Published in final edited form as:

Methods Mol Biol. 2016 ; 1427: 277–290. doi:10.1007/978-1-4939-3615-1_16.

Rapid and Semi-Automated Extraction of Neuronal Cell Bodies and Nuclei from Electron Microscopy Image Stacks

Paul S. Holcomb¹, Michael Morehead², Gianfranco Doretto², Peter Chen², Stuart Berg³, Stephen Plaza³, and George Spirou^{1,4,5}

¹ Center for Neuroscience, West Virginia University School of Medicine, Morgantown, West Virginia 26506-9303

² Lane Department of Computer Science and Electrical Engineering, Benjamin M. Statler College of Engineering and Mineral Resources, West Virginia University, Morgantown, West Virginia, 26506-6070

³ Howard Hughes Medical Institute, Janelia Research Campus, Ashburn, VA, 20147

⁴ Department of Physiology and Pharmacology, West Virginia University School of Medicine, Morgantown, West Virginia 26506-9303

⁵ Department of Otolaryngology, West Virginia University School of Medicine, Morgantown, West Virginia 26506-9303

Abstract

Connectomics—the study of how neurons wire together in the brain—is at the forefront of modern neuroscience research. However, many connectomics studies are limited by the time and precision needed to correctly segment large volumes of electron microscopy (EM) image data. We present here a semi-automated segmentation pipeline using freely available software that can significantly decrease segmentation time for extracting both nuclei and cell bodies from EM image volumes.

Keywords

Connectomics; segmentation; 3D reconstruction; electron microscopy; ilastik

1. Introduction

Connectomics is at the forefront of modern neuroscience research. With the advent of programs such as the BRAIN Initiative and the Human Brain Project, increasing numbers of research groups are attempting large-scale mapping of the mammalian nervous system. However, these research endeavors present significant challenges in data acquisition, storage, segmentation and interpretation (1). In particular, the segmentation of electron microscopy (EM) images for accurate nanoscale reconstruction of neurons and related structures requires significant time and effort on the part of researchers. Automated software

for segmentation is being developed (2–5), but still requires training data in the form of precise—and usually manual—segmentation of subsets of the data to be analyzed, known as ground truth. There exists a host of software tools for performing semi-automated segmentation, each with a unique set of tools, which can make difficult the selection of software best suited for solving a particular segmentation problem. Indeed, it is quite common that several different software solutions must be applied in order to converge on a solution. Additionally, most semi-automated software techniques perform segmentation on an entire image volume, whereas the specific research questions being asked may only require local segmentation near features of interest. Here, we detail one method for rapid and semi-automated segmentation of nuclei and cell bodies consisting of four steps: image conversion (if necessary), pixel classification, cell identification, and cell segmentation. Using this segmentation pipeline, a cell body and nucleus segmentation that manually may take up to 30 hours can be accomplished in as little as 1 hour.

2. Materials

2.1. Hardware

Hardware requirements will vary widely depending upon the size and resolution of the data set. Overall, a large amount of fast access memory (32GB+, DDR3/4) and solid state disk (SSD) storage for fast read/write times will significantly decrease the major bottlenecks of dealing with large data volumes. Processor speed, number of cores, and multithreading capability are also important, but secondary to RAM and SSD in boosting performance. This section will provide details on the baseline system that our lab uses, which can be tailored to suit experimental requirements.

1. SuperMicro ATX Server Motherboard Socket G34 AMD (Model: MBD-H8SGL-O).
2. AMD Opteron 6320 Abu Dhabi 2.8GHz 8 core Server Processor (Model: OS6320WKT8GHKWOF).
3. 128GB Kingston 240-pin DDR3 SDRAM (8×16 GB, Model: KVR16R11D4K4/64).
4. XFX HD 5000 Radeon HD 5450 1GB Video Card (Model: HD-545X-ZQH2).
5. hec 585W ATX12V Power Supply (Model: HP585DB).
6. Corsair Force LS 480GB SATA Solid State Drive (Model: CSSD-F480GBLSB).
7. Seagate Barracuda 7200.14 3TB SATA Hard Drive (Model: ST3000DM001).

