# EMAN2 Reconstruction Tutorial
## Using the Workflow interface

Note that this tutorial was updated on 8/25/2010 to reflect any changes in the interface, etc. It should not be used with versions of EMAN2 prior to 2.0RC3.

## Getting Started

- Make sure you have the latest available version of EMAN2 installed. We recommend using the current snapshot version, and only reverting to the last stable release if you experience problems.

➡ EMAN2 documentation is largely provided via the Wiki at: http://blake.bcm.edu If you wish to edit the Wiki, create an account for yourself, then send email to sludtke@bcm.edu and we will adjust permissions so you can edit. Previously this was an 'open' wiki, but we had serious spam problems, and now have to individually approve new users. You don't need an account to browse the current contents, of course.

➡ **GUI Tips:** Interactions with the interface in EMAN2 are quite similar to EMAN1 in many ways. EMAN2 will work best with a 3-button scroll mouse, though there are alternatives using keyboard modifiers for people using one button mice on Macs.
- In most display windows (plots, 2-D images and 3-D volume display), the middle mouse button will open a 'control panel' for the widget with many options to control the display
- The right mouse button is used for panning in 2-D or 3-D image windows, and can be used to zoom (by shift+dragging), and to reset the zoom (clicking) in plot windows.
- The left mouse button has various purposes in various contexts.
- The scroll-wheel will generally act as a zoom. Use the control-panel for more precise control
- If you have a one button mouse, one of the modifier keys (depending on platform) combined with a mouse click will serve the same role as a middle-click. You may need to try them (alt, command, ctrl, shift) to discover which works on your machine.
- In the control panels, and other places in the EMAN2 interface you may encounter 'ValSliders'. These are widgets where a slider is attached to a text-box with a number in it. Dragging the slider controls the number, and entering a number will change the slider. In addition, the text-box can be used to control the range of the slider and get more precise control. By typing '<value' or '>value' in the text box you can change the limits of the slider. Note that it is also possible to enter values which are outside the current slider range.

## Introduction
EMAN2 can be used at many different levels ranging from high-level task-based workflow interface to writing code in C++ or Python. In this tutorial, we will be focusing primarily on the Workflow interface, which will guide you through the single particle reconstruction process, and keep track of appropriate intermediate data generated during the process in a self-consistent way.

- Open e2workflow.py
  - On Windows
    - double click the appropriate icon on your desktop
    - in the 'EMAN2 Tasks' window, go to the options under 'Utilities' and choose the 'Working Directory' task
    - Browse to the eman2_demo/raw_data directory and hit OK
  - On Mac and Linux

    - open a terminal window
    - cd to the eman2_demo/raw_data directory
    - type the name of the program

➡ Windows users - While you can use the traditional icons to run EMAN2 GUI programs under windows, if you run the commands from the windows command-line (as described for Mac/Linux) you will be able to more easily monitor error messages, and things are less likely to go wrong.

➡ If you did this step successfully you will have two windows : 'Running Tasks' and the other will be titled 'EMAN2 Tasks'. If you hover your mouse over 'Directory', you should see a tooltip showing that you are in the eman2_demo/raw_data directory. The 'Running Tasks' window might be hidden behind the the workflow window.

**Data Storage in EMAN2, Projects**

When you launch the EMAN2 workflow, it will establish a Project in the local directory. Initially this will just be an EMAN2DB directory, but as you proceed, you will see many other files and subdirectories appear. For this reason it is a very bad idea to run the workflow from your home directory. When starting a new project, make an empty directory and run e2workflow.py from there (as described below).

EMAN2 Projects store image data and metadata (data is considered to be images themselves, metadata are parameters, such as defocus, associated with the images) in an internal database rather than the flat files in MRC, IMAGIC, SPIDER, ... format. These are not big 'relational databases' like Oracle or MySQL, but rather embedded databases which exist in regular files on your hard drive. There is no database 'server' process for you to worry about. For most purposes, you will interact with these databases as if they were regular files in on your computer. These database files live in directories called EMAN2DB. It is VERY important that you not go into EMAN2DB directories and manipulate these files directly, except under very specific situations. You can get your data out of the database into regular files in any of the standard file formats at any time using one of several mechanisms (left-clicking in the browser and saying save-as is a popular method). There is a complete description of the database, why it exists, and how to deal with it in the EMAN2 wiki at http://blake.bcm.edu/ Please take a moment and read it. If you do need to manually move or manipulate EMAN2DB directories, it is critically important that you run 'e2bdb.py -c' before doing so, or data loss/corruption may result.

## Setup Project and Import Data

The workflow interface is an expandable tree. Each level of the tree has a form with associated information or parameters, even the levels which just appear to be containers for the levels below them. The first item in the workflow is **Single Particle Reconstruction**. This item can be expanded (press the little '+' button) into a list of the individual stages of the single particle reconstruction process, but it should also be selected itself to provide information about the overall reconstruction you are performing in the local directory.

➡ Note that the database used to store the information about your project resides in the local directory (wherever you are when you run e2workflow.py), so each project must reside in its own directory. The workflow will make a number of named subdirectories (folders) to store specific types of information.

- Select **Single Particle Reconstruction**
    - This will open a window where you enter basic information about your project. The demo data for the workshop has the following parameters : 2.0 A/pix, 200 kV, Cs=1.0, mass=800kDa.
    - For a laptop you may want to use only 1 CPU, even if you have a dual core machine, to reduce heating, but it will make things run slower.

- The workflow will attempt to detect how much RAM you have but this may or may not work on all platforms. This parameter isn't really used at the moment, but you may as well enter the correct value, as it may be used in future.
- Click 'OK' once you have entered the parameters. If you select a different workflow item without saying 'OK', your values will not be stored.

• Expand **Single Particle Reconstruction**, causing 9 subtasks to appear: **Raw Data, Particles, CTF, Particle Sets, Reference Free Class Averages, Initial Model, 3D Refinement, Resolution, and Eulers**.

➡ In most steps you have a choice between using data already in the project, or importing outside data in image files. For example, if you wish to use particles already boxed out using another (non-EMAN) application, you can skip forward to the **Particles** step. However, the earlier in the process you import your data, the more flexibility you will have in recording relevant metadata about the processing you're doing. For example, if you import your CCD frames or micrographs, when particles are selected, EMAN2 will keep track of where each particle came from in each micrograph, offering many possibilities for future analyses that could be performed. This flexibility would be lost if you skip forward in the process.

