Cross-validation of cryo-EM maps

A guide for monitoring overfitting in cryo-EM

S. Ortiz, L. Stanisic, B.A. Rodriguez, M. Rampp, G. Hummer and P. Cossio.
Max Planck Institute of Biophysics, Germany.
University of Antioquia, Colombia.

Contents

Chapter 1

Cryo-EM cross-validation tutorial

1.1 Introduction

The purpose of this tutorial is to provide a step-by-step guide for performing the cross-validation tests for cryo-EM maps described in ref. [1]. The validation tests are performed over a small independent set of particle images not used in the cryo-EM refinement. The main objective is to monitor how the posterior probability evolves over the control particle set as a function of a low-pass filter frequency cutoff and the refinement iteration (for details see [1]).

The cross-validation tests are complementary to the gold-standard reconstruction procedure [2]. Therefore, the standard methods for cryo-EM refinement are still used [3]. Figure 1.1 shows an outline of the cross-validation procedure. First, it is necessary to randomly select a small set of particles not used for the refinement. This will be the "control set". Then, the gold-standard refinement is performed by excluding these particles, and generating two maps for each iteration step of refinement. Afterwards, two BioEM [4, 5] rounds are performed. The objective of these rounds is to calculate the posterior probability efficiently. The first round is used to obtain the best orientations for each particle. In the second round, the maps from each refinement iteration are low-passed filtered and the BioEM posterior probability is calculated for each using the orientations from the first round. Finally, the BioEM posteriors are used to calculate the cumulative log-posterior and the

normalized Jensen Shannon divergence (NJSD) as a function of the low-pass frequency cutoff and refinement iteration steps (see ref. [1]).

In the following section, we will describe the prerequisite programs and libraries. Then, we will describe the `Cross-Validation_Tutorial` files. After, we will provide the tutorials for computing the BioEM rounds 1 and 2, and the post-processing analysis (last three steps, red boxes in Figure 1.1). Since there are excellent programs and tutorials for cryo-EM refinement already available [6, 7, 8], it is assumed that the user has sufficient knowledge of the cryo-EM reconstruction steps.

