Tutorial: High-Resolution Subtomogram Averaging with IMOD and PEET

John M. Heumann
Last Modified: 2/22/2023

Introduction

Recent hardware and software developments enable near-atomic (sub 4 Å) resolution with suitable samples using either single particle analysis (SPA) or subtomogram averaging (STA). STA is more challenging than SPA, because of the additional processing steps required, and because STA it is often reserved for noisier *in situ* or irregular samples where SPA is not directly applicable. Near-atomic resolution with STA has been demonstrated using several freely available software packages at various stages (*e.g.,* see https://teamtomo.org/). Unfortunately, inconsistent conventions and formats can make switching between packages challenging. In this tutorial, we demonstrate near-atomic resolution STA in the IMOD / PEET framework. Specifically, motion correction, CTF estimation, tomogram reconstruction with dose-weighting and 3D CTF correction, and modeling can all be done in IMOD, while STA to 3.8 Å is done using PEET. Provision is also made for easily exporting to (or importing from) Relion 4.0 STA which provides aberration correction and "polishing" steps not available in PEET, leading to even better 3.3 Å resolution.

Reasonable familiarity with IMOD and PEET Is assumed. More basic tutorials and documentation providing the necessary background are available on the IMOD and PEET home pages at https://bio3d.colorado.edu/. Processing steps and optional settings are described here. In the interest of brevity and maintainability, however, step-by-step instructions at the level of "Press this button…" are mostly omitted.

Extensively commented Linux bash shell scripts are used throughout. Most should be understandable even to those with no prior shell scripting experience so long as they have basic familiarity with Linux command line operations. Readers wishing to customize the approach described here or to generate new ones should become familiar with writing and editing their own scripts, either in bash or in an alternative scripting language.

Prerequisites

This tutorial uses the 5-tilt series subset of the EMPIAR-10164 HIV-1 dMACANC VLP dataset which has become a standard STA benchmark. Movies and mdoc files for tilt series 01, 03, 43, 45, and 54 can be downloaded directly from https://www.ebi.ac.uk/empiar/EMPIAR-10164/ if desired. However, these raw data are included in https://bio3d.colorado.edu/PEET/EMPIAR-10164-Tutorial.tgz, which also provides these instructions and the relevant scripts, intermediate, and output files. Download to an appropriate location and extract with "tar xvzf EMPIAR-10164-Tutorial.tgz". The 600 GB download may take considerable time, and approximately 1.2 TB will be required after extraction.

Recent IMOD (≥ 4.12.35) and PEET (≥ 1.16.0) versions are required. Tomogram reconstruction and other IMOD steps are best done on a system with at least one and preferably multiple Nvidia GPUs. PEET steps do not utilize GPUs and are best done on a cluster server, high core count system, or group of networked nodes.

Directory Structure

The top-level directory, EMPIAR-10164-Tutorial/, in addition to this file, contains an xml description of the dataset, and a script, described below, to do motion correction. Subdirectories frames/, mdoc/, and MotionCorrected/, respectively, contain raw tilt series movies, corresponding acquisition logs, and the motion corrected tilt series. Tomograms/ contains reconstructed tomograms plus the scripts and settings used to generate them and the resulting logs. Models/ contains the initial IMOD models, and PEET/ contains scripts, settings, and results from various stages of subtomogram averaging.

Tomogram Generation

Motion Correction

The original mdoc files supplied with EMPIAR-10164 are outdated and contain incorrect “ExposureDose = 0” entries. The original files have been renamed to mdoc/*.mrc.mdoc.orig and the correct ExposureDoses (2.4 e-/ Å^2 for tilt series 01 and 03, 3.1 e-/ Å^2 for 43 and 45, and 3.0 e-/ Å^2 for 54) have been entered in mdoc/*.mrc.mdoc. After this correction, motion corrected tilt series in MotionCorrected/ and an update mdoc file can then be generated by executing “runAlignframes.sh 2>&1 | tee alignFrames.log”. (Bash style shell commands are used throughout this tutorial; minor modification may be necessary to run under other shells).

Tomogram reconstruction is done as described below for each of the 5 tomograms in Tomograms/TS<nn>. This could be automated with batchruntomo, but that has not been done here.

Setup

The first frame is bad in both tilt series 01 and 03 and is marked for exclusion.

Preprocessing and Coarse Alignment

X-ray removal and coarse alignment are performed in the usual fashion.

Fiducial Model Generation

Under Seed Model, select “Make seed and track” and “Generate seed model automatically”. Request 20 seed points, allowing clustered beads. Under Track Beads, track the initial seed model, fix the fiducial model, and then iteratively track with the fiducial model as seed and fix, as usual, until you are satisfied with the results.

Fine Alignment

Recent IMOD versions can use cross-validation to choose alignment variable settings in a nearly automatic fashion. The current dataset is quite insensitive to the complexity of the fitted model, yielding good results with no geometric parameters at all or with simple rigid rotation. Nevertheless, we illustrate the use of cross-validation here to demonstrate choosing a model which fits the data well but avoids overfitting. On the Fine Alignment / General tab, ensure “Compute prediction errors for points left out of test fits” is checked, and that robust fitting and local alignments are disabled. On the Global

Variables tab, select “Rotation Solution Type / Solve for all rotations”, “Magnification Solution Type / Solve for all magnifications”, “Tilt Angle Solution Type / Group tilt angles”, “Distortion Solution Type / Disabled”, and under Single Variables select “Solve for beam tilt” and “Solve for single stretch during projection”. Press “Compute Alignment” and note the resulting global leave-out (aka cross-validation) error in the project log view window or the alignment log. You may also wish to try disabling “Solve for beam tilt” and enabling “Distortion Solution Type / Full solution”. Choose whichever gives the lowest leave-out error. At this point, you have a complex model which probably overfits the data. Go back to the General tab, and press “Run Cross-validation”. This will try a series of progressively simpler models, looking for the minimum leave-out error. If desired, open the restrictalign log for additional details. The restrictalign search is heuristic rather than exhaustive, but typically returns quite good solutions. You can also explore varying individual settings after or in conjunction with cross-validation. The goal is to minimize the cross-validation error while preferring the simplest model possible.

