BoutonAnalyzer, User Manual

BoutonAnalyzer is software for detection and tracking of structural changes in *en passant* boutons in time-lapse light-microscopy stacks of images. Technical details and validation of the algorithms implemented in this software can be found in [1].

1. System Requirements and Installation

BoutonAnalyzer is designed to run on Windows and Mac installations of MATLAB (versions 2015a to 2017a). The software can be downloaded at https://github.com/neurogeometry/BoutonAnalyzer. Start MATLAB, navigate to the software folder, and run BoutonAnalyzer in the MATLAB command window. Do not change MATLAB path while running the software.

2. Directory Structure, File Formats, and Naming Conventions

BoutonAnalyzer requires paths for loading image files and traces, and for saving axon intensity profiles and results. The *Optimize Trace & Generate Profile* GUI (Figure 4) uses an image stack and an axon trace as inputs and generates an output file containing optimized trace and axon intensity profile. The *Detect & Track Boutons* GUI (Figure 5) loads these data for an axon in multiple imaging sessions and allows the user to detect, edit, and match boutons across sessions. A strict file naming convention must be followed to allow *BoutonAnalyzer* to search and organize data. In the following, we describe the various allowed file types and naming conventions for images, traces, and profiles.

Images

Image stacks can be loaded in .mat, multi-layer .tif (or .tiff), and single-layer .tif (or .tiff) file formats (Figure 1). The .mat file is expected to contain grayscale intensity values of the image stack voxels in a 3d array named IM. Image file names must include animal identity (e.g. DL1), order within the time-lapse sequence (e.g. B), image stack name (e.g. S2), and channel (e.g. Gr). Each identifier must begin with an upper case letter and can be followed by an arbitrary number of alphanumeric characters or underscores. Two examples of image names consistent with this format are DL1BS2Gr.mat and DL_01BS02G.mat.

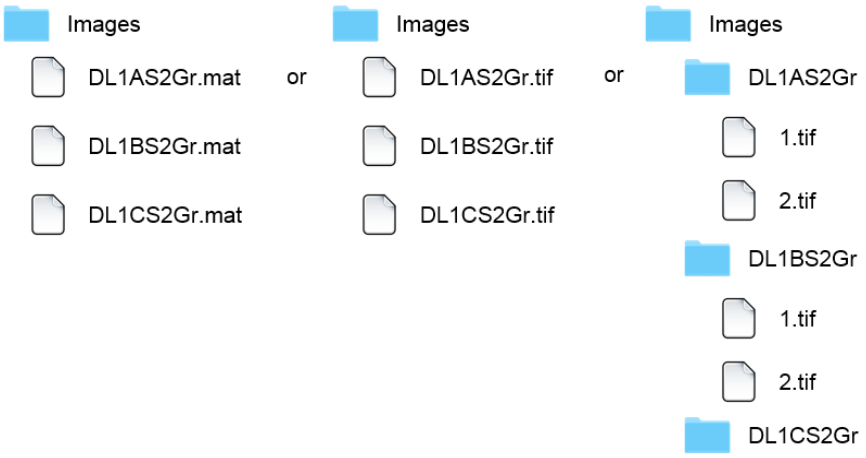

Figure 1: Compatible image stack file formats and naming conventions

Traces

Traces of axons are expected to be in the .swc file format with coordinates given in voxel units. NCTracer [2, 3] or other software can be used to create traces and export them as .swc files. Each .swc file should correspond to a unique axon segment without branch points. Axons traced in a single image stack must have names in the range A001.swc to A999.swc. Name of the parent folder should correspond to the name of the image stack without the channel identifier. An axon traced in multiple imaging sessions must have the same name in all folders corresponding to these imaging sessions. For example, axon traces A002.swc in folders DL1**A**S2 and DL1**B**S2 correspond to the same axon imaged in sessions **A** and **B** (Figure 2).

Figure 2: Naming convention for axon traces

Profiles

Axon intensity profiles are .mat files created by *BoutonAnalyzer*. The naming convention and directory structure of profile files are the same as those for traces.