• There are 2 possible forms for bringing raw data (micrographs or CCD frames) into the project. You can click directly on the **Raw Data** task, or you can use the **Filter Raw Data** subtask. The first option will leave the original data where it is, and simply reference it. There may be some limitations to this approach, and if you ever move the original data, you will no longer be able to access it from the workflow. If you use the **Filter Raw Data** task instead, the data will be (optionally) filtered, and imported (copied) into the project. After this, if you wish, the original files can be deleted. If you keep the original files as well, this approach has the side effect of taking 2x as much disk space at this stage. Select one of the following:
  - Select the **Raw Data** task. You will see a form appear with an empty table. This form is a status display showing you all of the frames which have already been imported. You can add raw data to this form by hitting the 'Browse To Add' button or by right clicking and choosing 'Add'. Note that the right click menu also has 'Remove' and 'Save As' options.
  - Select the **Filter Raw Data** subtask. Click on the 'Browse To Add' button. You should see the sample micrographs in the raw_data directory. Select them all by first clicking on image 1160.mrc, holding shift and then clicking on 1792.mrc (or click in the upper left corner of the table). This should highlight all images. Then hit OK. The browser dialog should close and the selected images should appear in the table on the first form. Make sure that the 'Edge norm' and 'Thumbnails' and 'Associate with project' options are selected. If the particles in your images appear dark on a light background, select the 'invert' option. In addition, if the images are CCD frames you may wish to use the x-ray pixel filter as well. When options are selected, hit 'OK'.

➡ Do not use the x-ray pixel filter option with phase-plate data (if you are lucky enough to have a phase-plate). The filter will not work properly.

➡ The Edge norm option will adjust the images so the mean value around the edge of the frame is zero and the standard deviation of the pixel values is 1. This normalization helps regularize the data for later processing and minimizes problems with brightness and contrast, though additional per-particle normalization may occur later.

➡ Generating thumbnails will save time later when e2boxer is used for boxing.

➡ The invert option should be used if necessary to make your particles white on a dark background. Depending on acquisition method and whether your data is in ice or stain, you may or may not need this. For the workshop demo data, you do not need to invert.

➡ The 'Associate with project' options adds the filtered images to the list of frames in the project - you can see list of the frames in the project by going back to the **Raw Data** task.

➡ The 'Inplace processing' option writes the filtered images back to the input files, which saves on disk space, but has the disadvantages discussed above with simply adding images to the project without importing.

- Check that the micrographs have been correctly imported into the project using the **Raw Data** task. This task will now display the images you just added, so long as you selected 'Associate with project'. If you imported more images later, they would be added to this list. Double-clicking on one of the images in the list will display the image on the screen. You may wish to double check to make sure your particles are white on a dark background (even for negative stain data).

## Particle selection

- Select the **Particles** task. You will see a form appear with an empty data list. This form is a status display showing you all of particles stacks currently associated with the project. You can add particle data to this form by hitting the 'Browse To Add' button or by right clicking and choosing 'Add'. Note that right click menu also has 'Remove' and 'Save As' options. If you have already boxed out your particles using some other program, you can import them to the project here, then skip ahead to the CTF stage.

➡ If importing boxed particles from EMAN1, you should import the raw particles from each micrograph separately. The particles should also NOT have been phase-flipped in EMAN1. The CTF correction method in EMAN2 is completely new, and not compatible with EMAN1. While you could import the phase flipped particles, then disable CTF correction in EMAN2, and produce a phase-only corrected reconstruction, this is not a recommended approach.

➡ Note that whenever possible we have designed the forms to the be flexible enough for you to add and remove data 'on the fly'. For example, you could bypass the earlier **Raw Data** form and go straight to the **Interactive Boxing** and 'Add' the raw data there.

- Expand the **Particles** entry in the EMAN2 tasks window. You should see : **Interactive Boxing, Auto Boxing,** and **Generate Output**.
- Go to the **Interactive Boxing – e2boxer** task and hit OK.
- You should see the e2boxer interface form appear which displays a table. The left-most column lists the image names of the micrographs in the project. The other column, most probably blank at this point, is titled Stored Boxes.

➡ Note that you can double click on any of the image names in the left column and the associated image will appear in an interface for viewing (Try it).

- If you have a decent workstation with a reasonable amount of RAM, you could open all of the micrographs at once, but on a laptop, this probably isn't wise. Choose 2 or 3 images (or just 1 if you have problems with more, implying your machine is really not adequate for this tutorial), enter a boxsize of 128, then hit OK.
- Wait a moment while e2boxer loads.

### e2boxer – the GUI

- You should see at least 3 windows appear.You will need to arrange the windows yourself. On a laptop with a low-resolution display, this can be a challenge. The windows are:
  - The main controller - looks like a regular window; it has an assortment of buttons and text entry boxes
  - the main image display window - shows the 2D micrograph currently selected
  - the particle display window - (with nothing in it) that will eventually show the boxed particles.
  - micrograph thumbnails - This will only appear if you selected 2 or more micrographs in the previous step. Clicking on one of these will select the current image to be boxed.

➡ Note that you can get help in many of EMAN2's interfaces by hitting F1. This will cause your web browser to open an EMAN2 wiki page displaying relevant information.

## Getting started with interactive autoboxing

The particle picker (e2boxer.py when started by itself from the command line) has 3 modes of operation: 'Manual', 'Erase', and 'Swarm'. The program will start in Manual mode, in which particles are picked by hand with no option for automatic selection. Erase mode permits you to mark large regions of the image which should be excluded from automatic picking. Finally, Swarm mode is the interactive automatic particle picker. In Swarm mode you select a few particles manually, then the remainder of the image is automatically selected based on parameters determined from your manually selected images. The more particles you select manually, in general, the more accurate the automatic picking should become.

- ➡ Note that particles picked in Manual mode have no impact on the Swarm picker at all, so if the automatic picker is doing a good job, but it missed a few particles you'd like to add, rather than adding them to the reference set in Swarm mode, you may elect to switch to Manual mode instead, to pick up the missing particles.
- ➡ Also note that the Swarm picker works well on some projects, and less well on others. While we do intend to add other automatic picking options in future, at the moment Swarm is the only option in EMAN2.

