-CSIC, 28049 Madrid, Spain*
R. Sánchez-García – *CNB-CSIC, 28049 Madrid, Spain*
D. Strelak – *CNB-CSIC, 28049 Madrid, Spain; Institute of Computer Science, Masaryk University, 60200 Brno, Czech Republic*
J. J. Conesa – *CNB-CSIC, 28049 Madrid, Spain*
E. Fernández-Giménez – *CNB-CSIC, 28049 Madrid, Spain*
F. de Isidro – *CNB-CSIC, 28049 Madrid, Spain*
C. O. S. Sorzano – *CNB-CSIC, 28049 Madrid, Spain*
J. M. Carazo – *CNB-CSIC, 28049 Madrid, Spain*
 orcid.org/0000-0003-0788-8447

Complete contact information is available at:
<https://pubs.acs.org/10.1021/acs.jcim.9b01032>

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors acknowledge economical support from The Spanish Ministry of Economy and Competitiveness through Grants BIO2016-76400-R (AEI/FEDER, UE) and BFU2016-74868-P (AEI/FEDER, UE), and the “Comunidad Autónoma de Madrid” through Grant S2017/BMD-3817. D.S. holds a fellowship from “la Caixa” Foundation (ID 100010434) with code LCF/BQ/IN18/11660021. The authors acknowledge the support and the use of resources of Instruct-ERIC.

REFERENCES

- (1) Kuhlbrandt, W. Biochemistry. The resolution revolution. *Science* **2014**, *343*, 1443–1444.
- (2) Kuhlbrandt, W. Cryo-EM enters a new era. *eLife* **2014**, *3*, e03678.
- (3) Liebschner, D.; Afonine, P. V.; Baker, M. L.; Bunkoczi, G.; Chen, V. B.; Croll, T. I.; Hintze, B.; Hung, L. W.; Jain, S.; McCoy, A. J.; Moriarty, N. W.; Oeffner, R. D.; Poon, B. K.; Prisant, M. G.; Read, R. J.; Richardson, J. S.; Richardson, D. C.; Sammito, M. D.; Sobolev, O.

V.; Stockwell, D. H.; Terwilliger, T. C.; Urzhumtsev, A. G.; Videau, L. L.; Williams, C. J.; Adams, P. D. Macromolecular structure determination using X-rays, neutrons and electrons: recent developments in Phenix. *Acta Crystallogr. D Struct. Biol.* **2019**, *75*, 861–877.

(4) Cassidy, C. K.; Himes, B. A.; Luthey-Schulten, Z.; Zhang, P. CryoEM-based hybrid modeling approaches for structure determination. *Curr. Opin. Microbiol.* **2018**, *43*, 14–23.

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

(6) Emsley, P.; Lohkamp, B.; Scott, W. G.; Cowtan, K. Features and development of Coot. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **2010**, *66*, 486–501.

(7) Baker, M. L.; Baker, M. R.; Hryc, C. F.; Ju, T.; Chiu, W. Gorgon and pathwalking: macromolecular modeling tools for subnanometer resolution density maps. *Biopolymers* **2012**, *97*, 655–668.

(8) Wriggers, W.; Milligan, R. A.; McCammon, J. A. Situs: A package for docking crystal structures into low-resolution maps from electron microscopy. *J. Struct. Biol.* **1999**, *125*, 185–195.

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

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

(11) Burnley, T.; Palmer, C. M.; Winn, M. Recent developments in the CCP-EM software suite. *Acta Crystallogr. D Struct. Biol.* **2017**, *73*, 469–477.

(12) Adams, P. D.; Afonine, P. V.; Bunkoczi, G.; Chen, V. B.; Davis, I. W.; Echols, N.; Headd, J. J.; Hung, L. W.; Kapral, G. J.; Grosse-Kunstleve, R. W.; McCoy, A. J.; Moriarty, N. W.; Oeffner, R.; Read, R. J.; Richardson, D. C.; Richardson, J. S.; Terwilliger, T. C.; Zwart, P. H. Phenix: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **2010**, *66*, 213–221.

(13) de la Rosa-Trevin, J. M.; Quintana, A.; Del Cano, L.; Zaldivar, A.; Foche, I.; Gutierrez, J.; Gomez-Blanco, J.; Burguet-Castell, J.; Cuenca-Alba, J.; Abrishami, V.; Vargas, J.; Oton, J.; Sharov, G.; Vilas, J. L.; Navas, J.; Conesa, P.; Kazemi, M.; Marabini, R.; Sorzano, C. O.; Carazo, J. M. Scipion: A software framework toward integration, reproducibility and validation in 3D electron microscopy. *J. Struct. Biol.* **2016**, *195*, 93–99.

(14) Maluenda, D.; Majtner, T.; Horvath, P.; Vilas, J. L.; Jimenez-Moreno, A.; Mota, J.; Ramirez-Aportela, E.; Sanchez-Garcia, R.; Conesa, P.; Del Cano, L.; Rancel, Y.; Fonseca, Y.; Martinez, M.; Sharov, G.; Garcia, C. A.; Strelak, D.; Melero, R.; Marabini, R.; Carazo, J. M.; Sorzano, C. O. S. Flexible workflows for on-the-fly electron-microscopy single-particle image processing using Scipion. *Acta Crystallogr. D Struct. Biol.* **2019**, *75*, 882–894.

