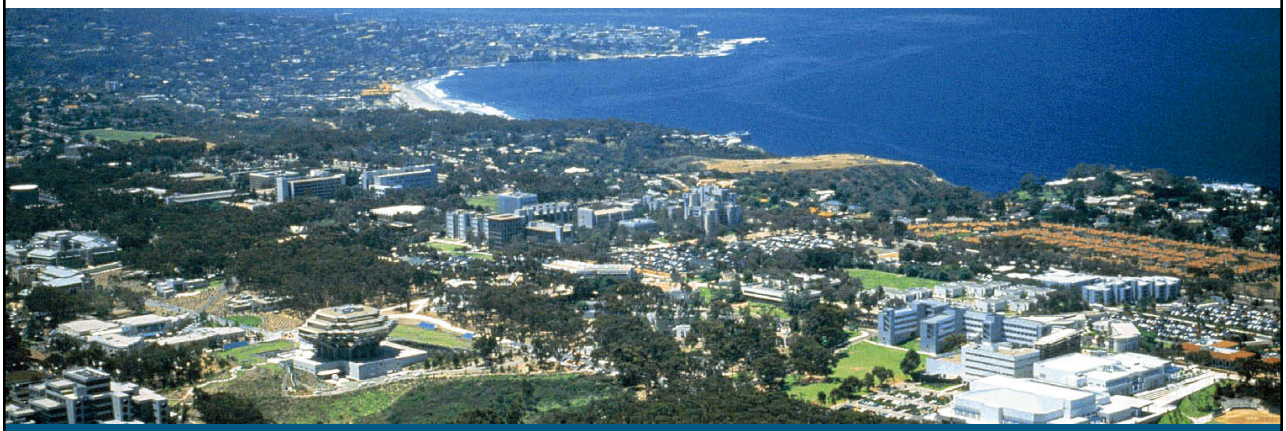
Segmenting Large Electron Microscopic Image Volumes

An Introduction to NBCR image analysis and segmentation tools



National Biomedical Computation Resource Summer Training Program @ UC San Diego

Chris Churas & Matthias Haberl (original instructions created by Alex Perez P.H.D)
August 7th, 2017
churas@ncmir.ucsd.edu & mhaberl@ucsd.edu

Hands-On Session 4

The Post-processing of Automatically Generated Segmentations

Goals

By the end of this session, you will be able to:

1. Convert the output of CHM automatic segmentation jobs to 3D models
2. Apply filters to the automatic output to improve its accuracy
3. Merge models of different organelles
4. Obtain quantitative metrics (volume, surface area, centroid) of all segmented organelles

1. Convert Mitochondrial Probability Maps to MRC

A. Download the *download_dataset_4.zip* file from https://github.com/CRBS/nbcrtrainingvm/blob/master/download_dataset_4.zip?raw=true and unzip it as before. Enter this new directory.

B. The directory will contain three items:

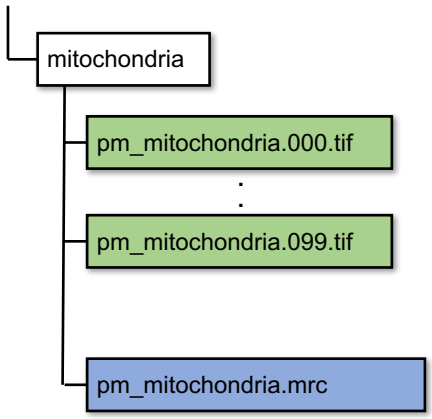
   i. An MRC stack, *sbem_neuron_2dbin10.mrc* that has 100 slices and is centered on the soma of a neuron. To keep file sizes manageable, the data were downsampled laterally by a factor of 10.
   ii. A directory, *mitochondria*, which contains the CHM probability maps for mitochondria.
   iii. A directory, *nuclei*, which contains the CHM probability maps for the nuclei in this dataset.

C. The probability maps have already been converted from PNG to TIF. As of version 4.7, most IMOD programs can process TIF images in the same manner as MRC images. Therefore, we can directly append the TIF files to a single MRC stack. Use the following commands:

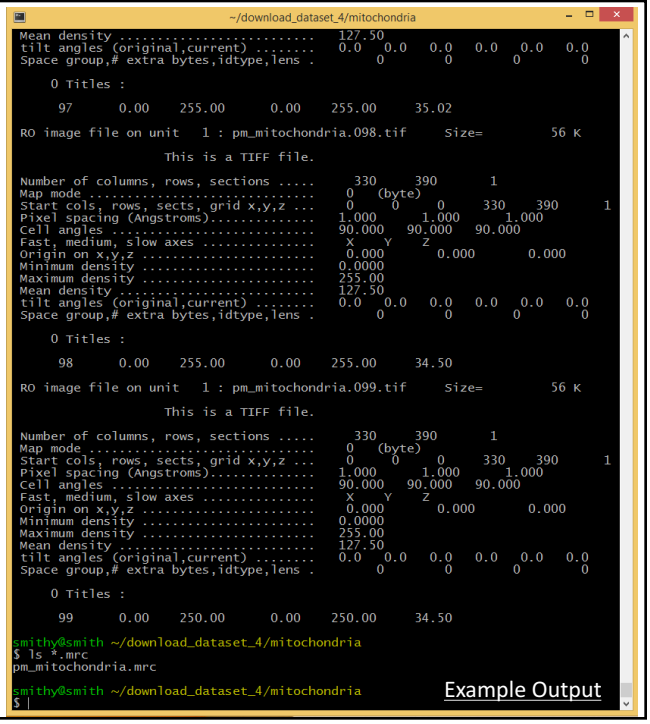
```
cd mitochondria
newstack *.tif pm_mitochondria.mrc
```

This will append all of the newly made TIF files, in alphanumerical order, to a new MRC stack.

Directory Structure:



Note: The set of TIF files and the MRC stack are interconvertible forms of the same data. If this were a full SBEM dataset, each of these could approach 100s of GBs to TBs of disk space. Therefore, it would make sense to delete the set of TIFS after stacking. For the sake of this tutorial, we'll just keep all files since they are relatively small.



