Resolving macromolecular structures from electron cryo-tomography data using subtomogram averaging in RELION

Tanmay A M Bharat & Sjors H W Scheres

Structural Studies Division, MRC Laboratory of Molecular Biology, Cambridge, UK. Correspondence should be addressed to T.A.M.B. (tbharat@mrc-lmb.cam.ac.uk) or S.H.W.S. (scheres@mrc-lmb.cam.ac.uk).

Published online 29 September 2016; doi:10.1038/nprot.2016.124

Electron cryo-tomography (cryo-ET) is a technique that is used to produce 3D pictures (tomograms) of complex objects such as asymmetric viruses, cellular organelles or whole cells from a series of tilted electron cryo-microscopy (cryo-EM) images. Averaging of macromolecular complexes found within tomograms is known as subtomogram averaging, and this technique allows structure determination of macromolecular complexes *in situ*. Subtomogram averaging is also gaining in popularity for the calculation of initial models for single-particle analysis. We describe herein a protocol for subtomogram averaging from cryo-ET data using the RELION software (http://www2.mrc-lmb.cam.ac.uk/relion). RELION was originally developed for cryo-EM single-particle analysis, and the subtomogram averaging approach presented in this protocol has been implemented in the existing workflow for single-particle analysis so that users may conveniently tap into existing capabilities of the RELION software. We describe how to calculate 3D models for the contrast transfer function (CTF) that describe the transfer of information in the imaging process, and we illustrate the results of classification and subtomogram averaging refinement for cryo-ET data of purified hepatitis B capsid particles and *Saccharomyces cerevisiae* 80S ribosomes. Using the steps described in this protocol, along with the troubleshooting and optimization guidelines, high-resolution maps can be obtained in which secondary structure elements are resolved subtomogram.

INTRODUCTION

The use of an electron microscope to image macromolecular complexes that are frozen in a thin layer of vitreous ice (cryo-EM) is rapidly gaining in popularity. A single cryo-EM image may contain 2D projections of many copies of the same complex in different orientations, and these 2D projections may be combined in a 3D reconstruction of its scattering potential. This technique, which is known as single-particle analysis, has recently undergone substantial progress with the development of highly efficient direct-electron detectors and improved image processing software. Notably, this technique now allows near-atomic resolution structures to be calculated without the need for crystallization and from as little as 10–100 μg of purified material[1,2].

In ET, multiple images are taken of the same sample region at different tilt angles of the microscope. From such a series of tilted images, a 3D reconstruction, or tomogram, of a single 3D object such as an entire cell[3] may be obtained. This technique thereby provides the unique possibility to image complexes in their native environments. Moreover, if many copies of a complex of interest are present in tomograms, then the reconstructed 3D density corresponding to each complex may be computationally extracted, and the resulting 3D 'subtomograms' may be averaged together to increase the signal-to-noise ratio and thereby produce a higher-resolution 3D structure[4]. This technique is called subtomogram averaging, and it has been successfully applied in numerous cases to reveal biological structures *in situ* or in environments that are otherwise not amenable to single-particle analysis[5–9].

To date, the use of subtomogram averaging is not as widespread as that of single-particle analysis. An important limitation of subtomogram averaging is that the best-resolved structures by this technique are markedly lower in resolution than those from single-particle analysis[4]. Tomographic data collection is slower, and subtomogram averaging requires more complicated image processing, as tomographic reconstruction needs to be followed by alignment and classification of the subtomograms. Furthermore, due to increased effective specimen thickness at high tilt angles, the sample cannot be imaged at high tilt angles, which leads to

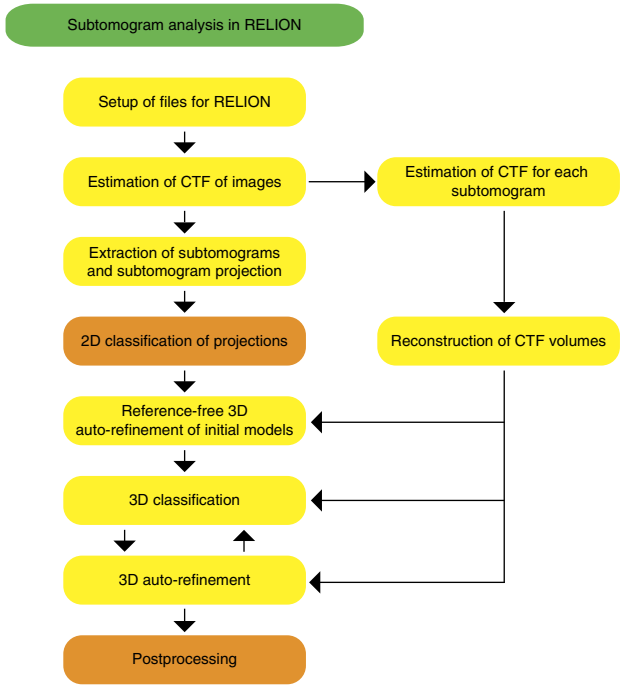

Figure 1 | Workflow of the image processing protocol. A schematic representation of the recommended workflow for subtomogram analysis using RELION presented in this protocol. The main difference between single-particle analysis and subtomogram analysis in RELION is related to CTF estimation and the new 3D CTF model. This 3D CTF model also compensates for the missing wedge and is used in both 3D classification and 3D auto-refinement. Steps highlighted in orange are unchanged from the single-particle analysis workflow.

a wedge-shaped region in the Fourier domain where data are absent. This 'missing wedge' leads to blurring artifacts in tomograms. Still, the advantage of being able to study macromolecules *in situ* (e.g., inside an entire cell) remains extremely attractive. This is powerfully illustrated by the recent application of subnanometer-resolution cryo-ET subtomogram averaging to the HIV-1 capsid[10] and to membrane-bound ribosomes[11]. Further development of both experimental data acquisition procedures[12] and image processing algorithms[13] will continue to drive this technique toward higher resolutions and wider applicability.

Recently, we introduced a new image processing approach to subtomogram averaging[14] that is based on a regularized likelihood optimization algorithm in the RELION program[15,16]. This program was originally designed for single-particle analysis and has been used to calculate numerous near-atomic resolution structures[1]. Because the subtomogram averaging approach in RELION was modeled on the single-particle analysis workflow, existing RELION users will find many similarities (**Fig. 1**). The main deviation from the single-particle workflow lies in the generation of a 3D model for the information transfer in each subtomogram, which is used to compensate for both the missing wedge and the effects of the CTF in the tomogram[14]. A substantial effort was made to build on existing tools inside RELION, rather than writing new tools specifically for subtomogram averaging. This facilitates transitioning between subtomogram averaging and single-particle analysis, and thus it naturally supports a hybrid approach of combining cryo-EM and cryo-ET data[17–19].

In this protocol, we describe the practical use of RELION for subtomogram averaging. Our approach complements various single-particle analysis software packages that also offer functionalities for subtomogram averaging[20,21], as well as multiple specialized packages for subtomogram averaging[8,9,13,22–24]. As many structure determination projects in practice resort to a combination of different software packages, we will explicitly indicate those points in the workflow that are probable points of conversion between alternative approaches. Recommended procedures for single-particle analysis in RELION are described in detail in the online documentation (the RELION wiki: http://www2.mrc-lmb.cam.ac.uk/relion). Using the protocol described here, together with the online documentation, novice users should be able to conduct subtomogram averaging for their own projects using RELION. We assume basic familiarity with Unix/Linux-based systems, and the ability to run provided scripts from the command line.

