frontiers in Synaptic Neuroscience

METHODS
published: 18 December 2020
doi: 10.3389/fnsyn.2020.584549

SynapsEM: Computer-Assisted Synapse Morphometry

Shigeki Watanabe[1,2], M. Wayne Davis[3], Grant F. Kusick[1], Janet Iwasa[4] and Erik M. Jorgensen[3]**

[1] Department of Cell Biology, Johns Hopkins University School of Medicine, Baltimore, MD, United States, [2] Solomon H. Snyder Department of Neuroscience, Johns Hopkins University School of Medicine, Baltimore, MD, United States, [3] Department of Biology and Howard Hughes Medical Institute, University of Utah, Salt Lake City, UT, United States, [4] Department of Biochemistry, University of Utah, Salt Lake City, UT, United States

The structural features of a synapse help determine its function. Synapses are extremely small and tightly packed with vesicles and other organelles. Visualizing synaptic structure requires imaging by electron microscopy, and the features in micrographs must be quantified, a process called morphometry. Three parameters are typically assessed from each specimen: (1) the sizes of individual vesicles and organelles; (2) the absolute number and densities of organelles; and (3) distances between organelles and key features at synapses, such as active zone membranes and dense projections. For data to be meaningful, the analysis must be repeated from hundreds to thousands of images from several biological replicates, a daunting task. Here we report a custom computer program to analyze key structural features of synapses: SynapsEM. In short, we developed ImageJ/Fiji macros to record x,y-coordinates of segmented structures. The coordinates are then exported as text files. Independent investigators can reload the images and text files to reexamine the segmentation using ImageJ. The Matlab program then calculates and reports key synaptic parameters from the coordinates. Since the values are calculated from coordinates, rather than measured from each micrograph, other parameters such as locations of docked vesicles relative to the center of an active zone can be extracted in Matlab by additional scripting. Thus, this program can accelerate the morphometry of synapses and promote a more comprehensive analysis of synaptic ultrastructure.

Keywords: electron microscopy, synapse, morphometry, ultrastructural analysis, SynapsEM

OPEN ACCESS

Edited by:
Xiaobing Chen,
National Institutes of Health (NIH),
United States

Reviewed by:
Thomas Uram,
Computing, Environment and Life Sciences Directorate, Argonne National Laboratory, United States
Jae Hoon Jung,
National Institutes of Health (NIH),
United States

****Correspondence:***
Shigeki Watanabe
shigeki.watanabe@jhmi.edu
Erik M. Jorgensen
jorgensen@biology.utah.edu

Received: *17 June 2020*
Accepted: *30 November 2020*
Published: *18 December 2020*

Citation:
Watanabe S, Davis MW, Kusick GF, Iwasa J and Jorgensen EM (2020) SynapsEM: Computer-Assisted Synapse Morphometry. Front. Synaptic Neurosci. 12:584549. doi: 10.3389/fnsyn.2020.584549

INTRODUCTION

Understanding the mechanisms of synaptic transmission requires detailed characterizations of synapses at the ultrastructural level. To release neurotransmitters, synaptic vesicles fuse at a specialized membrane domain of the presynaptic terminal called the active zone (Couteaux and Pécot-Dechavassine, 1970; Heuser et al., 1979). A subset of vesicles are docked, that is, in contact with the active zone membrane by morphology (Schikorski and Stevens, 1997; Hammarlund et al., 2007; Imig et al., 2014), and fuse in response to calcium influx (Heuser et al., 1979; Heuser and Reese, 1981). Following fusion, these vesicles are recycled locally *via* endocytosis and components sorted in an endosome to sustain synaptic transmission (Ceccarelli et al., 1972; Heuser and Reese, 1973; Dittman and Ryan, 2009; Saheki and De Camilli, 2012; Watanabe et al., 2013a,b, 2014; Kononenko and Haucke, 2015). However, the structures involved in synaptic membrane trafficking are extremely small. For example, synaptic vesicles are 30–50 nm in diameter (Zhang et al., 1998), and a few hundred vesicles are clustered within a synaptic bouton (Schikorski and Stevens, 1997; Shepherd and Harris, 1998), which is only ∼0.5–1 μm in diameter. Moreover, a vesicle may move only a few nanometers to fully engage the active zone membrane during docking (Hammarlund et al., 2007; Imig et al., 2014), and this state is quite dynamic (Chang et al., 2018; Kusick et al., 2020). Given these dimensions, synaptic morphometry requires the resolution of electron microscopy.

Morphometry is typically performed on an electron micrograph of a single synaptic profile from a 30–70 nm-thick section. Analyses from ∼200 synaptic profiles are then summed, and results are compared between controls and experimental samples, such as mutants or drug-treated neurons. In each image, the following features are analyzed: the size of membrane-bound structures such as vesicles and other organelles, the number of these structures, and the distance of these structures from the active zone membrane, the plasma membrane, and if apparent, electron-dense cytomatrix (dense projection) that presumably harbors calcium channels. The organelles identified in serial electron micrographs can be "segmented" by tracing and characterized manually by measuring the sizes of the organelles and distances of these structures to the relevant membranes. Thus, manually measuring features in electron micrographs is labor-intensive, requiring extra effort to record annotated features in such a way that they can be easily reexamined.

To overcome these issues, we developed an analysis workflow, SynapsEM, that integrates ImageJ macros and Matlab scripts for the morphometry of synapses from electron micrographs (**Figure 1**). Specifically, the Matlab scripts first shuffle images from different conditions, which are pooled into a single folder (**Figure 1A**). This shuffling procedure reduces potential bias during annotation. These images are imported into ImageJ as a sequence (**Figure 1A**). With the freehand line tool, the contours of the plasma membrane and the active zone membrane are traced, and their x,y-coordinates are recorded in the ROI manager (**Figure 1B**). Then, the diameters of vesicles are traced using a straight line tool (**Figure 1B** inset), and the x,y-coordinates at the centroid of vesicles are recorded. Additionally, the membrane of endosomes can be traced with a freehand selection tool, and x,y coordinates of the contour line recorded. Once all structures are annotated from an image, the values in the ROI are exported as a text file (**Figures 1B,C**). The text file can be imported back to ImageJ for re-evaluation by independent researchers. This re-evaluation step is not required but independent confirmation of segmentation calls makes the annotation more accurate. When all images are analyzed, the resulting text files are unblinded and imported into Matlab (**Figure 1C**). The custom scripts then calculate the distances of membrane-bound structures to the active zone membrane as well as the plasma membrane. The numbers and diameters of these structures are also determined. These data are then compared between different conditions computationally; the researcher remains blinded to individual images. The outputs of the scripts can be saved or exported to other programs for statistical analysis. SynapsEM can be applied to serial-section data, but one must be cautious about overcounting structures that span in multiple sections. Overall, this workflow expedites the analysis of synaptic ultrastructure, reduces the experimental biases associated with manual image annotation, and unifies the method of analysis across many labs.

MATERIALS AND EQUIPMENTS

The following materials are required for the procedures described in this manuscript.

- Electron micrographs, preferably acquired with the same camera setting on the same microscope for a set of experiments.
- A computer (no specific requirements as long as it meets the system requirement for the Matlab program).
- A computer keyboard with a numeric keypad.
- Matlab, MathWorks (The scripts were originally written in Matlab 2008, and have been added to in versions up to Matlab 2020).
- Matlab custom scripts (available from https://github.com/shigekiwatanabe/SynapsEM).
- Fiji (or ImageJ).
- Macros (available from https://github.com/shigekiwatanabe/SynapsEM).
- Maya for 3-D rendering (optional).
- Wacom tablet (Cintiq 22HD), or another pen tablet (optional, but makes annotating many images easier).

METHODS

Randomizing Images

Images from multiple samples should be analyzed in a batch to minimize the potential bias during the analysis. Shuffling removes any possibility of conscious or unconscious bias between samples, eliminating any variables in analysis apart from different samples themselves. Even with blinded, but not scrambled images, the analysis can be skewed when they are analyzed in different batches: this can range from an obvious phenotype making it clear which condition a sample is, to simple day-to-day differences in the segmentation, to a novice improving the accuracy of their segmentation as they work through more images in an experiment. For this purpose, all images for single experiments are pooled into one folder (**Figure 1A**). These images should be duplicates of the original images to keep the original data intact. To ensure the original images are safe, the program asks whether the images should be duplicated when the "randomize" code is run in Matlab. Answering "yes" to this question will make copies of the images and randomize the duplicated images, leaving the original data untouched. In the popup window, select the directory (folder) that contains all images for an experiment. After the selection, another window pops up, prompting the user to select all images. Select all images to be randomized. At the end of the program, randomized images are found in the "randomized" folder, nested in the directory where the original images are (**Figure 1A**). The key is named "key.mat" and is also saved in the "randomized"