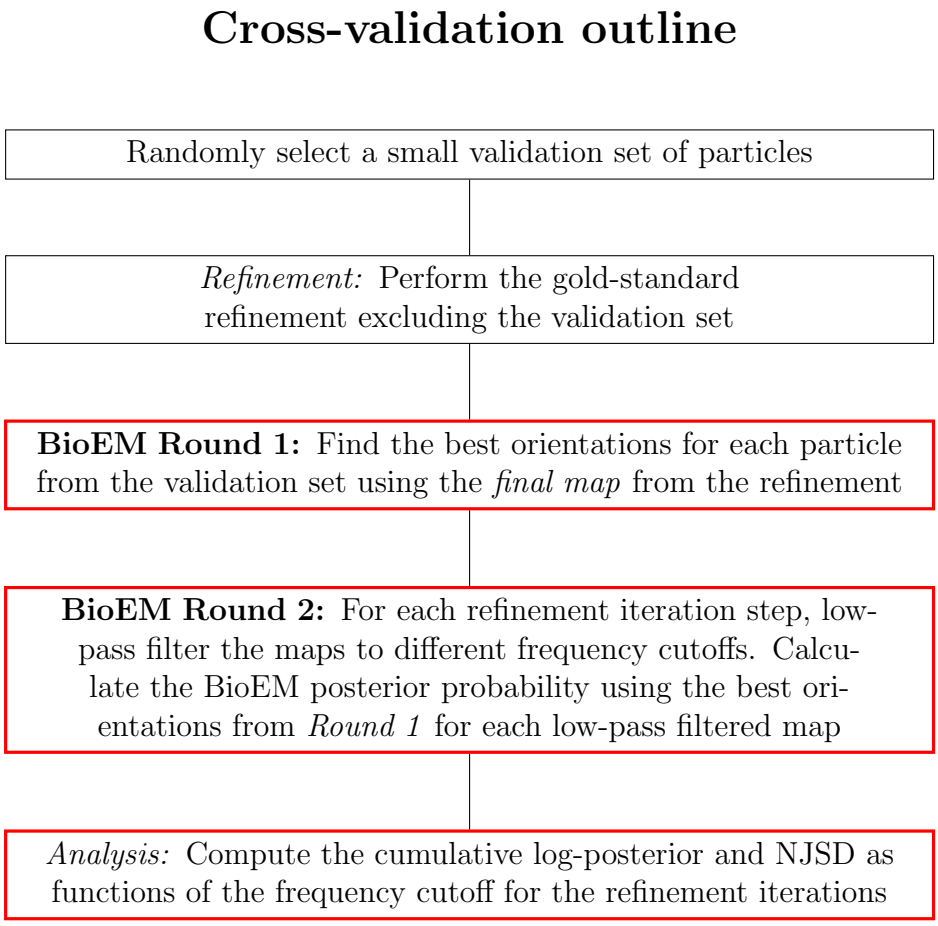

Figure 1.1: Outline for the cross-validation of cryo-EM maps. For the last three steps (red boxes), we provide a detailed tutorial.

1.2 Prerequisite programs and libraries

Before running the tutorial the following programs should be installed on a compute node:

- *BioEM:* A compressed directory of the BioEM software can be downloaded from

  ```
  https://github.com/bio-phys/BioEM
  ```

  The instructions for compiling and running BioEM are therein provided.

- *FFTW library:* is a subroutine library for computing the discrete Fourier transform. Specifically, it is used to calculate the Fourier transform of the maps for the low-pass filtering program (see 3D-map utilities below). FFTW can be downloaded from the webpage www.fftw.org.

- *3D-map utilities:* We use modified versions of some programs from the 3D-map utilities from the Rubienstien lab

  (https://sites.google.com/site/rubinsteingroup/3-d-map-utilities).

  The programs that we have modified are `apply_mask_bioem.f90` and `lowpassmap_fftw.f90`. These are provided in the `programs` directory, which is included in the tutorial directory. To compile these two programs please execute:

  ```
  module load gcc fftw/gcc/3.3.4
  ./build_3Dmap.sh
  ```

  Note: This automatic building script is based on a static URL for 3D-map utilities. If the link changes or you experience any other issue, please modify the script or install manually.

1.3 Cross-validation tutorial directory

The cross-validation tutorial can be downloaded from https://github.com/bio-phys/BioEM-tutorials, which provides the `Cross-Validation_Tutorial` directory. In this directory, you will find the following scripts:

- `Apply_MaskRound1.sh`: A script for masking and converting a map in MRC format into the BioEM format.

- `run_Round1_sge.sh`: The principal script for running the first BioEM round (see Section 1.4), which searches for the best orientations of the particles from the control set.

- `Apply_Filter.sh`: A script that low-pass filters a map (in MRC format), masks and converts it into to the BioEM format.

- `Write_ListMap.sh`: This script generates the `List_MapFilter` file which contains the information from all the maps and the filtering-radii that will be analyzed in the second stage of the protocol.

- `Create_ParamInput.sh`: A script that generates the individual BioEM parameter-input file for running BioEM round 2, using the defocus information of the individual particles.

- `run_Round2_sge.sh`: The principal script for running the second BioEM round (see section 1.5.3), which calculates the posterior probabilities for each filtered map over the control set.

- `Analysis.sh`: A script for analyzing the results obtained from the BioEM round 2. It calculates the cumulative log-posterior and the NJSD for all iterations as a function of the filtering frequency.

In the `Cross-Validation_Tutorial`, you will find the following example files:

- `run_class001.mrc`, `run_it001_half1_class001.mrc` and `run_it001_half2_class001.mrc`: MRC maps for running the tutorial. The first map is used for BioEM round 1, which is created by joining the two maps from the final iteration of RELION [6]. The other two maps are examples for BioEM round 2, which consist of the maps created from the first and second half of the data, respectively, for one iteration.

- `mask.mrc`: Reference mask used to create the map input for the BioEM analysis.

- `particles_file.txt`: A file containing the path for all the independent particles (first column) from the control set and their corresponding defocus in micrometers (second column).

- `QUATERNION_LIST_4608_Orient`: A grid of 4608 quaternions which samples uniformly the orientation space.

- `Param_BioEM`: A file with the necessary parameters for executing the BioEM program (see BioEM manual [5] for details).

- `Input_Cross-Val`: This file specifies the parameters and inputs necessary to execute the cross-validation tutorial (see below for details on the necessary keywords).

- `map_iteration.txt`: This file contains the path to the root names of the maps for the iterations to analyze. For this tutorial, it would have `run_it001`.