MATERIALS

EQUIPMENT

- A computer or a computing cluster with at least 64–128 processors, 120 GB of shared memory and 1 TB of disk space
- A RELION installation (version 1.4; available for free from http://www2.mrc-lmb.cam.ac.uk/relion)
- A Python installation (version 2.6 or newer; available from https://www.python.org)
- An IMOD installation (version 4.7 or newer; available for free from http://bio3d.colorado.edu/imod/)
- A CTFFIND installation (version 3 or newer; available for free from http://grigoriefflab.janelia.org/ctffind4)

EQUIPMENT SETUP

Data: tomograms To enter the RELION workflow, we assume that tomograms have been generated in MRC format[25]. Tomogram calculation is not done inside RELION but instead relies on software packages such as IMOD[24], Tomo3D (ref. 26), pyTOM[13] or Bsoft[21]. In the examples below, we used the IMOD package for tilt series alignment, and we used Tomo3D for tomographic reconstruction.

Data: subtomogram coordinates To find the positions of macromolecular complexes within tomograms we either used the MolMatch software for template matching[8] or manually picked particles using `3dmod`[24] and converted the resulting model into a text file using the IMOD `model2point` command.

Data: aligned tilt series The final aligned tilt series in MRC format is required for CTF estimation in RELION for 3D classification and refinement.

PROCEDURE

Arrangement of input files and directories

1| The directory from which all RELION commands and the graphical user interface (GUI) is launched will be known as the project directory (`./`). Make a folder inside the project directory that is called `./Tomograms/`, where all the input data will be stored.

2| Generate a subdirectory in the `./Tomograms/` directory with the name of each tomogram in the data set, and copy all tomograms to their respective subdirectories. For example, if there are two tomograms in the data set, then the locations of those tomograms should be as follows:

```
./Tomograms/tomogram1/tomogram1.mrc
./Tomograms/tomogram2/tomogram2.mrc
```

3| Save the coordinates of the centers of all macromolecular complexes, or particles, in a tomogram in the same subdirectories as the corresponding `.mrc` file. The coordinate files should have the suffix `.coords`, and the prefix should be the same as the prefix of the tomogram name. In our two-tomogram example, the name and location of the coordinate files should be

```
./Tomograms/tomogram1/tomogram1.coords
./Tomograms/tomogram2/tomogram2.coords
```

The coordinates within each file should be written out in a three-column ASCII format, corresponding to the *x*–*y*–*z* position of the center of each subtomogram in pixels. The origin of the tomogram is in the lower left corner (when displayed in IMOD). Each coordinate file should contain as many lines as there are particles in the tomogram. For example,

```
100.0          355.0      200.0
2,034.0      1,100.0      561.0
3,011.0      2,539.0      321.0
```

4| For CTF estimation, place the aligned tilt series in the same subdirectories. For each tomogram, this should be a single MRC stack, and the suffix of the files should be `.mrcs` . It is important that this is the exact same aligned tilt series as the one used for tomographic reconstruction. Again, in our example, these should be called

```
./Tomograms/tomogram1/tomogram1.mrcs
./Tomograms/tomogram2/tomogram2.mrcs
```

5| Along with the aligned stack, also provide the final tilt angles used for tomographic reconstruction (e.g., from IMOD) in a separate text file. Each line in these text files should contain one number corresponding to the final tilt angle assigned to a particular image during alignment. The order of the lines should correspond to the order of the images in the aligned stack.

```
59.44
56.44
53.43
```

Copy these angle files into the same directory.

```
./Tomograms/tomogram1/tomogram1.tlt
./Tomograms/tomogram2/tomogram2.tlt
```

If `.tlt` files with angle values are not provided, then the tilt angles will be read from the extended header of the `.mrcs` file, if this exists, in Step 9 (see below).

6| For each aligned tilt series, create a text file that lists the tilt angles and the accumulated radiation for each image in the tilt series. If the dose on the detector was calibrated, then the values for the accumulated dose can be read from output log files of the microscope data acquisition software—for example, the `.mdoc` file written out by SerialEM[27]. This information will be used to calculate a dose-dependent 3D CTF model (**Fig. 2a**,**b**) that also accounts for radiation-induced damage of the specimen. This text file should have as many lines as there are images in the tilt series. The text file should have two columns: the first one for the refined tilt angle (after tilt series alignment), and the second one for the total accumulated dose in electrons per angstrom squared before collecting the image. If one provides the nominal tilt angles rather than the refined tilt angles, the Python setup script in Step 9 will assign accumulated dose values to the closest refined tilt angle from the `.tlt` file. For example,

```
61.1 0.0
57.3 2.0
54.2 4.0
```

Save these files in the same subdirectories, with the suffix `.order` -

```
./Tomograms/tomogram1/tomogram1.order
./Tomograms/tomogram2/tomogram2.order
```

7| Generate a RELION-type metadata file in the STAR format[16,28], called `all_tomograms.star`, using the following command line:

```
relion_star_loopheader rlnMicrographName
> all_tomograms.star
ls ./Tomograms/*/*.mrc >> all_tomograms.
star
```

The output file lists all the tomograms in the data set, and it should have the following format:

```
data_
loop_
_rlnMicrographName
./Tomograms/tomogram1/tomogram1.mrc
./Tomograms/tomogram1/tomogram2.mrc
```

Calculation of 3D CTF models for each subtomogram

▲ **CRITICAL** For routine application of the steps in this part of the protocol, one can use a single Python script called `relion_prepare_subtomograms.py`, found on the RELION wiki. This script takes the `all_tomograms.star` file as input, and carries out Steps 8–11, depending on the options that the user specifies in its header. For specialized applications or in case of troubleshooting, any of the steps (8–11) may be carried out independently.

8| CTF correction depends on the estimation of a defocus value for each image of the tilt series (i.e., for all the individual images in the `.mrcs` stacks mentioned above). We produce these estimations by calling CTFFIND[29] through the wrapper provided in RELION: `relion_run_ctffind`. Executing this program will be done automatically by the `relion_prepare_subtomograms.py` script. Change the input parameters for CTFFIND in the header of the Python script: specify inputs such as voltage, spherical aberration and detector pixel size based on your data collection experiment.