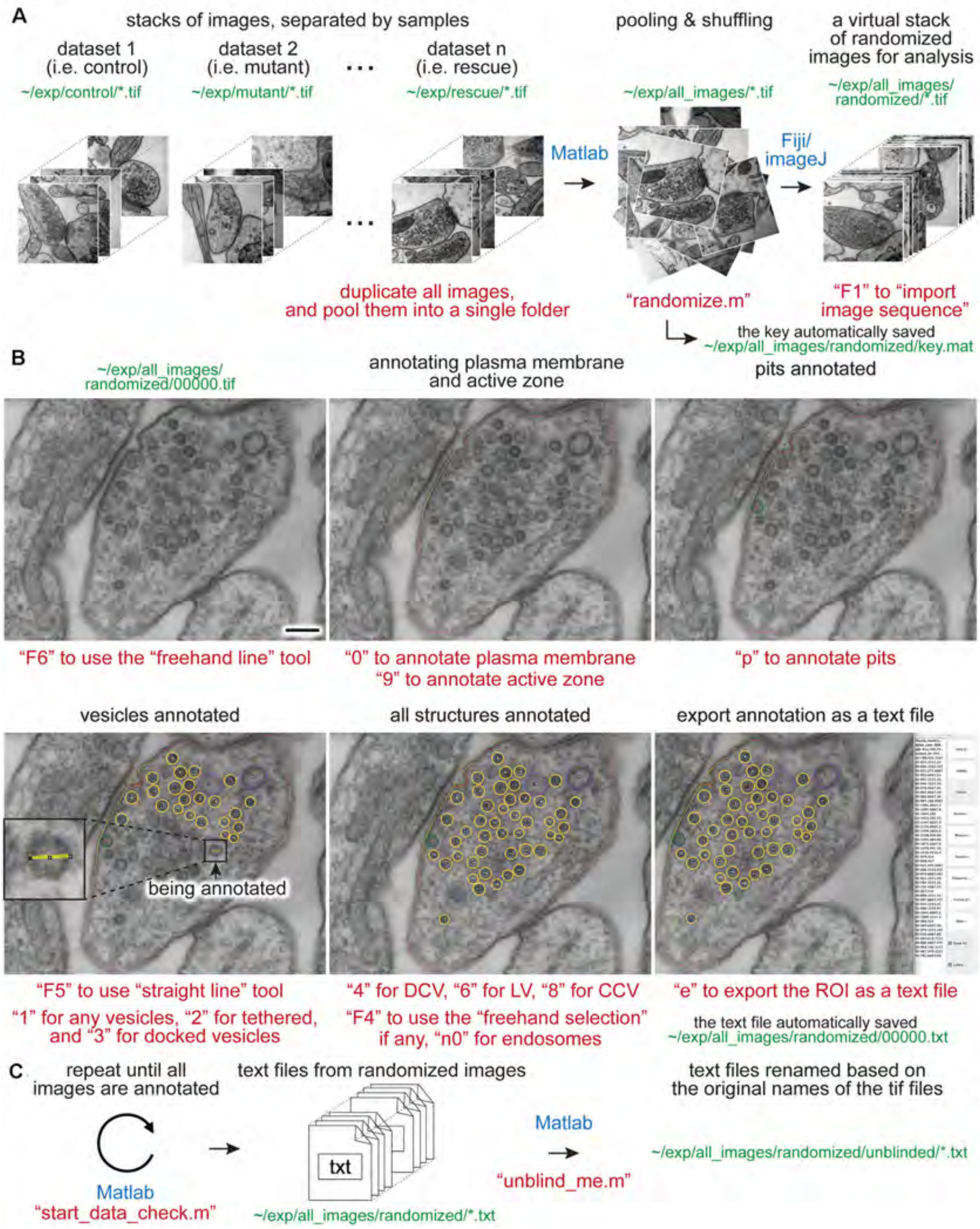

A
stacks of images, separated by samples
dataset 1
(i.e. control)
dataset 2
(i.e. mutant)
dataset n
(i.e. rescue)
~/exp/control/*.tif
~/exp/mutant/*.tif
~/exp/rescue/*.tif
pooling & shuffling
~/exp/all_images/*.tif
a virtual stack of randomized images for analysis
~/exp/all_images/randomized/*.tif
Matlab
Fiji/imageJ
duplicate all images, and pool them into a single folder
"randomize.m"
"F1" to "import image sequence"
the key automatically saved
~/exp/all_images/randomized/key.mat
B
~/exp/all_images/randomized/00000.tif
annotating plasma membrane and active zone
pits annotated
"F6" to use the "freehand line" tool
"0" to annotate plasma membrane
"9" to annotate active zone
"p" to annotate pits
vesicles annotated
all structures annotated
export annotation as a text file
being annotated
"F5" to use "straight line" tool
"1" for any vesicles, "2" for tethered, and "3" for docked vesicles
"4" for DCV, "6" for LV, "8" for CCV
"F4" to use the "freehand selection" if any, "n0" for endosomes
"e" to export the ROI as a text file
the text file automatically saved
~/exp/all_images/randomized/00000.txt
C
repeat until all images are annotated
Matlab
"start_data_check.m"
text files from randomized images
txt
~/exp/all_images/randomized/*.txt
Matlab
"unblind_me.m"
text files renamed based on the original names of the tif files
~/exp/all_images/randomized/unblinded/*.txt

FIGURE 1 | Synapse morphometry using synapsEM. **(A)** A schematic showing the image randomization procedure. In each experiment, images are collected from multiple samples (i.e., control, mutants, drug-treated). Ideally, image acquisition should also be done blinded. These images are pooled into the same folder and then randomized using the "randomize.m" Matlab script. Running this program creates a new folder, named "randomized," and transfers the images into this folder with the randomized number assigned to each image. These images should be opened in ImageJ or Fiji as a virtual stack using the F1 key after installing the "synapsEM_analysis_macro.txt" macro. **(B)** A schematic showing the procedure for image annotation. After opening the images in ImageJ/Fiji, membranes can be traced using specific tools and hot keys (as denoted in red). Note that all structures annotated are recorded into the ROI manager. For vesicles, draw a line across the vesicle membrane as shown in the inset. After pressing a hot key corresponding to the structure (i.e., "1" for synaptic vesicle in this case), the macro draws a circle using the drawn line as a diameter for the vesicle. When the annotation is complete, the structures listed in the ROI manager can be exported by simply pressing "e" on the computer keyboard. **(C)** A schematic showing the final steps of the morphometry. After all images in the dataset are annotated, the resulting text files must be first checked for errors using the "start_data_check" script in Matlab. After corrections, the text files can be decoded based on the original names of the corresponding images using the "unblind_me" script. This script duplicates the text files, decode the copied files, and move them to the "unblinded" folder. The decoded text files can be further processed for analysis. The sentences in red, green, and blue indicate the commands, the directory of files, and the software used, respectively. Scale bars = 100 nm.

folder (**Figure 1A**). This file must be kept safe until the analysis is complete.

Opening Images

Morphometry is performed in Fiji. Start Fiji, and install the "synapsEM_analysis_macro.txt" by navigating to the plugin dropdown menu, clicking on "Macros" and then "Install," and selecting the "synapsEM_analysis_macro.txt" file. For easy access to the file, it is highly recommended to store this macro in the ∼/Fiji/macros directory so it is readily accessible. To skip this installation procedure, the contents of "synapsEM_analysis_macro.txt" can be copied into the StartupMacros.fiji.ijm, which is found in the ∼/Fiji/macros directory, since all the macros in this file are activated as Fiji starts up.

To start the analysis, the randomized images should be opened as a stack (**Figure 1A**). Press the "F1" key to import the image sequence as a virtual stack—the pixel size information on tif files will be converted such that each pixel is one arbitrary unit when the "F1" hotkey is used. If images are opened through "File, Import, Image sequence" or simply dragging a folder to Fiji, it is important to "set scale" through the "Analyze" drop-down menu and type in 1 for the "distance in pixels" and 1 for the "known distance." This conversion of the pixel size can also be performed by pressing "F2." When working with a large dataset, it is highly recommended that images are opened as a virtual stack.

Segmenting Images

The eventual goal of segmentation is to determine the size of vesicles and other membrane-bound organelles, the numbers or density of these structures, and the distribution of these structures relative to the plasma membrane or active zone membrane. The macros are set up to record x,y-coordinates of the structures of interest and their size information in the ROI manager window. These macros are accessed through hot keys, as listed in **Tables 1**, **2**. Three sets of line tools are used to trace different features at synapses. A straight line tool (the Fiji tool #1) is accessed with "F5" and used to annotate closed and near-uniformly circular structures like synaptic vesicles, large vesicles, and dense-core vesicles. A freehand selection tool (the Fiji tool #3, "F4") is used to segment closed and irregularly-shaped structures such as endosomes. A freehand line tool (the Fiji tool #7, "F6") is used to segment open-ended structures, such as plasma membranes, active zone membranes, and pits.

When an appropriate tool in Fiji is selected (**Table 1**), objects in the micrograph can be segmented using hot keys listed in **Table 2** (**Figure 1**). For open-ended membranes, trace the contour of the membranes as closely as possible using the freehand line tool. From each micrograph, one plasma membrane ("0") and at least one active zone membrane ("9") must be segmented. Since some synapses have multiple active zones (**Figures 2A,B**), judged based on the presence of postsynaptic density in the juxtaposed membranes, more than one active zone can be defined per micrograph. However, a single plasma membrane should be drawn in each image. In some synapses, presynaptic dense projections are prominent in the active zone (Zhai and Bellen, 2004; Watanabe et al., 2013a), and they can be traced using the same tool and added to the ROI Manager by pressing "d" for dense projection and "r" for the synaptic ribbon. If membranes are deflected inward, towards the cytoplasm (**Figures 1A–C**, **2E,F**), they can be traced as pits ("U"), although the exact nature of these membrane invaginations, whether exocytic, endocytic, or simple membrane ruffles, must be determined with careful experiments (Watanabe et al., 2013a,b, 2014). If any of these pits are clearly covered with electron-dense materials indicative of clathrin-coats (Heuser and Reese, 1973), they can be annotated as clathrin-coated pits ("7"). Note that the active zone membrane is drawn under the pit where the plasma membrane would have been before exocytosis when prominent pits like the one in **Figure 1B** are present within the active zone. Pits are classified as being "within the active zone" (**Figure 1B**) if their segmentation overlaps with the traced active zone membrane at two points or if both ends are within 5 nm of the active zone membrane (that is, the entire pit is in the active zone). Otherwise, pits are classified as being "outside the active zone" (**Figures 2E,F**). Thus, these features must be traced carefully so that pits inside and outside the active zone are properly distinguished from each other.

For vesicles, use the straight line tool to draw a line across the outer edges of a vesicle (**Figure 1D**). By doing so, the diameter of a vesicle as well as the x,y-coordinates at the center of the line are recorded in the ROI manager. A circle is drawn on the vesicle based on the diameter. Synaptic vesicles can be manually categorized (**Figures 2G,H**) into docked—no lighter pixels between vesicle membrane and plasma membrane ("3"), tethered when a vesicle is close but not docked and has visible tethers to the plasma membrane ("2"), and all other vesicles in the terminal ("1"; **Figures 2G,H**). If not categorized, docking will be determined by the overlap between the vesicle membrane and the plasma membrane. However, tethered vesicles must be annotated by visual inspection, since they are defined as having a physical tether to the plasma membrane (Watanabe et al., 2013b). The same line tool is used to annotate other types of vesicles including clathrin-coated vesicles ("8," **Figures 2C,D**), dense-core vesicles ("4" and "5" if docked, **Figures 2E–H**), and large vesicles ("6," **Figure 1B**). Large vesicles are clear-core vesicles with a diameter of 60–100 nm that may be involved in exocytosis (He et al., 2009; Borges-Merjane et al., 2020; Maus et al., 2020), endocytosis (Watanabe et al., 2013a,b, 2014; Kononenko et al., 2014), and cargo trafficking (Ou et al., 2010; Vukoja et al., 2018). If preferred, all vesicles can be annotated as synaptic vesicles using "1."

Endosomes are also quite prevalent at presynaptic terminals. Although their identity is difficult to determine from single profiles, we define structures as endosomes if they are larger than 100 nm by visual inspection or have irregular shapes (**Figures 2E,F**; Watanabe et al., 2014, 2018). To trace endosomes, the freehand selection tool ("F4") is used to follow the contour of the putative endosomal membrane, and press "0" from the numeric keypad to add the coordinates ("n0"—hereafter, when a number is preceded by n, it will refer to the number key on the numeric keypad). Their areas are also recorded

TABLE 1 | A list of hot keys, encoded in macros for ImageJ/Fiji.

Analysis tools in ImageJ/Fiji	Tool#	Hot keys
Freehand selection tool	3	F4
Straight line tool	1	F5
Freehand line tool	7	F6

TABLE 2 | A list of hot keys, encoded in macros for ImageJ/Fiji.

Structures to be segmented	Hot keys	Analysis tools in ImageJ/Fiji
Synaptic vesicles	1	Straight line tool
Tethered synaptic vesicles	2	Straight line tool
Docked synaptic vesicles	3	Straight line tool
Dense-core vesicles	4	Straight line tool
Docked dense-core vesicles	5	Straight line tool
Large vesicles	6	Straight line tool
Clathrin-coated pits	7	Freehand line tool
Clathrin-coated vesicles	8	Straight line tool
Active zone membrane	9	Freehand line tool
Plasma membrane	0	Freehand line tool
Endosomes	n0	Freehand selection tool
Ferritin+ synaptic vesicles	n1	Straight line tool
Ferritin+ tethered synaptic vesicles	n2	Straight line tool
Ferritin+ docked synaptic vesicles	n3	Straight line tool
Ferritin+ large vesicles	n4	Straight line tool
Ferritin+ clathrin-coated vesicles	n5	Straight line tool
Ferritin+ pits	n6	Freehand line tool
Ferritin+ multivesicular bodies	n7	Freehand selection tool
Ferritin+ endosomes	n8	Freehand selection tool
Ferritin+ buds on endosomes	n9	Freehand line tool
Pits	u	freehand line tool
Multivesicular bodies	m	Freehand selection tool
Particles	p	Straight line tool
Import a text file	i	N/A
Export ROI as a text file	e	N/A
Synaptic ribbon	r	Freehand line tool
Dense projection	d	Freehand line tool
Buds on endosomes	j	Freehand line tool

in the ROI manager. For multivesicular bodies (MVBs), press "m." Currently, endosomes and MVBs are the only such structures traced in our study, but this analysis can be extended to other irregularly shaped structures like mitochondria and autophagosomes.