Example Output

2. Alter the Header Info of the Converted MRC Stack

A. View the header information of the raw image stack:

header ../sbem_neuron_2dbin10.mrc

Note that the Pixel Spacing entry is "389.9 389.9 300" and the Origin entry is "0 0 0". While the origin is the same as that of the newly created *pm_mitochondria.mrc*, the pixel spacing is not (it is set to "1 1 1" by default). We will correct this using the IMOD program *alterheader*.

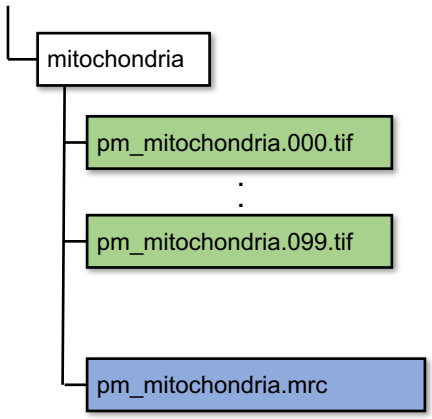
B. At the command line, enter the following: *alterheader pm_mitochondria.mrc*

C. This will launch an interactive prompt. Choose to change the pixel spacing by typing the option *del*, and hitting enter. Then, enter the desired pixel spacing values, separated by spaces, and hit enter. To finalize, type *done* and hit enter. You should be returned to the command line.

To recapitulate, the entries for the entire process are:

alterheader pm_mitochondria.mrc
del <Enter>
389.9 389.9 300 <Enter>
done <Enter>

Directory Structure:



Example Output

```
~/download_dataset_4/mitochondria

smithy@smith ~/download_dataset_4/mitochondria
$ alterheader pm_mitochondria.mrc

 OLD image file on unit   2 : pm_mitochondria.mrc     Size=      12569 K

 Number of columns, rows, sections .....     330     390     100
 Map mode ..............................       0   (bytes - signed in file)
 Start cols, rows, sects, grid x,y,z ...       0       0       0     330     390     100
 Pixel spacing (Angstroms)..............   1.000   1.000   1.000
 Cell angles ...........................  90.000  90.000  90.000
 Fast, medium, slow axes ...............       X       Y       Z
 Origin on x,y,z .......................   0.000       0.000       0.000
 Minimum density .......................  0.0000     (  -128.00     in file)
 Maximum density .......................  255.00     (   127.00     in file)
 Mean density ..........................  33.180     (  -94.820     in file)
 tilt angles (original,current) ........   0.0   0.0   0.0   0.0   0.0   0.0
 Space group,# extra bytes,idtype,lens .       0       0       0       0

     1 Titles :
NEWSTACK: Images copied                                  26-Jul-17  11:16:55

 If you make a mistake, interrupt with Ctrl-C instead of exiting with DONE
 Options: org, cel, dat, del, map, sam, tlt, tlt_orig, tlt_rot, lab, mmm,
 rms, fixpixel, feipixel, fixextra, fixmode, invertorg, setmmm, real, fft,
 ispg, 4bit, toggleorg, fixgrid, help, OR done
 Enter option: del
 Alter delta - changes cell sizes to achieve desired pixel spacing
 Current delta x, y, z:    1.0000       1.0000       1.0000
New delta x, y, z: 389.9 389.9 300
 Options: org, cel, dat, del, map, sam, tlt, tlt_orig, tlt_rot, lab, mmm,
 rms, fixpixel, feipixel, fixextra, fixmode, invertorg, setmmm, real, fft,
 ispg, 4bit, toggleorg, fixgrid, help, OR done
 Enter option: done

 RO image file on unit   3 : pm_mitochondria.mrc     Size=      12569 K

 Number of columns, rows, sections .....     330     390     100
 Map mode ..............................       0   (bytes - signed in file)
 Start cols, rows, sects, grid x,y,z ...       0       0       0     330     390     100
 Pixel spacing (Angstroms)..............   389.9   389.9   300.0
 Cell angles ...........................  90.000  90.000  90.000
 Fast, medium, slow axes ...............       X       Y       Z
 Origin on x,y,z .......................   0.000       0.000       0.000
 Minimum density .......................  0.0000     (  -128.00     in file)
 Maximum density .......................  255.00     (   127.00     in file)
 Mean density ..........................  33.180     (  -94.820     in file)
 tilt angles (original,current) ........   0.0   0.0   0.0   0.0   0.0   0.0
 Space group,# extra bytes,idtype,lens .       0       0       0       0

     1 Titles :
NEWSTACK: Images copied                                  26-Jul-17  11:16:55

smithy@smith ~/download_dataset_4/mitochondria
$
```

3. Generate Contours From the Probability Map Stack

A. For this course, we will apply a single-level threshold to the probability maps and generate contours around the output. The IMOD program *imodauto* is used to do this. The following *imodauto* arguments are useful to use and understand for this purpose:

- -h Pixel intensity value for the high threshold. Contours will be drawn about all pixels above this value. Since the data are 8-bit, values ranging from 0 to 255 are valid.
- -u Uses unscaled pixel intensity values for the threshold cutoff. Highly recommended.
- -R Proportion of points to remove from the final contours. This can have the effects of producing smoother contours and reducing file size. Useful values typically range from 0.25 – 0.5.
- -k Smooths the probability map stack with a kernel filter whose Gaussian sigma is given by the entered value.