? TROUBLESHOOTING

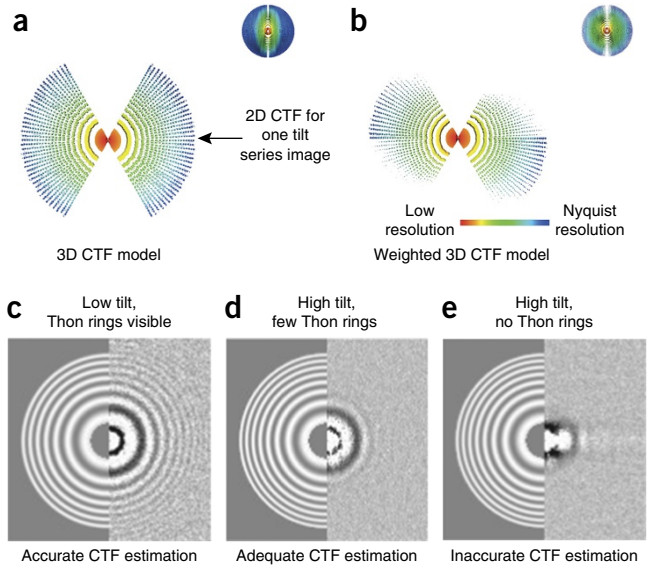

Figure 2 | CTF estimation for the 3D CTF model. (**a**) The unweighted 3D CTF model used in RELION. This model is constructed by placing the 2D CTFs of each image in the tilt series into a 3D volume in Fourier space, with the correct orientation depending on the tilt angle. Therefore, this model also compensates for the missing wedge. (**b**) The weighted 3D CTF model used in RELION. The weighted model accounts for increase in noise at high tilts, and for radiation-induced damage. The volume is colored from red (at low resolution) to dark blue (at the highest obtainable (Nyquist) resolution). (**c**) A diagnostic output file of CTFFIND3 from a low-tilt tilt series image. There is no visible radiation-induced motion, and many Thon rings are visible, making CTF estimation accurate. (**d**) Corresponding diagnostic file from a high-tilt image. Fewer Thon rings are visible because of increased specimen thickness. CTF estimation in this case is adequate but not as accurate as in **c**. (**e**) A diagnostic file from a high-tilt image, in which no Thon rings are visible. CTF estimation is not possible from this image, and it should be removed from the tilt series or the data collection strategy should be modified to include the recording of additional images for CTF estimation on either side of the target region[14].

9| Run the provided Python script from the RELION project directory by typing the following text at the command line:

```
python relion_prepare_subtomograms.py
```

As in single-particle analysis, this script will create a `./Particles/Tomograms/` directory with one subdirectory per tomogram. After CTF estimation using CTFFIND, in each tomogram subdirectory `.star` files corresponding to the CTF model for each subtomogram will be written out. A `particles_subtomo.star` file that lists all the subtomograms, as well as their corresponding 3D CTF models, will also be written to the project directory. This file will be the input for 3D classification and 3D auto-refinement in RELION.

? TROUBLESHOOTING

10| Inspect the diagnostic output of CTFFIND for each image of every tilt series in the data set (**Fig. 2c–e**). This may be conveniently done by using the 'Display' button in the RELION GUI (**Fig. 3**), and selecting the output `_ctffind.star` file for each tomogram—e.g., `./Tomograms/tomogram1/ctffind/tomogram1_ctffind.star`.

▲ **CRITICAL STEP** If CTF estimation failed for some images (**Fig. 2e**), then `relion_run_ctffind` should be executed again for those images, using different parameters. We recommend that either the entire Python script be run again (Step 9)

or users look in the `relion_subtomo_commands.txt` file for the relevant commands for individual tilt series and run them again. Alternatively, one could use an external CTF estimation program, possibly one that allows manual steering, such as EMAN2's e2ctf.py[20], to find a defocus value that fits the observed power spectrum of the image. In that case, the resulting defocus value should be inserted manually into the output `_ctffind.star` file by changing the corresponding values using a text-file editor. In fact, a complete set of defocus values (for all images in the tilt series) pre-estimated using another software could be manually entered into this `_ctffind.star` file.

? TROUBLESHOOTING

11 | At this point, there are three extra options that can be used to improve CTF estimation accuracy. For flat samples, the effective thickness of the ice that the electron beam passes through increases with increasing tilt angle. This results in lower signal-to-noise ratios for the higher-tilt images, which may preclude reliable CTF estimation. If this is the case, the average defocus value measured for the lower-tilt images may be applied to the higher-tilt images, especially if the applied defocus value is stable throughout the tilt series[10]. To do this, set the `UseOnlyLowerTiltDefoci` variable to `True` in the header of the `relion_prepare_subtomograms.py` script and provide a threshold value for the lower tilt.

Another way to estimate the CTF parameters of higher-tilt images more accurately is to acquire two extra images for each setting of the stage in the tilt series. We collected these images along the direction of the tilt axis, and they were spaced equally on either side[14,30] of the region of interest. If such images were collected, then save the extra images in MRC format with a `.trial` suffix in the relevant tomogram subdirectories:

```
./Tomograms/tomogram1/tomogram1.trial
./Tomograms/tomogram2/tomogram2.trial
```

The number of images in the `.trial` stack should be exactly double the number of images in the original tilt series stack, and the order of the images should be the same as that of the aligned tilt series (60°, 60°, 57°, 57° and so on). We do not recommend collecting a single extra image on one side of the region of interest because there could be a systematic focus offset between the extra image and the tilt series image. To run CTFFIND on these extra images, set the `UseTrialsForCtffind` variable in the `relion_prepare_subtomograms.py` script to `True`.

The last method of improving the 3D CTF model is to apply a linear, dose-dependent B factor to the data (also see Bharat *et al.*[14]). On the basis of the observations made for single-particle data sets[31], we increased the B factor by 4 Å^2 for each 1 e$^-$/Å^2 of accumulated dose for both examples described in this article. Users may want to select a different value depending on the radiation sensitivity of their specimen. Change the B-factor variable in the header of the Python script. Note that the tilt-dependent scale factor and the position-dependent defocus value of each particle will be calculated automatically for each image in the tilt series.

12| The command to reconstruct each 3D CTF model is written out in a run script called `do_all_reconstruct_ctfs.sh`, which is automatically generated in Step 9. Run this script from the RELION project directory:

```
./do_all_reconstruct_ctfs.sh 200
```

In the above command, the parameter '`200`' determines the size of the generated CTF volumes in pixels. This should be set to the same values as that used in the 'Particle box size' command in Step 13—i.e., the size of the extracted subtomograms. The above script is a text file containing single commands on each line, and it may thus be split into shorter text files for convenient parallelization—for example, using the `split` command in Unix.

▲ **CRITICAL STEP** The CTF models generated by the above commands will follow the IMOD convention, with the tilt axis of the aligned tilt series being the *y* axis of the image. If the tilt axis is in a different orientation, then the Euler angles used to generate the CTF models will need to be changed accordingly.

Particle extraction

▲ **CRITICAL** Particle extraction can also be performed using an external program such as EMAN2, Bsoft or Xmipp. In this case, Step 13 may be skipped. However, Steps 1–12 will still need to be adequately completed and a `.coords` text file will still need to be provided to generate position-dependent 3D CTF models for each subtomogram.

13| Extract each particle into a subtomogram by performing a 3D window operation on the corresponding tomogram. On the 'General' job-type of the RELION GUI (**Fig. 3**), provide the tomogram pixel size in angstroms, and the diameter of a spherical mask (in angstroms) that will be used for calculating the average and standard deviation of the background pixel values. Make