Ferritin or gold particles, as well as organelles that contain them, can also be tracked as distinct structures (**Figures 2I–L**). Ferritin is typically used as a fluid phase marker to track recently endocytosed membranes (Watanabe et al., 2013b, 2014, 2018), while gold particles are often conjugated with antibodies or other moieties for affinity interaction to probe proteins of interest (Li et al., 2020). In addition to the plasma membrane and active zone membrane, pits with particles can be annotated using the freehand line tool and recorded with "n6" on the numeric keypad. For particle-containing vesicles, use the straight line tool, and add them to the ROI manager by pressing "n1" for any vesicles, "n2" for tethered vesicles, "n3" for docked vesicles, "n4" for large vesicles, and "n5" for clathrin-coated vesicles (**Table 2**). The particles themselves can be annotated using the same tool and hitting the "p" on the keyboard. Particle-positive endosomes and MVBs are traced using the freehand selection tool and recorded with "n8" and "n7," respectively.

For other structures, the best practice is to annotate using hot keys of structures similar to the structures of interest. For example, mitochondria can be marked as "particle-positive MVBs (n7)" or "particle-positive endosomes (n8)" for the analysis purpose. Likewise, any vesicular structures can be annotated with particle-positive vesicles (n1-n5) since these structures are normally not annotated unless ferritin or something similar is used in the study. Be sure to take notes on what keys are used.

As a cautionary note, the distance between membranes is calculated at the minimum, that is, from the outer leaflet of a vesicle to the inner leaflet of the plasma membrane or active zone membrane. Accordingly, it is highly recommended to use images acquired at a sufficient spatial resolution (i.e., less than 1 nm/pixel) and use a pen tablet to trace objects. Since electron micrographs might not be acquired at the same settings, the contrast should be adjusted in each image to make the features-of-interest appear clear for analysis.

Exporting as a Text File

When segmentation is completed from an image, the annotated structures in the ROI manager can be exported as a text file. Press "e" on the keyboard (**Figure 1B**). This macro then generates a text file containing all the segmented structures in the order of the ROI manager list. The text file is named after the image and automatically saved in the folder where the image is. The Fiji screen advances to the next image after the text file is saved. In the text file, the record of each structure is organized as follows: the tool used to annotate (by its Fiji tool number), the name of a structure, area or length of a structure if it is not a vesicle, x-coordinate(s), y-coordinate(s), and radius of a vesicle if it is a vesicle. These values are separated by a tab character (ASCII 09). The record in the ROI manager will be erased after the export is complete.

Importing a Text File

After completing the analysis for all images, it is important to check whether all the essential components are segmented in every image and the text files are compatible with the Matlab codes. To check, run the "start_data_check" function in Matlab (**Figure 1C**). If any data are missing or more than one plasma membrane are annotated, this function returns the names of the files and the description of the problems associated with the files in the command window. To fix the problems or reevaluate the annotations, the records in a text file can be imported back to the correct image in a stack. On the image of interest, press "i" to import the text file. If there is a text file corresponding to this image in the folder, the segmentation will automatically pop up and the records in the ROI manager. If the list in the ROI manager is modified, press "e" again to export the modified list to the text file. Note that this process will overwrite the existing file.

Unblinding the Text Files

After ensuring that all images are annotated correctly, the resulting text files can be unblinded for further analysis by the Matlab scripts (**Figure 1C**). To unblind the shuffled text files, use the "unblind_me" script in Matlab. This script prompts users to define the directory where the key is and where the text files are.

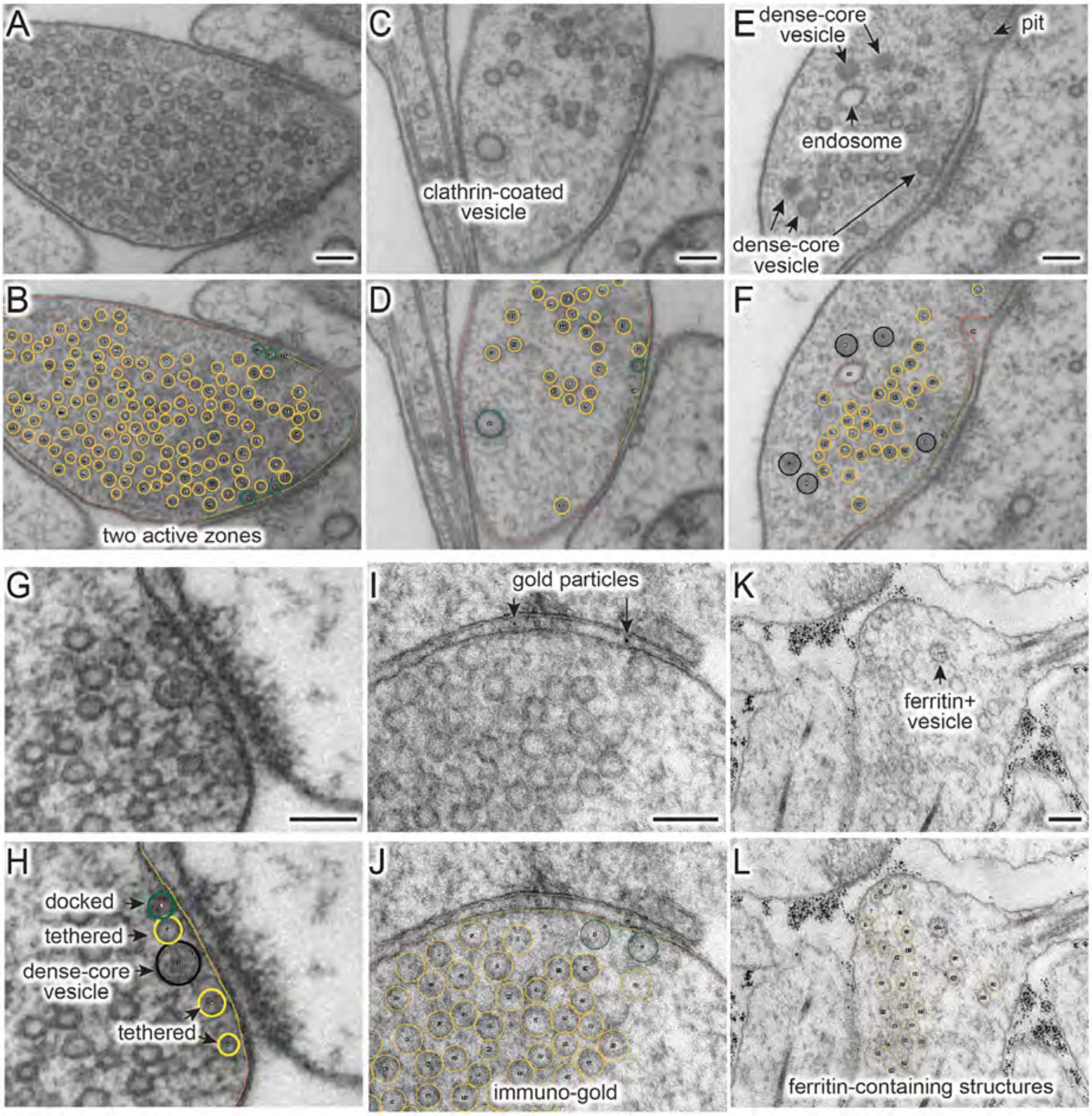


FIGURE 2 | (A–L) Example micrographs **(A,C,E,G,I,K)** and their annotations **(B,D,G,H,J,L)**, showing structures that can be traced using the ImageJ/Fiji macros ("synapsEM_analysis_macro.txt"). **(A,B)** Multiple active zones in a single synaptic profile. **(C,D)** Clathrin-coated vesicle. **(E,F)** Dense-core vesicles and endosomes. **(G,H)** Docked or tethered synaptic vesicles in active zones. **(I,J)** Gold particles. **(K,L)** Ferritin-containing vesicles. See **Table 2** for the full list of structures and hot keys to enter into the ROI manager. Scale bars = 100 nm.

After selecting the files, the text files will be renamed based on the names of the original images and copied into a new folder, called "unblinded," nested within the folder with the shuffled text files. Matlab analysis should be performed using the unblinded text files.

Running Matlab Scripts

The Matlab scripts are designed to use the spatial coordinates and size information of annotated structures to calculate the distances among them and extract count data for annotated features (numbers of synaptic vesicles, docked vesicles, endosomes, etc). To start, select the directory where all the scripts are located, and then type in "start_analysis" in the command line. Since the data analysis for each sample must be run separately, we typically define the name of the sample at this stage (i.e., control_1 = start_analysis;). The program prompts users to input the pixel size for the images (nm/pixel) and the size of the bin (i.e., 50 nm), which is used for plotting the distribution data such as locations of vesicles relative to the active zone. Then, a user interface pops up, first asking to choose the directory where the text files are and then to select all the text files. The files are then loaded into Matlab for processing.

The scripts are designed to perform the following three calculations: size, number, and distribution of distances from the plasma membrane and active zone, of the vesicles and endosomes. First, the sizes of the structures are calculated based on the pixel size, and the mean and median diameter of vesicles, as well as 1-D surface area of and 2-D space within endosomes, are determined from the sample. These numbers are available as an average in a single image or an average for the sample. For pits, the diameter is calculated at the full-width half-maximum. Besides, the scripts also calculate the depth (height), the width at the base, and the surface area of the pits. These data are available as a number array in the final dataset.

Second, the total numbers of structures are calculated from each profile and then their mean and median are determined. The total numbers are additionally sorted based on their locations relative to the active zone and plasma membrane. For example, if a vesicle is within 30 nm of the plasma membrane and also within 30 nm of the active zone, this vesicle is counted towards the vesicle-associated with the active zone. If a vesicle is within 30 nm of the plasma membrane but not associated with the active zone membrane (>30 nm), this vesicle will be categorized as a vesicle in the periactive zone. If neither condition is met, the object is considered cytoplasmic. The distinction between vesicles above the active zone vs. the periactive zone is somewhat arbitrary but useful for detecting certain vesicle pools. For example, synaptic vesicles within 30 nm of the active zone membrane (about two rows of vesicles) are thought to contribute to the readily releasable pool (Schikorski and Stevens, 2001), and their numbers are often reported (Richmond et al., 2001; Hammarlund et al., 2007). Vesicles or pits in the periactive zone reflect endocytic events since they correlate with the internalization of fluid phase markers and are typically observed hundreds of milliseconds after an action potential (Watanabe et al., 2013b, 2014, 2018). In contrast, pits within the active zone represent fusing vesicles since they appear a few milliseconds after an action potential (Watanabe et al., 2013a,b; Kusick et al., 2018). The distance threshold in nanometers can be moved by changing the number in line number 65 in the source code (vesicle_count.m) from 30 to the desired number. After the analysis, the number of data is available in the "vesicle_number" table as a number array. The key to interpreting the array is listed in **Supplementary Table 2**.

Third, they calculate the minimal distance from each structure to the plasma membrane and active zone membrane and determine the distribution of each structure relative to these membranes. For a vesicle, the distance from the center to every point on the plasma membrane, active zone, and if annotated, dense projection is calculated, and the radius of the vesicle is subtracted such that the distance is determined from the outer edge of the vesicle to the membrane. Then, the minimal distance is reported as the final distance. If the distance is calculated to be 0 nm from the plasma membrane, the vesicles are considered docked, and they will be categorized into the docked pool for the numerical calculations. For endosomes and other irregularly shaped organelles, to determine the distance to the plasma membrane and active zone, we calculate numerous distances from the organelle membrane to the synaptic membrane and find their minimum. The distances can be plotted as continuous frequency distribution with no binning if enough data are collected. However, the distribution of structures is typically determined by calculating the number of the structures at certain distances away from the active zone membrane or if annotated, the dense projection based on the bin the user specifies (i.e., 50 nm). The resulting tables show their average number and normalized abundance at each bin (**Supplementary Table 1**).