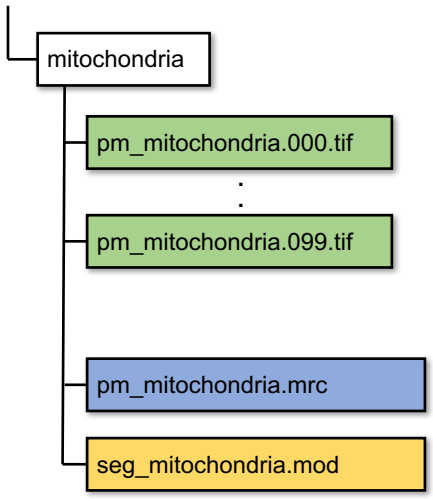
For these data, a good segmentation can be achieved using a 50% cutoff for pixel intensity (= 128):

imodauto –h 128 –u pm_mitochondria.mrc seg_mitochondria.mod

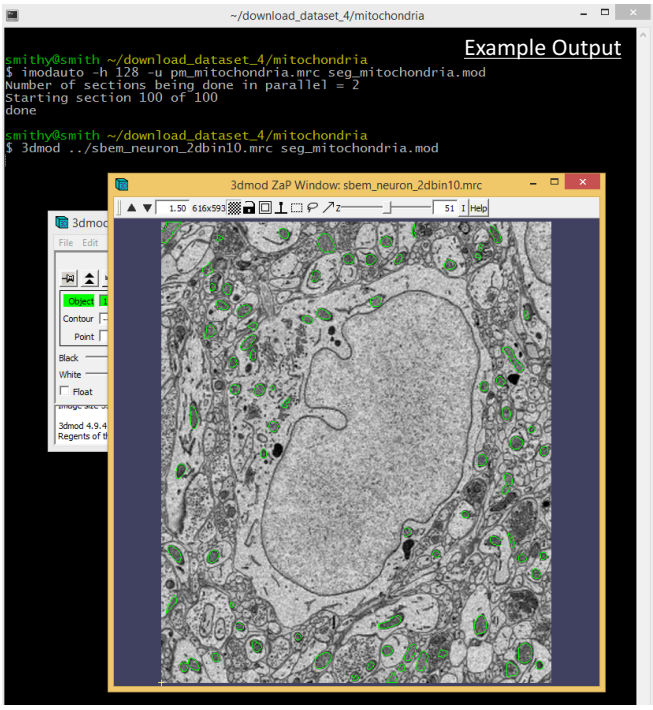
B. Though the above command will yield good output, I recommend taking some time experimenting with different combinations of arguments and visualizing how they affect the output.

For example, increasing the high threshold value will reduce false positives, but may also increase false negatives by being too stringent. Similarly, check how the *–R* and *–k* arguments change the output.

Directory Structure:

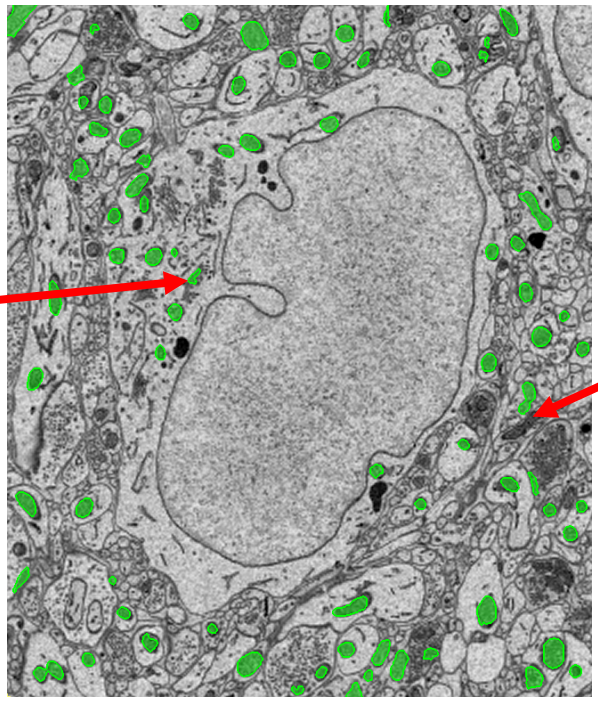


Example Output



Some Common Segmentation Errors:

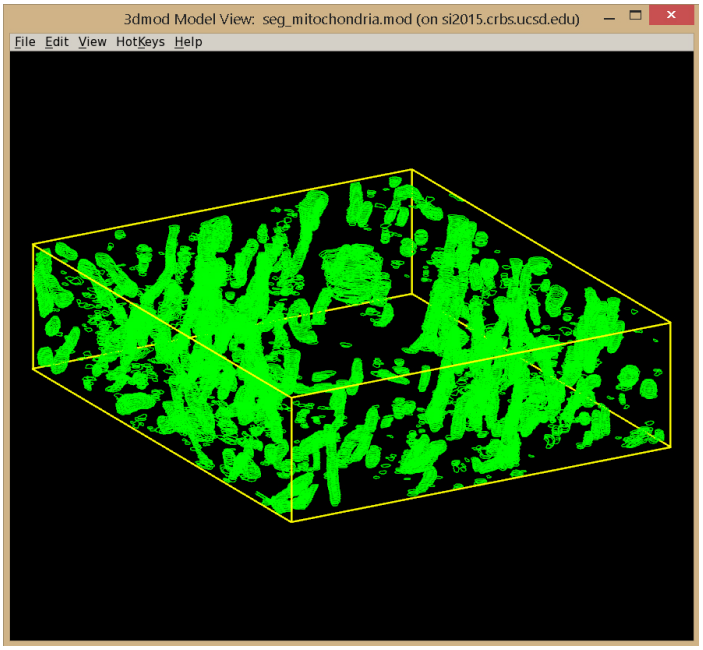
False positive. In this case, a piece of the Golgi apparatus was erroneously segmented as a mitochondrion.



False negative. In this case, a mitochondrion in a neurite that should have been segmented has not been.

Model View:

Note: Bring up the 3dmod Model View by going to Image → Model View, or by pressing the 'v' hotkey.



If you right-click on a contour in the 3D Model View, the 2D ZaP Window will automatically re-direct to its slice and contour. You can use this to see what some of the errors are (e.g. 3D objects with only one contour).