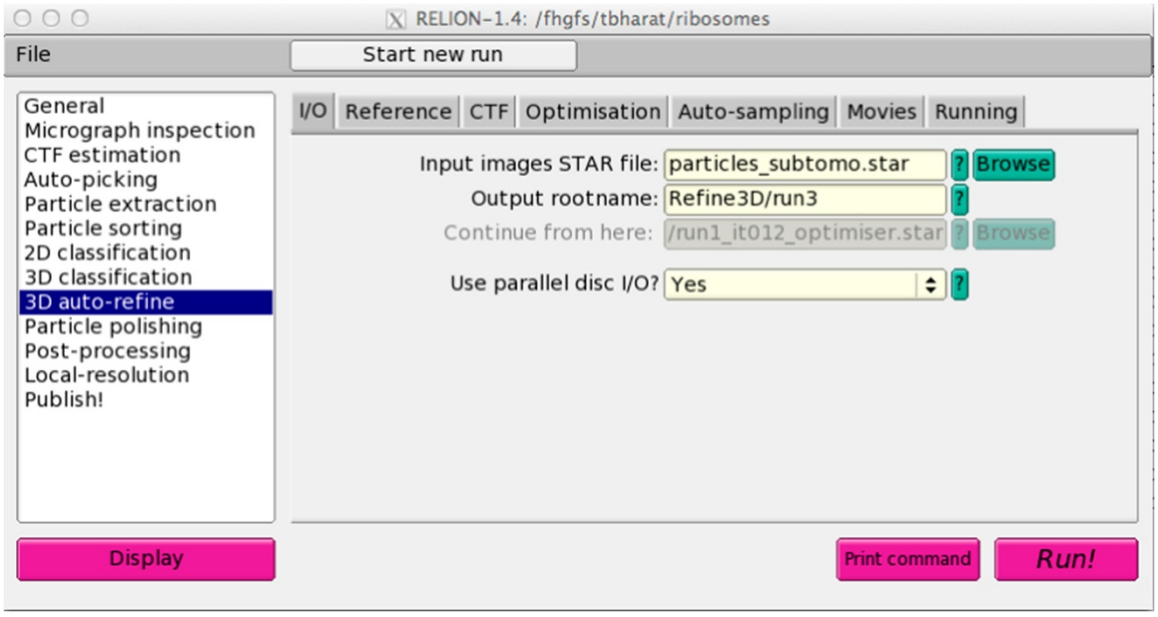

Figure 3 | The RELION-1.4 graphical user interface. After CTF parameters have been estimated for each particle in each image of the tilt series and 3D CTF models have been reconstructed, the actual tasks of subtomogram analysis may all be performed using the RELION graphical user interface. The 3D auto-refine page of this user interface is shown. The white column on the left shows different 'job-types', which are ordered according to the natural workflow from top to bottom. On the main panel, the '3D auto-refine' job-type is shown. This job-type has tabs for 'I/O', 'Reference', 'CTF', 'Optimisation', 'Auto-sampling', 'Movies' and 'Running', where users should enter the input parameters as described in the main text. The 'Display', 'Print command' and 'Run!' buttons are used to view images, commands and launch jobs, respectively.

sure that this mask is slightly larger than the longest diameter of the particle. For the 'Particle extraction' job-type, provide the `all_tomograms.star` file (from Step 7) as 'micrograph STAR file'; set the 'Coordinate-file suffix' to `.coords`; and the 'rootname' to the same as the `RootName` variable in the `relion_prepare_subtomograms.py` script. Set the 'Particle box size' (which is given in pixels) to reflect 150–200% of the particle's longest dimension. If the particles in the tomogram are black, invert the contrast of the subtomograms. If the tomograms were taken with a smaller pixel size than necessary for the target resolution, use the rescaling option to downscale the extracted subtomograms, in order to save computational resources. Running this job (by pressing the 'Run!' button on the GUI) will write out the subtomograms as individual `.mrc` files in the `./Particles/Tomograms/tomogram?/` directories that were created by the Python setup script in Step 9.

14| Along with ordinary subtomograms (which are 3D volumes), 2D projections of all subtomograms (along the *z* axis) may also be calculated by again using the 'Particle extraction' job-type. To do this, run the same job as in Step 13, but provide the extra option `--project3d` on the 'Additional arguments' line of the 'Running' tab. By setting the 'Extract rootname' on the 'I/O' tab to `subtomo_proj2d`, an additional `.star` file called `./subtomo_proj2d.star` will be generated that lists the 2D projections of all subtomograms.

Subtomogram classification and refinement

15| The 2D projections may be used in reference-free 2D classification of the data, much as one would use them in single-particle analysis, which is a computationally cheaper alternative to classification of the subtomograms[14,32]. On the 'I/O' tab of the '2D classification' job-type, provide the `./subtomo_proj2d.star` as the input `.star` file, and set the 'Output rootname', for example, to `Class2D/run1`. The number of classes will depend on the number of subtomograms and the expected heterogeneity in the data. As a rule of thumb, we typically use at least on average 30 subtomograms per class, and we hardly ever use more than 50–100 classes. Because the 2D projections do not have a CTF model, switch CTF-correction off on the 'CTF' tab. Go to the 'Optimisation' tab to select parameters for the classification. We typically perform 25 iterations, and use a regularization parameter of 2 for 2D classification[15]. Typically, we mask particles with zeros, and provide a limit on the resolution of ~10–15 Å to include in the expectation (E-) step of the algorithm. On the 'Sampling' tab, the default parameters are suitable for most cases. Change angular and translational search ranges, if desired. Execute this job by pressing the 'Run!' button. This will result in multiple output files for each iteration in the `./Class2D` directory. The final class averages are stored in a `.mrcs` stack file called `./Class2D/run1_it025_classes.mrcs`. The `./Class2D/run1_it025_model.star` file contains information about the final classes, such as their relative size and the estimated resolution of each class average.

16| Visualize the resulting 2D class averages from Step 15 using the 'Display' button on the GUI and selecting the `Class2D/run1_it025_model.star` file. In the subsequent pop-up window, it is useful to reverse-sort the classes on `rlnClassDistribution`, which will place the largest class averages on the top of the display window. Select good-looking class averages (which reveal recognizable protein features; e.g., **Fig. 4**) by double-clicking the class averages, which will put a red border around the image. Save the new `.star` file containing only the particles that correspond to the good classes by right-clicking the display window and selecting the 'Save STAR with particles from the selected classes' option from the pop-up menu as `subtomo_proj2d_sel.star`.

17| Use the selected 2D classes to write a `.star` file containing only the corresponding subtomograms. We again provide a Python script on the RELION wiki to facilitate this. Run this script from the project directory:

```
python relion_2Dto3D_star.py subtomo_
proj2d_sel.star particles_subtomo.star
```

This script will take two input `.star` files; the first is the `subtomo_proj2d_sel.star` file, containing selected particles from Step 16, and the second is the `particles_subtomo.star` file, containing all subtomograms from Step 9. This script will write out the subset of the subtomograms selected in Step 16 into `output_2Dto3D.star`.

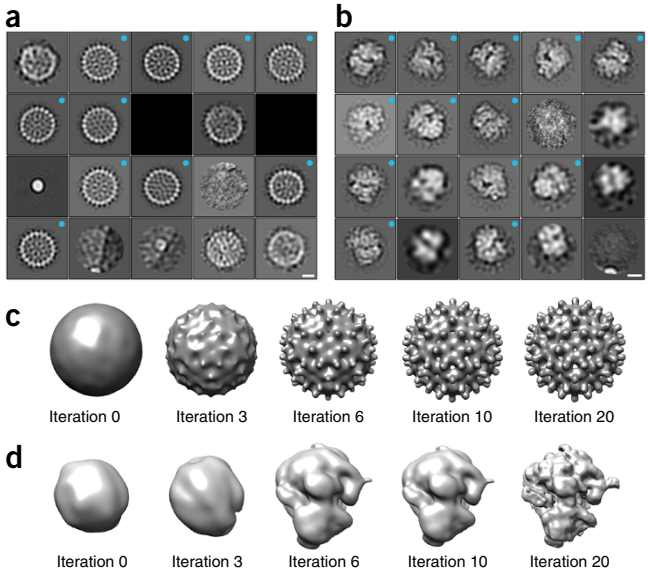

Figure 4 | 2D Classification and initial model generation. (**a**) 2D classification of projected subtomograms of the HBV capsid particles. Particles were selected from a template-matching procedure, and 2D classification helped in removing bad particles—for example, ones that correspond to 10-nm gold fiducials. Good classes that were selected for further processing are marked with blue dots. (**b**) 2D classification of projected subtomograms of *S. cerevisiae* 80S ribosomes. These data were picked manually in IMOD. Classes of ribosomes selected for refinement are marked with blue dots. (**c**) Reference-free refinement of the HBV capsid data set. Subtomograms were assigned random Euler angles initially (in iteration 0) and then refinement was commenced. (**d**) Reference-free refinement of the *S. cerevisiae* 80S ribosome particles, again starting from random orientations. Initial models described in panels (**c** and **d**) may then be used to begin 3D refinements within RELION. Scale bars, 100 Å.