The output of the Matlab scripts appears in the workspace as a structure array and can be compared between different conditions or samples (e.g control vs. mutants, or glutamatergic vs. GABAergic neurons). To compare, all other samples in a single experiment should be processed by the same procedure (i.e., mutant_1 = start_analysis; in the command line). After all the samples are processed, the workspace should be saved in .mat format.

For plotting the data and statistical analysis, we export the data to Prism. Several optional scripts are available to re-organize the data for exporting. Please refer to the **Supplementary Information**. Step-by-step protocols are also available in the **Supplementary Information**.

RESULTS

To validate program scripts, the computed data were compared to manually segmented data. Specifically, we segmented 10 images using the procedures described above and calculated distances from ∼25–30 vesicles to the active zone membrane using the Matlab scripts (**Figure 3A**: diameter of synaptic vesicles). Then, we manually measured the distances from those vesicles to the nearest active zone membrane based on visual inspection. We repeated the measurements three times to estimate errors caused by manual measurement. We then plotted the disparities between distances determined by the different methods. On average, the difference between the calculated and measured distances was 1.4 nm (**Figure 3B**; median and 95% CI). This number is similar to the error made by repeating the manual measurements three times on the same set of images (1.3 nm median, **Figure 3B**). The overall distribution of synaptic vesicles is also unchanged (**Figure 3C**). Thus, the calculations based on the x,y-coordinates of structures are valid and produce data as accurate as manual measurements.

Based on the data from these 10 images, we determined key synaptic parameters of synapses from cultured mouse hippocampal neurons (14 days *in vitro*). The diameter of synaptic vesicles was 39.6 ± 0.2 nm (mean ± SEM) when all vesicles in the set were pooled, and 40.1 ± 0.6 nm (mean ± SEM), when the numbers were first averaged per profile and then the mean of means was determined from the entire dataset (**Figure 3D**). The mean of means would better represent the population average. The average number of synaptic vesicles per profile was 51 (**Figure 3E**, median, and 95% CI shown). About four vesicles were found within 30 nm of the active zone membrane; of these, ∼2 vesicles were docked

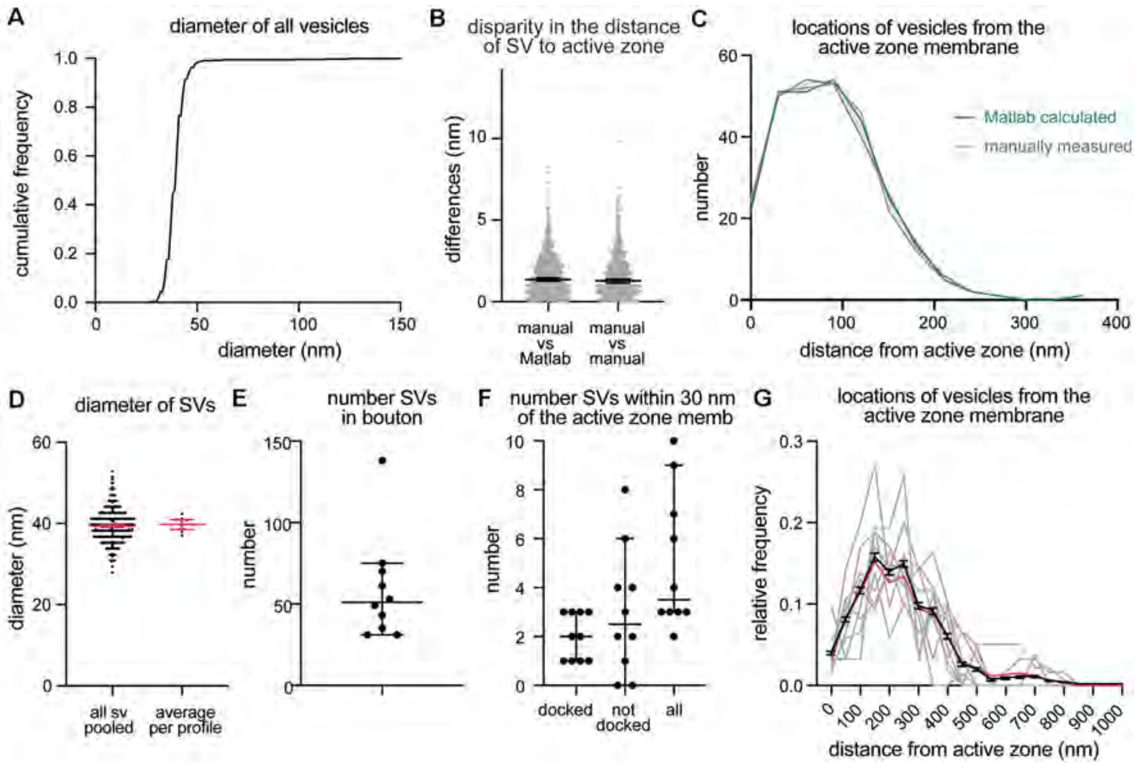


FIGURE 3 | Example plots that can be readily generated with the synapsEM. **(A)** A cumulative plot showing the diameter of all vesicles annotated from 10 sample images used in this study. A total of 590 vesicles are annotated. **(B)** A scatter plot showing the disparity in distances of vesicles to the active zone membrane between the Matlab calculated and manually measured (left) or among three manually measured (right). Each dot = one measurement. The medians are 1.4 nm and 1.3 nm, respectively, (p = 0.8, unpaired t-test). **(C)** A plot showing the distribution of vesicle distances from the active zone membrane. No difference is observed between the Matlab calculated and manually measured ($p > 0.99$, Kruskal–Wallis test with Dunn's multiple comparison test). **(D)** A plot showing the diameter of synaptic vesicles, averaged from all vesicles pooled (left, 39.6 ± 0.2 nm, mean ± SEM) and means of each profile (right, 40.1 ± 0.6 nm, mean ± SEM). **(E)** A plot showing the number of vesicles in the terminal. Each dot = the number in each profile. **(F)** A plot showing the number of vesicles docked or tethered at the active zone or all vesicles within 30 nm of the active zone membrane. Each dot = the number in each profile. **(G)** A plot showing the distribution of synaptic vesicle distances to the active zone. Gray lines = the normalized distance distribution from each profile. Black line = average from 10 profiles. Red line = the normalized abundance from the pooled data.

on average (**Figure 3F**). It is sometimes useful to normalize vesicle numbers and docking to the size of the active zone. To accommodate this calculation, the length of active zones is accessible in the output (**Supplementary Table 2**), and the average is also determined (median = 385 nm). Although n is small, these numbers are surprisingly close to the numbers we have obtained from thousands of images across many experiments (Watanabe et al., 2013b, 2014; Kusick et al., 2018; Li et al., 2020). The distribution of distances from synaptic vesicles to the active zone can be mapped from each synaptic profile (gray lines), averaged numbers per profile (black line, mean ± SEM), and the data all combined (**Figure 3G**, red line). Thus, typical synaptic parameters can be measured and plotted using SynapsEM.

SynapsEM was used to segment data from other model systems. We performed the same analysis using serial sections of *C. elegans* neuromuscular junctions (**Figure 4**, n = 5 reconstructed synapses). In these reconstructions, the numbers can be calculated per synaptic profile containing a dense projection or per fully reconstructed synapse (end-to-end of a synaptic bouton defined by the presence of synaptic vesicles). The average number of synaptic vesicles per profile and per reconstructed synapse were 35.5 and 394, respectively (**Figures 4A–C,E**). Of the ∼5 vesicles within 30 nm of the active zone membrane in single profiles, an average of 3 are docked per synaptic profile (**Figure 4D**), thus, 8.5% of the total vesicle pool are docked in the active zone. Similarly, 9% of the total vesicles in the reconstructed boutons were docked (**Figure 4F**). Thus, the docked pool can be estimated from the synaptic profile data, as has been done in previous publications (Hammarlund et al., 2007; Watanabe et al., 2013a). Since dense projections are apparent at these synapses (**Figure 4A**), distances between synaptic vesicles and docked vesicles and the dense projection can be calculated (**Figures 4G,H**). The median radial distance from the dense projection for all vesicles was 140 nm, while the median radial distance from the dense projection for docked vesicles was 67 nm, suggesting that vesicles tend to dock near the dense projections, where voltage-gated calcium channels are harbored (Gracheva et al., 2008). Thus, SynapsEM works with both 2D and 3D datasets from multiple model systems.