Tomogram Positioning

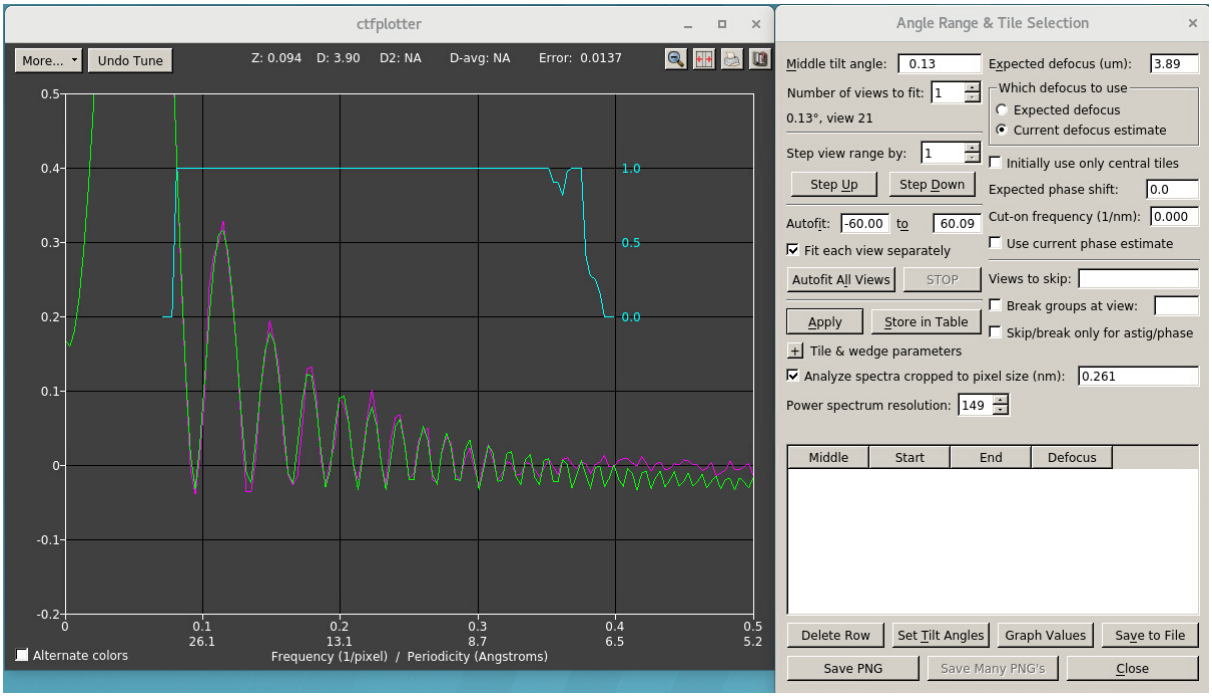
Large positioning tomogram thickness is needed for these data. Choose boundaries which do not clip usable data. Since the edges of the ice are difficult to see, we place all 3 Boundary model lines for a given (upper or lower) surface at the same Z height in the XY plane, resulting in 0° tilt angle offset. After positioning, create the Final Alignment as usual.

Final Aligned Stack

On the Create tab, press “Create Full Aligned Stack”. After viewing the aligned stack to ensure that everything looks good, proceed sequentially to the Correct CTF, Erase Gold, and 2D filter tabs, as described below. 3D CTF correction, gold erasure, and filtration will be performed during tomogram reconstruction, so at this stage we only need to generate the necessary settings, not to perform the actual operations nor to press the “Use…” buttons. (If you should do so, however, don’t worry! It’s harmless).

CTF Estimation

On the Correct CTF tab, ensure that the “Voltage”, “Spherical Aberration”, and “Expected Defocus” entries are correct, and then press “Run CTF Plotter”. In previous versions, defocus estimation with ctfplotter required an iterative process; with reasonably good data, it is now almost automatic. In the Angle Range & Tile Selection window, change “Step view range by” to 1 and set the “Middle tilt angle” to the value nearest to 0. In the Ctfplotter window, press “Autotune”. If the resulting green and magenta curves are in good agreement and show multiple CTF peaks / zeros, as shown in the example below, you’ve finished tuning and are ready to do defocus estimation. If not, press “Undo Tune” and try again with a different initial defocus estimate or starting tilt(s) (still close to 0).



in the Fitting Params window, check “Find astigmatism”. In the Angle Range & Tile Selection window, check “Fit each view separately”, and press “Autofit All Views”. Press “Yes” if you are prompted whether to remove existing table entries. Monitor the plots during fitting. Assuming the green and magenta curves look good and are in reasonable agreement, press “Save to File” in the Angle Range & Tile Selection window. Close ctfplotter and proceed to the Erase Gold tab.

Erase Gold

Finding and erasing gold beads is done as usual. In this case, you will want to actually Erase Beads for verification. Check that you’re satisfied with erased bead locations, correcting the model if not. When satisfied, proceed to the 2D Filter tab. It is not necessary to press “Use Erased Stack”.

2D Filter

On the 2D Filter tab, select “Apply dose weighting”. Press done when finished to proceed to the Tomogram Generation.

Tomogram Generation

Select “3D CTF” as the type of tomogram generation. Under 3D CTF Correction Parameters, check “Compute slabs in parallel”, “Erase gold”, “Apply 2D filter”, and “Reconstruct from raw images”. Set “Thickness of slab for each CTF correction” to 15 nm. Additional recommended settings are available in Tilt Parameters advanced mode. (Press the A at the upper right of the Tilt Parameters pane if visible). Selecting “Super-sample by” and specifying 3 will reduce aliasing artifacts in the tomogram. Specifying a “Linear density scaling factor” which yields gray values ranging from ~ -8000 to 8000 in the raw CTF-corrected tomogram will minimize quantization errors. One way to estimate the required scaling value is to first generate a trial CTF-corrected tomogram with the default settings and to check the resulting min and max gray values using the header command. When satisfied, generate the final, full CTF corrected tomogram, press Use CTF-corrected Tomogram and Done.