2.2. Software

This chapter assumes that you have the following software installed:

1. Python 2.7. Python is the second most commonly used programming language(4), and is the major backbone of all of the software used in this chapter.
2. ImageMagick (<http://www.imagemagick.org>). ImageMagick is a lightweight and versatile tool for converting images between different file formats.
3. ilastik 1.1.6 or later (<http://ilastik.org>). The ilastik software(5), developed by the Heidelberg Collaboratory for Image Processing at the University of Heidelberg, provides a graphical user interface (GUI)-driven method for segmentation of image data using multiple classification methods.
4. CellSeeker (<http://www.github.com/SpirouLab/CellSeeker>). CellSeeker, developed by the authors, utilizes EM images and nucleus segmentation from ilastik to provide rapid proofreading of nuclei and extraction of EM subvolumes for further processing in ilastik.

3. Methods

The method described below assumes a high-resolution (<10nm) fully registered EM image stack. Lower resolutions or misaligned images may significantly affect results.

3.1 Image Conversion, If Necessary

Convert EM image stack to tiff format.

1. ImageMagick should be installed and added to the system PATH (see installation instructions at <http://www.imagemagick.org/script/binary-releases.php>).
2. Open a console window.
 - For Windows, press **Windows Key + r**, then type “cmd” (no quotation marks).
 - For Linux, most distributions will use the **Ctrl + Alt + t** shortcut to open a terminal window.
 - For Mac, in OS X and later, navigate to Applications -> Utilities and double-click on Terminal.
3. Change the directory to where your EM images are stored.
4. Type “convert [input directory/file] [output file]” (no quotation marks) and press Enter. The input directory/file can be either a single file (in the case of a single non-tiff image stack) or a directory containing sequential image files (ie. image01.tiff, image02.tiff, etc.). If a directory is specified, an asterisk (*, denoting ‘any file’) and the file type of the images must also be supplied. For example, a set of TIFF files would be specified as “*directory/**.tiff.” The output should be a single multi-page tiff file specified with the TIFF file extension (ie, “*directory/output.tiff*”). EM images may also be

reduced in size and resolution by adding the “-scale #” command before the input file, where “#” is the scale factor desired (i.e., to scale images by a factor of 2, a scale factor of 0.5 would be used). This may be desirable to speed up subsequent processing (see **Note 1**).

3.2. Pixel Classification

1. Import EM images into ilastik's “Pixel Classification” workflow. Pixel classification utilizes pixel information gained by applying multiple image filters (Gaussian blur, gradient magnitude, etc.) to build a feature vector representing each pixel in an image. Based on training data provided by a user, pixel classification attempts to group pixels with similar feature vectors into subsets that, when taken together, completely segment the image.
2. Run ilastik and select the “Pixel Classification” workflow.
3. Provide a project name and location to save the project.
4. On the Input Data screen, under the Raw Data tab, click “Add New” and select the “Add Separate Images” option.
5. Navigate to the location of the EM image data and select the file (or folder, for a directory of sequential images), then click Open.
6. In the “Raw Data” tab, double click on the axis labels under the “Axes” column header.
7. In the axes field, define axes as either “zyxc” for isotropic data or “tyxc” for anisotropic data (see **Note 2**) and click OK.

3.2.1. Training ilastik to Recognize Cell Nuclei in EM Images (Fig. 1)

1. After the data have been successfully imported, click the “Feature Selection” tab in the menu options on the left side of the screen.
2. Under “Feature Selection”, click “Select Features...”.
3. In the “Features” dialog box, select the filters to use for training the classifier (see **Note 3**). A good rule of thumb is to start with as many

¹As nuclei are large compared to other organelles, have a clear border (the nuclear envelope) and have a generally uniform internal texture in images, a high-resolution image is not needed to identify them. In order to increase the speed of cell identification, we recommend binning EM images to a low resolution before beginning this process. For example, our original data set for testing was a 250GB image stack with dimensions of 11000 × 8000 × 1420 at an {x, y} pixel dimension of 11.4 nm and a slice thickness of 60 nm. Before training in ilastik, we binned our images (using either the “-scale” command in ImageMagick or the “newstack -bin” command in the IMOD software (University of Colorado, Boulder; <http://bio3d.colorado.edu/imod/>). When binned by a factor of 9, the resulting image stack was a more manageable 3.1GB in size and sufficient for identifying nuclei in ilastik. If higher resolution subvolumes are desired, use the “Use different EM volume for subvolumes” option during the subvolume extraction step in CellSeeker.