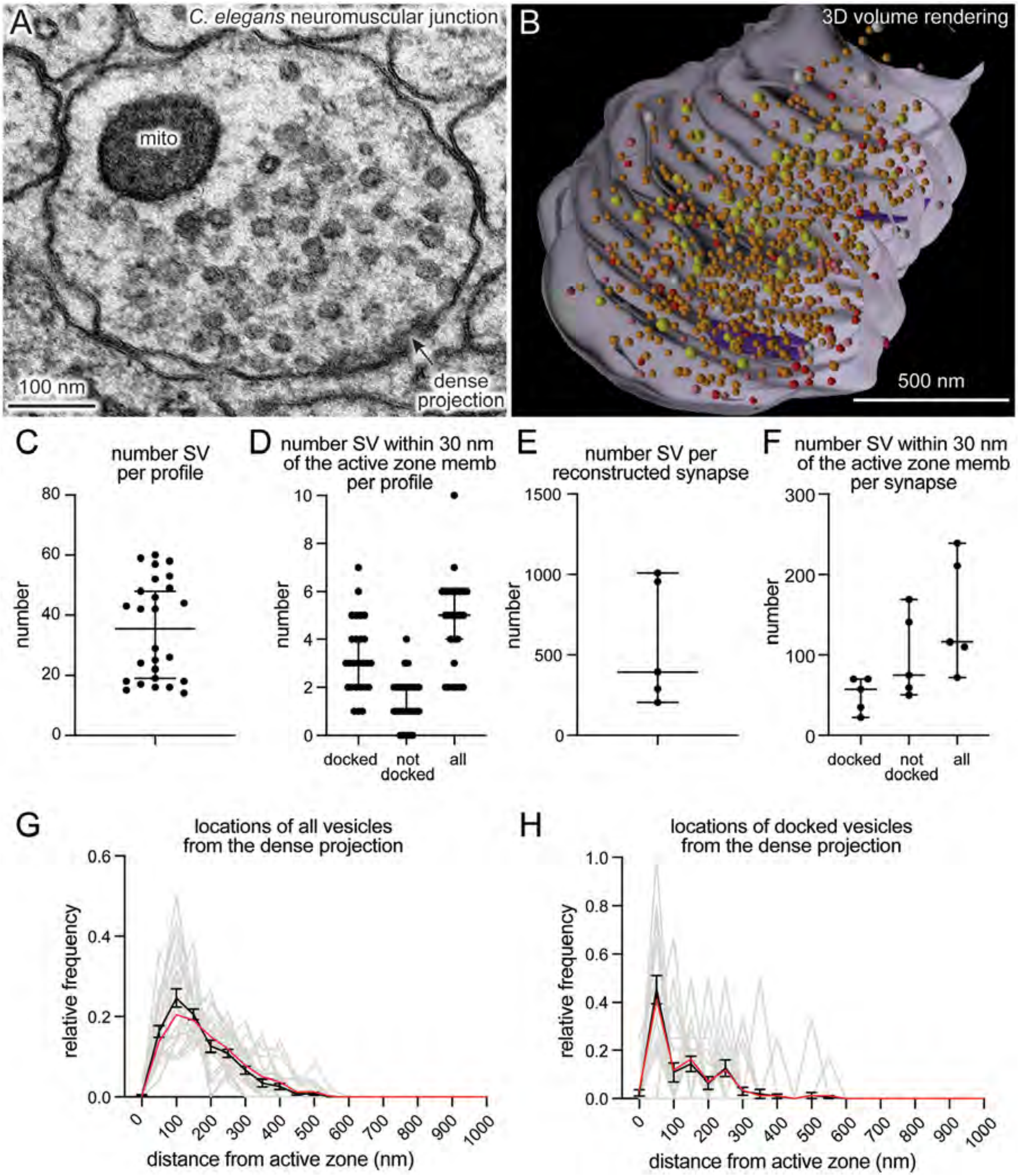


FIGURE 4 | SynapsEM works with other model systems and with 3D reconstruction programs. **(A)** An example micrograph from a serial reconstruction of a *C. elegans* neuromuscular junction, shown in **(B)**. Fourty-eight raw electron micrographs (33 nm each) in series are segmented in Fiji, and x,y-coordinates of each structure exported into text files (i.e., x,y-coordinates from the contour of endosomes, dense projections, plasma membrane). Using x,y-coordinates, polygonal meshes on membranes and dense projections are created using the "loft" command, and vesicles created using the "sphere" command in Maya (see **Supplementary Information**). The z-coordinates for all structures in each slice were determined based on the slice number and increments of 33 nm. **(B)** A snap-shot of the reconstructed synapse using Maya, based on the x,y-coordinates collected from each profile in this study. Red = docked synaptic vesicles; Orange = synaptic vesicles; yellow = dense-core vesicles; white = large vesicles; purple = dense projections. **(C)** A plot showing the number of vesicles in single synaptic profiles. Each dot = the number in a profile. **(D)** A plot showing the number of vesicles within 30 nm of the active zone membrane per profile, and the number of those vesicles docked or tethered at the active zone. Each dot = the number in each profile. **(E)** The total number of synaptic vesicles from synapses fully reconstructed from serial electron micrographs. **(F)** the number of vesicles within 30 nm of the active zone membrane per profile and the number of those vesicles docked or tethered at the active zone from fully reconstructed synapses. **(G)** A plot showing the distribution of synaptic vesicle distances to the dense projection. **(H)** A plot showing the distribution of docked synaptic vesicle distances to the dense projection. In **(G,H)**, gray lines = the normalized distribution from each profile. Black line = average from 26 profiles. Red line = the normalized abundance from the pooled data.

DISCUSSION

Ultrastructural analysis of synapses has been performed by many labs over the years. Excellent programs, for example, IMOD (Kremer et al., 1996), TrakEM2 (Cardona et al., 2012), and Reconstruct (Fiala, 2005; SynapseWeb, Kristen Harris), provide visualization software for data acquired from either serial sections or tomograms. SynapseEM is a Fiji plug-in designed to quantify morphometric data from electron micrographs. IMOD, TrakEM2, and Reconstruct can all perform similar measurements but focus more on 3-D data, with in-depth features for handling and rendering serial images or tomograms not included in SynapsEM. TrakEM2 in particular offers similar quantification procedures; SynapsEM's benefit is an all-in-one package, from scrambling raw images to outputting final data tables, that is fast and easy to use even for those completely unfamiliar with Fiji or Matlab, but also easily modified. These procedures have been used by everyone from novices to experts to quantitate tens of thousands of 2-D images and over a thousand 3-D reconstructions (Watanabe et al., 2013b, 2014, 2018; Kusick et al., 2018; Li et al., 2020). Although SynapsEM handles serial-section data, it is difficult to annotate structures like endosomes and multivesicular bodies that span across multiple sections, since images are randomized. Thus, for serial-section data, careful re-evaluation is necessary after unblinding. Alternatively, one can skip the randomization step if desired.

Several features streamline and standardize the characterization of synaptic features. First, this approach allows multiple experimenters to assess the validity of annotations, reducing potential errors in the analysis. Second, automated shuffling of images from different conditions reduces potential bias in the analysis. Third, additional parameters can be extracted from the dataset *post hoc*, since the positions of structures are all recorded. For example, the locations of pits relative to the center of an active zone can be calculated based on their coordinates (Kusick et al., 2018; Li et al., 2020; also see **Supplementary Information**). Fourth, this approach can also be applied to serial-sections to calculate distances in three-dimensions (Kusick et al., 2018), similar to IMOD (Kremer et al., 1996) and Reconstruct (Fiala, 2005; SynapseWeb, Kristen Harris). Fifth, the 3D dataset can be rendered into a segmented volume in Maya based on the x,y-coordinates of the structures and their sizes (**Figure 4B**; see **Supplementary Information** for the procedure). Thus, SynapsEM is a versatile approach for the morphometry of synapses. For truly universal and automated analysis, the implementation of the machine-learning-based algorithms into the SynapsEM platform is awaited. It is hoped that SynapsEM will promote data sharing and consistent morphometry of synaptic ultrastructure among labs.

DATA AVAILABILITY STATEMENT

All the data underlying this study can be found in the **Supplementary Material**. Data Sheet 2 includes text files generated by Fiji, Data Sheet 3 includes the Matlab files generated by running analysis on these text files, and Data Sheet 4 includes final data tables used to generate the figures.

AUTHOR CONTRIBUTIONS

SW, MD, and EJ conceived and designed the experiments and wrote the manuscript. GK and SW wrote the protocols. MD wrote the original macro for ImageJ/Fiji. SW wrote the Matlab scripts. JI wrote the Maya code for volume rendering. All authors contributed to the article and approved the submitted version.

FUNDING

SW and this work were supported by start-up funds from the Johns Hopkins University School of Medicine, Johns Hopkins Discovery funds, Johns Hopkins Catalyst Award, the National Science Foundation (1727260), the National Institutes of Health (1DP2 NS111133-01 and 1R01 NS105810-01A1) awarded to SW. SW is an Alfred P. Sloan fellow, McKnight Foundation Scholar, and Klingenstein and Simons Foundation scholar. GK was supported by a grant from the National Institutes of Health to the Biochemistry, Cellular and Molecular Biology program of the Johns Hopkins University School of Medicine (T32 GM007445) and is a National Science Foundation Graduate Research Fellow (2016217537). MD's work was supported by the National Institutes of Health (R01 NS034307) awarded to EJ. EJ is an Investigator of the Howard Hughes Medical Institute.

ACKNOWLEDGMENTS

We thank H. James de St. Germain for training SW on Matlab coding and Marc Hammarlund for discussion. We are also indebted to Edward Hujber, Thien Vu, Morven Chien, and Stephen Alexander Lee for additional MatLab routines.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnsyn.2020.584549/full#supplementary-material.

REFERENCES

Borges-Merjane, C.,Kim, O., and Jonas, P. (2020). Functional electron microscopy ("Flash and Freeze") of identified cortical synapses in acute brain slices. *Neuron* 105, 992–1006.e6. doi: 10.1016/j.neuron.2019.12.022

Cardona, A., Saalfeld, S., Schindelin, J., Arganda-Carreras, I., Preibisch, S., Longair, M., et al. (2012). TrakEM2 software for neural circuit reconstruction. *PLoS One* 7:e38011. doi: 10.1371/journal.pone.0038011

Ceccarelli, B., Hurlbut, W. P., and Mauro, A. (1972). Depletion of vesicles from frog neuromuscular junctions by prolonged tetanic stimulation. *J. Cell Biol.* 54, 30–38. doi: 10.1083/jcb.54.1.30

Chang, S., Trimbuch, T., and Rosenmund, C. (2018). Synaptotagmin-1 drives synchronous Ca^{2+} triggered fusion by C2B domain-mediated synaptic vesicle-membrane attachment. *Nat. Neurosci.* 21, 33–40. doi: 10.1038/s41593-017-0037-5

Couteaux, R., and Pécot-Dechavassine, M. (1970). Synaptic vesicles and pouches at the level of "active zones" of the neuromuscular junction. *C. R. Hebd. Seances Acad. Sci. Ser. Sci. Nat.* 271, 2346–2349.

Dittman, J., and Ryan, T. A. (2009). Molecular circuitry of endocytosis at nerve terminals. *Annu. Rev. Cell Dev. Biol.* 25, 133–160. doi: 10.1146/annurev.cellbio.042308.113302

Fiala, J. C. (2005). Reconstruct: a free editor for serial section microscopy. *J. Microsc.* 218, 52–61. doi: 10.1111/j.1365-2818.2005.01466.x

Gracheva, E. O., Hadwiger, G., Nonet, M. L., and Richmond, J. E. (2008). Direct interactions between *C. elegans* RAB-3 and Rim provide a mechanism to target vesicles to the presynaptic density. *Neurosci. Lett.* 444, 137–142. doi: 10.1016/j.neulet.2008.08.026

Hammarlund, M., Palfreyman, M. T., Watanabe, S., Olsen, S., and Jorgensen, E. M. (2007). Open syntaxin docks synaptic vesicles. *PLoS Biol.* 5:e198. doi: 10.1371/journal.pbio.0050198

He, L., Xue, L., Xu, J., McNeil, B. D., Bai, L., Melicoff, E., et al. (2009). Compound vesicle fusion increases quantal size and potentiates synaptic transmission. *Nature* 459, 93–97. doi: 10.1038/nature07860

Heuser, J. E., and Reese, T. S. (1973). Evidence for recycling of synaptic vesicle membrane during transmitter release at the frog neuromuscular junction. *J. Cell Biol.* 57, 315–344. doi: 10.1083/jcb.57.2.315

Heuser, J. E., and Reese, T. S. (1981). Structural changes after transmitter release at the frog neuromuscular junction. *J. Cell Biol.* 88, 564–580. doi: 10.1083/jcb.88.3.564

Heuser, J. E., Reese, T. S., Dennis, M. J., Jan, Y., Jan, L., and Evans, L. (1979). Synaptic vesicle exocytosis captured by quick freezing and correlated with quantal transmitter release. *J. Cell Biol.* 81, 275–300. doi: 10.1083/jcb.81.2.275

Imig, C., Min, S.-W., Krinner, S., Arancillo, M., Rosenmund, C., Südhof, T. C., et al. (2014). The morphological and molecular nature of synaptic vesicle priming at presynaptic active zones. *Neuron* 84, 416–431. doi: 10.1016/j.neuron.2014.10.009

Kononenko, N. L., and Haucke, V. (2015). Molecular mechanisms of presynaptic membrane retrieval and synaptic vesicle reformation. *Neuron* 85, 484–496. doi: 10.1016/j.neuron.2014.12.016

Kononenko, N. L., Puchkov, D., Classen, G. A., Walter, A. M., Pechstein, A., Sawade, L., et al. (2014). Clathrin/AP-2 mediate synaptic vesicle reformation from endosome-like vacuoles but are not essential for membrane retrieval at central synapses. *Neuron* 82, 981–988. doi: 10.1016/j.neuron.2014.05.007

Kremer, J. R., Mastronarde, D. N., and McIntosh, J. R. (1996). Computer visualization of three-dimensional image data using IMOD. *J. Struct. Biol.* 116, 71–76. doi: 10.1006/jsbi.1996.0013

Kusick, G. F., Chin, M., Lippmann, K., Adula, K. P., Davis, M. W., Jorgensen, E. M., et al. (2018). Synaptic vesicles undock and then transiently dock after an action potential. *BioRxiv*

Kusick, G. F., Chin, M., Raychaudhuri, S., Lippmann, K., Adula, K. P., Hujber, E. J., et al. (2020). Synaptic vesicles transiently dock to refill release sites. *Nat. Neurosci.*, 23, 1329–1338. doi: 10.1038/s41593-020-00716-1

Li, S., Raychaudhuri, S., Lee, S. A., Wang, J., Kusick, G. F., Prater, C., et al. (2020). Release sites are positioned to activate NMDA receptors. *BioRxiv* [Preprint]. doi: 10.1101/2020.05.01.072157