You likely want to start with the 'Swarm' picking mode. Select this mode, and start by manually selecting 2-3 particles. You should see a bunch of other boxes appear automatically. The more manual references you pick, the better the automatic selection should become. Some particles may cause the picker to pick too much. In this case, generally picking a few more references manually will help. You can also change the method to 'More Selective' to get fewer particles. Note also that the 'Interactive Threshold' slider may not be 100% functional. We have seen some bugs associated with it, but it is on the list of things to fix.

- ➡ You can delete 'bad particles' by holding down shift and clicking on them. This includes particles that were automatically selected. If you delete one of the references you selected, it will update the autopicking, of course.
- ➡ 'Particle Diameter' is NOT the box size, it is the actual size of the particle, used by the algorithm for picking. You may improve results by changing this value.
- ➡ Bad particles or boxes that contain just noise can be damaging to your reconstruction, as they permit noise/model bias to become stronger. It is much better to miss a few good particles if it permits you to exclude obvious 'bad' particles.
- ➡ One caveat to the above, if the good particles that get excluded are all in one orientation, that would be a bad thing
- ➡ Note that this isn't your only opportunity to eliminate bad particles. Try not to be overly liberal, but realize that you will have another chance to clean up any false positives after CTF phase-flipping.
- ➡ When picking, it can sometimes help for purposes of more accurately centering and identifying particles, to use a box size smaller than the final size you plan to use for processing. This is absolutely fine. You can use whatever box size you like during the picking process. When you get to 'generate output' later, you will be given a chance to select the final box size to be used for processing.

### Building up a boxed particle set with e2boxer

- Now that you are familiar with how e2boxer works, the idea is to select a set of references that results in accurate autoboxing of as many good particles as possible. Once you are satisfied the autoboxing results are as good as you are going to make them, you go through the particles in either window and manually delete bad particles or manually add missed good particles

## Boxing multiple images interactively

➡ Note that this section applies only if you have more than one image loaded into the e2boxer interface.

- Choose the next image in the image thumbnails window. This should load the new image into the main display window, and if you used swarm, you should also observe that autoboxing has occurred, based on whatever autoboxing was done on the last selected image.

## Finishing e2boxer

- Once you are done boxing the images you may be tempted to press the 'generate output' button in the e2boxer control panel, but don't. When you have finished boxing each set of micrographs, simply click Done in the main controller and return to the e2workflow interface. We will deal with 'generate output' later.

- If you have run e2boxer correctly you should be able to click on the Interactive boxing task and observe that the Stored Boxes column now has entries in it. These entries should correspond to the total number of boxes currently stored in the database for the given image.

➡ For the purposes of the workshop before proceeding to the next step, you should have boxed out all of the micrographs in the project.

- Generate boxed particle output by going to the **Generate Output (e2boxer)** task in the **Particles** section of the workflow. You should see a form appear that lists the micrographs you have in the project as well as the number of boxes you have stored for each image in the database.
  - Select all of the images
  - enter a boxsize of 140.
  - make sure the normalize.edgemean option is chosen
  - make sure the output image format is "bdb"
  - Hit Ok. Once again this will spawn processes on your operating system that you can monitor in the Running tasks window.

➡ The box size here, 140, is appropriate for the workshop demo data. When processing your own data, you need to select an appropriate box size yourself. It is critical for accurate CTF correction that the box size be substantially larger than the particle, ideally almost 2x. That is, if a box that just barely contains your particle has a size of 128, you should use a final box size in the 192-256 range. This is larger than typically used in EMAN1, but there is a good reason for it, as you will see later. See: http://blake.bcm.edu/emanwiki/EMAN2/BoxSize for information on 'good' sizes to use to optimize processing speed.

➡ On lower end/old machines , especially Windows machines, you may experience problems and be forced to run the output generating tasks one at time.

- Check that your particle images exist and are stored in the database by going to the **Particles** task. This will display a table that tells you precisely which particles exist, how many there are and what dimensions they have.
- If this table is complete and the dimensions of the particles are correct (140x140) you are ready to go the next stage in the workflow.

➡ Assuming you chose the 'bdb' format your boxed particle output you'll note that the filenames that appear in this list aren't simply "1160_ptcls", but much longer and complicated looking 'bdb:particles#1160_ptcls'. The files embodying this database are stored in the 'particles/EMAN2DB' directory. The specifications for bdb database access are fairly straightforward:

- 'bdb:dbname' which refers to the database (think of it as an image file) in the local directory named 'dbname'. 'dbname' can contain individual images, numbered sets of images, named images and other named metadata
- 'bdb:/path/to#dbname' allows you to specify a database residing in a different directory. Note that '#' is used for the final separation between the path and the database name
- files in the EMAN2DB directory should NEVER be renamed, individually deleted, moved, etc. It can cause the system to become 'confused' and produce a range of seemingly unrelated errors.
- If you really insist on deleting files in the BDB directory, make sure you aren't running any EMAN2 programs, run 'e2bdb.py -c' and only then erase the files. Note that the easiest way to remove files from an EMAN2 database is using the EMAN2 browser.
- For more info on the database please see: http://blake.bcm.edu/emanwiki/Eman2DataStorage

## CTF

EMAN2 uses a substantially different CTF model than EMAN1. EMAN2 uses a data-based background curve. For this process to work properly, and to have phase-flipping work optimally, it is important that the box size used for the particles is somewhat larger than the actual particles. This is the reason for expanding the 128x128 boxes used for particle picking to the larger 140x140 box size. It can be very painful to try to go back and change the box-size after going through the CTF step, so I urge you to use a good box-size from the start. The edges of the images are used in determining the background noise spectrum. After determining the background automatic fitting of defocus and B-factor takes place. The uncertainty in B-factor tends to be fairly substantial, largely because it is only a very approximate representation of the true envelope function of the data coming out of modern FEG microscopes.

➡ EMAN1 used a complicated 10-parameter CTF model, and required a lot of painstaking manual fitting, with several complicated difficult to describe tasks related to determining a structure factor. Almost all of this work has been completely eliminated in EMAN2, but the models are not compatible. While EMAN2 can understand EMAN1 CTF parameters as well, if you want to work with EMAN2, you are much better off going back to non-phase-flipped particles and let EMAN2 deal with the CTF entirely.