In the `Cross-Validation_Tutorial`, there are the following sub-directories:

- `programs`: It contains all auxiliary Fortran and Python programs used in the tutorial.

- `particles`: It contains four example particles in MRC format.

1.3.1 Cross-validation input

A file containing the information for running the tutorial is necessary. As an example, the `Input_Cross-Val` file is provided in the tutorial folder. It should have the following keywords:

- `MAP_ROUND1 path`: where *path* directs to the map (in MRC format) used in BioEM round 1 for searching for the best orientations.

- `MASK path`: where *path* directs to the mask (in MRC format) that defines the voxels used for the BioEM map format.

- `PARTICLES_FILE path`: where *path* directs to a file which contains a column with the path of the independent particles, and a second column with their corresponding defocus in μm.

- `ORIENT_FILE path:` where *path* directs to the quaternions list used for the first BioEM round, which samples uniformly the orientation space.
- `MAP_ITERATION_FILE path:` where *path* directs to a file listing the root names of the maps generated from the iterations in the refinement procedure. For example, using the RELION format, it will be similar to

  Refine3D/job010/run_it004
  Refine3D/job010/run_it008
  Refine3D/job010/run_it012
- `NUM_FREQ int:` where *int* is the number of frequencies that will be selected for filtering the maps.

1.4 BioEM Round 1: Searching for the best orientations

In this stage, we search for the best orientations of each particle among a quaternion set which samples uniformly the orientation space. To perform this search, it is recommended to use the *final map* obtained in the refinement procedure, which is created by joining the reconstructions from set 1 and set 2 from the gold-standard procedure. For example, in the RELION [6] workflow, this map is usually named `run_class001.mrc`. As an example, this file is provided in the `Cross-Validation_Tutorial` directory.

To run BioEM round 1, the following steps should be executed:

1.4.1 Masking and converting the map

The first step is to convert the *.mrc* map into the BioEM format. For this, it necessary to provide a mask (in mrc format) that will define the voxels which will be taken into account to create the map format for BioEM. To do so, in the `Cross-Validation_Tutorial` directory run

```
./Apply_MaskRound1.sh Param_BioEM Input_Cross-Val
```

where `Param_BioEM` is the parameter input file and `Input_Cross-Val` is the input file for the cross-validation tutorial (see above). If successful, this

execution should produce `map_ROUND1.txt` file, which is the input map for BioEM.

1.4.2 Executing BioEM Round 1

The BioEM program is executed using the parameters specified in the file `Param_BioEM`. To extract the best orientations please execute

```
qsub run_Round1_sge.sh path_to_BioEM_executable
Param_BioEM Input_Cross-Val
```

Note 1: It takes around 108 seconds to execute this run on a single node from the phys cluster (24 cores Haswell CPU + 2 $\times GeForceGTX1080$).

This command runs the BioEM program with the `ALGORITHM=1` (see BioEM manual [5]) over GPUs. This script generates a small grid of quaternions around the best orientations obtained for each particle from the BioEM analysis. These lists of quaternions are found in the `orientations` directory with name `ANG_particle-name`. As an example, in this tutorial, the best 10 orientations for each particle are taken, and a finer grid of 125 quaternions around each is used, producing a list with 1250 quaternions for each particle. These files will be used in BioEM round 2 to perform a local search around the best orientations, increasing the accuracy of the probabilities.

Note 2: If there are many particles or the defocus range of the particles is large, it is recommended to divide the particle set into defocus subgroups and execute round 1 for the different defocus subgroups by modifying the CTF defocus range parameter in the `Param_BioEM` file.

Note 3: For a more accurate orientation search, it is recommended to use the `QUATERNION_LIST_36864_Orient` orientation file, which is provided in the `BioEM/Quaternions` folder.

1.5 BioEM Round 2: Cross-validation around the best orientations

The second phase of the protocol is the core of the cross-validation tests. The objective is to calculate the BioEM posterior over the control particle set for different maps generated with different low-pass frequency cutoffs and

refinement iterations. Before executing BioEM round 2, it is necessary to create a list that contains the map name and filtering-radius for the selected iterations. It is also necessary to create the individual BioEM input files for each particle image.