18| Use the particle subset generated in Step 17 to get a subtomogram averaging structure using the '3D auto-refine' job-type. The output rootname on the 'I/O' tab could be, for example, `./Refine3D/run1`. Whereas single-particle analysis requires an input 3D reference map, subtomogram averaging in either the 3D auto-refine or 3D-classification (see next step) job-types may be performed without an initial reference by stating `None` at the 'Reference map' entry on the 'Reference' tab. If one is using an external 3D reference, indicate whether it is on the correct absolute grayscale. In general, maps created by RELION from the same data will be on the correct grayscale, whereas maps coming from elsewhere may not be. Back on the 'I/O' tab, provide the resolution of an initial low-pass filter that will be applied to the input 3D map. To prevent model bias, we typically use relatively harsh filters—e.g., in the range of 40–100 Å. On the 'CTF' tab, turn on CTF correction (provided Steps 8–12 were performed). If CTF correction is desired and a 3D map is provided as initial reference, then indicate whether the input map has been CTF-corrected. For maps from RELION that have been CTF-corrected (depending on whether CTF correction was performed in previous runs), and also for maps that were generated from an atomic model (i.e., reflecting that they do not suffer from CTF artifacts), answer 'Yes' to the question 'Has reference been CTF corrected?'. Go to the 'Auto-sampling' tab to set the initial search step and range. The angular and offset sampling rates and ranges will be used only in the first several iterations. After that the autosampling algorithm will automatically use finer samplings and smaller ranges until the refinement converges[16]. The default parameters will be suitable for most projects, perhaps with the exception of particles with icosahedral symmetry, for which initial angular sampling rates of 3.7 degrees and local angular searches from 0.9 degrees may yield better results. Note that the auto-refinement will divide the data into two random half-sets, each of which will be refined independently, so that in Step 23 'gold-standard' resolution estimates may be calculated[33,34]. Executing this job by pressing the 'Run!' button will output multiple files for each iteration in the `./Refine3D` directory. The final `./Refine3D/run1_half[1,2]_class001_unfil.mrc` files will be used in the postprocessing, as described in Step 23.

? TROUBLESHOOTING

19| Once a 3D reconstruction has been obtained using subtomogram averaging, classify the subtomograms using the '3D classification' job-type on the GUI to detect different conformational states of the specimen. On the 'I/O' tab, select the original `particles_subtomo.star` file, or the `output_2Dto3D.star` from the 2D classification described in Step 17. The output rootname could be, for example, `Class3D/run1`. Because of computational costs, we often use fewer classes for 3D classification than for 2D classification, with typical values in the range of 3–10. On the 'Reference' tab, use the final reconstruction from Step 18 (`./Refine3D/run1_class001.mrc`) as the reference map. This map is now on the correct absolute grayscale, and a similar initial low-pass filter as in Step 18 may be applied. On the 'CTF' tab, indicate that the reference has been CTF-corrected (if this was indeed done in Step 18), and select 'Do CTF correction' to use the combined missing wedge and CTF model (**Fig. 2a**,**b**). On the 'Sampling' tab, the default parameters are again suitable for most projects. Edit the entries in this tab if any changes are desired. Go to the 'Optimisation' tab to enter parameters for the refinement. We typically use similar options as for 2D classification runs, with the exception of the regularization parameter, which should be set to 4 for 3D classification. Execute this job by pressing the 'Run!' button. This will result in multiple output files for each iteration in the `./Class3D` directory. The final reconstructions for each class are stored in `.mrc` files called `./Class3D/run1_it025_class0??.mrc`. The `./Class3D/run1_it025_model.star` file contains information about the final classes, such as their relative size and the estimated resolution of each reconstruction.

? TROUBLESHOOTING

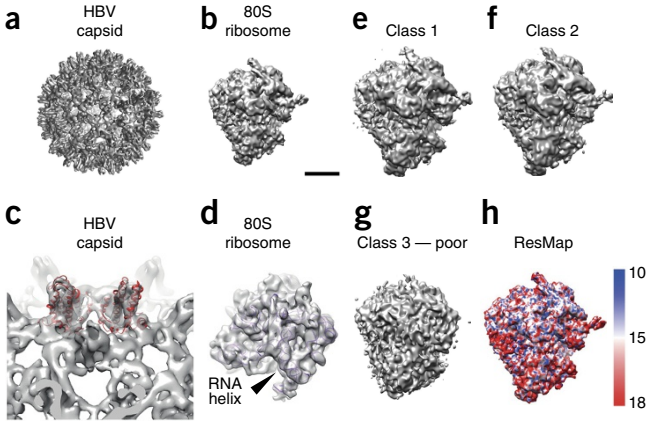

Figure 5 | 3D auto-refinement and classification using the regularized-likelihood algorithm in RELION. (**a**) Output of the 3D auto-refinement procedure from RELION for the HBV capsid data set. (**b**) Output of 3D auto-refinement for the 80S ribosome data set (this map has been deposited at the EMDB under the accession number EMD-3228). Scale bar (applies to **a**,**b** and **e**–**h**), 100 Å. (**c**) Secondary structure features (α-helices) are resolved in the HBV capsid map. Atomic coordinates fitted into the subtomogram average highlight the positions of the helices. (**d**) RNA helices resolved in the 80S ribosome map. The atomic coordinates have been fitted into this map as rigid bodies for visualization. (**e**–**g**) 3D classification of the ribosome data set (with the combined 3D missing wedge and CTF model applied) into three classes (**e**, class 1; **f**, class 2; and **g**, class 3) reveals a subset of particles (~15% of the data set) in **g** that show a poor subtomogram average. (**h**) Removing these particles in a second 3D auto-refinement leads to a cleaner map. The result of ResMap refinement is plotted onto the final density, showing somewhat lower resolution in the small subunit of the ribosome. The color map is defined from blue (10 Å) to red (18 Å).

20| The user again needs to decide which of the classes look good. Use the 'Display' button on the GUI to show 2D slices through each of the 3D maps (e.g., select `./Class3D/run1_it025_class001.mrc` to visualize the first class). Visualization of all classes together in UCSF Chimera[35] is also useful. Good classes typically yield expected protein-like features in the 3D maps, whereas bad classes are often noisy, both inside the region of the particle and in its surrounding solvent background (e.g., **Fig. 5**). Resolution in itself is not necessarily a good indicator of class quality, as smaller classes will be calculated to lower resolution by the intrinsic filtering of the regularized likelihood algorithm.

21| Once good classes have been identified, select the `./Class3D/run1_it025_model.star` file from the 'Display' button on the GUI to display (central slices through) the maps of all classes. Save a `.star` file with the particles belonging to the good classes, as described in Step 16.