(15) Gomez-Blanco, J.; de la Rosa-Trevin, J. M.; Marabini, R.; Del Cano, L.; Jimenez, A.; Martinez, M.; Melero, R.; Majtner, T.; Maluenda, D.; Mota, J.; Rancel, Y.; Ramirez-Aportela, E.; Vilas, J. L.; Carroni, M.; Fleischmann, S.; Lindahl, E.; Ashton, A. W.; Basham, M.; Clare, D. K.; Savage, K.; Siebert, C. A.; Sharov, G. G.; Sorzano, C. O. S.; Conesa, P.; Carazo, J. M. Using Scipion for stream image processing at Cryo-EM facilities. *J. Struct. Biol.* **2018**, *204*, 457–463.

(16) Conesa Mingo, P.; Gutierrez, J.; Quintana, A.; de la Rosa Trevin, J. M.; Zaldivar-Peraza, A.; Cuenca Alba, J.; Kazemi, M.; Vargas, J.; Del Cano, L.; Segura, J.; Sorzano, C. O. S.; Carazo, J. M. Scipion web tools: Easy to use cryo-EM image processing over the web. *Protein Sci.* **2018**, *27*, 269–275.

(17) Cuenca-Alba, J.; Del Cano, L.; Gomez Blanco, J.; de la Rosa Trevin, J. M.; Conesa Mingo, P.; Marabini, R.; Sorzano, C. O. S.; Carazo, J. M. ScipionCloud: An integrative and interactive gateway for large scale cryo electron microscopy image processing on commercial and academic clouds. *J. Struct. Biol.* **2017**, *200*, 20–27.

(18) Marabini, R.; Macias, J. R.; Vargas, J.; Quintana, A.; Sorzano, C. O.; Carazo, J. M. On the development of three new tools for organizing and sharing information in three-dimensional electron microscopy. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **2013**, *69*, 695–700.

(19) Protein Data Bank, MAXIT Suite, <https://sw-tools.rcsb.org/apps/MAXIT/index.html>.

(20) de la Rosa-Trevin, J. M.; Oton, J.; Marabini, R.; Zaldivar, A.; Vargas, J.; Carazo, J. M.; Sorzano, C. O. Xmipp 3.0: an improved software suite for image processing in electron microscopy. *J. Struct. Biol.* **2013**, *184*, 321–328.

(21) Scheres, S. H. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J. Struct. Biol.* **2012**, *180*, 519–530.

(22) 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. MonoRes: Automatic and Accurate Estimation of Local Resolution for Electron Microscopy Maps. *Structure* **2018**, *26*, 337–344.

(23) Kucukelbir, A.; Sigworth, F.; Tagare, H. Quantifying the local resolution of cryo-EM density maps. *Nat. Methods* **2014**, *11*, 63–65.

(24) Cardone, G.; Heymann, J. B.; Steven, A. C. One number does not fit all: mapping local variations in resolution in cryo-EM reconstructions. *J. Struct. Biol.* **2013**, *184*, 226–236.

(25) Ramirez-Aportela, E.; Mota, J.; Conesa, P.; Carazo, J. M.; Sorzano, C. O. S. DeepRes: a new deep-learning- and aspect-based local resolution method for electron-microscopy maps. *IUCr. J.* **2019**, *6*, 1054–1063.

(26) Ramirez-Aportela, E.; Vilas, J. L.; Glukhova, A.; Melero, R.; Conesa, P.; Martinez, M.; Maluenda, D.; Mota, J.; Jimenez, A.; Vargas, J.; Marabini, R.; Sexton, P. M.; Carazo, J. M.; Sorzano, C. O. S. Automatic local resolution-based sharpening of cryo-EM maps. *Bioinformatics* **2019**, btz671.

(27) Webb, B.; Sali, A. Comparative Protein Structure Modeling Using MODELLER. *Curr. Protoc. Protein. Sci.* **2016**, *86*, 1–2.

(28) van Zundert, G. C. P.; Bonvin, A. M. J. J. Fast and sensitive rigid-body fitting into cryo-EM density maps with PowerFit. *AIMS Biophysics* **2015**, *2*, 73–87.

(29) Cock, P. J.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; de Hoon, M. J. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422–1423.

(30) Cheng, A.; Henderson, R.; Mastronarde, D.; Ludtke, S. J.; Schoenmakers, R. H.; Short, J.; Marabini, R.; Dallakyan, S.; Agard, D.; Winn, M. MRC2014: Extensions to the MRC format header for electron cryo-microscopy and tomography. *J. Struct. Biol.* **2015**, *192*, 146–150.

(31) Frank, J.; Radermacher, M.; Penczek, P.; Zhu, J.; Li, Y.; Ladjadj, M.; Leith, A. SPIDER and WEB: processing and visualization of images in 3D electron microscopy and related fields. *J. Struct. Biol.* **1996**, *116*, 190–199.