1.5.1 Creating the map-filtered list

In this section, we will generate a list which contains the path to the map and the filtering-radius in $\mathring{A}$ for the desired iterations. This list is created using the information from the `Param_BioEM` and `Input_Cross-Val` files. In particular, it uses the `MAP_ITERATION_FILE` which contains a list with the root name of the map files from the selected iterations. **Important:** Note that we are using RELION's [6] notation, where the root name is typically `run_itY` (where `Y` is the iteration) and the names are finalized with `half1_class001.mrc` and `half2_class001.mrc` for sets 1 and 2, respectively. The number of frequencies (inverse of the filtering-radius) is set with the flag `NUM_FREQ` in the input file `Input_Cross-Val`.

To generate the list, please execute:

```
./Write_ListMap.sh Param_BioEM Input_Cross-Val
```

The output is `List_MapFilter`. This file should contain $[2 \times \#iterations \times \#filter - radius]$ lines. The first column of this file is the path to the map and the second is the filtering radius.

By default, the filtering-frequencies are distributed uniformly from $2/(PixelSize \times ImageSize)$ to $1/(3 \times PixelSize)$. The user can modify the range of frequencies by changing the parameters `fmin` and `fmax` in the file `Write_ListMap.sh`.

1.5.2 Creating the individual parameter-input files

Before running the second round of BioEM, it is necessary to create the parameter-input file for each image. These files will contain the specific value of the defocus for each individual particle, which is extracted from the `PARTICLES_FILE`. Please execute:

```
./Create_ParamInput.sh Param_BioEM Input_Cross-Val
```

This command should generate in the `tmp_files` directory the files called `param_particle-name`.

1.5.3 Executing BioEM Round 2

In this section, we use the `List_MapFilter` and parameter-input files previously generated. The script `run_Round2_sge.sh` filters each map at the specific filtering-radius, masks and converts the map into BioEM format. This is done on-the-fly using the script `Apply_Filter.sh`. Then, it runs the BioEM program with `ALGORITHM=2` using the the local orientation-grid and parameter-input for each particle. To execute the script, use the following command:

```
qsub run_Round2_sge.sh path_to_BioEM_executable
Param_BioEM Input_Cross-Val
```

Note: It takes around 30 seconds to execute a single instance of this example on a single core of a single node from the phys cluster (24 cores Haswell CPU).

The `run_Round2_sge.sh` script will be executed over a single core for each map and filter-radius (i.e., each line of `List_MapFilter`) using the job-arrays. After execution, the files `outputfiles/output_X` are generated. The output file contains the log-posterior probability for each particle calculated for the map and filtering-radius corresponding to line `X` in the `List_MapFilter` file.

Running round 2 on GPUs: If there are many particles, it might be recommended to run round 2 on GPUs. In the tutorial directory, there is an example file `run_Round2_sge-gpus.sh` which runs round 2 over GPUs with the filtered maps in series. To execute this script run
`qsub run_Round2_sge-gpus.sh path_to_BioEM_executable Param_BioEM Input_Cross-Val X Y`
where `X` and `Y` are the starting and final lines of the `List_MapFilter` file, which indicate the filtered maps to be computed.

1.6 Analyzing the results

With the log-posterior outputfiles it is possible to calculate the cumulative posterior and the normalized Jansen-Shannon divergence NJSD (see [1]) as

a function of the frequency cutoff for each iteration. We provide a script to perform these calculations. To do so, run:

```
./Analysis.sh
```

This script will generate the folder `results`, where the files `CUM_itY_half_1` and `CUM_itY_half_2` contain the cumulative posterior (for sets 1 and 2, respectively), and the file `NJSD_itY` contains the NJSD, all as a function of the frequency cutoff for iteration `Y`.

If there are no signs of overfitting in the data, both the cumulative posterior and the NJSD should increase as a function of the frequency cutoff. For details on the interpretation of the results see ref. [1].

1.7 Summary

A graphical summary for the implementation of the cross-validation protocol is showed in figures 1.2 and 1.3 for Round 1 and Round 2, respectively. Yellow boxes are codes or process to execute; the blue boxes show the output files, and the white boxes are input files that must be edited by the user.