Maus, L., Lee, C., Altas, B., Sertel, S. M., Weyand, K., Rizzoli, S. O., et al. (2020). Ultrastructural correlates of presynaptic functional heterogeneity in hippocampal synapses. *Cell Rep.* 30, 3632–3643.e8. doi: 10.1016/j.celrep.2020.02.083

Ou, C.-Y., Poon, V. Y., Maeder, C. I., Watanabe, S., Lehrman, E. K., Fu, A. K. Y., et al. (2010). Two cyclin-dependent kinase pathways are essential for polarized trafficking of presynaptic components. *Cell* 141, 846–858. doi: 10.1016/j.cell.2010.04.011

Richmond, J. E., Weimer, R. M., and Jorgensen, E. M. (2001). An open form of syntaxin bypasses the requirement for UNC-13 in vesicle priming. *Nature* 412, 338–341. doi: 10.1038/35085583

Saheki, Y., and De Camilli, P. (2012). Synaptic vesicle endocytosis. *Cold Spring Harb. Perspect. Biol.* 4:a005645. doi: 10.1101/cshperspect.a005645

Schikorski, T., and Stevens, C. F. (1997). Quantitative ultrastructural analysis of hippocampal excitatory synapses. *J. Neurosci.* 17, 5858–5867. doi: 10.1523/JNEUROSCI.17-15-05858.1997

Schikorski, T., and Stevens, C. F. (2001). Morphological correlates of functionally defined synaptic vesicle populations. *Nat. Neurosci.* 4, 391–395. doi: 10.1038/86042

Shepherd, G. M. G., and Harris, K. M. (1998). Three-dimensional structure and composition of CA3→CA1 axons in rat hippocampal slices: implications for presynaptic connectivity and compartmentalization. *J. Neurosci.* 18, 8300–8310. doi: 10.1523/JNEUROSCI.18-20-08300.1998

Vukoja, A., Rey, U., Petzoldt, A. G., Ott, C., Vollweiter, D., Quentin, C., et al. (2018). Presynaptic biogenesis requires axonal transport of lysosome-related vesicles. *Neuron* 99, 1216–1232.e7. doi: 10.1016/j.neuron.2018.08.004

Watanabe, S., Liu, Q., Davis, M. W., Hollopeter, G., Thomas, N., Jorgensen, N. B., et al. (2013a). Ultrafast endocytosis at Caenorhabditis elegans neuromuscular junctions. *eLife* 2:e00723. doi: 10.7554/eLife.00723

Watanabe, S., Rost, B. R., Camacho-Pérez, M., Davis, M. W., Söhl-Kielczynski, B., Rosenmund, C., et al. (2013b). Ultrafast endocytosis at mouse hippocampal synapses. *Nature* 504, 242–247. doi: 10.1038/nature12809

Watanabe, S., Mamer, L. E., Raychaudhuri, S., Luvsanjav, D., Eisen, J., Trimbuch, T., et al. (2018). Synaptojanin and endophilin mediate neck formation during ultrafast endocytosis. *Neuron* 98, 1184–1197.e6. doi: 10.1016/j.neuron.2018.06.005

Watanabe, S., Trimbuch, T., Camacho-Pérez, M., Rost, B. R., Brokowski, B., Söhl-Kielczynski, B., et al. (2014). Clathrin regenerates synaptic vesicles from endosomes. *Nature* 515, 228–233. doi: 10.1038/nature13846

Zhai, R. G., and Bellen, H. J. (2004). The architecture of the active zone in the presynaptic nerve terminal. *Physiology* 19, 262–270. doi: 10.1152/physiol.00014.2004

Zhang, B., Koh, Y. H., Beckstead, R. B., Budnik, V., Ganetzky, B., and Bellen, H. J. (1998). Synaptic vesicle size and number are regulated by a clathrin adaptor protein required for endocytosis. *Neuron* 21, 1465–1475. doi: 10.1016/S0896-6273(00)80664-9

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Watanabe, Davis, Kusick, Iwasa and Jorgensen. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

Supplementary information

Step-by-step protocols for running SynapsEM are described below. The **bold** font indicates the actions you need to take. More descriptions are in regular fonts and also found in the Methods section of the manuscript. The texts in red are cautionary notes.

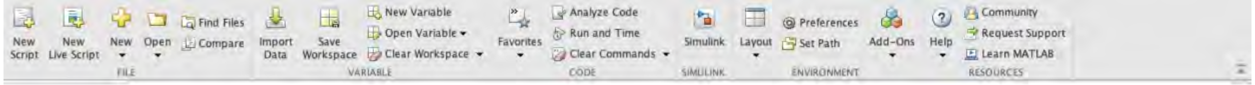
Randomizing images

1. Getting started with Matlab:

1.1 Open Matlab.

-At the center is the Command Window, where code is run. Scripts are called by typing in their name (for example, for "randomize.m", type "randomize") and pressing enter/return. Scripts will sometimes ask for input while they are running (for example, a yes/no question: "do you want to duplicate the images y/n", or asking for a number input "what is the pixel size?"). Simply type the response into the Command Window and press enter/return to continue running the code.

-At the right is the Workspace. This is where Matlab stores "variables": for SynapsEM, this is where your data will be stored. The "Save Workspace" button in the "Home" panel of the top toolbar will save all the current variables as a .mat file.

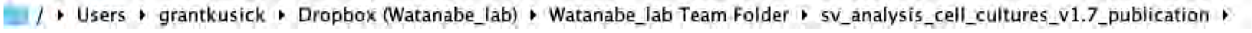


-At the left is the Current Folder. This is the directory where the Command Window can call scripts (.m files) from. Before you get started, you will need to set the directory to where you have SynapsEM code stored.

-When you open up something in the Current Folder (to edit/look at scripts) or the Workspace (to look at data), the Command Window will move to the bottom of the screen.

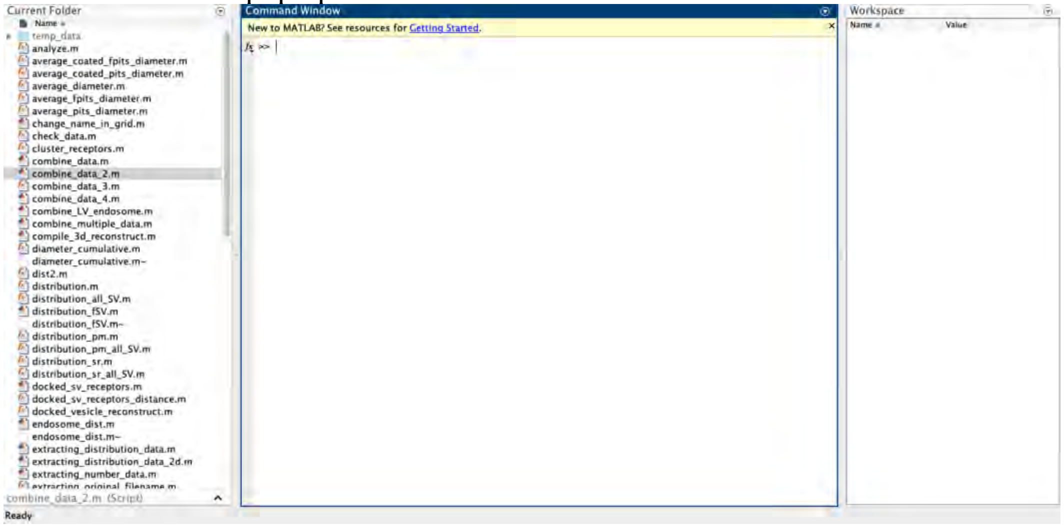
2. Setting the Current Folder to where you have the code stored:

2.1 At the top of the screen, below the toolbar, you will see the file path to where the Current Folder is currently set to. **Click through the file path until it ends with the folder where you have all SynapsEM code stored**. For example:



/ ▸ Users ▸ grantkusick ▸ Dropbox (Watanabe_lab) ▸ Watanabe_lab Team Folder ▸ sv_analysis_cell_cultures_v1.7_publication ▸

-The codes should pop up under Current Folder. Your screen should look like this:

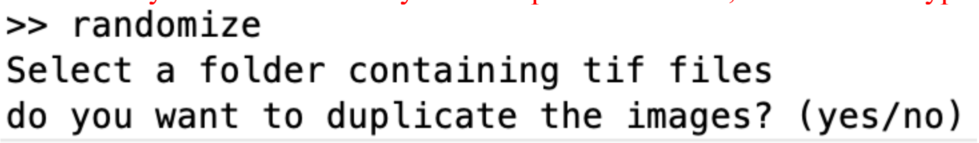


New to MATLAB? See resources for Getting Started.

```
>>
```

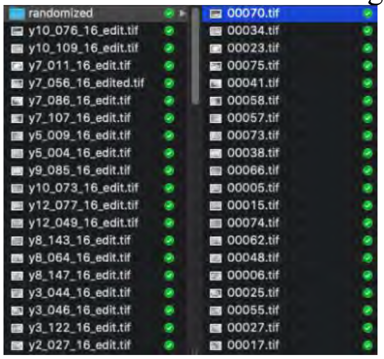
-Matlab will remember your Current Folder after closing, so this step doesn't need to be repeated each time it's opened.
-Scripts can be inspected or edited by double clicking on the .m files. This will open a "script editor" panel. SynapsEM scripts are heavily commented (in green text), so inspecting these files can be helpful if you are troubleshooting or simply want to know more about how the scripts work.

3. Randomizing images:
3.1 Move all your image files for a single experiment into a single folder, or duplicates of these files.
3.2 In Matlab, run the "randomize" script*.
3.3 Select the folder that contains the images
3.4 Select all the images that you want to randomize.
-The code will ask whether you want to duplicate the images. Answering yes will make a copy of the files and randomize them. Answering no will randomize the names of the files you selected
Warning: if you answer "no" the code will overwrite the original file names of the files you selected. If you have not already made copies of the files, make sure to type "yes".



```
>> randomize
Select a folder containing tif files
do you want to duplicate the images? (yes/no)
```

-The code will create a new folder called "randomized" within the folder you selected. This should contain the same images you selected but with scrambled file names.

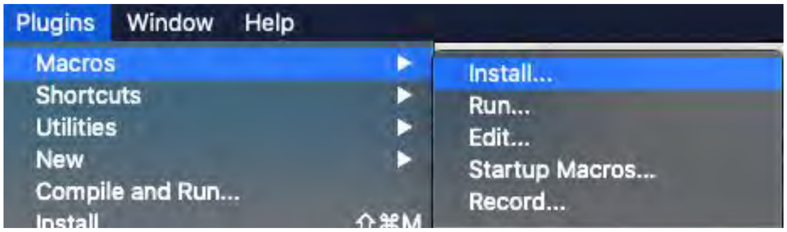


randomized
y10_076_16_edit.tif
y10_109_16_edit.tif
y7_011_16_edit.tif
y7_056_16_edited.tif
y7_086_16_edit.tif
y7_107_16_edit.tif
y5_009_16_edit.tif
y5_004_16_edit.tif
y9_085_16_edit.tif
y10_073_16_edit.tif
y12_077_16_edit.tif
y12_049_16_edit.tif
y8_143_16_edit.tif
y8_064_16_edit.tif
y8_147_16_edit.tif
y3_044_16_edit.tif
y3_046_16_edit.tif
y3_122_16_edit.tif
y2_027_16_edit.tif

00070.tif
00034.tif
00023.tif
00075.tif
00041.tif
00058.tif
00057.tif
00073.tif
00038.tif
00066.tif
00005.tif
00015.tif
00074.tif
00062.tif
00048.tif
00006.tif
00025.tif
00055.tif
00027.tif
00017.tif

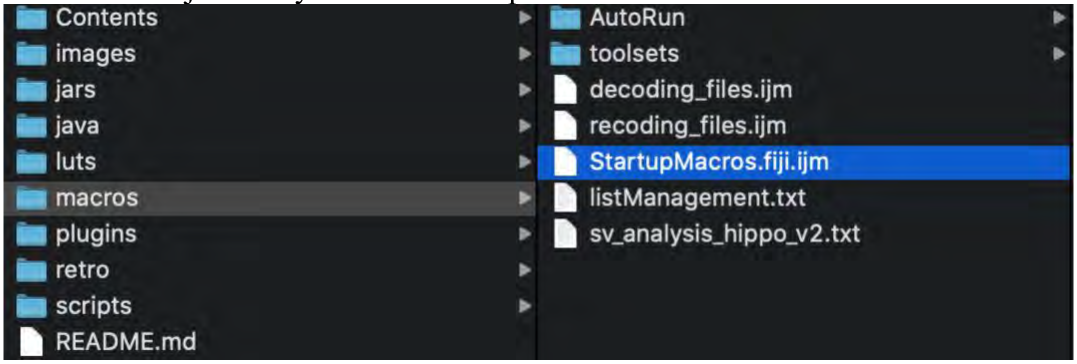
-Important: the randomized folder should also contain a Matlab file called "key". Do not lose this file: you will need it to unscramble your text files later.