4. Create 3D Meshes from Contours

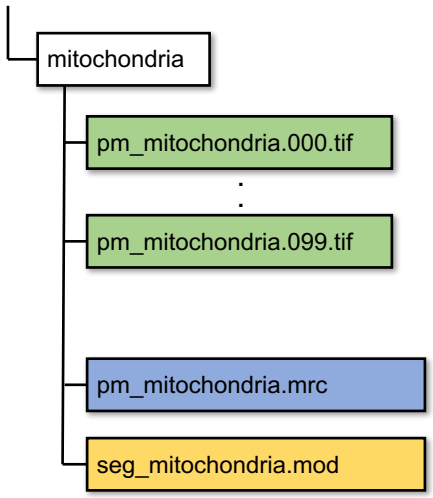
A. You've probably noticed that *imodauto* outputs all contours as part of one, massive object. This is not what we want; ideally, we'd like each spatially separated mitochondrion to be its own object. To achieve this, we first need to generate a mesh for the model file. This is done using the program *imodmesh*.

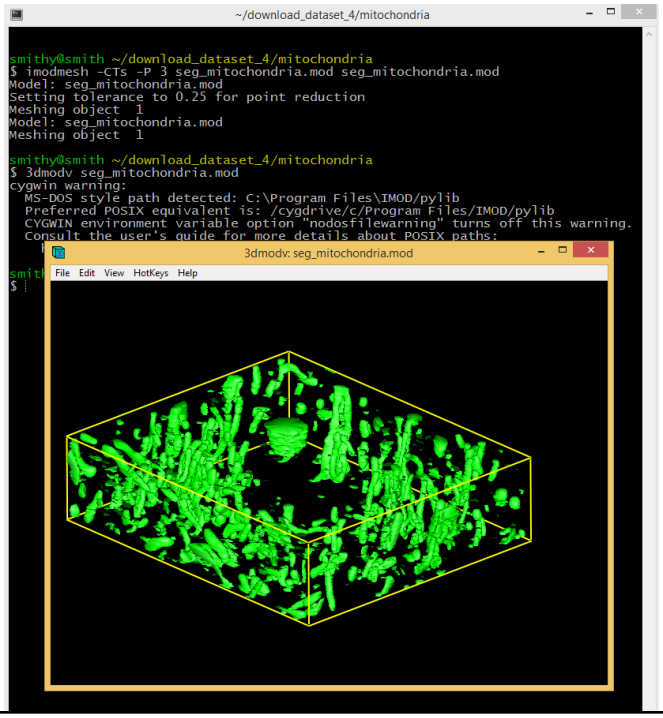
There are many optional arguments for *imodmesh*, and it is a good idea to review them on the program's man page. For now, let's mesh the contours using the following command:

imodmesh –CTs –P 3 seg_mitochondria.mod seg_mitochondria.mod

The most important options to note here are the –s and –P arguments. The combination of both will causes meshes to be made across slices of the object that have missing data. An object containing a slice with missing data is common, and may occur due to artifacts of the imaging process. The integer value entered after the –P flag tells imodmesh to skip up to three sections when forming meshes. If this value is too large, you run the risk of erroneously connecting spatially separated objects that shouldn't be joined together (false merge error).

Directory Structure:





5. Split the Model into Spatially Separated Objects

A. Once the single object has mesh information, it can be split into separate objects based on 3D connectivity by the *imodsortsurf* algorithm. Let's run *imodsortsurf* on the data, using the –s argument to create new objects:

imodsortsurf –s seg_mitochondria.mod seg_mitochondria_sort.mod

B. Once the process has completed, you can visualize the split objects using *3dmod*. But first, you will need to re-mesh the objects to update the mesh info to reflect the colors of the new, sorted objects:

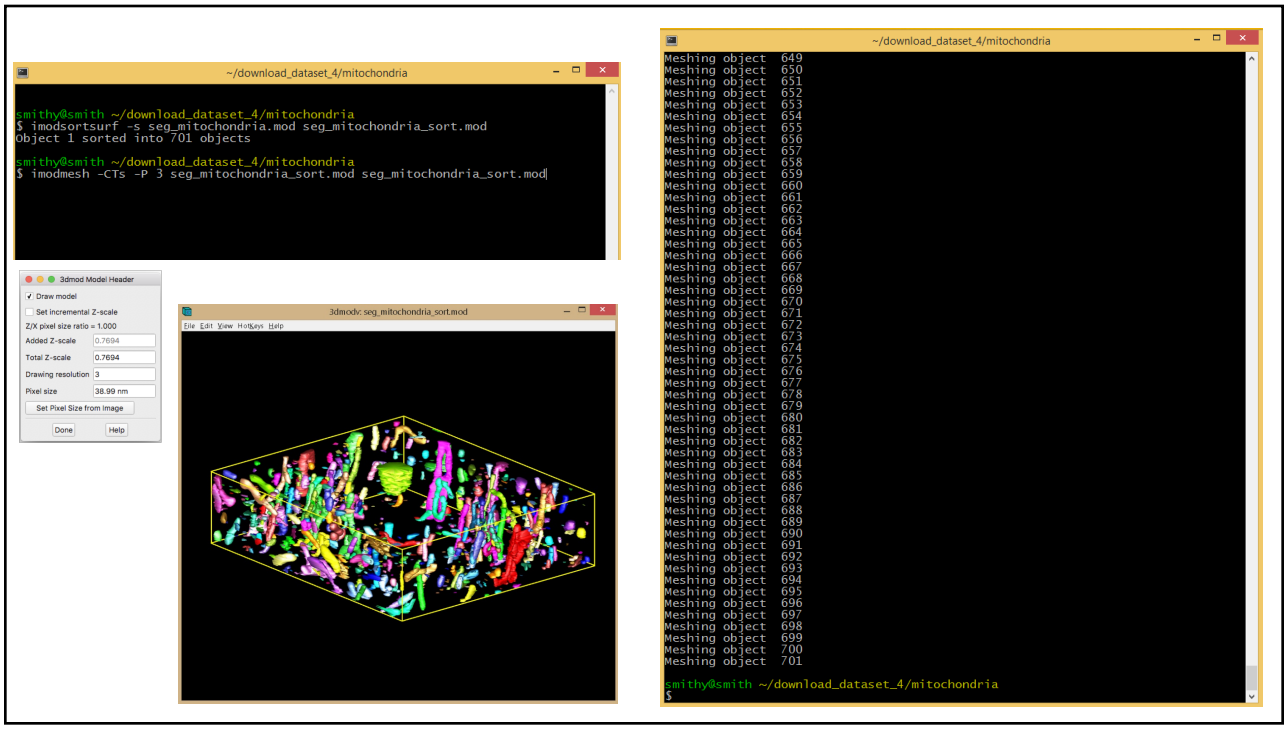
imodmesh –CTs –P 3 seg_mitochondria_sort.mod seg_mitochondria_sort.mod