²For anisotropic data, you may want to force ilastik to see your data as individual 2D images rather than a 3D image stack to get better segmentation during nucleus training. You can do this during the data import step (see above, “Import EM images into ilastik's ‘Pixel Classification’ workflow”). In the “Raw Images” tab, next to the data set you will be working with, double-click in the “Axes” column. This will open a “Dataset Properties” window, with one of the options being “Axes”. Change the axes from “zyx” to “tyx” and click OK. If successful, you will see a “Time” indicator at the bottom-right corner of the ilastik window, with a maximum equal to the number of images in your dataset. This tricks ilastik into treating your data as a set of 2D images taken at different time points, rather than a 3D volume. All other steps in the process are completed as written.

classifiers as your data set will allow (a dialog box will provide a warning if a classifier is out of range) and reduce the features if necessary as training progresses to get a better segmentation. In our hands, the number of classifiers used is usually 18-22. Click OK to compute the selected features (*see Note 4*).

4. Click the “Training” tab in the menu options on the left side of the screen. Click “Add Label” twice to create a training label for nuclei and background segmentation. The color of the label is customizable by clicking the colored square next to each label. Change label names, if necessary, by double-clicking the label name.
5. Using Shift + left-click on the mouse to pan, Shift + the mouse wheel to scroll through images, and the “+” and “-” keys to zoom in and out respectively, locate a nucleus within the EM image.
6. Left-click on the label for the nucleus, and select the Paintbrush button (underneath the “Add Label” button on the left side of the screen). Adjust your brush size using the “Size” drop-down menu to a size that will allow for quick painting of the interior of the nucleus without going outside of the nuclear envelope.
7. Left-click and hold inside the nucleus and paint the interior (**Fig. 1**).
8. Left-click on the label for the background, and repeat the two previous steps for the background outside of the nucleus.
9. Zoom in to a small section of the EM image using the “+” key. Click “Live update” to see the training results. ilastik computes the training results only for the area of the image visible on the screen, so to reduce the time it takes to see a result, keep the visible area relatively small (single nuclei, for example). Ideally, if only a small area is visible, this process should take one minute or less (*see Note 5*). Visibility and opacity of the results can be controlled under the “Group Visibility” section in the lower left of the screen.
10. Modify the nucleus and background labels by painting in areas trained incorrectly (nucleus in background area, and vice versa, white and black arrowheads, respectively, in **Figure 1**) until obtaining an acceptable segmentation of the nucleus.

³The feature sets provided by ilastik for training the pixel classifier are “Color/Intensity”, “Edge”, and “Texture”. For nuclei in electron microscopy images, usually all three of these categories apply; nuclei in neurons tend to be lighter in color, have clearly defined edges, and a unique speckled texture. Each feature also has an associated range of pixel sizes to choose from, the choice of which will vary depending on the resolution of the EM used. When in doubt, select as many combinations of features and pixel sizes as ilastik will allow, and refine these settings after some pixel classifier training has been completed.

⁴Feature selection during the ilastik pixel classification step has a significant impact on the resulting nucleus segmentation. Typically, an iterative approach works best: select features, add training labels, check the segmentation, change the feature selection and see if the segmentation improves. More details on feature selection can be found in the ilastik documentation (<http://ilastik.org/documentation/pixelclassification/pixelclassification.html#selecting-good-features>).

⁵As often as possible, stay zoomed in to a small area in ilastik when “Live Update” is on. To save computational time, ilastik only computes the segmentation for the area currently viewed, so the larger the area, the longer the computation time. When verifying that all nuclei are being segmented appropriately, it may take several minutes for the segmentation to appear in ilastik.

11. Zoom out to view an entire image slice (use the upward-pointing triangle key in the upper-right corner of the image).
12. Ensure that other nuclei in the image are also being accurately segmented. If not, zoom in (“+” key) to trouble areas and paint using the nucleus and background labels until an acceptable segmentation is obtained.
13. Switch to an image in another section of the dataset (use Shift + mouse wheel or manually input the slice number in the Z location box in the top left of the image screen) to ensure that training is uniform throughout the volume.

3.2.2. Exporting Ilastik Nucleus Segmentation

1. Under “Group Visibility” in the lower left area of the screen, locate the nucleus segmentation entry (e.g., Segmentation (Nucleus)).
2. Right-click on the nucleus segmentation entry, and select “Export...”.
3. Under “Output File Info”, click the “Select...” button next to the “File:” entry.
4. In the navigation window, specify a folder and filename for the nucleus segmentation. Click OK.
5. Click OK in the “Image Export Options” window. ilastik should now automatically export the segmentation, which may take several minutes for larger datasets (*see Note 6*).