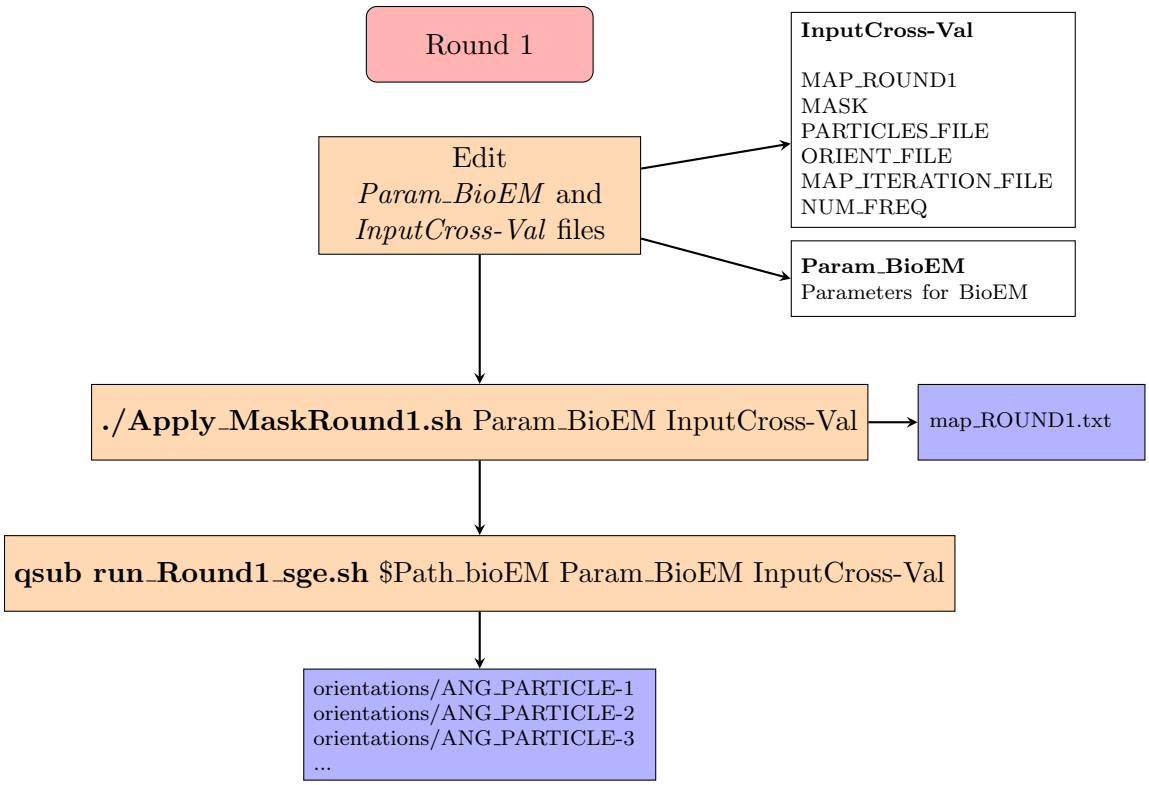

Figure 1.2: Summary of BioEM round 1 for the cross-validation protocol.