Post-processing and Clean Up

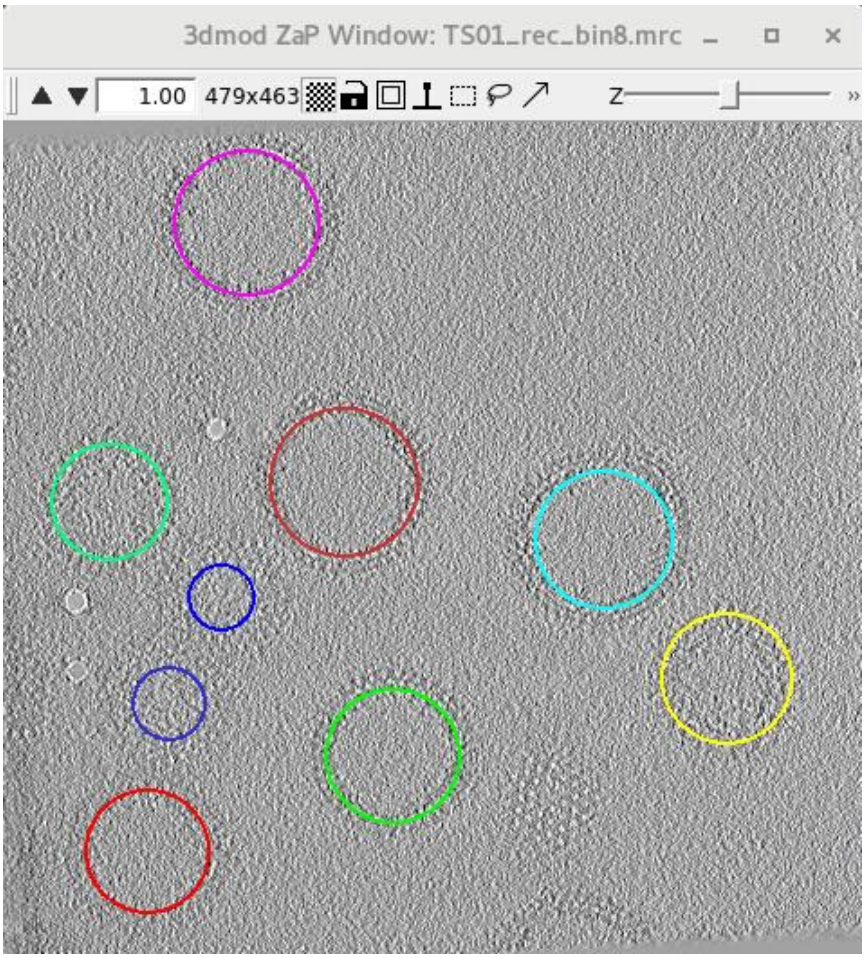
Post-processing and clean up are performed as usual.

Binning

For STA, we will begin processing with highly binned data to suppress noise and improve throughput. From each directory Tomograms/TS<nn>, invoke "../binVolumes.sh" to generate 2, 4, and 8X binned volumes. Tomogram TS01 is the highest defocus, highest contrast volume in this dataset. Open the bin8 version in 3dmod and scroll in Z to get a feel for theses samples.