Results

Once boutons have been detected and matched for a given axon across multiple imaging sessions, files containing bouton weights (*w*) and bouton probabilities (*p*) are created in both .txt and .mat formats. The .txt file can be opened with a text editor, and the values can be imported in Excel for the analysis of structural plasticity. The .mat file contains additional information related to flags assigned during matching (*flag*), 3D position of boutons in the stack (*rx*, *ry*, *rz*), bouton positions along the axon (*d*), and other variables that can be used for a more detailed analysis.

3. Bouton Analysis Workflow

BoutonAnalyzer can be used to perform the following operations:

1. Trace optimization
2. Generation of axon intensity profiles
3. Registration of traces across multiple imaging sessions
4. Annotation of traces
5. Detection, editing, and matching boutons in multiple imaging sessions

Steps 1-2 are performed in the *Optimize Trace & Generate Profile* GUI, and steps 3-5 are completed in the *Detect & Track Boutons* GUI. Running BoutonAnalyzer in the MATLAB command window opens the main *BoutonAnalyzer* window, which can be used to set paths and launch one of the two GUIs (Figure 3).

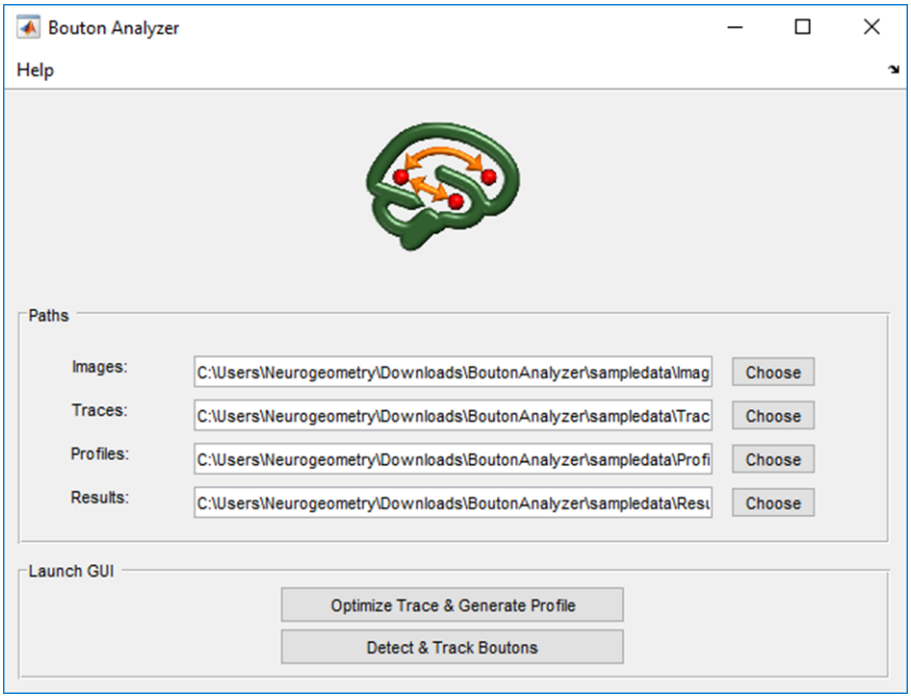

Figure 3: *BoutonAnalyzer* main window.

4. Optimize Trace & Generate Profile GUI

Pressing the Optimize Trace & Generate Profile button opens a window to specify the animal, imaging session, image stack, channel, and axon identifiers. A single axon trace and the corresponding image stack (up to 3 stacks if data from more than one channel is available) can be loaded (Figure 4).