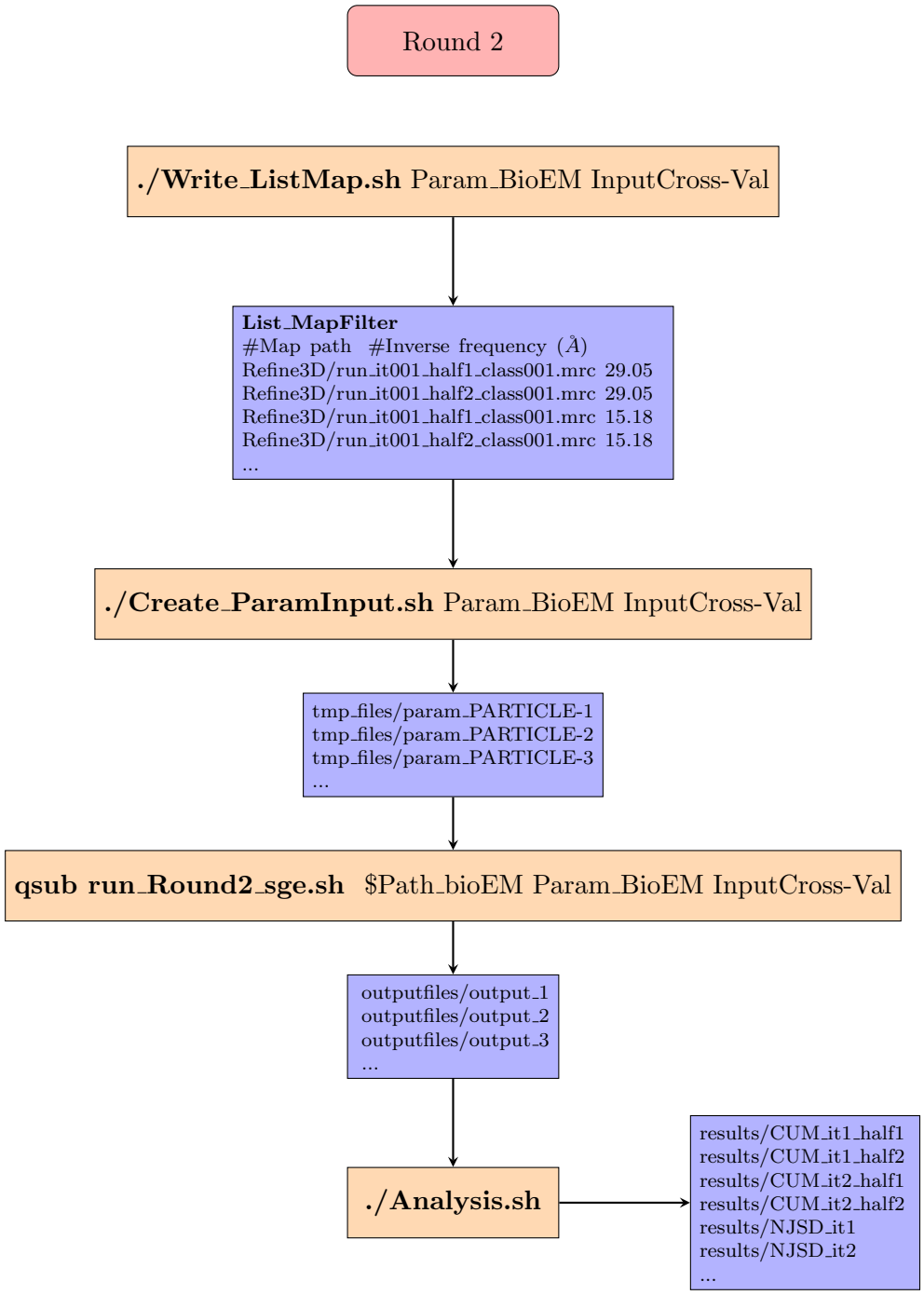

Figure 1.3: Summary of BioEM Round 2 for the cross-validation protocol.

Bibliography

[1] Sebastian Ortiz, Luka Stanisic, Boris Rodriguez, Markus Rampp, Gerhard Hummer, and Pilar Cossio. Cross-validation tests for cryo-EM maps using an independent particle set. *arXiv*.

[2] Sjors H W Scheres and Shaoxia Chen. Prevention of overfitting in cryo-EM structure determination. *Nature methods*, 9(9):853, 2012.

[3] Pilar Cossio and Gerhard Hummer. Likelihood-based structural analysis of electron microscopy images. *Current Opinion in Structural Biology*, 49:162–168, apr 2018.

[4] Pilar Cossio and Gerhard Hummer. Bayesian analysis of individual electron microscopy images: Towards structures of dynamic and heterogeneous biomolecular assemblies. *Journal of structural biology*, 184(3):427–437, 2013.

[5] Pilar Cossio, David Rohr, Fabio Baruffa, Markus Rampp, Volker Lindenstruth, and Gerhard Hummer. BioEM: GPU-accelerated computing of Bayesian inference of electron microscopy images. *Computer Physics Communications*, 210:163–171, 2017.

[6] Sjors H W Scheres. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *Journal of structural biology*, 180(3):519–530, 2012.

[7] Guang Tang, Liwei Peng, Philip R. Baldwin, Deepinder S. Mann, Wen Jiang, Ian Rees, and Steven J. Ludtke. EMAN2: An extensible image processing suite for electron microscopy. *Journal of Structural Biology*, 157(1):38–46, jan 2007.

[8] C.O.S. Sorzano, R. Marabini, J. Velázquez-Muriel, J.R. Bilbao-Castro, S.H.W. Scheres, J.M. Carazo, and A. Pascual-Montano. XMIPP: a new generation of an open-source image processing package for electron microscopy. *Journal of Structural Biology*, 148(2):194–204, nov 2004.