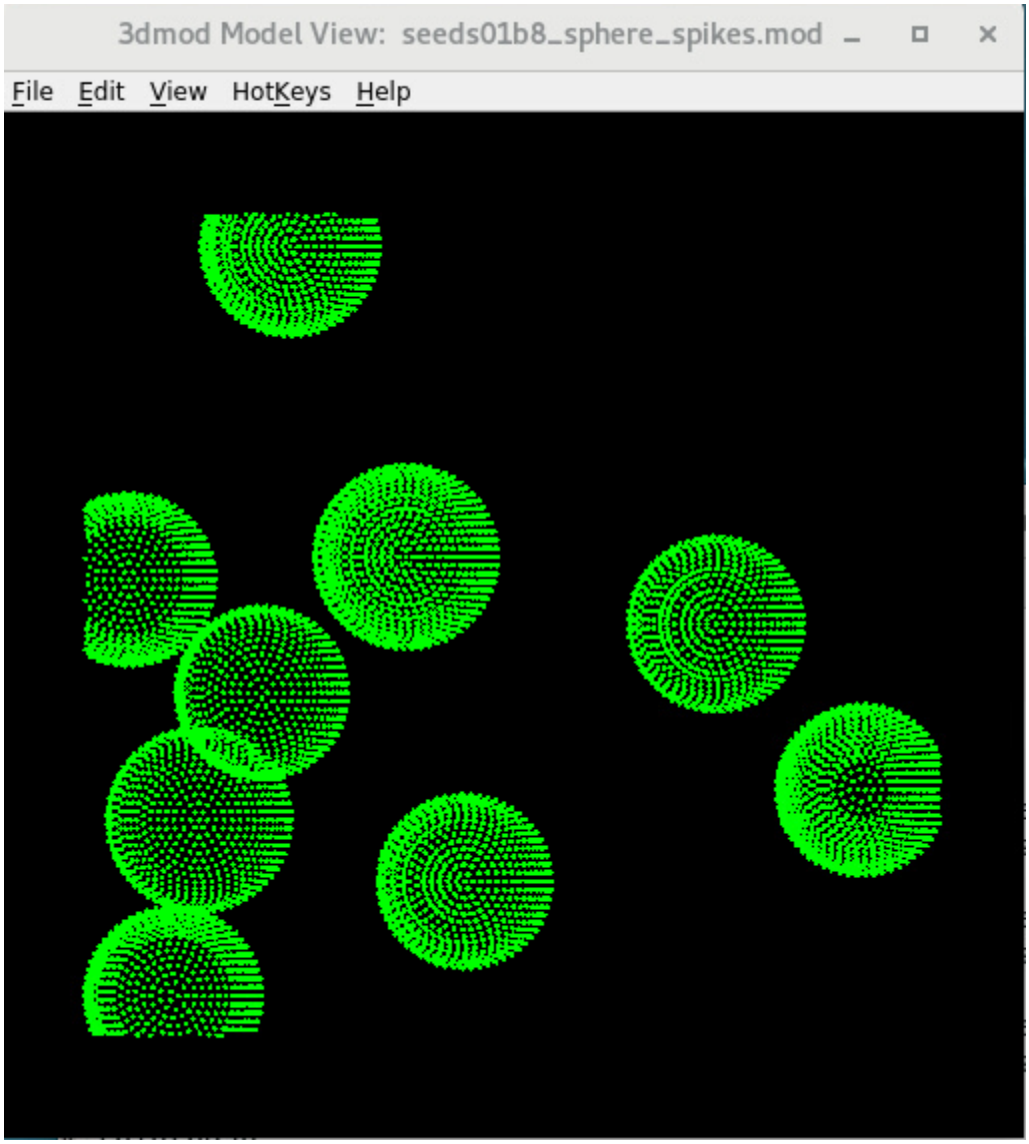
Modeling

We need to create a separate IMOD model for each Tomogram describing the location and size of its VLP's. Each model will contain an object for every VLP (or fragment) to be processed from that tomogram. Each object will contain a single, scattered contour with 1 point whose position and radius approximately match those of the VLP. Appropriate models have been supplied. To view the model for TS01, for example, from the EMPIAR-10164-Tutorial directory, execute "3dmod Tomograms/TS01/TS01_rec_bin8.mrc Models/TS01/vlps01b8.mrc", and scroll up and down in Z.



Notice that where a slice grazes a VLP tangentially (*e.g.,* at the lower right center in the image above) we see evidence for hexagonally packed subunits with approximately 8 nm spacing. Our goal is to align and

average these subunits rather than entire VLPs, since the latter are irregular. In each Model/TS<nn> directory, execute “seedSpikes.sh”. These scripts use PEET programs seedSpikes and spikeInit to place candidate particles, oversampled to about twice the expected density, and then to generate models, initial motive lists, and rot axes files (named seeds<nn>b8_*) suitable for PEET alignment and averaging. A model view of seeds01b8_sphere_spikes.mod a is show below. These model files can also be opened on their corresponding volumes in 3dmod for verification, and the radially outward rotation axes displayed with plotRotAxes.



Initial Reference Generation

Points have been seeded at ~2X the expected density in a regular pattern with positions unrelated to actual VLP subunits. To remedy this, we will need to generate a template which will be used to identify desirable locations. The template will be determined using TS01, since it is the highest defocus, highest contrast tomogram. At least 3 strategies can be used in PEET for this purpose.

First, we could simply generate a starting reference using a no-search iteration from the initial model points and orientations. This would not be expected to show structural details, since those will have been blurred during averaging of incorrect locations. Overall VLP curvature would be preserved, however, and the following iteration would select for locations where the curvature matches the average well. Additional details can be expected to emerge with subsequent iterations of alignment and averaging.

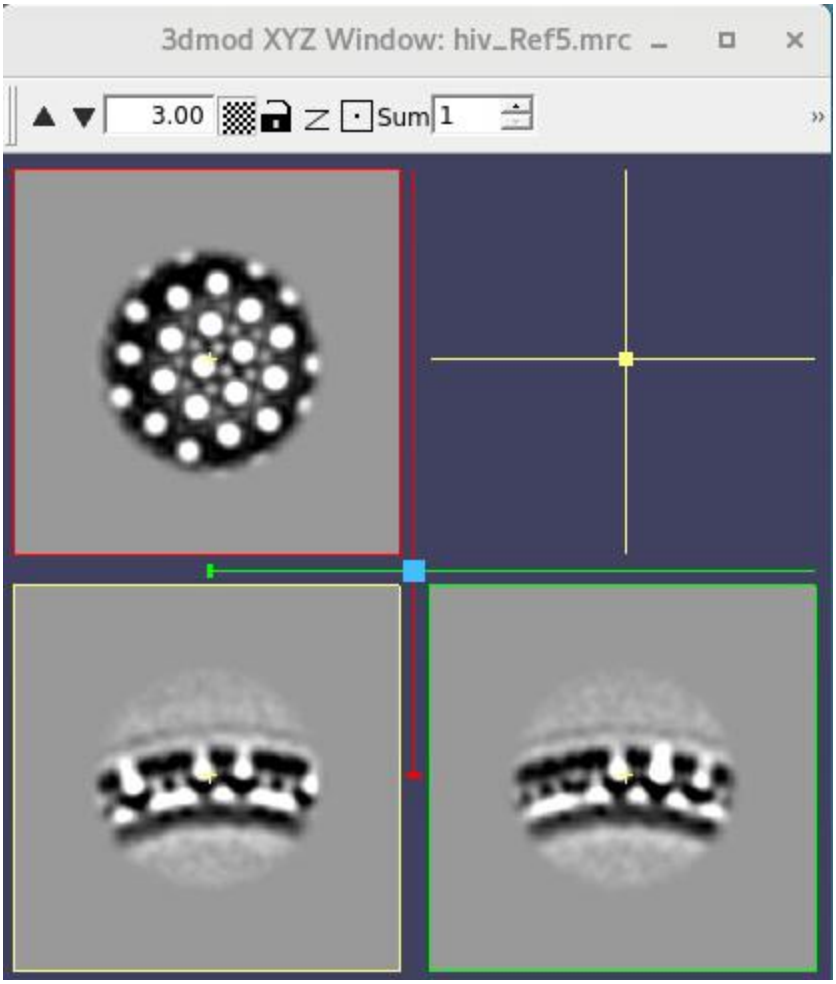
Second, we could choose (or manually add) a specific single point as the reference for the first iteration, again expecting initial details to emerge with repeated iteration. A starting point can be chosen whose location and orientation closely match the underlying VLP, and which is located on what appears to be a radially oriented spike in the original tomogram. Several starting points could be tried to verify that results are representative and not excessively biased.

Third, PEET provides an option to generate a multiparticle reference. Briefly, 2^k particles are chosen pseudo-randomly from the starting model and aligned and averaged in a pairwise, hierarchical manner until a single starting reference is generated. This reference is then refined iteratively.

Each of these 3 approaches can be used successfully with this and similar datasets. In each case, strong low-pass filtration would be used to minimize overfitting. In this tutorial, we've chosen the 2^{nd}, single point approach. Settings, intermediate files, and output files can be found in EMPIAR-10164-Tutorial/PEET/SingleTomogram/InitialReference. In this directory, linkToRotAxes.sh was run before running the PEET alignment to create symbolic links to the rotation axes files generated by spikeInit.