3.3. Cell Identification

Process nucleus segmentation using CellSeeker to identify cells and create EM subvolumes.

1. Download CellSeeker (<http://www.github.com/SpirouLab/CellSeeker>).
2. Open a terminal window (see above) and navigate to the CellSeeker folder.
3. Run CellSeeker by typing “python cellseeker.py” and press Enter.
4. Within the “CellSeeker:Initialization” window that appears, provide the following:
 - The location of the nucleus segmentation, either by typing the path in the text box beside “Nucleus Segmentation:” or clicking the “Search...” button to the right of the text box, navigating to and selecting the file and clicking “OK”.

⁶Exporting the nucleus segmentation from ilastik may take approximately 1 hour per gigabyte of original image data. This will produce a nucleus segmentation file in Hierarchical Data Format (HDF5), which will be used in subsequent steps in the segmentation pipeline. This format, designed by the National Center for Supercomputing Applications, is designed for efficient storage and access of large three-dimensional datasets. The HDF5 file format has both benefits and drawbacks. For saving masked or binary information, the amount of compression obtained by using this format is significant (> 30x). For more complex images, such as EM images, the compression is more modest (~2x). In our hands, the HDF5 format is also slower to load into a Numpy array in Python, increasing the wait time in CellSeeker. A mixture of HDF5 and TIFF formats is allowed in CellSeeker, so it is possible to load EM images from a TIFF stack and nucleus segmentation from a HDF5 file.

- The location of the EM images in the “EM Images:” field by following the instructions detailed for the nucleus segmentation.
 - The linear pixel dimension in nanometers (nm) for the EM images (i.e., for a $5\text{ nm} \times 5\text{ nm}$ pixel size, this value would be 5) in the text box next to “Pixel dimension (nm):”. CellSeeker assumes isotropic pixel dimensions in x and y.
 - The distance between images in z in nm in the text box next to “Z-slice thickness (nm)”.
 - The estimated cell diameter in micrometers (μm) of cells being extracted from the data volume. This value will be used to calculate the extents in x, y, and z for cropping the EM volume and for filtering (*see Note 7*).
5. Click “OK”. CellSeeker will perform a size filtering in 2D and 3D based on the largest object found in each image, followed by a connected component analysis to enumerate the presumptive nuclei.
 6. When the filtering step completes, CellSeeker will present an image of a possible extracted nucleus in 2D in all three planes ($\{X, Y\}$, $\{X, Z\}$, $\{Y, Z\}$) overlaid on the associated EM image.
 - If the segmentation has correctly identified the nucleus, click the “Approve” button or press the up arrow key (**Fig. 2A**).
 - If the segmentation has incorrectly identified a structure as a nucleus, click the “Reject” button or press the down arrow key (**Figs. 2B, C**).
 - If a classification was incorrectly performed, click the “Back” button or press the left arrow key to return to the previous set of images and reclassify it.
 7. Repeat the verification process for each possible nucleus identified.
 8. After proofreading is complete, click the “Extract Cell Volumes” button.
 9. Specify a location where EM subvolumes representing individual cells should be extracted. If desired, a higher resolution version of the EM volume can be substituted at this point by checking the “Use different EM volume for subvolumes” box and supplying the path to the new EM volume. The new volume *must* be a multiple of the original volume in the $\{x, y\}$ axis (i.e., a $400 \times 400 \times 200$ volume could be substituted for a $100 \times 100 \times 200$ volume), or an error will occur.

⁷If the size of the cell or cells is unknown, we recommend performing a rough measurement in a program such as ImageJ to get an estimate of the cell size. In this case, it is best to err the cell diameter on the smaller side; although this may produce more false positives, it is also more likely to result in complete identification of all cells in the volume.

10. Click the OK button. CellSeeker will automatically create subfolders within the directory for each identified cell (“Cell_1”, “Cell_2”, etc.), as well as folders within each cell folder to hold later segmentation (“Nucleus”, “CellBody”, “Axon”, “Dendrites”, “Inputs”, and “AdditionalStructures”). Within the “CellBody” folder for each cell, CellSeeker produces a multipage TIFF file, containing the EM images for that cell, a text file (“coordinates.txt”), defining where the subvolume was extracted from in the context of the full EM volume, and an ilastik project file, containing the preprocessed images for use in the ilastik “Carving” workflow. This may take some time depending on the data volume size and relative size of each cell.
11. At this point, for each verified cell, a subvolume of EM data and an associated ilastik project file for carving has been produced.