22| Sometimes, performing multiple iterations of Steps 19–21, in which selected classes from one run are reclassified in the next run, helps to obtain more homogeneous classes. Thus, refine each class identified in Steps 19–21 again to high resolution using the 3D auto-refinement as described in Step 18.

Postprocessing and resolution estimation

23| After completion of the final 3D auto-refinement for each class, use the 'Postprocessing' job-type to obtain resolution estimates that have been corrected for the influence of a solvent mask[36] in order to calculate for the modulation transfer function (MTF) of the detector and to sharpen the final map. On the 'I/O' tab, one of the two unfiltered half-maps (calculated from the two half data sets described in Step 18) should be provided as `./Refine3D/run1_half1_class001_unfil.mrc`; the output rootname can be set to `run1_post`. Go to the 'Mask' tab to provide parameters for the calculation of an automated solvent mask (automask). The average of the two half-maps will be used to make a binary mask at the specified initial threshold and will be extended by the provided number of pixels; finally, a cosine-shaped soft edge with the specified width

will be added to the mask. Choose an initial threshold value such that the automask does not contain isolated white regions in the solvent area. Often a good estimate for the initial threshold value is the threshold at which a display of the `./Refine3D/run1_class001.mrc` map in UCSF Chimera is noise-free in the region around the particle. Once a suitable automask has been created, it can also be provided as input on the 'Mask' tab instead of calculating a new automask in subsequent post-processing runs. Go to the 'Sharpen' tab to provide an MTF curve for the detector. Curves for some detectors may be downloaded from the RELION wiki. For other detectors, the manufacturer may provide MTF curves. If no curve is available, this entry may also be left empty, in which case MTF correction will be emulated by additional B-factor sharpening. For maps with resolutions beyond 9–10 Å, automated B-factor sharpening[37] may be performed. Alternatively, a user-defined value may be provided. Go to the 'Filter' tab if one wants to skip Fourier Shell Correlation (FSC) weighting. Typically, this option is not used. Execute the post-processing step by pressing the 'Run!' button. This will generate the final map (`./Refine3D/run1_post.mrc`), the automask (`./Refine3D/run1_post_automask.mrc`) and a `.star` file with the applied B factor, the resolution estimate and the corrected FSC curve (`./Refine3D/run1_post.star`). In addition, the corrected FSC curve will be written out as a file called `./Refine3D/run1_post_fsc.xml`, which can be directly uploaded to the Electron Microscopy Data Bank (EMDB).
? TROUBLESHOOTING

24| Because many macromolecular complexes are inherently flexible, even after classification, data sets will often still contain some extent of structural heterogeneity. This will lead to local variations in resolution in the refined map. To estimate local resolution variations, the 'Local-resolution' job-type implements a wrapper to the ResMap program[38]. On the 'I/O' tab, again provide one of the two unfiltered half-maps, and provide the range and step size of the resolutions to be tested by ResMap. One typically does not change the default *P* value. ResMap will provide much more reliable resolution estimates if one provides a suitable mask. The automask calculated in the previous step typically performs well. Click the 'Run!' button to launch the GUI from the ResMap program. In most cases one can just hit 'Continue' inside ResMap, i.e., without adjusting any of its parameters. The result is a file called `./Refine3D/run1_half1_class001_unfil_resmap.mrc`, which can be used inside UCSF Chimera to color the `./Refine3D/run1_post.mrc` map, using menu options `Tools -> Surface Color -> by volume data value`.

25| Sometimes, a refined map still shows signs of large amounts of structural heterogeneity; i.e., it contains regions of relatively low local resolution. In this case, perform a focused classification on that specific region. Provide the `./Refine3D/run1_data.star` file as input for a new 3D classification run, as explained under Step 19. In this run, one could use a solvent mask that is white only in the disordered region. This mask can be generated within RELION by using the command `relion_mask_create` or by using an external program such as `beditimg` from Bsoft[21] or `e2proc3d.py` from EMAN2 (ref. 20). In addition, one could skip orientational searches (through the corresponding option on the 'Sampling' tab), or one could perform relatively fine, but local, angular searches around the input orientations. One could iterate two or more times through Steps 19–25.
? TROUBLESHOOTING

? TROUBLESHOOTING
Troubleshooting advice can be found in **Table 1**.

TABLE 1 | Troubleshooting table.

Step	Problem	Possible reason	Solution
All	RELION executables not found	RELION was not installed correctly, or the shell environment variables were not set correctly	Install RELION 1.4 on your system, and confirm that the shell environment variables are set correctly as indicated on the RELION wiki
	IMOD executables not found	IMOD is not installed on the system	Install IMOD version 4.7 or newer on your system
8	CTF correction is not desired	The target resolution is lower than the spatial resolution of the first zero of the CTF, or it is impossible to estimate and compensate for the CTF	Set the variable `SkipCTFCorrection` in the `relion_prepare_subtomograms.py` script to `True`. In this case a wedge-shaped 3D 'CTF' model will be created that is weighted appropriately depending on the tilt angle and the accumulated radiation
9	The Python setup script crashes with the message 'File not found'	One or more of the input files are missing	Please ensure that all the files needed for Steps 1–7 are present in the correct locations. The .tlt file may be omitted if the .mrcs file has the correct angles stored in its extended header

(continued)

TABLE 1 | Troubleshooting table (continued).

Step	Problem	Possible reason	Solution
	The Python script exits with the warning that the number of tilt angles and the number of images are different	Some images were omitted in the tomogram reconstruction, but the tilt angle values were not updated	Use only the tilt angles for the images that are included in the aligned tilt series and thus in tomogram reconstruction
10	CTF cannot be estimated for some images of the tilt series	Low signal-to-noise ratio in the image, leading to poor visibility of Thon rings	Either follow one of the strategies in Step 11 for accurate CTF estimation or remove the offending images from the aligned tilt series stack. If the latter option is followed, the tomogram will have to be regenerated with that image removed
18 and 19	The job crashes with the error message 'Cannot allocate memory'	The job requires more computer memory than is available, or too many parallel processes were run on the same node	If multiple MPI processes were run on a single computer, run multiple threads instead to share its available memory. More detailed instructions on the hybrid parallelization scheme are available on the RELION wiki
23	The postprocessing program complains that the masked FSC never drops below 0.8	The automasking procedure yielded an all-black, or otherwise unsuitable, mask	Adjust the parameters of the automasking procedure, or make a mask outside the postprocessing procedure. For the latter, one could use any program outside RELION, or use the relion_mask_create program
	Even though the estimated resolution of the reconstruction is high, the reconstruction appears to be low-pass-filtered with no high-resolution features	The tomogram reconstruction was conducted with a simultaneous iterative reconstruction technique (SIRT) algorithm	Try using weighted back projection for tomogram generation, and re-extract the subtomograms, or apply a negative (*ad hoc*) B factor in the postprocessing

● TIMING

The time taken for the procedure depends approximately linearly on the number of tomograms and the number of subtomograms, and it is inversely proportional to the number of processors used on the cluster. Here, for the hepatitis B virus (HBV) capsids, we analyzed a data set containing 15 tomograms (each occupying ~75 GB of hard-disk space). 1,851 capsid particles were extracted in boxes of 240^3 pixels from these tomograms, and the data were processed on a computing cluster with four hyper-threaded 12-core Intel Xeon nodes at 2.9 GHz, each with at least 32 GB of RAM. The nodes were interconnected by a 10-GB Ethernet network. CTF volume reconstruction (distributed over 15 cores, one per tomogram) took 2 h. Subtomogram extraction and projection of the extracted subtomograms took 30 min using a single core. 2D classification of the data in 20 classes took ~3 h on two nodes. 3D auto-refinement was concluded in 24 h, using four nodes. Postprocessing was performed on a single core in 10 min.