Several observations regarding the chosen setting are in order. Angular search limits have been chosen based on the suspected C6 symmetry suggested by the raw tomograms, but we do not impose symmetry at this stage, preferring to let it emerge (or not) from the data. With 64 cores, this run took roughly 3 hours and 50 minutes. Particles near the VLP poles have been found to be helpful for obtaining high-resolution from this and similar datasets. These views tend to be under-represented unless steps are taken to avoid this. As a result, we use an option to adjust cross-correlation scores based on elevation angle prior to particle selection for averaging. Finally, we initially use subvolume and mask sizes larger than might be expected. Specifically, since we suspect hexagonal packing, it's tempting to assume that a radius large enough to include a central subunit and its 6 nearest neighbors would suffice. In practice, a larger radius encompassing not just the nearest neighbors but the next ring out is helpful, although not absolutely required. One reason is that VLPs have variable curvature, and the larger radius allows more accurately choosing particles closely matching the average curvature. This also helps with retention of particles near the poles.

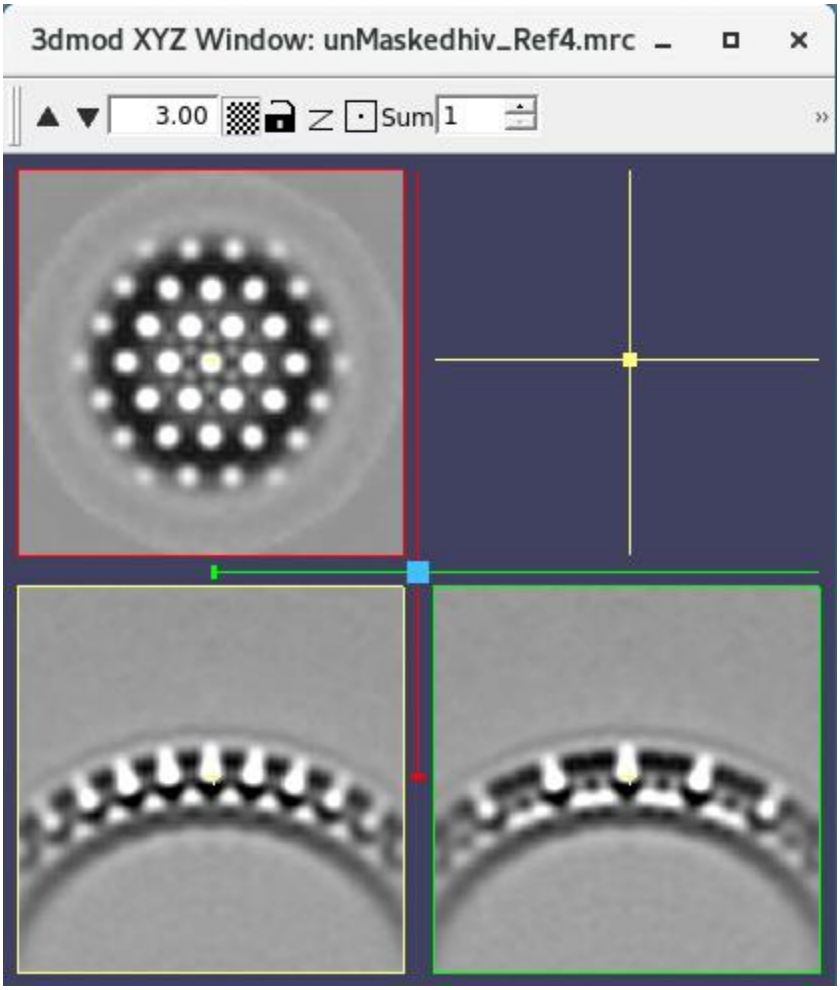
A masked reference resulting from this initial run confirms our suspicion of c6 symmetry in addition to illustrating the size of the masked region used for alignment.



Using the 3dmod Slicer window, we find rotations and translations to center the nearest C6 symmetry axis and orient it along the Y axis. These parameters are then edited into reorientAndCenter.sh, which runs modifyMotiveList to generate a new motive list. In this case, we've chosen to do the rotations and translations separately in 2 successive calls to modifyMotiveList. It is possible to find (different) rotations and translations which will accomplish the desired transformation in a single call, but these can be a little more difficult to determine.

Having edited and run reorientAndCenter.sh, linkToRotAxes.sh is now run in …/PEET/SingleTomogram/InitialReferenceC6 and a new alignment started. (LinkToRotAxes.sh uses "ln -s" to create symbolic links. Here and throughout this tutorial, users Windows / Cygwin users will need to replace "ln -s" with "ln" to generate hard links). The first iteration is no-search, and the corresponding output reference s(hiv_Ref2.mrc or unMaskedhiv_Ref2.mrc) can be checked as soon as they are created to verify that the previously chosen transformation has oriented and centered the particle as desired; if not, the run can be cancelled, and the transformation adjusted until the results are satisfactory. Then the run can be allowed to complete, with additional iterations included to refine the alignment.