3.4. Cell Segmentation

Carve nucleus and cell body for each cell in ilastik. The carving workflow in ilastik provides a fast and straightforward method for segmentation of single objects such as cell bodies (8). Using an oversegmentation of the data into supervoxels (automated groupings of pixels based on intensity and intensity gradient characteristics) and training data supplied by the user, ilastik combines the supervoxels into “object” and “background” segmentations.

1. Run ilastik.
2. Click the “Project” menu option at the top of the screen and select “Open Project...” or use the Ctrl + o (Command + o for Mac) shortcut key.
3. In the “Open ilastik Project” window, navigate to the folders created by CellSeeker containing the preprocessed cells.
4. Select the project file for the cell to be segmented and click “Open”.
5. When the project has finished loading, click the “Labeling” tab in the menu options on the left side of the screen.

3.4.1. Segmenting the Nucleus and Cell Body (Fig. 3)

1. Left click on the “Object” label under the “Problem specific labels” heading in the “Labeling” tab.
2. Left click and hold to paint the interior of the cell nucleus in the X, Y, and Z slices displayed in the image windows. Cover the interior of the nucleus as thoroughly as possible while staying within the nuclear envelope.
3. Left click on the “Background” label under the “Problem specific labels” heading in the “Labeling” tab.
4. Left click and hold in the EM image slices displayed in the image windows to paint the area surrounding the nucleus as background in the X, Y, and Z. Make sure the background label covers a subset of pixels that

- represent the majority of the background (i.e., non-nuclear) image (*see Note 8*).
5. Left click on the “Segment” button under the “Current Object” heading in the “Labeling” tab. ilastik will produce an initial segmentation of the nucleus based on the training data provided, as well as a three-dimensional representation of the object.
 6. Using the slice number controls in the top-left of each image window, navigate to trouble spots in the segmentation (nucleus incorrectly labeled as background and vice-versa) and add additional labels to refine the segmentation. Once changes are made, the “Segment” button must be clicked again to update the segmentation.
 7. Once an acceptable nucleus segmentation is obtained (*see Note 9*), click the “Save current object” button under the “Current Object” heading in the “Labeling” tab. Name the object “nucleus” and click the “OK” button.
 8. In the list of layers at the bottom-left corner of the window, right-click on the “Segmentation” layer and click “Export...”.
 9. In the “Image Export Options” window, under the “Output File Info” header, click the “Select...” button next to the “File” option. Select a directory and filename for the nucleus segmentation and click “Save”. Click “OK” on the “Image Export Options” window to output the nucleus segmentation to an HDF5 file (*see Note 10*). Click “Finished!” when the export is complete.
 10. Click the “Clear” button under the “Current Object” heading in the “Labeling” tab, then click “Clear” to clear the nucleus segmentation.
 11. Repeat **steps 1–10, Section 3.4.1** to segment and save the cell body.
 12. To create three-dimensional mesh objects for the nucleus and cell body (for use in programs such as MeshLab, Blender, or any software that supports the OBJ file format) click the “Export All Meshes” button under the “All Objects:” header. Select a folder for output and click “OK” (*see Note 11*).

⁸For the initial training step, when carving a cell in ilastik, we have found that training on 3 images—one containing a cell body near the “top” of the image stack, one through the center of the cell, and one near the “bottom” of the image stack—in each image plane ($\{x, y\}, \{x, z\}, \{y, z\}$) is sufficient to obtain a reasonable segmentation quickly (usually < 5 minutes). The amount of time for refinement of the segmentation will depend on the image resolution and level of required accuracy.

⁹The number of iterations of training, proofing, and retraining will depend on the resolution of the image data and accuracy of training. For the nucleus and cell depicted in the lower panel of Figure 3, three and five retraining iterations were necessary, respectively.

¹⁰Although HDF5 is our recommended format for export of masks, it is advisable to make sure that the software you plan to use for post-processing can read HDF5 format. If it cannot, make sure to change the export setting in ilastik from HDF5 to your preferred format (ie. multipage TIFF, JPEG, etc.).