4. Getting started with Fiji:
4.1 Open Fiji
4.2 Select "Install"

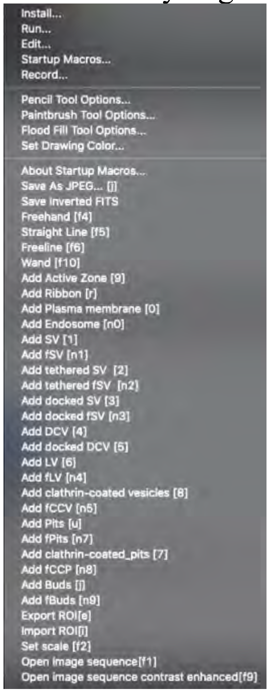


4.3 **Install the file “synapsEM_analysis_macro.txt”**

-This process has to be done each time you boot up Fiji. To avoid this, you can copy the text in synapsEM_analysis_macro.txt into the text file “StartupMacros.fiji.ijm”. This is the list of macros that Fiji calls by default when opened.



-Now when you go to “Macros” it should look like this:



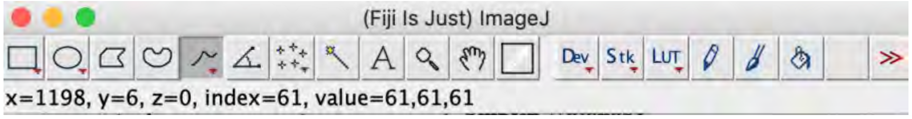
This is a list of the macros and the keyboard shortcuts used for each. “n” indicates the number pad. For example pressing “0” on the top number row of your keyboard will execute the add plasma membrane macro, whereas pressing 9 on the number pad will execute the add fBuds macro. Refer **Table 2** for a complete set of macros.

4.4 **Open a stack of your scrambled images by pressing F1.** Important: you cannot simply drag and drop images into Fiji, you must use the F1 macro. Note: if you are not sure if you used F1 to open images, press F2 to set the scale before stating the annotation.

5. Segmenting images

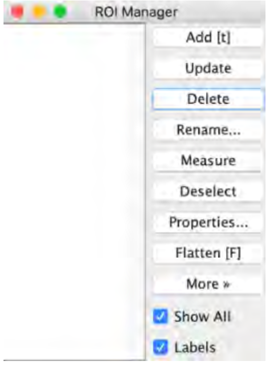
Segment features as described in the Methods section and corresponding figures.

5.1 Press F6 to activate the freehand line tool.



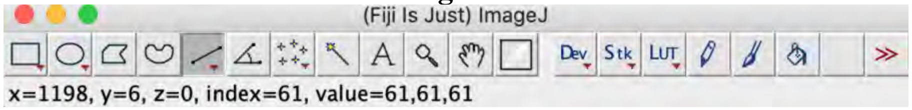
x=1198, y=6, z=0, index=61, value=61,61,61

5.2 Trace the contour of the plasma membrane and press “0” on the keyboard. Important: you can only annotate one plasma membrane per image. Make sure “show all” and “labels” are selected in the ROI manager so you know which structures are already segmented.



5.3 Trace the contour of the active zone membrane and press “9” on the keyboard. You can select multiple active zones if necessary.

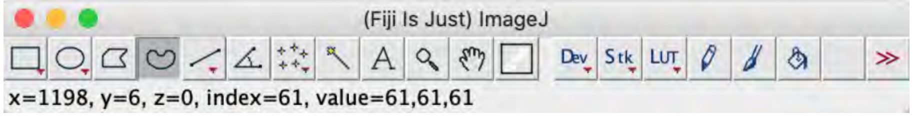
5.4 Press F5 to activate the straight line tool.



x=1198, y=6, z=0, index=61, value=61,61,61

5.5 Draw a line across a vesicle (outer edge to outer edge) and press appropriate numbers on the keyboard (see Table 2). For examples, press “1” for a synaptic vesicle and “3” for docked synaptic vesicles. **Repeat this procedure until all vesicular structures are annotated.**

5.6 Press F4 to activate the freehand selection tool.



x=1198, y=6, z=0, index=61, value=61,61,61

5.7 Trace the contour of an endosome or multivesicular body and press “n0” on the numerical keypad for an endosome and “m” for multivesicular body.

The order of 5.1-5.6 is not important, but for the Matlab program, exactly one plasma membrane and at least one active zone must be annotated.

6. Exporting text files

6.1 Press “e” on the keyboard. The macro generates a text file containing all the segmented structures in the order of the ROI manager list. The text file is named after the image and automatically saved in the folder where the image is. The Fiji screen advances to the next image, and the text file is saved. In the text file, the record of each structure is organized as follows: the tool used to annotate (by its Fiji tool number), the name of a structure, area or length of a structure if it is not a vesicle, x-coordinate(s), y-coordinate(s), and radius of a vesicle if it is a vesicle. These values are separated by a tab character (ASCII 09). The record in the ROI manager will be erased after the export is complete.

Repeat 5-6 until all images are annotated.

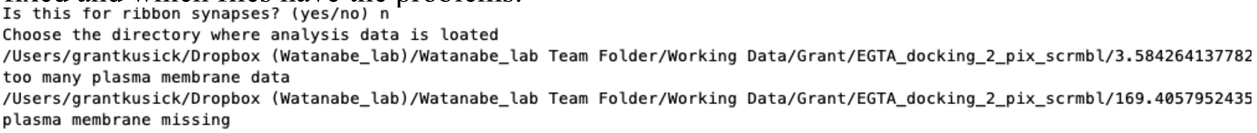
7. Checking data

Before running the Matlab analysis script, you need to check that no errors are in the data that would cause the code to fail to run (missing active zone, multiple plasma membranes, etc.).

7.1 Run the script 'start_data_check'.

7.2 Select the folder with the files and then the files themselves.

-The script will scan through all the files to see if there are any issues that would cause the analysis code to fail: if the analysis code (described below) has an error caused by the text file, it will not tell you what caused it or which text file had the issue, but start_data_check will. Below is an example of the script running and detecting errors, indicating the type of problem to be fixed and which files have the problems.



```
Is this for ribbon synapses? (yes/no) n
Choose the directory where analysis data is loated
/Users/grantkusick/Dropbox (Watanabe_lab)/Watanabe_lab Team Folder/Working Data/Grant/EGTA_docking_2_pix_scrmbl/3.584264137782
too many plasma membrane data
/Users/grantkusick/Dropbox (Watanabe_lab)/Watanabe_lab Team Folder/Working Data/Grant/EGTA_docking_2_pix_scrmbl/169.4057952435
plasma membrane missing
```

-Fix any errors in the text files as needed (see section **8**).

8. Importing text files (re-annotating images/correcting analysis)

8.1 Press 'i'. Provided the text files and images are still in the same folder and still have the same name, the segmentation will appear on the image. You can now edit the segmentation as desired. Re-checking segmentation, ideally by another member of the lab, is important to ensure accuracy. You can also use this procedure to annotate features that you ignored in your first round of segmentation.

-To view segmentation that has already been done on an image, you must first open an image stack using the f1 macro.

-Export by pressing 'e'. Important: this will automatically overwrite the original text file. If you wish to keep the original files, make copies of them and move them to another folder before re-annotating images.

Repeat 7 to make sure all the text files are compatible.

9. Unscrambling text files

9.1 Open Matlab and call up the SynapsEM scripts as described in section 2.

9.2 Run the script 'unblind_me' by typing in the command window.

9.3 Select the folder containing the text files and decoding key.

9.4 Select the text files.

-The text files will be renamed based on the names of the original images and copied into a new folder, called "unblinded", nested within the folder with the shuffled text files. Further analysis should be performed with the unblinded text files.

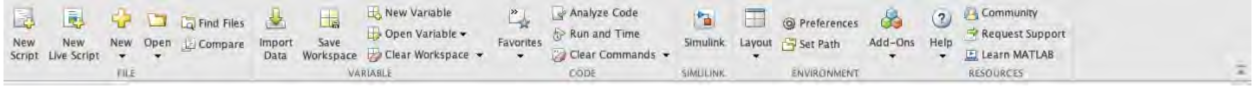
10. Running analysis scripts

10.1 Run 'start_analysis'. This generates all of the data you will need for most experiments, and these numbers serve as the basis for any other analysis scripts you might want to run. When prompted, choose all the text files corresponding to a single sample: for each run MATLAB generates an object in the workspace that contains all the data for that sample. You will be asked for 1) the pixel size in nm (all data in MATLAB will be in nm) and 2) the bin size, which for

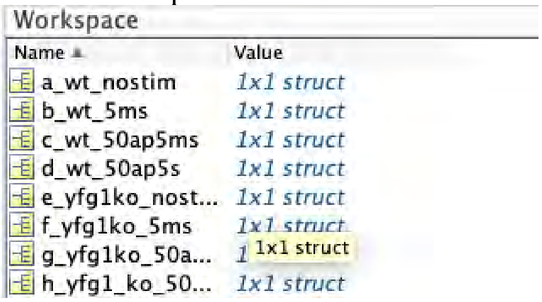
data relating to the distance of objects (SVs, LVs, etc.) to the active zone/plasma membrane will determine how these distances are binned. (in 2 nm increments, in 20 nm increments, etc).
- Name the object after that sample, either by changing it after it's generated (control click on the folder in the Variables window and select rename) or by indicating it in running the code, for example "mutant_1= start_analysis". Important: if you do not specify a name, the data set will have the name "ans" by default. If you run the script again for a new set of txt files, it will overwrite the first data set. Therefore, make sure to name the data set before adding the next data set in.

10.2 Repeat for each sample in the experiment, making sure the name of each data set is correct.

10.3 Save the data by using the "Save Workspace button", in the Home window of the top toolbar.



-This will save a .mat file, which is all of these data sets ("Variables") saved together. Whenever opening a .mat file, you will be asked which of the variables you want to load in; all will be selected by default. The easiest way to keep things organized is to keep one .mat file for each experiment.
-Your Workspace should now look something like this:

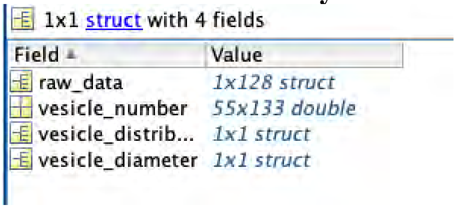

Workspace

Name ▲	Value
a_wt_nostim	1x1 struct
b_wt_5ms	1x1 struct
c_wt_50ap5ms	1x1 struct
d_wt_50ap5s	1x1 struct
e_yfg1ko_nost...	1x1 struct
f_yfg1ko_5ms	1x1 struct
g_yfg1ko_50a...	1x1 struct
h_yfg1_ko_50...	1x1 struct

-Once saved, you can remove folders from the Variables window to reduce clutter as desired, or clear the Variables window by inputting the command "clear all".