C. Open the model file in *3dmod*:

3dmod seg_mitochondria_sort.mod

D. Make sure the header information for the model file is set properly. From the 3dmod window, navigate to Edit → Model → Header. Set the 'Pixel Size' and 'Total Z-scale' to correspond with the values of the MRC stack entered when the *alterheader* program was run. These values should be (yes type *nm*):

Pixel Size = 38.99 nm
Total Z-scale = 0.7694 (which is equal to 300 / 389.9, the ratio of Z:XY pixel sizes)



6a. Filter the Objects Based on Morphology

Using the suggested parameters, the above step split the model file into 701 putative mitochondria. Obviously, there are not that many mitochondria in this small volume. Many of these objects are false positives, and can be removed by applying some simple morphological filters. As a starting point, we'll remove all objects that have less than a certain number of contours.

Doing this through the IMOD GUI is very tedious, and IMOD doesn't have any command line programs to automate the process. We will use a set of Python classes to automate this filtering on the Rocce cluster. First, we need to copy our model file to this virtual machine.

A. Copy the file to your folder on Rocce:

scp seg_mitochondria_sort.mod <USER>@rocce.ucsd.edu:/data/scratch/<USER>

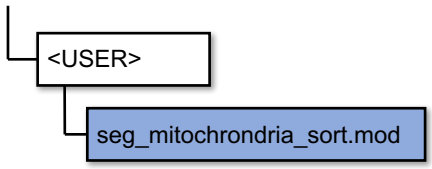
Replace the "<USER>" with your assigned name. Enter your password when prompted.

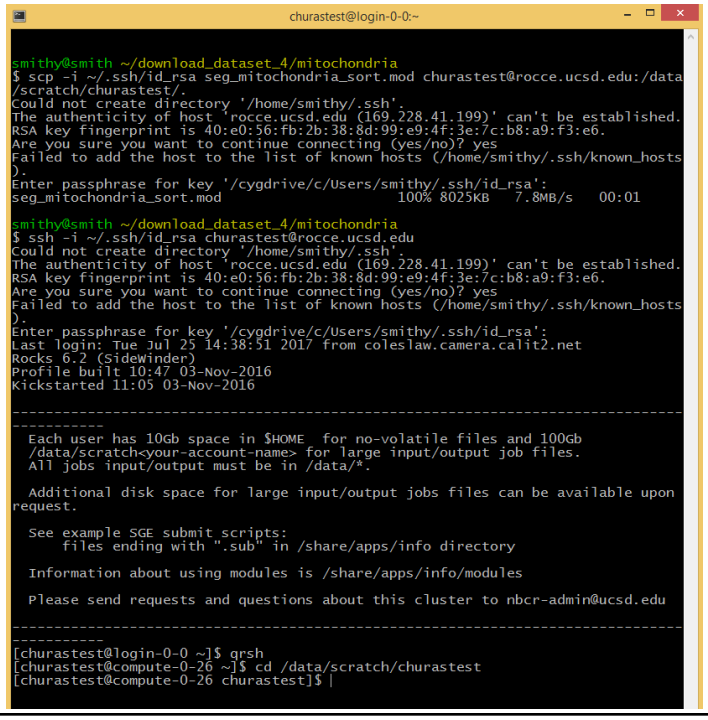
B. Connect to the virtual machine and connect to compute node on Rocce cluster:

ssh <USER>@rocce.ucsd.edu
qrsh
cd /data/scratch/<USER>

Again, enter your password when prompted and replace "<USER>" with your assigned name.

Directory Structure:





6b. Filter the Objects Based on Morphology

C. Clone the PyIMOD repository from GitHub:

git clone https://github.com/CRBS/PyIMOD.git

D. Install numpy (Only need to do this once)

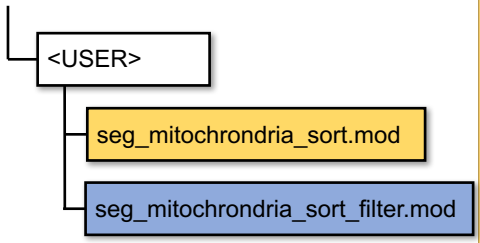
pip install numpy

E. Start Python, import PyIMOD, and perform some basic filtering of the model file (don't type the >>> characters they are the Python prompts):

```
python
>>> import PyIMOD
>>> mod = PyIMOD.ImodModel('seg_mitochondria_sort.mod')
>>> mod.filterByNContours('>', 3)
>>> PyIMOD.ImodWrite(mod, 'seg_mitochondria_sort_filter.mod')
>>> exit()
```

Starting with the 'import PyIMOD' line, this will import the PyIMOD module you cloned. Next, the model file you copied to the virtual machine is read into a ImodModel Python object. The next line performs the filtering. This particular command will keep all objects that have greater than 3 contours, and remove all objects that have less than or equal to 3 contours. This is a very quick and dirty way to filter out potential false positives. Finally, the filtered model file is written to a new file under a different name, and then Python is exited.