- Select the **CTF** task. This will display the list of particles in the project along with any determined CTF parameters (defocus, bfactor, etc) and/or any CTF related output (phase flipped or Wiener filtered data). As per usual, you can 'Browse To Add' particle stacks to this form, and remove unwanted data (right click menu). Note that the table displayed by the **CTF** task is the same as the one displayed by **Particles** task, except that extra columns have been added to display CTF related information. The CTF-related columns are filled in as you determine CTF parameters and write CTF output.

### Determining CTF parameters and generating CTF/Wiener filtered particles

- Expand the **CTF** entry in the workflow (the **CTF** under **Single Particle Reconstruction**, not the standalone **CTF** entry near the bottom of the workflow). You should see four entries appear underneath the CTF entry called: **Automated Fitting , Interactive Tuning, Generate Output, and Generate Structure Factor.**
- Start by running automated CTF determination on your particle data by choosing the **Automating Fitting (e2ctf)** task.
  - A form should appear with a table that lists the particles that you generated in the previous stage of the workflow. Other columns (which are probably blank at this point) are titled Particles On Disk, Particle Dims, Defocus, B factor, SNR and Sampling .
  - You must decide whether to check the 'Auto high pass' checkbox. If selected, this will modify the SNR curve to eliminate the first sharp peak that appears in virtually all single particle data. This

sharp peak can cause some issues leading to incorrect 2-D alignment, and hence classification. However, it is also responsible for some of the contrast which makes particles visible by eye. Do NOT check this box for negative stain data. For purposes of the rest of this tutorial, please check this box.
  - You should also decide on an oversampling factor. For purposes of the demo, 2 is a good value. If you have very far from focus images or small particles, larger values may produce more accurate CTF fitting. For particularly large particles, 1 may be fine. Note that in a later step we will bring this number back to 1 regardless of what you used here.
  - To proceed, select all of the images, make sure the microscope voltage, spherical aberration and the angstrom per pixel parameters are correct, and hit OK. This will cause the workflow to spawn a set of jobs to complete automatic CTF determination. You can monitor the progress of the task (s) in the Running Tasks window.

➡ Note: if you found some of the image parameters to be incorrect, you probably forgot to fill in the form you get when clicking on the top level 'Single Particle Reconstruction' item in the workflow.

- When the fitting tasks are complete, click on the **CTF** task in the workflow. This should display a form that has been updated with automatically determined defocus, bfactor, ... parameters.

➡ Proceed to the next step when you certain that automatic CTF parameters have been generated for all particle data in the project. If necessary, you can rerun automatic fitting on one or more sets.

- Select the **Interactive Tuning (e2ctf)** task. This will display a (by now) familiar looking table that lists particle file names and CTF data. Check this list carefully for any outliers which may indicate a misfit. I recommend quickly checking all of the fits, but at the very least, check any fits with unusual B-factors or defocus values. I also like to check over any images with low integrated SNR values.
- The next thing we want to do is to create an estimated 1-D structure factor function from our data. We don't need to use all of the data for this purpose. You'll want at least 5 images covering a range of defocuses, but you may use as many as you like. Generally using images with high SNR values will produce better results. Look at the list of fit values (you can sort the list by clicking on a column header in the usual way), and select the images you want to use for structure factor creation, and open the interactive CTF tuning interface.
- We'd like the images we use for generating a structure factor to be fit as well as possible. While EMAN2 is not generally sensitive to accurate B-factor values (EMAN1 was), it isn't a bad idea to get it pretty close for the images used in this step. When adjusting B-factors, focus on getting the high-resolution amplitudes to match as well as possible, even if the low resolution amplitudes don't fit well. For good quality images from a mid-range 200kev FEG scope, B-factors will generally be in the 300-600 range. On a high-end 300kev FEG scope you may find B factors as low as the ~200 range. On a low-end non-FEG scope, you may find appropriate B factors to be 2000 or more, depending on many factors.
- If you find an image with an incorrectly determined defocus, adjust the defocus to roughly the correct value, then press the 'refit' button. This will do a local optimization of parameters, and store the new results. If you manually change parameters without pressing refit, then be sure to hit the Save parms button, which will store the changed parameters in the EMAN2 database. The interactive interface also gives you a variety of tools for assessing the quality of the individual images.
- Note that there is also a quality slider, which allows you to manually assign a quality value to each image. This value defaults to 5. A value of 0 indicates an image that you would definitely want to exclude from the reconstruction. In any case, you are not required to use the quality slider at all, but it may help if you have some images which, for example, have a lot of drift or astigmatism you wish to exclude later from the reconstructions. You might wish to consider increasing the images you use for structure factor generation to a '6' so you know which ones you've used.
- Once you have the parameters for the selected images adjusted, close the interactive interface (close the control panel window), and select 'Generate Structure Factor' from the workflow. Select the same

images in this dialog as you just used for fitting (if you set the quality to 6, they will be easy to find), and again, set the oversampling value to 2, then hit ok. This process takes almost no time.

- Due to an unresolved issue in the workflow, once the structure factor generation completes (the running task window will be empty again), exit the workflow completely (close all the workflow and running tasks windows), and launch the workflow again.
- If you like, you can look at the generated structure factor by opening the browser, and double clicking on 'strucfac.txt'. You will probably want to middle-click on the resulting plot window and make the Y axis on a log scale. There isn't much interesting to see here unless you are familiar with small-angle x-ray scattering, but you can do it. Please note that the strucfac.txt file is output-only. That is, if you replace it with a different structure factor file it will NOT be used internally. At the moment there is no easy way to import an external structure factor into the EMAN2 refinement process.
- Now that you've generated a structure factor, go back and rerun the automatic fitting process on all of your data. While this will undo any of the manual fitting you've already done, in most cases it will improve results overall.
- Once the automatic fitting is complete, you may wish to quickly go through the fitting results on all of the images, at the very least, to insure that the defocus values were determined accurately. The automatic fitting routine will be very accurate for most projects, but occasionally you can find a project where some aspect of the images throws the fitting process off. This evaluation can be done very quickly by opening the interactive interface with all frames selected, then clicking on the first image in the interactive interface, and using the down-arrow button to move quickly through them. If the size/zoom/contrast is set well in the 2-D power spectrum display, it is generally pretty easy to see any fitting problems here, then stop and correct them.