11. Examining data

11.1 Double click on any of the folders in the Variables window to open up that data set.


1x1 struct with 4 fields

Field ▲	Value
raw_data	1x128 struct
vesicle_number	55x133 double
vesicle_distrib...	1x1 struct
vesicle_diameter	1x1 struct

-Refer to Table 4 for how the features are arranged in the vesicle_number data table.
-Refer to Table 1 for how features are arranged in the vesicle_distribution tables.
-Move data into your statistical analysis/data visualization platform of choice
-Enjoy the thrill of discovery!

12. Specialized scripts: making data into tables for export

You can export count data manually out of the spreadsheets, but this can be streamlined using the 'extracting_number_data' script.

12.1 Once you run through all your data, change the names in the workspace so that they are in the order you want to have in your final graph. For example:
control no stim
control 100 ms
control 1s
control 10 s
kd no stim
kd 100 ms
kd 1s
kd 10s

-The easiest way to make it in this order is to add alphabet in the beginning.

a_control no stim
b_control 100 ms
c_control 1s
d_control 10 s
e_kd no stim
f_kd 100 ms
g_kd 1s
d_kd 10s

12.2 Once you do so, save the workspace (see 10.3).
12.3 Then in the command window, type in extracting_number_data. It will prompt you to answer 2 questions. Answer yes to the first question if you need usual numbers like docked synaptic vesicles, pits, large vesicles, endosomes, etc. Answer yes to the second question if you want to get ferritin-containing structures, too.

If you would like something special like clathrin-coated pits, say no to both questions. It will ask you to type in the row number you want to get the values from. In the case of clathrin-coated pits, this would be 46.

It should give you sample_out and table in the workspace. If you answered yes to either question, go to sample_out. If not, go to table. The first row is the total number of sections you analyzed for the particular dataset. The columns are organized according to how the names were organized in the workspace (see above, i.e. 1st column, control no stimulation, 2nd, control 100 ms, etc). You can copy the whole table except for the first row.

*The only problem is that the matlab will fill in empty slots with 0s, so if the total number of images analyzed is not uniform across, you will need to remove those extra 0s based on the total number you analyzed for each group. For example, out of the 8 samples mentioned above, if control no stim had 90 images, but others had 100, the Matlab will add ten zeros to the control no stim. Remove those 0s after you copy the data out.

12.4. To extract the size data, type in "extracting_size_data" in the command window and select the directory and saved dataset (.mat). Sample_out shows up in the workspace. Double-click on sample_out and navigate through the variables to access the data. For example, for synaptic vesicle diameter, click on "SV". Two tables are available: "pooled_raw_number" and "mean_profile". The pooled_raw_number pools diameter of all vesicles from every profile in the sample, which is used to generate a plot in **Figure 3C**. Each column represents every vesicle in this case. The mean_profile lists the mean of synaptic vesicles from every synaptic profile. Each column represents the mean diameter of all synaptic vesicles in a profile. The first row is the total number of sections you analyzed for the particular dataset in both cases. The columns are organized according to how the names were organized in the workspace (see above, i.e. 1st column, control no stimulation, 2nd, control 100 ms, etc). You can copy the whole table except for the first row. Again the 0s will need to be eliminated (see 12.3). The diameter data for vesicles are also available as cumulative frequency, as shown in **Figure 3A**. In this table, The columns are also organized according to how the names were organized in the workspace (see above, i.e. 1st column, control no stimulation, 2nd, control 100 ms, etc). Each row is 1-nm bin.

12.5. To extract the distribution data, type in "extracting_sv_distribution_from_pm_data", or "extracting_sv_distribution_from_dp_data", or "extracting_sv_distribution_from_az_memb_data". As the names of the scripts suggest, these would summarize the distribution of vesicles relative to plasma membrane, dense projection, or active zone membrane. Two tables are available in each case: "average" and "norm_abundance". As the names suggest, average shows the mean number of vesicles at each bin. The binning is shown in the column 1, and the numbers from each sample starts from the second column in the same order as how the names were organized in the workspace (see above, i.e. 1st column, control no stimulation, 2nd, control 100 ms, etc). The norm_abundance showns the normalized abundance at each distance instead. These data can be simply copied and pasted into x,y spreadsheet of the Prism with the first column in the x column. Since all the samples are processed with the same bin size, there will be no 0s at the end to remove. Note that the last row is any vesicle over 1000 nm from the plasma membrane, dense projection, or active zone.

13. Specialized scripts: determining the location of features within the active zone/postsynaptic density

Are you interested in whether your gold-labelled protein of interest is concentrated in the center of the post-synaptic density, or distributed throughout? Are you interested in where vesicle fusions are located within the active zone? These codes output count data on these features, either for 2-D or 3-D serial-section analysis and, more importantly, tell you where each feature is relative to the center of the active zone/PSD, and also calculate pairwise distances between your features of interest See Li et al., 2020 and Kusick et al., 2020 for example applications of these scripts.

13.1 Make sure the data you are interested in have already had "start_analysis" run, saved as a .mat file (i.e. you've gone through part 10 of the main protocol). Make sure there are no other variables in the Workspace, as these scripts by default will run on every variable in the Workspace, and if any are not in the expected format, it will trigger an error and the script will fail.

When running the start_xxx_reconstruct codes, make sure you don't have any important unsaved variables in your Workspace. If this code experiences any errors, it will wipe everything currently in the Workspace

13.2 Run the start_xxx_reconstruct code (where xxx are either pits or docked_sv). There are various scripts that determine the locations of different objects within the active zone/post-synaptic density, which all function basically the same way. For this example, we'll use **"start_pit_reconstruct"**.

13.3 Select the folder and then the .mat file you are analyzing. You will be asked if this is for 3D reconstructions. See part 14 for info on dealing with serial-section data.

13.4 Input pixel size.

13.5 Input the thickness of your sections. This is only used if using 3D data, but it will be asked regardless.

13.6 Input pit cutoff height. This will be asked regardless of if you are looking at pits or not. If looking at pits, this will simply sort the pits (indicated by a 1 or 0 in the data table) by height/depth (ie. deep vs shallow), it will still include data on all pits in the active zone regardless of their depth/height.

The code will run through, in this case calculating data for each pit in the active zone. If the code runs properly, you should see "Sample in" and "Sample out" variables in your workspace.

13.7 Open "Sample Out"

In "xxx_Data" you will find a data table (ignore rows that are only 0s). Each row is a single active zone/synapseThe columns are as follows:

1.Distance to the edge of the active zone/PSD

2. Distance to the center of the active zone/PSD

3. *Normalized distance to the center. This is the value you are most likely interested in*

4. Pit height

5. the side index (which side of the active zone pits or docked vesicles are)

If there are another 5 columns in the same row, this is another pit in the same active zone.

13.8 "Distance matrix" data table: this contains all pairwise distances between pits/docked synaptic vesicles, etc. that are in the same active zone. These values are duplicated, mirrored along a diagonal. Simply copy out data from one side of the diagonal.

14. Specialized scripts: Serial-sectioning analysis

14.1 Run "start_analysis" for each synapse (stack of images from a single reconstruction) from a particular sample/condition, then save the workspace. You will have a workspace populated by all the different serial section stacks, each as an individual variable (data set), each named something like synapse 1, synapse 2, etc

14.2 If interested in data from start_analysis: Run the extracting_xxx_data code of your choice, depending on what you are interested in. You can also go through each synapse and pull out numbers manually, but with many serial section stacks this is tedious.

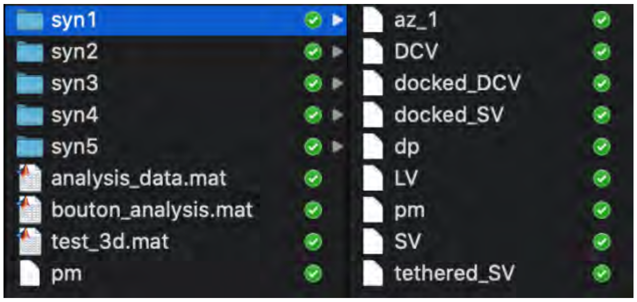
14.3 If interested in reconstruct (location within active zone) data:

Run the start_xxx_reconstruct code of your choice, depending on what you are interested in. The reconstruct codes can give you data on locations of pits, docked vesicles, etc. from serial section stacks. In fact, these codes were originally written to calculate positions of these objects within the active zone in 3D.

Do the same as in **13**, but when asked “is this for 3D reconstruction?” answer yes.
14.4 Delete the “Sample In” variable and save the workspace. This will allow the compile_3d_reconstruct code in the next part to run: you must have not variables in your workspace other than what you get from “Sample out”. Going through each synapse individually is cumbersome, but the next step will compile them into a single data set.
14. Compile the synapses from one sample into a single data set with “compile_3d_reconstruct”. Simply run compile_3d_reconstruct and choose the .mat file that you saved in 14.4. This will pool the pit/vesicle counts from all the synapses from a sample, as well as their locations, into a single data set.

15. Specialized scripts: Aligning pits/docked synaptic vesicles with receptors.
If you marked receptors with gold particles (Li et al., 2020), you can align pits or docked synaptic vesicles to those gold particles, found in the synaptic cleft.
15.1 “start_pits_reconstruct” as instructed above (13.2-13.6). Delete the sample_in and rename sample_out as “pits” or “docked_sv”. Save the workspace.
15.2 Run “start_docked_sv_reconstruct” as instructed above (13.2-13.6). Delete the sample_in and rename sample_out as “docked_sv”. Save the workspace.
15.3 Run “start_particle_recustruct as instructed above (13.2-13.6). Delete the sample_in and rename sample_out as “receptor”. Save the workspace.
15.4 Load all three .mat files into the workspace so that pits, docked_sv, and receptor are available in the workspace. Save the workspace.
15.5 Run “start_alignment_analysis”. Follow the instructions in the command window. The data will be stored in the sample_out. Click on either “xxx_pits.PitDist” or “xxx_docked_sv.SVDist”. The table lists distances from every pit/docked synaptic vesicles to every gold particles (if three gold particles are in the synaptic cleft and 4 docked vesicles are found, a total of 12 distances will be listed).

16. Specialized scripts: Volume rendering. For volume rendering of the 3D reconstruction data as shown in Figure 4B, the location data need to be exported back to the text files. For Maya, the same structures must be in the same text file. To do so, **type “text_convert_maya” in the command window. When prompted, type in the thickness of sections (i.e. 40 nm).** This program will generate new folders with the name of the samples in the same directory where the .mat file is. These text files then can be imported into Maya for volume rendering.



syn1
syn2
syn3
syn4
syn5
analysis_data.mat
bouton_analysis.mat
test_3d.mat
pm

az_1
DCV
docked_DCV
docked_SV
dp
LV
pm
SV
tethered_SV

17. Specialized protocols: Maya 3D modeling and rendering.
17.1 Modeling cells: In Maya, closed curved are generated for each Z section. This is done by reading through the text file and running a “curve” command on x y points that share the same z

coordinate. To create a lofted surface, it is necessary for all of the curves to share the same number of vertices and to have their “seams” aligned. To accomplish this, the curves are selected and retopologized using the “rebuild” command with a specified number of control vertices (for example, 20), and their seams moved to face the same direction using the “move seam” command. The curves are then selected and used to create a polygonal mesh using the “loft” command.

17.2 Modeling vesicles: In Maya, a sphere is generated using the “sphere” command and moved to the x y z coordinates specified by the data file. The radius of the sphere is also changed as specified.

The completed model can then be rendered as an image, or exported from Maya as a 3D geometry file (e.g. STL, obj, etc).