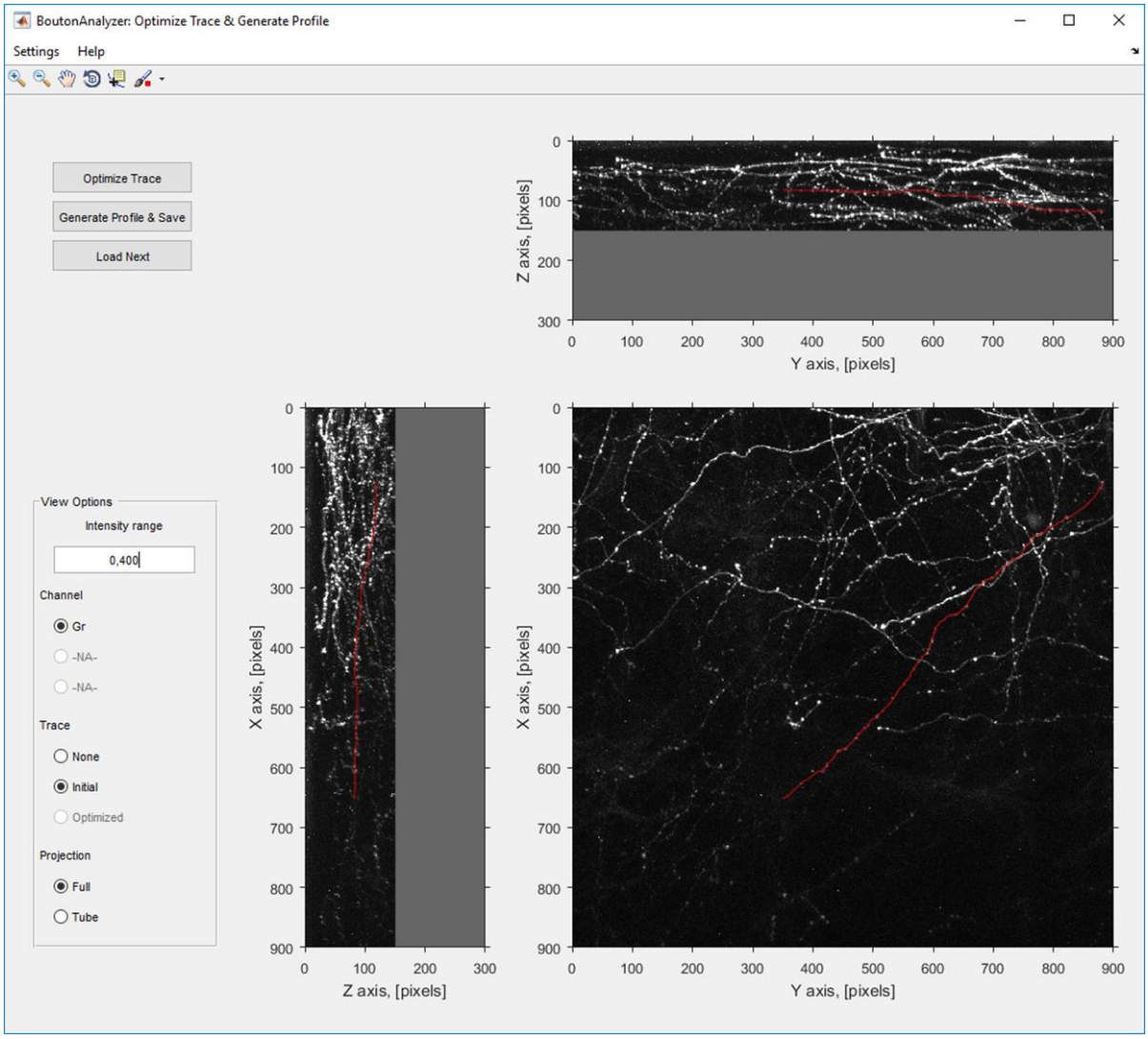

Figure 4: Optimize Trace & Generate Profile GUI.

Navigation

Default zoom and pan figure tools of MATLAB can be used to independently focus on different regions of interest on the three projections. **Note that all shortcut keys are disabled if any MATLAB figure tool is active.** Alternatively, mouse scroll and arrow keys can also be used to perform zoom and pan operations (see Table 1 for the complete list of shortcuts). The View Options panel contains additional display options. Full and Tube Projections show the full stack projection or the background-removed

projection of the axon. Brightness can be adjusted by changing values in the Intensity Range field and pressing the return key, or by using the "–" and "=" shortcut keys. The Channel option will only be activated if multiple channels are loaded.