➡ You might be tempted at this point to delete images which seem to be not quite as good as the others. The preferred approach in EMAN2 is to simply set them with a low quality. When you generate output from the CTF stage, you can then exclude any with low quality settings. This way, you have to option of going back later and experimenting with using some of this data, and also have some examples of 'bad' data around, which can sometimes be useful.

➡ Note that unlike in EMAN1, accurate per-micrograph B-factors are *not* critical for a good reconstruction or proper CTF correction (though it isn't a bad idea). The automatic fitting routine currently has an internal bias towards B-factors in the ~500 range. We plan in future to provide a way to shift this bias for automatic fitting, but haven't gotten around to it yet.

- When you are happy with all of the fits, select the **Generate Output (e2ctf)** task. In the subsequently appearing form select all of the images you are likely to want to include in your first reconstruction. You may elect to exclude any images you have set a low quality setting for. For purposes of this tutorial so you can see the differences, select all of the check boxes, set the oversampling factor to 1, and hit OK. This will spawn output writing processes that you can monitor in the Running tasks window.

➡ Regardless of what oversampling value you used earlier, use an oversampling of 1 in the generate output step. If you do oversampling in this stage, it will make the phase-flipping process irreversible, and reduce some options you may have at later points in the processing.

- Finally check that your phase flipped and Wiener filtered data exist and are correctly stored in the EMAN2 database. Do this by choosing the **CTF** task and carefully inspecting the displayed table. The number of regular, phase flipped and Wiener filtered particles should match for all image entries in the table as should all particle image dimensions. You may also use the browser to look in the 'particles' directory and examine the raw particles, the phase-flipped particles and the Wiener filtered particles.

## Getting rid of bad particles and selecting data to use

- Expand the **Particle Sets** entry. Select “Build Particle Sets”.
- This interface serves a dual purpose. First, it allows you to manually identify ‘bad’ particles, and second, it is used to generate ‘sets’ of images which are then used for reconstruction. These sets allow you to try doing reconstructions with various subsets of your data without taking large amounts of disk space, and keeping track of exactly what you’re doing. For example you may ask ‘do I get a better reconstruction if I use only the 10 best images, or if I use all 20 images’.
- First, we will learn how to eliminate bad particles. This is an optional process. If you did a very careful job of selecting particles in boxing, you may elect to trust your earlier results and skip this step. If you did little manual ‘fine tuning’ of the original particle selection, then this will give you the opportunity to identify ‘bad’ particles. One ‘bad’ particle can do far more harm than a single ‘good’ particle would do to aid the reconstruction.
- When you select Build Particle Sets, it will first prompt you for a type of particle to use. This is NOT asking you what type of data to use for reconstruction, but is only asking what type of particle you want to look at when interactively identifying bad particles. Wiener filtered particles are almost universally the best choice.
- You will now see a window showing statistics for all of the images you have processed so far. Double-click on one of these images, and a tiled image display window will open.
- Look through the Wiener filtered particles. It should now be much easier to see bad particles as well as particles with other particles too close to them in the box. Left click on each bad particle that you see. You will see a blue mark appear on the particle, and if you look back in the (still open) Build Particle Sets window, you will see the bad particle count increase by one. If you accidentally mark a good particle, just click again to toggle the mark off. When you’re done, just double click on the next image in the Particle Sets window.
- Once you have marked as many bad particles as you like in all of the images, you are ready to make your first set. Close the window displaying particle images, but leave the ‘Build Particle Sets’ dialog open. In the list, you should now highlight all images you want to use for your first reconstruction (don’t double-click. Drag, or shift/control-drag in the usual way).

➡ There are several different schools of thought over what data to use, but there are a few general guidelines. First, if you are targeting high resolutions (towards 4-6 A or beyond), images too far from focus must be removed because the CTF phases cannot be corrected accurately past a certain (specimen dependent) limiting resolution. Second, while including images with very low SNR values will undoubtedly slightly improve the measured ‘resolution’ of your reconstruction, if they are too noisy they will actually just be contributing noise/model bias to your structure rather than actual useful information. As a general rule of thumb, if you can’t see the particles fairly clearly in the Wiener filtered data, it’s probably too noisy to use. Beyond this, you may need to experiment to determine the optimal data subset to use for your reconstruction.

- Once you have highlighted the sets you want to use, give your set some sort of identifying name, and hit OK. Generating a named output stack from this step will create a ‘virtual stack’ file for each of the types of files you have already prepared (original, phase flipped, etc.). The nice thing is that these stacks take very little disk space as they do NOT copy the image.

## Generating reference free class averages (2D refinement)

- Under **Reference Free Class Averages**, there is a single entry : **Generate Classes – e2refined2d**

➡ Note that if you click the **Reference Free Class Averages** task itself you will see a form that lists the reference free class averages currently associated with the project. You can add your own reference free class averages to this form using the usual mechanisms (Browse To Add etc). This can be useful if you want to try generating an initial model using class averages generated outside the workflow (with XMIPP, Spider, etc.).

- Select the **Generate Classes (e2refine2d)** task and hit OK. This will pop up a small form asking you to choose from your regular, phase flipped and Wiener filtered particles (you can also specify files, but ignore this option for the purposes of the workshop). Choose either the Wiener filtered or phase flipped-hp data and hit ok. This will pop up a form with three tabs asking you to specify the parameters required to run e2refine2d.py. The default parameters will produce reasonable results, but to make it run faster, you may want to reduce the number of alignment references to 3, reduce # basis fp to 5, and make sure that 'Shrink' is set to 2 under the Simmx tab. Select the data and hit OK. This will spawn a process that you can monitor in the Running tasks window.

➡ As a rule of thumb, there should be at least 10-20 particles per class, but more is also fine. This should guide your choice of '# classes'. In this case the default is probably ok.

➡ You can leave most of the options in the Simmx and Class Averaging tabs as they are. However, you do want to think about the shrink option in the Simmx page. For the reference free class averages you can shrink the data to save time. The time savings can be quite substantial with larger shrink values, but the quality degrades as well. In some cases, the next step (initial model generation) will work quite poorly with images shrunken by too much.

➡ Tool tips display useful information regarding the specific parameters. Just hold your mouse still over one of the parameters for a few seconds to see the tooltips. Also, don't forget the documentation text at the top of each panel in the dialog. If you want to obtain more information you can also go to the command line and type e2refine2d.py -h.