Directory Structure:




```
churastest@login-0-0:~
[churastest@compute-0-26 churastest]$ git clone https://github.com/CRBS/PyIMOD.g
it
Cloning into 'PyIMOD'...
remote: Counting objects: 643, done.
remote: Total 643 (delta 0), reused 0 (delta 0), pack-reused 643
Receiving objects: 100% (643/643), 116.70 KiB, done.
Resolving deltas: 100% (370/370), done.
[churastest@compute-0-26 churastest]$ pip install numpy
Collecting numpy
  Downloading numpy-1.13.1-cp27-cp27mu-manylinux1_x86_64.whl (16.6MB)
    100% |################################| 16.6MB 29kB/s
Installing collected packages: numpy
Successfully installed numpy-1.13.1
[churastest@compute-0-26 churastest]$ python
Python 2.7.13 |Continuum Analytics, Inc.| (default, Dec 20 2016, 23:09:15)
[GCC 4.4.7 20120313 (Red Hat 4.4.7-1)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
Anaconda is brought to you by Continuum Analytics.
Please check out: http://continuum.io/thanks and https://anaconda.org
>>> import PyIMOD
>>> mod = PyIMOD.ImodModel('seg_mitochondria_sort.mod')
>>> mod.filterByNContours('>',3)
>>> PyIMOD.ImodWrite(mod, 'seg_mitochondria_sort_filter.mod')
>>> exit()
[churastest@compute-0-26 churastest]$
```

6b. Filter the Objects Based on Morphology

E. Exit from compute and head node on Rocce and copy the filtered model file back to your laptop
Be sure to include a space and the period character at the end of the *scp* command below (yes:

exit
exit
scp <USER>@rocce.ucsd.edu:/data/scratch/<USER>/seg_mitochondria_sort_filter.mod .

F. Re-mesh the model file:

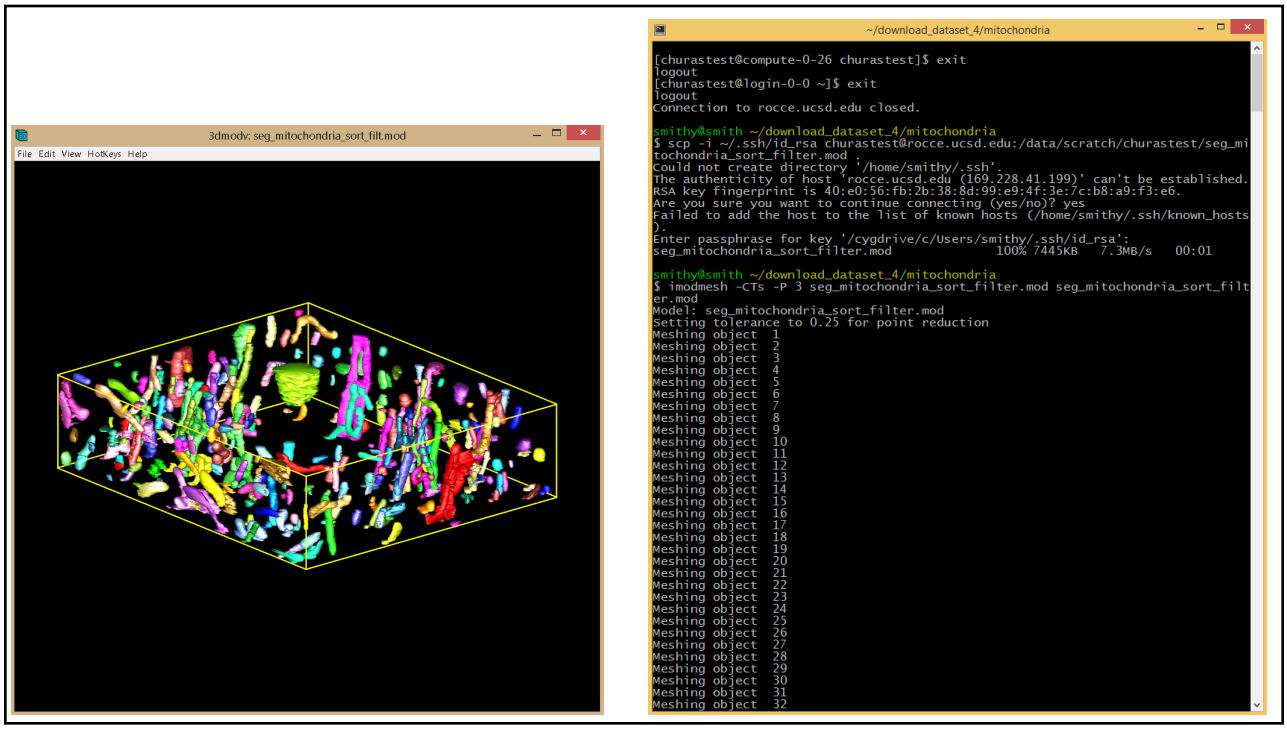
imodmesh –CTs –P 3 seg_mitochondria_sort_filter.mod seg_mitochondria_sort_filter.mod

G. Open the model file using the 3D model viewer. Note the differences between this model and the un-filtered version.

NOTE: If you have some extra time at the end of the demo, you can try experimenting with some of the different filters in PyIMOD. Instead of filtering by number of contours, you can, for example, filter by object volume using PyIMOD.filterByVolume.

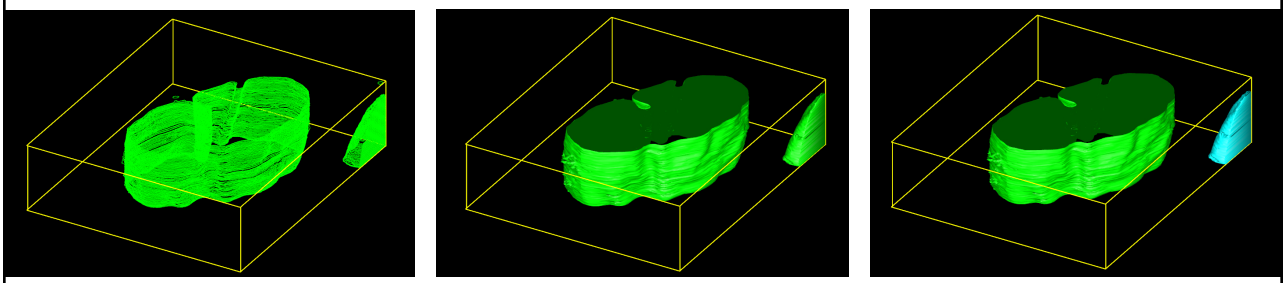
To remove all objects with volumes less than 0.01 um^3, for example, you would use:

PyIMOD.filterByVolume('>=', 0.01)



7. Repeat Steps 1-5 for the Nucleus Probability Maps

NOTE: Since nuclear segmentations are much more accurate than mitochondrial ones, there are very few false positives. Therefore, you can safely skip the filtering process of Step 6, and just manually delete false positives in 3dmod.