¹¹Although 3D models can be generated directly from ilastik, we recommend importing the nucleus and cell body masks along with the corresponding EM subvolume into a proofreading tool such as Seg3D (Scientific Computing and Imaging Institute, University of Utah, <http://www.sci.utah.edu/cibc-software/seg3d.html>). This software provides powerful tools for editing the masks created by ilastik to ensure the accuracy of the segmentation, as well as allowing for 3D mesh export.

13. When finished, click “Project” in the menu options at the top of the window, and select “Save Project”.
14. Repeat the carving process for each verified cell.
15. At this point, two additional HDF5 files (or other format specified by the user) containing either nucleus or cell body segmentation masks for each cell have been produced. Additionally, if “Export All Meshes” was used, two OBJ files containing mesh representations of the nucleus and cell body have been produced.

Acknowledgements

This work was supported by grants NIH F31 DC014393 to PSH, NIH R01 DC007695 to GAS, and NIH/NIGMS CoBRE P20 GM103503 to the WVU Center for Neuroscience.

References

1. Lichtman JW, Pfister H, Shavit N. The big data challenges of connectomics. *Nat Neurosci.* 2014; 17:1448–1454. [PubMed: 25349911]
2. Kaynig V, Vazquez-Reina a. Large-scale automatic reconstruction of neuronal processes from electron microscopy images. *IEEE Trans Med Imaging.* 2013; 1:1–14.
3. Seyedhosseini M, Sajjadi M, Tasdizen T. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. *2013 IEEE Int Conf Comput Vis.* 2013:2168–2175.
4. Jain V, Turaga SC, Briggman KL, et al. Learning to agglomerate superpixel hierarchies. *Advances in Neural Information Processing Systems.* 2011; 24:648–656.
5. Funke J, Andres B, Hamprecht FA, et al. Efficient automatic 3D-reconstruction of branching neurons from em data. *Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit.* 2012:1004–1011.
6. Kim BYL. 10 Most Popular Programming Languages Today. Inc. Mansueto Ventures. 2015
7. Sommer C, Straehle C, Ullrich K, Hamprecht FA. Interactive learning and segmentation toolkit. *Int Symp Biomed Imaging Proc.* 2011:230–233.
8. Straehle C, Köthe U, Knott G, Hamprecht FA. Carving: Scalable interactive segmentation of neural volume electron microscopy images. *Med Image Comput Comput Assist Interv.* 2011; 14:653–660. [PubMed: 22003674]

Fig. 1. ilastik Pixel Classification

Classification for nucleus segmentation performed on serial block-face scanning EM image stack. Nucleus (*solid red*) and background (*solid green*) training data are used to define pixel properties of each region. Pixel classification then sorts each pixel into either nucleus (*translucent red*) or background (*translucent green*) for segmentation. Additional training data can be added to misclassified pixels (i.e., nucleus as background (*black arrowheads*) or background as nucleus (*white arrowheads*)) to refine the segmentation. The output of this step is used as the input to CellSeeker for cell identification.

Fig. 2. Cell Verification in CellSeeker

EM images are presented in all three image planes with associated nuclear segmentations (*red*). The cell being identified should have its nucleus centered in the viewing windows (*centers marked with green circles*). **(A)** This panel demonstrates a successfully identified cell, which would be kept using the “Approve” button. False positives usually appear as either **(B)** non-nuclei incorrectly identified as nuclei or **(C)** two nuclei incorrectly fused together. In either case, the “cell” would be rejected using the “Reject” button. Users can navigate backwards to change decisions using the “Back” button. EM of cells verified in this manner will be processed and output for carving in ilastik.

Fig. 3. Carving Cells in ilastik (top)

Training data for cell interior (*solid green*) and background (*solid red*) in all three dimensions is used by ilastik to define the cell segmentation. The result of the segmentation is overlaid on the EM images (*translucent green*). Additional training data can be added to missed pieces of the cell (*white arrowheads*) and background that was incorrectly identified as cell (*black arrowheads*) in an iterative fashion to refine the segmentation. A 3D representation of the current segmentation is displayed in the lower right panel to assist in refinement. This process can produce both a mask of the cell segmentation for further refinement, or a 3D mesh in OBJ format (*bottom, shown with nucleus (red), cell body (tan)* and EM in the Seg3D software (see **Note 11**) for measurement and display.