➡ If you want more information on the particular parameters you can pass to the aligners and/or comparators (in the Simmx and Class Average tabs) go to the command line and type 'e2help.py aligners -v 2' or 'e2help.py cmps -v 2'. Omitting the '-v' will produce a less detailed listing.

➡ You can monitor the progress of the refinement in the 'Running Tasks' window, or by running 'e2history.py' in the appropriate directory.

- Once e2refine2d has finished you will want to view the reference free class averages that were generated. You can do this in a number of ways. The first approach is to choose the Browse option in the EMAN2 tasks window. When the browser pops up you should see a folder called r2d_00. Note that if you have run e2refine2d several times you will most likely see several folders that start with 'r2d_' and end with a two figure number (such as r2d_01,r2d_02). Navigate into the most recent r2d entry (mostly likely r2d_00 at this stage). The reference free class averages are called classes_init, classes_01, and classes_02 etc. The highest numbered classes file contains the final results.
- Another way to view the results of e2refine2d in the workflow is to select the **Reference Free Class Averages** task in the EMAN2 tasks window. There is one entry in this form for every time you have executed e2refine2d. This entry lists the most recently generated class averages (viewable by double clicking on the entry). Note that this form can also be used to monitor e2refine2d processes as they are running, by allowing you to view the most recently generated class averages during the run.

➡ For most projects, this is an ideal point at which to look for structural heterogeneity in your data (not an issue for the workshop sample data). If you see several class averages apparently in near-identical orientations, but with subtly different internal features, this may be a sign that your particle is moving in solution. For example, a molecule like mammalian fatty acid synthase can be observed to move as much as 50 A in solution. It can be difficult in some cases to distinguish between variability and differences in orientation, so you may not get a definitive answer to this question at this stage. If you do observe what appears to be structural variability, keep it in mind as you move on to subsequent steps.

## Making an initial model

There is a lot of controversy in the cryoEM community on this point. Some people feel that initial model generation is the most critical step in refinement, and you need to use difficult and time-consuming experimental methods such as random conical tilt or +-45 degree tilt methods to get an initial model before you can proceed. We disagree with this philosophy. In the vast majority of cases, the simple

approach used in EMAN can give a completely reliable initial model with no additional experiments required. However, there are a few caveats here:

- Handedness cannot be determined from single particle data without tilting. If determining handedness directly from your data is important, and you aren't at high enough resolution for local folds to help, you will have to do some sort of tilt experiment. Personally I find single particle tomography to be the most appealing approach, since it is becoming very standard in most labs nowadays, and issues with direction of rotation, etc. likely have already been dealt with. You simply need to collect a tomogram with sufficient particles that the handedness can be observed. EMAN2 is beginning to incorporate software for 'single particle tomography' as well.
- Heterogeneity is an issue. If you have a particle that is highly heterogeneous, the EMAN initial model strategy is likely to fail to produce a unique answer (since there isn't one). Again, single particle tomography may offer the best solution towards understanding the heterogeneity in your specimen. EMAN1 has a 'multirefine' procedure for refining data with many types of heterogeneity, but this has not yet been reimplemented in EMAN2 (it will be...). There IS a tool in EMAN2 for splitting data into two groups based on ligand presence.
- Poor angular distribution. If your particles have a strongly preferred orientation, especially if this is combined with a low symmetry, there may not be enough information to produce an unambiguous starting model. However, it is also important to note that in this situation, even if you get a good starting model, refinement will also tend to degrade rather than improving the model. To perform a proper 3-D reconstruction, you must have a reasonable number of particles in orientations either spanning the equator of the unit sphere, or along a line connecting the pole to the equator (corresponding to a complete tomographic series). We will discuss this point more in the workshop.

GroEL is actually a fairly difficult case for the EMAN approach, as it tends to be found predominantly in the side view orientation, with only a few end-on views present, and very few particles in between. This leads to a substantial number of potential bad starting models. However, as you will see, identifying bad starting models can be quite straightforward, and in most cases the correct starting model will be obvious, even without prior knowledge of the shape of your particle. In the case of GroEL or GroEL-like molecules, shrinking the data during 2-D class-averaging is particularly detrimental, as it can make incorrect starting models appear to be more reasonable.

- Under the **Initial Model** entry in the workflow, you should see **Make Model (e2initialmodel)**.

➡ If you wish to use an externally generated initial model instead of this process, you can add it/them using the **Initial Model** table using the 'Browse To Add' button.

- Choose **Make model (e2initialmodel)** and hit OK. This will pop up a form displaying the available reference free class averages. This form will also require you to input some parameters such as the number of iterations and the number of models to try (to generate). Enter d7 for the symmetry, select the class averages you wish to use for generating the initial model and hit OK. This will spawn a process (e2initialmodel) that you can monitor in the Running task window.

➡ In some cases it can be a good idea to select a subset of the class-averages produced in the last step. Sometimes there will be some obviously bad class averages, and in general, you only need one representation of any given orientation in the set of class averages. If you have several near-identical class-averages in your classes file, you can make initial model generation run much faster by just keeping 1 or 2 of them. To do this, use the browser and double click on the classes file you wish to reduce. Middle-click on the image window, then select the 'Del' mouse mode. Click on any averages you wish to eliminate. When you're done, hit the 'save' button, and write the results to 'classes-good.hdf' (name isn't important, but the format should be BDB or HDF). Then when you use the **Make model** dialog, add this new file to the list and select it.

- Once the e2initialmodel process has completed, go to the **Initial Models** task in the EMAN2 tasks window. This will show you a table listing the initial models that were produced by e2initialmodel. You can double click on entries to view the results. Before going to the next stage you should probably

decide on your best initial model, this will be used to seed 3D refinement in step 6. In theory, the initial models should be sorted in order of quality, so '_01' will be the best model. In most cases, you will see that the best scoring model will look very much like GroEL, however, due to the small number of 'top' views of GroEL, there is one false positive (obviously wrong) which can sometimes score better than the correct solution. By browsing through the results you can see some of the other failure mechanisms of this method.

- There are additional checks you should make when evaluating the quality of the initial models. Open the browser (from the tasks window, last item), and browse to the 'initial_models/BDB' directory. In addition to the models themselves you will also find a number of 'aptcl' files. These files will allow you to compare each class-average with the corresponding projection of the final 3-D model. While having the projections and class-averages agree well is not absolute proof of a good model, it is a very strong indicator, and if they don't match well, it is an indication of a bad result. You will notice for GroEL that bad models will have very poor agreement between class-averages and projections in certain views.