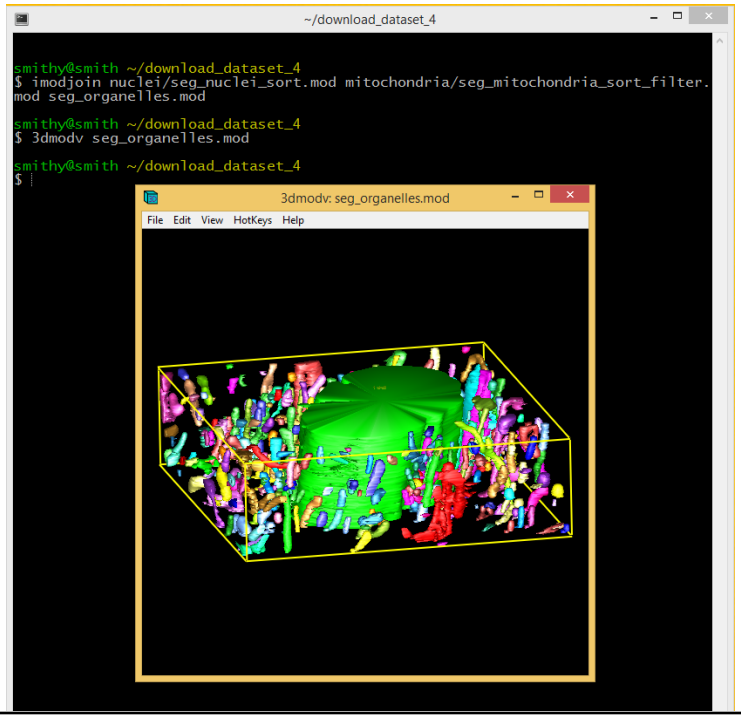


8. Join the Segmentations for Mitochondria and Nuclei

A. The program *imodjoin* will join model files together, as long as they have the same header information. To join the two models we've generated so far, you can use the following command from your unzipped directory (e.g. /home/aperez/download_dataset_4):

imodjoin nuclei/seg_nuclei_sort.mod mitochondria/seg_mitochondria_sort_filt.mod seg_organelles.mod

You can join together as many model files as you like using this command. The last entry in the space-separated list will be the name of the output model file.



9a. Quantifying the Results Using IMOD

A. Ensure that the model file has the proper header information (it usually won't, unless you enter it manually). With the model file open in 3dmod, go to Edit → Model → Header.

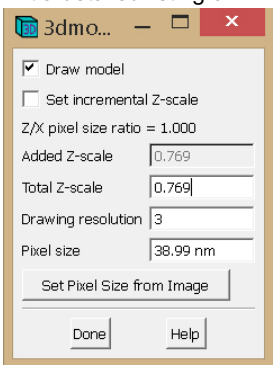
Under Pixel Size, enter the correct lateral pixel size, which is '38.99 nm' for this downsampled dataset.

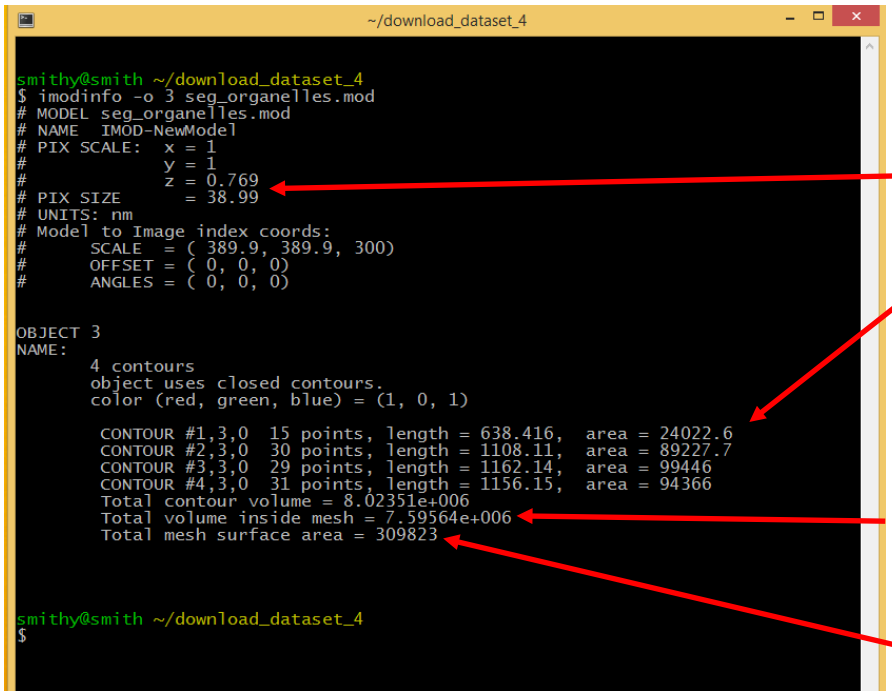
Under Total Z-Scale, enter the ratio of the axial pixel size to the lateral pixel size. For this dataset, this ratio is 30 / 38.99, or ~0.769.

B. Choose the object number corresponding to a mitochondrion (e.g., 3). We can print a detailed listing of metrics of this object using the program *imodinfo*. Run *imodinfo* with the –o flag specifying the appropriate object number, for example:

imodinfo –o 3 seg_organelles.mod

You should get an output to the terminal resembling the screen shot on the following slide.





