Presynaptic analysis example

April 26, 2020

This example shows an example of the complete presynaptic workflow.

1 Segmentation and analysis of individual datasets (synapses)

All relevant data is in examples/presynaptic_example/segmentation/cell_tomo-4/ .

1.1 Generating a boundary label image file

This step explains how a boundary label file can be generated. Here, synaptic vesicles are labeled in a semi-automated way, while other membranes, cellular structures and organelles are labeled manually.

1.1.1 Prerequisite

- tomogram (3d/tomo.mrc)

1.1.2 Steps - Direct method

- Segment the active zone membrane (viz/disks.raw, label id=2) based on the tomogram
- Segment presynaptic intracellular region (id=3) and cellular structures that should not be used for segmentation (microtubules, mitochondria, non-synaptic vesicles; ids = [35, 47, 72, 74, 75]) (viz/disks.raw)
- Segment equatorial circles (disks) of synaptic vesicles (ids = [36, 48, 49, 50, 52, 58, 59, 60, 61] (viz/disks.raw) visualized in the tomogram
- It is advisable that none of the labels generated in the previous steps is present on the faces
- Enter parameters in common/tomo_info.py (template pyto/scripts/tomo_info.py), the following label ids are set:
  - all_ids: all ids present in the labels file, if a label is present in the file, but is not specified in the list, it will be set to 0
  - boundary_ids: label ids for the AZ membrane and synaptic vesicles
  - vesicle ids: synaptic vesicle ids
  - segmentation_region: label id for the cytoplasm, this is the region where tethers and connectors are detected
  - distance_id: id of the AZ membrane
- Run viz/discs_to_balls.py script (template pyto/scripts/discs_to_balls.py) to generate a boundary label file (viz/labels.mrc) where equatorial discs are expanded to full balls

1.1.3 Steps - Iterative method (alternative)

- Segment the active zone membrane (viz/disks_1.raw, label id=2) in the tomogram
- Segment presynaptic intracellular region (id=3) and cellular structures that should not be used for segmentation (microtubules, mitochondria, non-synaptic vesicles; ids = [35, 47, 72, 74, 75] (viz/disks_1.raw)
- Segment equatorial circles (disks) of some synaptic vesicles (ids = [36, 48, 49, 50]) in the tomogram and save the segmentation as viz/disks_1.raw
- Enter parameters in viz/discs_to_balls_1.py script (template pyto/scripts/disks_to_balls.py) and run this script to generate viz/labels_1.raw where the labeled disks are expanded to full balls. Check the script to see how the parameters (especially label ids) are set.
- Enter parameters in viz/blank_1.py (template pyto/scripts/blank.py) and run this script to make a tomogram where the labeled synaptic vesicles are clearly marked (3d/tomo_1.mrc). This make easier the further segmentation of synaptic vesicle equatorial disks.
- Segment equatorial circles (disks) of (some) other synaptic vesicles (ids = [52, 58, 59, 60, 61]) based on the blanked tomogram (tomo_1.mrc). Depending on the segmentation program, labels_1.raw may need to be loaded. Save the segmentation as viz/disks_2.raw.
- Enter parameters in viz/discs_to_balls_2.py (template pyto/scripts/discs_to_balls.py) script and run this script to generate viz/labels-2.mrc where the newly labeled disks are expanded to full balls. Check the script to see how the parameters (especially label ids) are set.
- Continue with iterating blank and discs_to_balls until the final boundary labels file is generated, where all synaptic vesicles are fully labeled

1.1.4 Results

- Final boundary label image file (viz/labels.mrc, or viz/labels_n.mrc)

1.1.5 Notes

Other membrane segmentation tools and procedures can be used. However, the resulting boundary label file has to be consistent with the one provided here. The most important requirements are:

- Boundary label image has to contain integers
- Value 0 is reserved for the background
- Separate, specific values (ids) need to be used to label the following: the active zone region, presynaptic cytoplasm (segmentation region), synaptic vesicles (each vesicle has to have a unique id) and other organelles and cellular structures. These ids should not appear anywhere else in the label file.
- Labels should not touch tomogram sides

The most common reason for errors is the label values are not assigned properly.

1.2 Setting common parameters

If not done during the previous step, common parameters heve to be entered in common/tomo_info.py (template pyto/scripts/tomo_info.py)

1.3 Regions script

Basic greyscale statistics for the whole tomogram and for the labeled parts are calculated in this step.

1.3.1 Steps

- Run regions/regions.py script (template pyto/scripts/vesicles.py). Typically, no parameters need to be changed from the default values

1.3.2 Results

- The results are stored both in regions/tomo_regions.dat (plain text) and regions/tomo_regions.pkl (Python pickle file). The text file is used to quickly check the results. For more complicated analysis of the results, the pickle file can be loaded into a Python session.

1.4 Vesicles script

Basic greyscale, morphological and localization analysis of synaptic vesicles. Includes separate analyses for vesicle membrane, lumen and the whole vesicles.

1.4.1 Steps

- Enter parameters to regions/regions.py script (template pyto/scripts/vesicles.py). Typically, only membrane thickness (membrane_thick) need to be adjusted
- Execute the vesicles script

1.4.2 Results

Results are saved in the following files (all in regions/):

- tomo_vesicles.dat: All results in a plain text file
- tomo_vesicles.pkl, tomo_mem.pkl, tomo_lum.pkl: Pickle file results for whole vesicles, membranes and lumens, respectively. Used for the analysis of multiple datasets.

1.5 Layers script

Make layers inside the presynaptic terminals that start at and are parallel to the AZ membrane.

1.5.1 Steps

- Enter parameters to layers/layers.py script (template pyto/scripts/layers.py). Typically, no parameters need to be changed from the defaults
- Execute the layers script

1.5.2 Results

Results are saved in the following files (all in regions/):

- labels_layers.dat: All results in a plain text file, used also for the analysis of multiple datasets.
- layers.mrc: layers image

1.6 Connectors script

Detects, analyzes and classifies tethers that link synaptic vesicles to the active zone membrane and connectors that interlink synaptic vesicles.

1.6.1 Steps

- Enter parameters to connectors/connectors.py script (template pyto/scripts/connectors.py).
- Typically, the only parameter that need to be changed from the default value relates to the classification by volume (class_3_volumes). The exact values depend on the pixel size and on the expected volume
- Execute the script

1.6.2 Results

Results are saved in the following files (all in connectors/):

- tomo_thr-*.dat: Single threshold results in a plain text file
- tomo_thr-*.pkl: Single threshold results in a pickle file, used for the analysis of multiple datasets.
- tomo_thr-*.mrc: Single threshold segmentation image
- tomo.dat, tomo.pkl, tomo.mrc: Combined results at all thresholds before classification in a plain text file, pickle file and as an hierarchical segmentation image, respectively.
- tomo_new_[connectors, tethers]_[small, good, big].[dat, pkl, mrc]: Combined results at all thresholds after classification for all classes obtained as a combination of classification by contacted ids (tethers, connectors) and by volume (small, good, big). The extension determines whether the file is a plain text file, pickle file or segmentation image.

1.6.3 Results