Trace optimization

The Optimize button initiates trace optimization. Once trace optimization is completed, the radio button related to the optimized trace view will be enabled. Toggling between the Initial and Optimized trace view options will display the corresponding trace on all projections. The Settings menu contains a link to the parameter file, which can be edited to change the parameters of all *BoutonAnalyzer* functions.

Generation of axon intensity profiles

After trace optimization, intensity profiles can be generated and saved for all loaded channels by pressing the Generate Profile & Save button.

5. Detect & Track Boutons GUI

Pressing the Detect & Track Boutons button in the main *BoutonAnalyzer* window opens the load data panel for the selection of axon profiles in one or more imaging sessions. Axon traces will be displayed superimposed on the maximum intensity projection of the image stack (Figure 5).

Navigation

The zoom, pan, and rotate figure tools of MATLAB perform the expected operations. Mouse scroll and arrow keys can also be used to zoom and pan respectively. **Note that all shortcut keys are disabled if any MATLAB figure tool is active.** The Intensity Range field allows one to adjust the range of intensities displayed on the screen. Note that this operation only affects the display and not the data used to perform calculations. The Shift field makes it possible to change the relative positions of axons in the image. Axons can be viewed on raw and normalized intensity projections.

Registration of traces across multiple imaging sessions

Multiple traces can be registered by identifying corresponding vertices in all traces. The currently selected vertex will be indicated by a bright green circle. Dark green circles will appear on the remaining traces as suggested corresponding vertices. Positions of the selected vertices can be adjusted by using ","and "." keys. Upon satisfactory alignment of the selected vertices, traces can be registered by right-clicking on one of the selected vertices and choosing the Add Landmark option in the context menu.

Placing Landmarks every ~15µm along the traces should be sufficient for the subsequent automated matching of boutons.