Z scale and pixel size set

2D area of each contour, in nm^2

Volume of the mesh, in nm^3

Surface area of the mesh, in nm^2

9b. Quantifying the Results Using IMOD

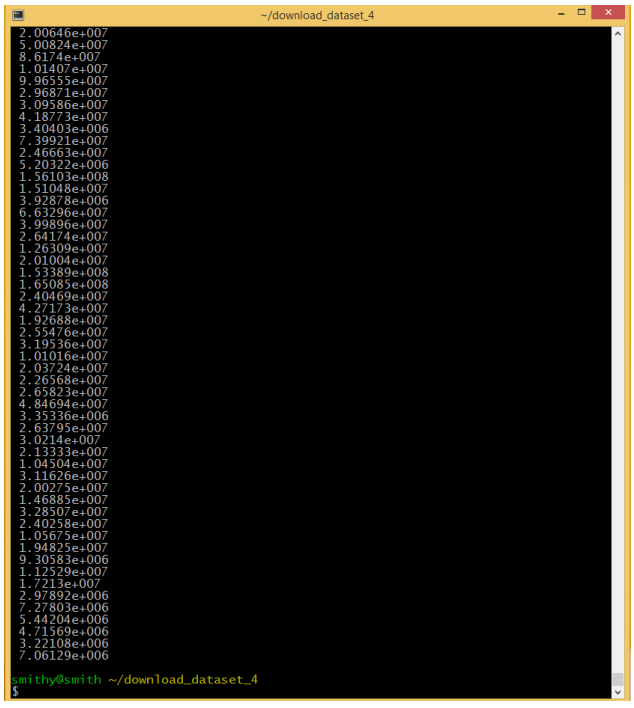
C. Of course, it would be nice to get the volume of say, every mitochondrion, printed out in a format that can be easily imported to a program like Excel. We can do this in a batch process using the following command, where objects 3-304 are the mitochondria:

imodinfo –o 3-304 seg_organelles.mod | grep 'volume inside mesh' | cut –d '=' –f2

For the surface area, the analogous command would be:

imodinfo –o 3-304 seg_organelles.mod | grep 'surface area' | cut –d '=' –f2

You should see an output to the terminal resembling the screenshot on the following slide, in which each line corresponds to the metric of each individual mitochondrion.



10. Generate Contours using thresholding

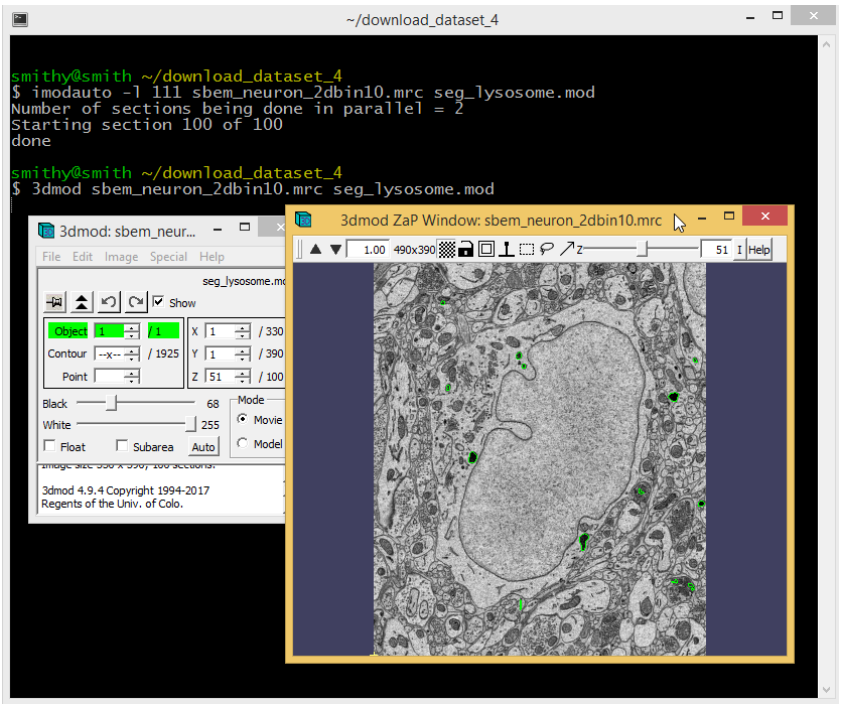
A. It is important to point out that one can use ***imodauto*** to attempt to segment features directly from the MRC file. Here we are using the *–l* flag (lower case L) to extract features darker then X pixel value:

-l Pixel intensity value for the low threshold. Contours will be drawn about all pixels below this value. Since the data are 8-bit, values ranging from 0 to 255 are valid.

For this dataset one can try to extract the lysosomes using a cutoff of 111 which was picked by moving the contrast sliders in IMOD graphical interface (Note: the *–u* is missing to disable raw thresholding):

imodauto –l 111 sbem_neuron_2dbin10.mrc seg_lysosome.mod

B. Once a model file has been created the same process applied earlier to the .mod files can be used to create 3d models. This method can work very well on data that stands out (very dark or very bright) obviating the need for more complex segmentation approaches.



11. Manual Proofreading/Correction of the Results

Since no automatic segmentation is perfect, the quantified results are, of course, not perfect.

The degree to which the automatic results need to be corrected depends upon the goals of your experiment and what you want to quantify.

For example, if you are only looking to quantify a metric like mitochondrial volume fraction (the sum volume of all mitochondria divided by the volume of the tissue), the purely automatic results may be accurate to within a few percent and may be sufficient.

If you are looking to quantify individual mitochondrial metrics, like the average mitochondrial volume, you may need to proofread the automatic output more carefully.

A. Let's spend the remaining time using the IMOD segmentation tools we learned about in the first session to correct our automatically generated model.

   We can also spend this time re-running some of the steps to check for their effects on the output.