➡ GroEL is an excellent demonstration case, as it has so many possible failure modes, particularly if a large shrink value is used (due to it's near square shape and strongly preferred orientation). If you don't shrink the data at all, these failure modes will be much harder to observe (a good thing normally). Interestingly, asymmetric particles, like ribosome, work virtually 100% of the time using this method, even with a lot of shrinking) because of its distinctive shape. Regardless, as you can see, detecting 'bad' answers is generally pretty trivial.

➡ You can use the I**nitial Models** table to view models that are generated as the e2initialmodel process is running. Try waiting until the e2initialmodel process has completed to 20% and then open this table.

## 3D Refinement

Under **3D Refinement**, **Run e2refine** should appear

Select the **Run e2refine** task. This will pop up a form asking you to choose the raw input data and optionally the 'usefilt' data. You can choose from your regular, phase flipped, phase flipped-hp and Wiener filtered particles (you can also specify files, but ignore this option for the time being). Choose the phase flipped-hp data for the raw data and (optionally) the Wiener filtered data for the usefilt option. Then hit ok. This will pop up a form consisting of 6 tabs that asks you to specify the parameters required to run e2refine.py.

➡ Running in parallel - Two mechanisms are now supported, local threads and distributed parallelism. The distributed method can be used on most clusters until real MPI support is provided. See the Wiki : http://blake.bcm.edu/emanwiki/EMAN2/Parallel

➡ There are a LOT of parameters. We have tried to select sensible defaults for all of them. Don't forget about tooltips when you need a little more help on a specific option, read the text at the top of each panel, and don't be shy about asking for help.

In the Particles tab:

- Chose the image set to use (probably is only one at the moment)
- Enter the number of refinement iterations (5-6 is good for now)
- Double check the angstrom per pixel and particle mass.
- Select the low mem option on low end machines.
- If you have multiple cores on your computer (and want to use them), you can put 'thread:<n>' in the parallel box (eg - thread:4 for a quad core machine). Otherwise leave it blank, or see the reference above.

In the Model tab:

- Choose the initial model that you think is best. If you forgot, double click on a model to look at it. the _01 model is most likely to be good.
- Make sure the symmetry is correct (d7).
- Select Auto Mask 3d and fill in the associated parameters. Tool tips should be informative. The threshold should probably be 0.8, and the mask dilations should both be about 5% of the width of your particle data. The radius parameter is particle specific, but 30 should be fine for GroEL. NMax can be set to zero or a small integer. See : http://blake.bcm.edu/emanwiki/EMAN2/FAQ/AutoMask

In the Project3D tab:

- Use the 'eman' orientation generator, and don't check the 'Include mirror' option.
- Leaving the orientation distribution method as angle based (at 5 degrees) is fine for this example. You can play with this at a later date. You may want to choose a coarser sampling (9 degrees, for example) when you're testing the workflow, as this will generate fewer projections and hence the refinement process will proceed much more rapidly.

➡ Using too large an angular step can produce insufficient sampling of orientations to produce a good reconstruction. This can often be observed in the FSC curve used to assess resolution after refinement as a curve which starts falling but doesn't reach all the way to zero (and often starts rising again at high resolution). If you observe this, it generally means you need a smaller angular step.

➡ There are several different ways to determine the required minimum angular step. The most conservative estimate is to compute 58 * resolution/particle size. This gives a pretty good lower limit for this value (ie - you shouldn't need to go below this value), but in practice is generally a lot smaller than you really need (which will make your refinements take longer to run, and may even degrade results). Practically speaking, 5-9 is good for initial reconstructions of typical objects at low

(15-25 A) resolution. Most projects will end up with a value in the 2-4 range for a 'typical' intermediate resolution reconstruction. Very high resolution work may require values towards 1.

On the Simmx page:
• For initial refinements with fairly large angular steps (>4), shrink=2 is good for most specimens. 2 Stage Simmx should be zero. Default values for the other parameters should be fine to start, though these can be very important when trying to push resolution.
• For smaller angular steps, you may consider shrink=1 and 2 Stage Simmx=2 (or 3 for large structures). This can make refinements with small angular steps run MUCH faster, but for large angular steps may introduce errors.

In the Class averaging page:
• For initial refinements, the default parameters are probably fine.
➡ Averaging iterations should generally be at least 2. Larger values will reduce initial model bias, but at a cost of somewhat reduced resolution. In early rounds of refinement setting this in the 3-6 range will help get a good low resolution model. It can be reduced to 2 thereafter.
➡ Class separation is like a poor-man's maximum likelihood method. It will take each particle and put it in the N best classes, instead of just the single best class. When used in conjunction with small angular steps, this can produce nicer looking reconstructions, even if resolution doesn't improve. If too large a value is used, measured resolution can be improved with this, but at a cost of rotational 'smearing' of the model. That is, the measured resolution will be better, but the actual resolved features will become worse. If you have a small data set, particularly with low symmetry, it may help to increase this to 2 or 3 even with larger angular steps.
➡ You have a choice of 'averagers' to use. 'ctf.auto' will perform CTF amplitude correction without Wiener filtration. 'ctfw.auto' will perform CTF amplitude correction WITH Wiener filtration on the class-averages. While this is not the most appropriate place to do Wiener filtration (3-D reconstruction is the correct place to do it), using 'ctfw.auto' will produce less noisy looking averages suitable for presentation. In either case, you should use the 'wiener_fourier' reconstructor on the next panel. You may also opt to use the 'mean' averager, which will not perform any CTF amplitude correction. This may be desirable for example, in the case of negative stain data.