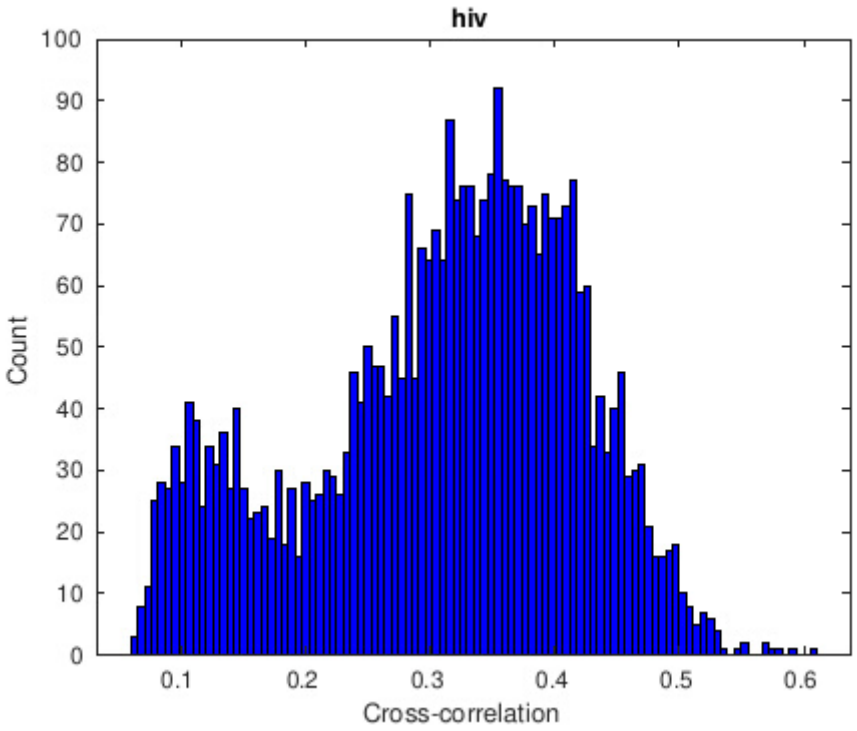
Hexagonal symmetry was confirmed in the first alignment and is now imposed during this and all subsequent PEET runs. Two methods of symmetrization are available. In previous PEET versions, symmetrization was accomplished by symmetry expansion of the particles followed by unconstrained alignment and averaging. This allows some deviation from strict symmetry, but also requires processing a larger, expanded dataset during both alignment and averaging. PEET 1.16.0 introduces a new symmetrization option for cylindrical symmetry like that here. With this option, symmetrization is imposed automatically during all averaging operations (including reference generation), but alignment against the symmetrized reference is done with the original (*i.e.,* unexpanded) dataset. This imposes strict symmetrization and speeds up alignment (although not averaging). The newer approach, termed cNSymmetricAveraging, is used here. This run took 2 hours and 19 minutes with 64 cores and resulted in a nicely centered reference as show below (in this case without masking). Purely for aesthetics, reorientAndCenter.sh was also set to orient the hexagonal cells with vertices along X and flat sides perpendicular to Y.



This reference will next be used to locate and pick candidate particles in all 5 tomograms by template matching. Script …/InitialReferenceC6/linkToInitialRef.sh is run to create symbolic link to this template in EMPIAR-10164-Tutorial/PEET where it can be easily accessed during template matching.

Template Matching

Each of the …/PEET/SingleTomogram subdirectories run1TS01_TemplateMatching to run5TS54_TemplateMatching are processed similarly. Script linkToRotAxes.sh creates a symbolic link to the rotation axes file, and then a PEET alignment is run in template matching mode. After alignment, createScoreHistogram.sh is run to display a histogram of cross-correlation scores, and a threshold to remove poorly matching particles chosen. For example, a threshold of 0.2 is chosen for TS43, shown below, and THRESH=0.2 edited into script split.sh. (Not all cases are clearly bimodal as this. That's especially true at later stages of processing).



Split.sh is then run to split the data pseudo-randomly into halves and to apply the chosen cross-correlation threshold. Matching half sets from all tomograms will be combined, and the 2 halves processed independently, enabling use of gold standard Fourier Shell Correlation. For TS43, for example, files corresponding to the 2 halves will be named with prefixes TS43Ab8C6 andTS43Bb8C6, respectively, where "b8" refers to 8X binning and "C6" to 6-fold cylindrical symmetrization. In the combined motive lists, the particles from TS43 will be assigned to classes 431 and 432 (*i.e.,* 10 * tomogram + half-set). This permits later selection of particles by either tomogram or half-set for operations such as thresholding or histogram computation.

Alignment and Averaging

Bin8 Processing of Half-sets

At this and subsequent stages, we process half sets from all 5 tomograms together rather than separately. In both …/PEET/Combined/run6Bin8C6/A and …/PEET/Combined/run6Bin8C6/B, we

- run linkToRotAxes.sh
- run the PEET alignment

- run createScoreHistograms.sh and chose cross-correlation thresholds
- edit these thresholds into prepareForBin4.sh, and
- Run prepareForBin4.sh to generate files for the next, less highly binned stage

A and B halves were run simultaneously using 64 cores each and took approximately 1 hour and 10 minutes.

An unmasked, gold standard FSC (GSFSC) can be created by running simpleFSC in the parent ../run6Bin8C6 directory and specifying the A and B *AvgVol*.mrc files with the maximum number of particles as inputs. Resolutions estimated by gold standard FSCs extend to Nyquist for all binned runs, however, and were not used to choose subsequent low-pass filtration at these stages.

Bin4, Bin2 and Initial Unbinned Processing

Bin4, Bin2, and Bin1 processing are done sequentially in a similar manner. Progressively tighter masks and smaller physical subvolumes are used both for speed and to focus on the highest resolution central region. Corresponding directories are …/Combined/run7Bin4C6 through …/Combined/run9Bin1C6. Run times using 64 cores each for the 2 half sets were 2 hours and 47 minutes, 6 hours and 37 minutes, and 62 hours and 24 minutes, respectively.

SimpleFSC yields a GSFSC resolution of ~4.4 Å for the unbinned run9 without masking or B-factor correction. A masked GSFSC, without B-factor compensation, could also be calculated with calcUnbaisedFSC. Instead, we run script ../Combined/run9Bin1C6/postprocess.sh which inverts contrast, standardizes gray values, and then uses Relion programs relion_mask_create and relion_postprocess to create a reasonably tight mask, estimate and correct for the B-factor, and to compute the resulting GSFSC. The resulting resolution estimate is 3.9 Å.