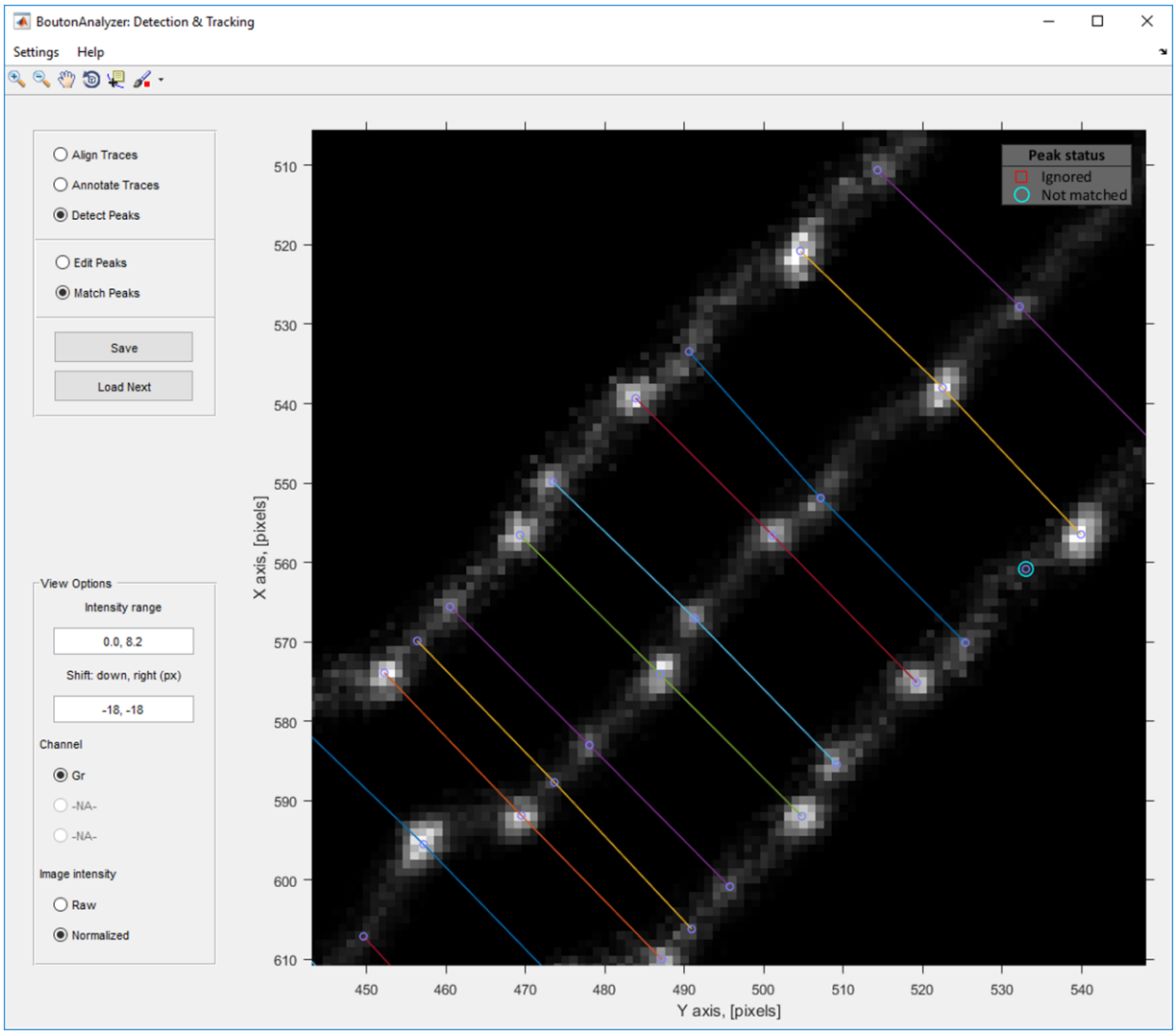

Figure 5: Detect & Track Boutons GUI.

Annotation of traces

Cross-over regions, terminal boutons, and locally noisy regions along an axon must be excluded from the analysis. Such regions can affect the normalization and, hence, detection and measurement of boutons. Such a region can be demarcated by placing a pair of vertices on one of the traces. Positions of these vertices can be adjusted by using “,”and “.” keys. Right-clicking on a selected vertex will open a context menu with options to ignore or add the enclosed region in one or all imaging sessions.

Detection, editing, and matching boutons in multiple imaging sessions

Boutons can be automatically detected as peaks along the intensity profiles by pressing the Detect Peaks button. Once bouton detection is completed, the Edit Peaks and Match Peaks modes will be enabled. In the Edit Peaks mode, positions of the detected peaks are indicated by triangles. Individual trace locations can be examined on the intensity profile by choosing the Show on Profile context menu option. A peak can be added or removed by selecting a vertex and using the appropriate context menu option. The detected peaks are matched automatically by a greedy nearest-neighbor algorithm using the landmarks provided in the trace alignment step. These matches can be reviewed in the Match Peaks mode. Matches can be broken by clicking on lines connecting pairs of peaks. New matches can be created by selecting multiple peaks, followed by choosing the Add Peaks context menu option. Upon saving changes, results are compiled in .txt and .mat files located in the \Results folder.

6. Shortcuts

Key	Optimize Trace & Generate Profile GUI	Detect & Track Boutons GUI
=	Increase brightness	Increase brightness
-	Decrease brightness	Decrease brightness
Mouse scroll	Zoom in/out	Zoom in/out
Up arrow	Pan up	Pan up
Down arrow	Pan down	Pan down
Left arrow	Pan left	Pan left
Right arrow	Pan right	Pan right
d		De-select all vertices
Comma (,) & period (.)		Move selected vertex along the trace
v		View/hide overlay elements
a		Add landmark, peak, or matches
r		Remove peak

7. List of Steps

1. Launch *BoutonAnalyzer* from MATLAB command window.
2. Set paths for Images, Traces, Profiles, and Results.
3. Choose the Optimize Trace & Generate Profile GUI.
4. Load a single trace and image stack(s).
5. Perform optimization, inspect the optimized trace, generate and save the profile.
6. Repeat step 5 for traces of the same axon in all available imaging sessions.
7. Launch *BoutonAnalyzer* again and choose the Detect & Track Boutons GUI.