For the 80S ribosomes, seven tomograms were used and 3,120 ribosomes were extracted in boxes of 200^3 pixels and projected into 2D in 20 min using a single core. CTF volume reconstruction took 30 min on all nodes; 2D classification into 20 classes took 30 min on two nodes; 3D classification into three classes took 10 h on four nodes; and 3D auto-refinement lasted 50 h on four nodes.

ANTICIPATED RESULTS

We illustrate the results of this protocol for two test data sets. The first set comprised 15 tomograms that were collected on a sample of purified HBV capsids. For this data set, two extra trial images on either side of the region of interest were used for CTF estimation. After optimizing the input parameters for CTFFIND (defocus search range, resolution search range and box size), CTF estimation from these extra images was found to be adequate even at high tilts (**Fig. 2d**). Thus, the arithmetic mean of the defoci was applied to the tilt series image at each tilt. The coordinate files for the HBV capsids were obtained by automated picking using the template-matching routines in the program MolMatch[8]. In addition to true HBV capsids, this program also picked up 10-nm gold fiducials and other undesirable features. Using Steps 13 and 14, HBV capsid particles were extracted in 240^3-pixel subtomograms, and the corresponding 2D projections along the *z* axis were generated. 2D classification, as described in Steps 15–17, readily separated HBV capsids from false positives of the automated picking procedure (**Fig. 4a**). 3D auto-refinement (Step 18), starting from random orientation assignments for all subtomograms,

followed by postprocessing (Step 23), yielded a final map to a resolution of 9.4 Å (**Figs. 4c** and **5a**) from 1,851 particles, in which secondary structure elements such as α-helices were clearly resolved (**Fig. 5c**).

The second test data set of seven tomograms was collected on a sample of purified 80S ribosomes from *S. cerevisiae*. We have deposited this data set together with the results described here at the EMPIAR database[39] under accession number EMPIAR-10045. The initial tomograms and the corresponding aligned tilt series, as well as the `.tlt`, `.order`, `.trial` and `.coords` files, as described in Steps 1–7, are all stored in the `./Tomograms` subdirectory of the EMPIAR entry. The results of all other steps are stored in the `./AnticipatedResults` subdirectory of the EMPIAR entry. Using these, novice users can follow the exact steps described above to replicate, and compare with, the results described here.

CTF estimation was attempted using the same procedures as those used for the HBV capsid data set. However, even after optimizing the inputs of CTFFIND, CTF estimation from high-tilt images was inaccurate in some instances. We believe that this is due to increased specimen thickness resulting from the deposition of a layer of carbon on the grid during sample preparation. Thus, we used the average of the defoci from the lower-tilt images and applied this average to all high-tilt images in Step 11. Manually picked particles were extracted as subtomograms, as well as 2D projections along the *z* axis. 2D classification of the projected subtomograms revealed multiple different views of the ribosomes, as well as some small classes that needed to be discarded (**Fig. 4b**). Analysis of the distribution of the ribosome particles in the tomograms showed that this sample often contained two layers of ribosomes, one at the top and the other at the bottom of the ice layer. Overlap of the projected densities of some of these ribosomes, which would complicate single-particle analysis, is not a problem in the tomographic approach (**Fig. 6**). In this case, 3D auto-refinement of the input 3,120 particles, again starting from random orientations, followed by postprocessing, led to a 13-Å reconstruction (**Figs. 4d** and **5b**). Typical features, such as grooves of RNA helices, were clearly visible in this map (**Fig. 5d**). Using the output of the 3D auto-refinement, we also conducted 3D classification (as in Step 19) of the entire data set into three classes (**Fig. 5e,g**). Although we could not identify different ratcheted states of the ribosomes, we identified a small subset of particles (class 3, **Fig. 5g**) that gave rise to a poor average. Removing these particles from the data set and subsequent 3D auto-refinement resulted in a somewhat cleaner output map, albeit at the same measured resolution. To assess local resolution variations in this map, ResMap analysis was performed as explained in Step 24. The resulting map shows somewhat lower resolution in the small subunit than in the large subunit (**Fig. 5h**).

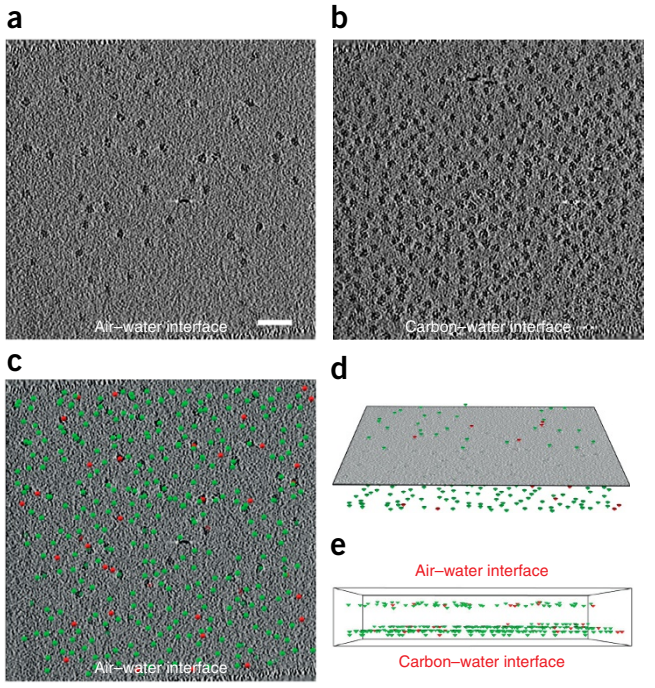

Figure 6 | Subtomogram analysis of particles at different *Z* heights. (**a**) The *S. cerevisiae* 80S ribosomes particles were found to localize at either the air–water or the carbon–water interface. This panel shows a small population of ribosomes at a *Z* slice corresponding to the air–water interface. Scale bar (applies to **a**–**c**), 1,000 Å. (**b**) A tomographic slice at the carbon–water interface showing a surface packed with ribosomes. (**c**) Same slice as in **a** is shown overlaid with a plot of the centers of all ribosomes in the tomogram. A green cone is placed at the center of a ribosome that is sorted into a good class (see **Fig. 4b**), and a red cone is placed at the center of a ribosome sorted into a bad class. This panel shows that in most tilt series images the signal from ribosomes in the top layer is superimposed on the signal from ribosomes in the bottom layer; therefore, tomography and subtomogram analysis is ideal for studying this sample. (**d**) The same image as **c** rotated to show the 3D arrangement of the sample. (**e**) A view of the tomogram along the *xy* plane with the plot of the centers of all ribosomes shown as in **c** and **d**. The edges of the tomogram have been demarcated with solid black lines.

ACKNOWLEDGMENTS We thank X.C. Bai, I.S. Fernandez and K. Vinothhumar for help with sample preparation; J. Grimmett and T. Darling for assistance with high-performance computing; S. Chen and C. Savva for assistance with electron microscopy; and J. Löwe for helpful discussions. This work was supported by funds from EMBO (ALTF 3-2013 and aALTF 778-2015 to T.A.M.B.) and the UK Medical Research Council (MC_UP_A025_1013 to S.H.W.S.).