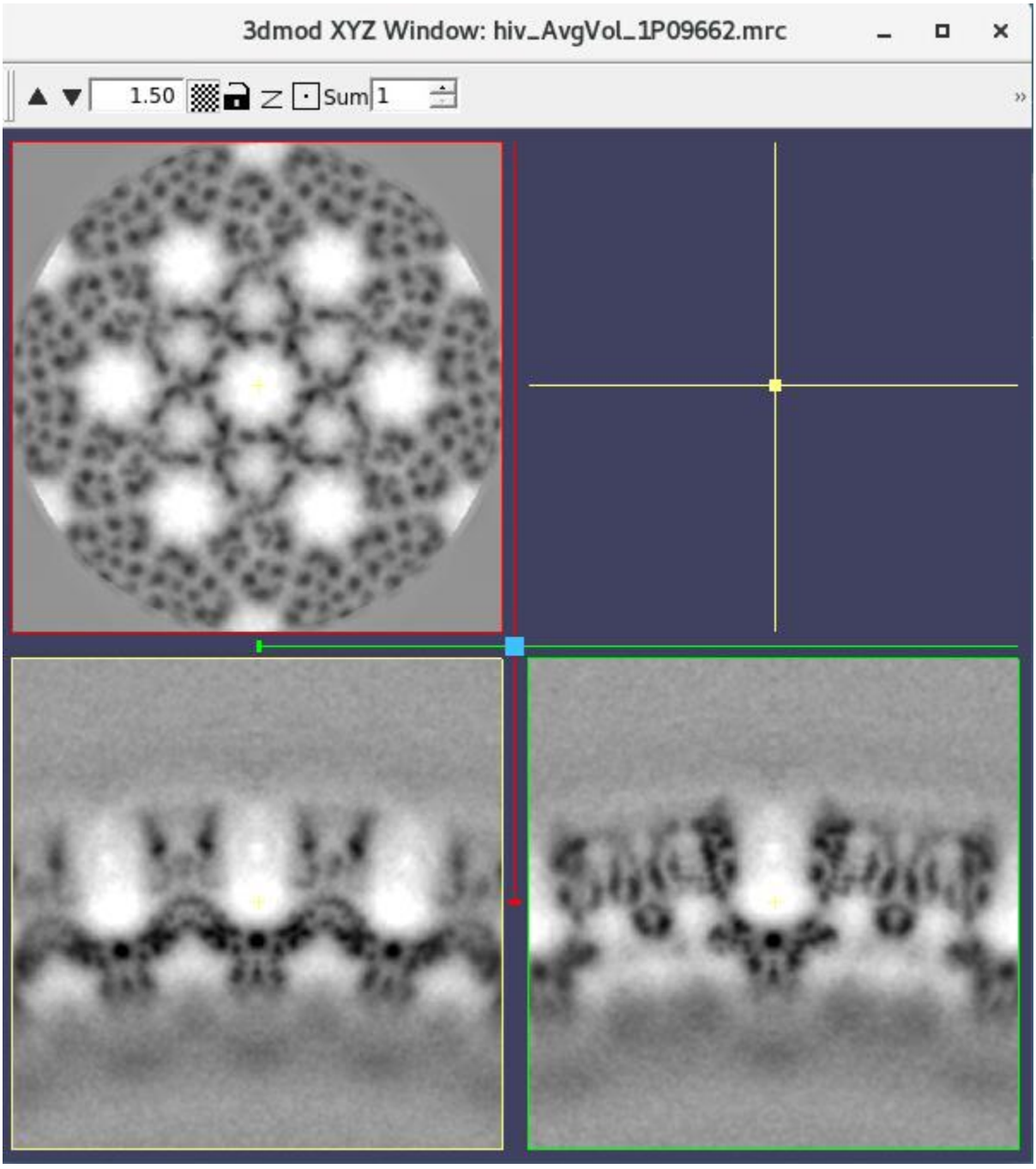
Optional PEET Processing

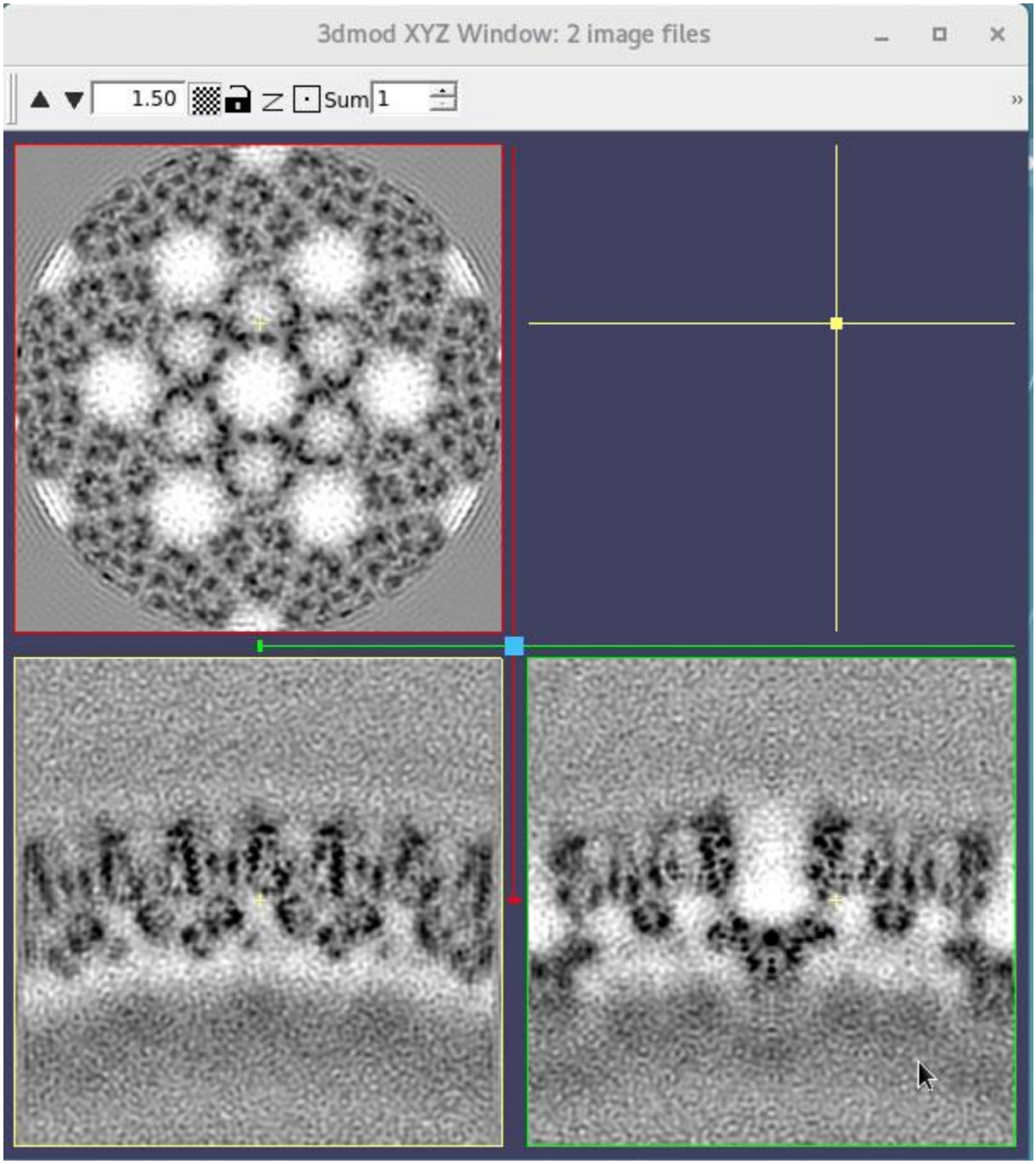
Except for the initial, highly filtered bin8 reference, the 2 half datasets have been refined here independently. This contrasts with the approach used by some other packages in which an appropriately low-pass filtered common reference is used at each binning. As a result, it is possible that the half set alignments here may have yielded slightly different orientations. Additionally, combining half-set averages can be done more accurately by including the known Fourier space weighting… either for the half set averages or the individual particles. Finally, we have not yet tuned the final low pass filtration to closely approximate the shape of the corrected GSFSC curve, minimizing high frequency noise. Jointly, these refinements have only a minor impact, resulting in a resolution improvement ≤ 0.1 Å in this case. Nevertheless, we demonstrate their use here for completeness.