8. Specify imaging sessions to simultaneously load profiles of an axon in multiple sessions. Upon loading the data, traces will be shown overlaid on the background-removed *xy* projections of the axon in different imaging sessions.
9. Add a landmark to align the traces. To do so, click on any vertex of one of the axon traces. A bright green circle will mark the currently selected location. Dark green circles will indicate guesses for the corresponding locations on the remaining traces.
10. Select and adjust the positions of these markers by using "," and "." keys until the correspondence is precise.
11. Right-click on one of the markers and select Add Landmark. A line connecting these positions will appear. Assigned landmark can be removed by right-clicking on one of the lines and selecting the Remove Landmark option.
12. Once at least two landmarks are added, the Annotate Traces mode will be enabled.
13. If there are regions on the axon that must be ignored from further analyses switch to the Annotate Traces mode. Select two vertices on a single trace that demarcate the region to be ignored. Right-click on one of the vertices to open the context menu. Select an option to either ignore the region in the current trace only or in all the loaded traces. Ignored regions can be readmitted by choosing the Add context menu options.
14. Select the Detect Peaks mode to fit the intensity profiles and detect boutons. The Edit Peaks and Match Peaks modes will be enabled after peak detection is completed.
15. In the Edit Peaks mode, detected peaks appear as triangles on the traces. Right-click on a trace vertex or a peak to open the context menu. Select the appropriate option to add new peaks or remove existing ones.
16. Inspect locations along the intensity profile using the Show on Profile option in the context menu.
17. The Match Peaks mode shows boutons that were matched based on the previously provided alignment.
18. Select multiple peaks on different traces. Right-click on a selected peak to match or flag the selected peaks. Click on lines that connect peaks to delete the matches.
19. Save the results.

8. Sample Data

Sample image stacks and axon traces can be found at http://www.northeastern.edu/neurogeometry/resources/bouton-analyzer/. Download SampleData.zip file and extract its contents. SampleData\Images folder contains three image stacks acquired under different microscopy conditions within a short period of time (< 1 hour). Details regarding the dataset can be found in [1]. Animal identifier is DL1, section identifier is S2, and channel identifier is Gr. The three imaging sessions are identified by letters A, B, and C. Three axons (A001, A002 and A003), manually traced in all three imaging sessions, are provided in the SampleData\Traces folder.

9. About

Contact information

BoutonAnalyzer is developed by the Neurogeometry group at the Department of Physics and the Center for Interdisciplinary Research on Complex Systems at Northeastern University, Boston. All inquiries should be addressed to Rohan Gala, rhngla@gmail.com, and Prof. Armen Stepanyants, a.stepanyants@neu.edu

Related publications

[1] Gala R., Lebrecht D., Sahlender D., Jorstad A., Knott G., Holtmaat A. and Stepanyants A., Computer assisted detection of axonal bouton structural plasticity in in vivo time-lapse images (submitted)

[2] Chothani P., Mehta V., and Stepanyants A., Automated tracing of neurites from light microscopy stacks of images, Neuroinformatics, 9(2-3): 263–278 (2011)

[3] Gala R., Chapeton J., Jitesh J., Bhavsar C., Stepanyants A., Active learning of neuron morphology for accurate automated tracing of neurites, Frontiers in Neuroanatomy, 8:37 (2014)

Acknowledgments

We would like to thank Seyed Mousavi Kahaki and Shih-Luen Wang for testing *BoutonAnalyzer*. This software is supported by the NIH grant R01 NS091421.