The data set of seven tomograms of 80S ribosomes from *S. cerevisiae* has been deposited at the EMPIAR database (EMPIAR-10045), and the final map from 3D auto-refinement of these data has been submitted to the EMDB (EMD-3228).

AUTHOR CONTRIBUTIONS T.A.M.B. performed tomographic data acquisition and data processing, and developed the Python script for pre-processing of the subtomograms. S.H.W.S. developed the subtomogram procedures inside RELION. Both authors contributed to writing the manuscript.

COMPETING FINANCIAL INTERESTS The authors declare no competing financial interests.

1. Bai, X.C., McMullan, G. & Scheres, S.H. How cryo-EM is revolutionizing structural biology. *Trends Biochem. Sci.* **40**, 49–57 (2015).

2. Cheng, Y. Single-particle cryo-EM at crystallographic resolution. *Cell* **161**, 450–457 (2015).
3. Baumeister, W. Electron tomography: towards visualizing the molecular organization of the cytoplasm. *Curr. Opin. Struct. Biol.* **12**, 679–684 (2002).
4. Briggs, J.A. Structural biology *in situ*—the potential of subtomogram averaging. *Curr. Opin. Struct. Biol.* **23**, 261–267 (2013).
5. Beck, M., Lucić, V., Förster, F., Baumeister, W. & Medalia, O. Snapshots of nuclear pore complexes in action captured by cryo-electron tomography. *Nature* **449**, 611–615 (2007).
6. Grünewald, K. *et al.* Three-dimensional structure of herpes simplex virus from cryo-electron tomography. *Science* **302**, 1396–1398 (2003).
7. Briggs, J.A. *et al.* Structure and assembly of immature HIV. *Proc. Natl. Acad. Sci. USA* **106**, 11090–11095 (2009).
8. Förster, F., Medalia, O., Zauberman, N., Baumeister, W. & Fass, D. Retrovirus envelope protein complex structure *in situ* studied by cryo-electron tomography. *Proc. Natl. Acad. Sci. USA* **102**, 4729–4734 (2005).
9. Nicastro, D. *et al.* The molecular architecture of axonemes revealed by cryoelectron tomography. *Science* **313**, 944–948 (2006).
10. Schur, F.K. *et al.* Structure of the immature HIV-1 capsid in intact virus particles at 8.8 Å resolution. *Nature* **517**, 505–508 (2015).
11. Pfeffer, S. *et al.* Structure of the native Sec61 protein-conducting channel. *Nat. Commun.* **6**, 8403 (2015).
12. Rigort, A. & Plitzko, J.M. Cryo-focused-ion-beam applications in structural biology. *Arch. Biochem. Biophys.* **581**, 122–130 (2015).
13. Hrabe, T. *et al.* PyTom: a Python-based toolbox for localization of macromolecules in cryo-electron tomograms and subtomogram analysis. *J. Struct. Biol.* **178**, 177–188 (2012).
14. Bharat, T.A., Russo, C.J., Löwe, J., Passmore, L.A. & Scheres, S.H. Advances in single-particle electron cryomicroscopy structure determination applied to sub-tomogram averaging. *Structure* **23**, 1743–1753 (2015).
15. Scheres, S.H. A Bayesian view on cryo-EM structure determination. *J. Mol. Biol.* **415**, 406–418 (2012).
16. Scheres, S.H. RELION: implementation of a Bayesian approach to cryo-EM structure determination. *J. Struct. Biol.* **180**, 519–530 (2012).
17. Bharat, T.A. *et al.* Cryo-electron microscopy of tubular arrays of HIV-1 Gag resolves structures essential for immature virus assembly. *Proc. Natl. Acad. Sci. USA* **111**, 8233–8238 (2014).
18. Bharat, T.A. *et al.* Structure of the immature retroviral capsid at 8 Å resolution by cryo-electron microscopy. *Nature* **487**, 385–389 (2012).
19. Bartesaghi, A., Lecumberry, F., Sapiro, G. & Subramaniam, S. Protein secondary structure determination by constrained single-particle cryo-electron tomography. *Structure* **20**, 2003–2013 (2012).
20. Tang, G. *et al.* EMAN2: an extensible image processing suite for electron microscopy. *J. Struct. Biol.* **157**, 38–46 (2007).
21. Heymann, J. & Belnap, D. Bsoft: image processing and molecular modeling for electron microscopy. *J. Struct. Biol.* **157**, 3–18 (2007).
22. Castano-Diez, D., Kudryashev, M., Arheit, M. & Stahlberg, H. Dynamo: a flexible, user-friendly development tool for subtomogram averaging of cryo-EM data in high-performance computing environments. *J. Struct. Biol.* **178**, 139–151 (2012).
23. Huiskonen, J.T. *et al.* Electron cryotomography of Tula hantavirus suggests a unique assembly paradigm for enveloped viruses. *J. Virol.* **84**, 4889–4897 (2010).
24. Kremer, J.R., Mastronarde, D.N. & McIntosh, J.R. Computer visualization of three-dimensional image data using IMOD. *J. Struct. Biol.* **116**, 71–76 (1996).
25. Crowther, R.A., Henderson, R. & Smith, J.M. MRC image processing programs. *J. Struct. Biol.* **116**, 9–16 (1996).
26. Agulleiro, J.I. & Fernandez, J.J. Fast tomographic reconstruction on multicore computers. *Bioinformatics* **27**, 582–583 (2011).
27. Mastronarde, D.N. Automated electron microscope tomography using robust prediction of specimen movements. *J. Struct. Biol.* **152**, 36–51 (2005).
28. Hall, S.R. The star file–a new format for electronic data transfer and archiving. *J. Chem. Inf. Comp. Sci.* **31**, 326–333 (1991).
29. Mindell, J.A. & Grigorieff, N. Accurate determination of local defocus and specimen tilt in electron microscopy. *J. Struct. Biol.* **142**, 334–347 (2003).
30. Eibauer, M. *et al.* Unraveling the structure of membrane proteins in situ by transfer function corrected cryo-electron tomography. *J. Struct. Biol.* **180**, 488–496 (2012).
31. Scheres, S.H. Beam-induced motion correction for sub-megadalton cryo-EM particles. *Elife* **3**, e03665 (2014).
32. Yu, L., Snapp, R.R., Ruiz, T. & Radermacher, M. Projection-based volume alignment. *J. Struct. Biol.* **182**, 93–105 (2013).
33. Henderson, R. *et al.* Outcome of the first electron microscopy validation task force meeting. *Structure* **20**, 205–214 (2012).
34. Scheres, S.H. & Chen, S. Prevention of overfitting in cryo-EM structure determination. *Nat. Methods* **9**, 853–854 (2012).
35. Pettersen, E.F. *et al.* UCSF Chimera—a visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612 (2004).
36. Chen, S. *et al.* High-resolution noise substitution to measure overfitting and validate resolution in 3D structure determination by single particle electron cryomicroscopy. *Ultramicroscopy* **135**, 24–35 (2013).
37. Rosenthal, P.B. & Henderson, R. Optimal determination of particle orientation, absolute hand, and contrast loss in single-particle electron cryomicroscopy. *J. Mol. Biol.* **333**, 721–745 (2003).
38. Kucukelbir, A., Sigworth, F.J. & Tagare, H.D. Quantifying the local resolution of cryo-EM density maps. *Nat. Methods* **11**, 63–65 (2014).
39. Patwardhan, A. *et al.* A 3D cellular context for the macromolecular world. *Nat. Struct. Mol. Biol.* **21**, 841–845 (2014).