In …/ run10CombineRun9Refs, the 2 half-set unmasked references are aligned, and a common reference generated. Low-pass filtration settings (0.17 / 0.06) were chosen based on the run9 GSFSC using PEET program resolutionAndFrequency to convert resolutions to relative spatial frequencies. Since there are only 2 “particles” this run takes just 20 minutes using 2 cores.

Next, half-dataset averages are created in …/run11Final/A and .../run11Final/B using a single iteration alignment against the common reference. These runs took 9 hours and 12 minutes using 64 cores. Running …/run11Final/postprocess.sh yields resolution and B-factor estimates of 3.8 Å and 150.3, respectively. Finally, the individual particles from the 2 halves are combined in …/run11Final/Both using slightly revised low-pass filtration settings of 0.175 / 0.06, and B-factor correction applied using

applyBFactor.sh. This run took 15 hours and 51 minutes. Results before and after B-factor sharpening are shown below.





Exporting and “Polishing”

Relion 4.0 STA (Zivanov *et al*, 2022) provides features not implemented in IMOD / PEET which can yield further improvements in resolution. These include “CTF refinement” and “Frame alignment” analogous to “polishing” in Relion SPA. We support directly exporting aligned data from PEET to Relion 4.0 (or vice versa) to provide access to these features.

Exporting to Relion

Relion 4.0 STA processing is done in a directory …/PEET/Combined/run11Final/Relion4STA. This directory should contain subdirectories:

- Inputs/ to which data will be exported from PEET and imported to Relion ,
- Masks/ containing files mask_align.mrc and mask_fsc.mrc from the Relion 4.0 STA tutorial, (https://relion.readthedocs.io/en/release-4.0/STA_tutorial/index.html), and
- Tomograms/ which is a symbolic link to ../../../../Tomograms (*i.e.*, …/EMPIAR-10164-Tutorial/Tomograms). (Windows / Cygwin users will need to copy individual TS<nn> directories or link to their contents, since hard links to directories are not allowed).

As above, we've prepared a directory with required input files and executed the following processing steps. Running exportToRelion4STA.sh populates Inputs/ with files TS<nn>{A|B}_coords.star ready for importing into Relion. This script uses PEET programs createAlignedModel to extract data for individual tomograms, modifyMotiveList to compensate for differing symmetry axis orientation in PEET and Relion, and toRelionCoords to do coordinate conversions.

Importing and Processing in Relion

Files …/Inputs/order_list.csv, describing the tilt series acquisition order, and …/tomograms_desc.star, giving tomogram and optics groups parameters, are required for importing, as described in the Relion 4.0 STA Tutorial. Subsequent processing is done via the Relion subtomogram averaging gui, which can be run from …/Relion4STA with "relion –tomo &". Tomogram descriptions are imported in job001, and half datasets from individual tomograms in job002 – job011. Job012 - job014 combine all the imported particles into a single star file. Subsequent processing essentially follows the steps described in the Relion 4.0 STA Tutorial, except that we work with unbinned data from the outset, since we already have an initial alignment from PEET.

Direct reconstruction from the imported data (job015 – job016) yields 4.0 Å resolution after postprocessing, ~0.2 Å worse than obtained in the previous PEEET run. This difference could be due to rounding errors during coordinate conversion or differences in masking and may not be significant. Relion 4.0 STA prefers refining alignments from what it calls "pseudo-subtomograms", which are optimized for computation rather than visualization. After initial pseudo-subtomogram construction in job017, a first 3D refinement is performed and postprocessed in job018 – job019, yielding 3.7 Å resolution. 3 cycles of "polishing" are next, followed by final refinement and postprocessing, ultimately yielding 3.3 Å. Compute settings provided are appropriate for a specific Slurm queue server on a cluster with a limited resources and will need to be altered for use on other systems .

Closing Remarks

We have described processing the 5-tomogram subset of EMPIAR-10164 from raw tilt series movie frames through subtomogram averaging. Motion correction, CTF estimation, and tomogram construction, and modeling were done in IMOD, followed by subtomogram averaging in PEET to 3.8 A. Further refinement in Relion 4.0 yielded 3.3 Å resolution.

Incremental improvements become progressively harder to attain at higher resolution, and the last few tenths of an Angstrom require disproportionate effort. Unfortunately, small (0.1 – 0.2 Å) differences in GSFSC resolution which do not correspond to significant map changes also occur. Among the possible

causes are minor mask changes, imperceptible changes in SNR at cutoff with a GSFSC that plateaus or is multimodal in that region, and the specific choice and number of particles in the half-sets. We recommend verifying improvements by checking for visibility of known features or use of independent measures such as map-vs-model FSC.

High-resolution STA is often followed by atomic modeling and refinement, which is beyond the scope of this tutorial. Nevertheless, it is worth noting that the sharpening B-factor (or B-factors) will have a major impact on the success or failure of this process. Multiple maps with different B-factors, or auto-sharpening (*e.g.*, in Phenix) can be helpful in successfully modeling different parts of a structure which may vary widely in resolution.