In the Make3D page:
• There are 2 possible reconstructors to use: 'fourier' and 'wiener_fourier'. Use 'fourier' with the 'mean' averager on the previous page. Use 'wiener_fourier' with the 2 'ctfc' averagers.
• Pad to should be set to some 'good' box size about 25-50% larger than your particle box size. In this example exercise, 140 -> 168 or 192. For large virus particles, system memory may be a consideration. Make sure you have at least 10 x $pad^3$ bytes of ram available on your computer (eg - for a pad size of 1024, you would want 10 G of ram).
• There are now several choices to make in how to filter the final reconstruction:
    • You can check 'Set SF', which will filter your particle to match the structure factor you determined during CTF correction. This option must be combined with 'filter.lowpass.gauss' (or another lowpass filter) with params = 'cutoff_freq=<x>' where <x> is 1/resolution in A you are trying to achieve. Note that this approach will appear to produce much more resolved structures than you might expect, due to the initial 'set sf' filter. In most single particle reconstructions, this low-pass filter is applied to a model which has already been low pass filtered by its B-factor. If the resolution is pushed past the limits of the data, in some cases this approach can cause some artifacts to gradually occur during refinement (your refinement may appear to grow 'hair'). Just keep in mind that the resolvability of a map is almost 2x the cutoff resolution of a Gaussian filter in Fourier space.
    • You can leave 'Set SF' unchecked, and perform some sort of post-processing filter to 'jib up' the high resolutions (a so called 'inverse B-factor correction'). This can be done with the filter.lowpass.gauss filter used with negative cutoff values.

• Alternatively you could try the 'filter.lowpass.autob' filter. This implements a variation on Richard Henderson's proposed B-factor correction method (basically to pick a B-factor to make the 3-10 A resolution range flat on average), but has caused some funny effects in practice when used iteratively. This matter needs more research still...

➡ The issues involved in the various refinement options are fairly complicated, and the correct choice to make can depend on many factors, including the shape of the particle, the size of the particle, the SNR of the data, and other factors. It is difficult to cover all of these issues in any detail in any single document. If you encounter any confusion on these issues, or are unsure how to optimize your reconstruction of a specific specimen, please feel free to email me (sludtke@bcm.edu), and I would be more than happy to give some problem-specific advice. If the question is general enough, I will add it to the FAQ: http://blake.bcm.edu/emanwiki/EMAN2/FAQ I would much rather spend a few minutes answering questions, than have you publish a bad structure using EMAN2.

Once you have filled in all of the parameters in the run e2refine form hit OK. This should spawn the e2refine process which you can monitor in the Running Tasks window.

As e2refine is running you can click on the 3D Refinement task in the EMAN2 tasks window to see if any reconstructions have been generated. This table will list the most recently generated 3D model for each 3D refinement you previously run or are currently running.

You can explore the output from e2refine using the browser and investigating the directories labeled refined_00, refine_01, etc.

## Resolution

After at least one refinement iteration has completed click on the **Resoution** task. This will display a table that has 4 columns titled 'Refinement Directory', 'Total Iterations', 'e2eotest', and 'e2resolution'. The Refinement Directory column lists directory that store refinement data, you can double click on any of these listed items to look at the convergence plots (FSC curves comparing subsequent ) for the particular refinement.

The total iterations column is just for your convenience, it displays how many iterations took place in the refinement directory.

The e2eotest column is the result of the most recent e2eotest that you have run. The number displayed is what you would get based on the 0.5 FSC criterion. Since we have not yet run e2eotest.py, this column will be blank at the moment. You do this by selecting the **Run e2eotest** task.

There is another alternative resolution measurement available called 'e2resolution', however, this is still considered highly experimental, and is not recommended for use yet.

**The 'Run e2otest' task**

Select **Run e2eotest**. This will launch a form with three tabs titled General, Class averaging, and Make3d. There are different levels at which an even/odd test can be performed. The basic idea is to split the raw data into 2 halves, even numbered particles and odd numbered particles. Ideally you would then run completely independent refinements using each half of the data and compare the results. This would test not only for resolution (noise level in the final model) but also to some extent test for initial model/noise bias. However, in practice this thorough test is rarely if ever used in practice.

The simplest version of the test is to keep all of the already-determined particle orientation/alignment parameters and recompute a 3-D reconstruction for each half of the data. This test will not make any assessments of model bias, but roughly assesses the noise levels present in the final model. Due to model bias, masking artifacts, inadequate sampling, etc., this method is subject to many artifacts, but is by far the most common method in use, due to it providing the highest measured resolutions for a given model.

The method used in EMAN2 is between these two extremes. Particle classification information from the entire data set is retained, but class-averages are recomputed for even and odd numbered particles in each class, and these averages are then reconstructed. So, while 2 of the 3 Euler angles are predetermined, the 2-D alignment parameters are recomputed independently for the 2 halves of the data, and model bias will have less influence on the final curve.

- In the General page choose your refinement directory and the iteration which will be used for the eotest. In general you should select usefilt if you used filtered data in the refinement. Enter the correct symmetry and check lowmem on less powerful machines.
- You can probably leave everything as is in the Class averaging page (in theory this should match the parameters you used during refinement)
- In the make3d page you can leave everything as is, but make sure to choose a good number for the padding parameter (see hints in the **3D Refinement** section above)

Once you're done hit OK. You can monitor progress of the task you just launched in the Running Tasks window. Once the e2eotest job is completed, open the **Resolution** form and double click on the directory where the eotest was just executed. This will enable you to view the newly generated FSC curve, along with any other convergence plots, in the EMAN2 plot window. The e2eotest result will appear as a thick line, while the convergence plot will use thin lines.

Ideally the resolution curve will start at 1.0 then at some resolution begin falling towards zero. It should actually reach zero, then will oscillate around zero to Nyquist frequency (the highest resolution on the plot). In practice the curve may not do this entirely. The most common issues are:

- The FSC plot will fall, and flattens out, but doesn't reach zero. The 'flat' region is at a higher value. If this is relatively close to zero (0.05 or 0.1), your resolution estimate is still probably reasonably accurate. This phenomenon is typically caused by using too tight a mask around your structure, or some sort of noise bias due to specific refinement parameters you used.
- The FSC plot falls to some value, then starts moving up to larger values at higher resolutions. In some cases it will actually fall all the way to zero before going up again. The typical culprit here is using too large an angular step. You will generally find that as you decrease the angular step, the curve becomes more and more 'healthy' looking. If the curve falls all the way to zero before rising, likely your resolution estimate is still fairly accurate, but prior to publication, you will likely need to deal with the issue properly (a smaller angular step), before reviewers will let you publish.
- The FSC plot is 'noisy' looking with a number of large, jagged peaks. You may find the curve falling to below 0.5, only to quickly rise above 0.5 again before finally falling towards zero. This is usually caused by using an inadequate number of different defocus values, and is a real effect rather than a reconstruction artifact. The only real 'solution' is to collect more data.
- There are other possible artifacts, and other possible causes for the artifacts listed above, but this list should help get you started.