

User Guide for NeuronID

Jikan Peng (pengjikan@westlake.edu.cn, Westlake University, China)

Running Environment: The initial version of NeuronID was developed using MATLAB 2023b (MathWorks, USA). To use this software, you can enter 'CSE' in the MATLAB command window, or directly launch the application by running CSE.mlapp. If you need to inspect or modify the code related to this software, you can open the CSE.mlapp and proceed with your changes.

Data Preparation: We strongly encourage you to create a standardized folder for each mouse, similar to the 'Mouse#1' example provided in this software. Within this folder, create a subfolder named 'Raw' to store all initial two-photon imaging datasets. Additionally, ensure that you have the basic information about your two-photon imaging data, which includes the pixel size of each frame, the physical size (in micrometers) of each frame, and the imaging frequency.

Operation Model Selections: We have provided several options for your data analysis.

To analyze the two-photon imaging dataset of a specific mouse, click the 'Open Mouse File' button to select the folder containing the mouse's data. Following this, input the necessary parameters, including the scan scale ratio—defined as the ratio between the pixel size and the physical size—and the duration (in milliseconds) required to image one frame. Additionally, you can click the 'Calculate' button to compute the Decay Index. After completing these steps, you can click the 'Mouse' button to perform a systematic and automatic analysis of your data, or you can follow the steps behind the 'Mouse' button to manually analyze your data. You can open the CSE.mlapp to view these steps.

To analyze the two-photon imaging datasets of multiple mice, first-firstly you should open the CSE.mlapp and modify the 'Path="Input your file pathway"' code behind the 'Mice' button. Ensuring that all mouse folders are arranged in the same directory and that their information is listed in a 'mouse.xlsx' file, for which we have provided a template in the 'Mice' folder. After completing these steps, you can click the 'Mice' button to perform a systematic and automatic analysis of your data.

Motion Correction: You can click the 'Motion Correction' button to correct motion artifacts within two-photon images. Alternatively, you can open this button to inspect the code and algorithms. The code associated with this feature is primarily based on the NoRMCorre algorithm ([GitHub - flatironinstitute/NoRMCorre: Matlab routines for online non-rigid motion correction of calcium imaging data](https://github.com/flatironinstitute/NoRMCorre)), with corresponding modifications described in our paper.

Note that this step requires sufficient internal storage to operate, so please ensure that you have adequate space before proceeding. The outcome of this step, or the two-photon images after motion correction, will be stored in a 'Registration' folder.

Noise Reduction: After motion correction, you can reduce independent noise within individual

frames using the DeepInterpolation algorithm ([DeepInterpolation-MATLAB/README.md at main · MATLAB-Community-Toolboxes-at-INCF/DeepInterpolation-MATLAB · GitHub](#)). The ‘**Normalize Parameters**’ and ‘**Extract Key Frames**’ buttons prepare materials for training your personalized DeepInterpolation models. The ‘**Train Personal Model**’ button allows you to customize the model through transferring learning from an existing model trained under 30Hz imaging conditions. You can adjust the number of epochs for training by modifying the ‘Epoch’ parameter input. After completing the training, you can click the ‘**Evaluate Personal Model**’ button to see the trend of loss over different epochs. Subsequently, you can click the ‘**Denoise Images**’ button to reduce noise in your two-photon images. You can also assess the effect of noise reduction by clicking the ‘**SNR after Denoise**’ and ‘**Evaluate Denoise (Reconstruction Loss)**’ buttons, the methods for which are detailed in our paper.

In the current version, we provide multiple fine-tuned modes for distinct imaging frequency conditions, listed in the ‘**AIModel**’ folder. If an existing model is available, NeuronID will automatically use this model for noise reduction, rather than generating a new one. The outcome of this step, including the two-photon images after noise reduction, will be stored in a ‘**Denoise**’ folder.

Neuronal Component Segmentation: You can use the functions behind the ‘**Periodic Mask**’, ‘**Temporal Mask**’, and ‘**Demixing Masks**’ modules to conduct the segmentation of neuronal components. The mathematical foundations and basis for these modules were introduced in our paper. Note that this step requires sufficient internal storage to operate, so please ensure that you have adequate space before proceeding. The outcome of this step, including the distinct masks, will be stored in a ‘**Segment**’ folder.

Particularly, you can click the ‘**Initialization**’ button to prepare for the following steps. Click the ‘**Periodic Masks**’ button to divide the whole recording sequence into different temporal blocks and generate the corresponding periodic masks. After that, click the ‘**Potential Location**’ button to identify the pixels associated with neuropil. Subsequently, click the ‘**t-SNE Projection**’ and ‘**Central Pixel**’ buttons to visualize the similarity among pixels. Then, click the ‘**Temporal Mask**’ button to analyze the relationship between the core pixels and other pixels, generating a temporal mask. Finally, click the ‘**Mask Overlap**’ and ‘**Combine Masks**’ buttons to calculate and blend the overlapping masks within the temporal mask, and click the ‘**Spatial Mask**’ button to develop a spatial mask.

Extraction of Neuronal Signal: You can extract and analyze the signal of individual neurons within the spatial mask using the ‘**Extract Neuronal Signal**’ module. The outcome of this step, including individual neuronal signals, will be stored in a ‘**Signal**’ folder. Specifically, the ‘**Neuronal Fluorescence**’ and ‘**Neuropil Fluorescence**’ buttons allow the calculation of the fluorescence signal of individual neurons and their associated neuropil. The ‘**Correct Fluorescence**’ button enables you to remove neuropil contamination from the neuronal signal and eliminate any polynomial trends within the data. The ‘**Calcium Signals**’ button allows you to normalize the signal and generate the calcium signal. The ‘**Event Signals**’ button provides a method to calculate the event signal of individual neurons, based on the Decay Index.

Other Functions: We have provided additional functions for analyzing your two-photon

imaging data. You can export the time labels for individual frames using the ‘**View Frames**’ and ‘**Export Timeline**’ buttons. Additionally, you can click the ‘**Evaluate Mask**’ button to compare different masks.

Authority and Ownership: The copyright of these codes is owned by Jikan Peng at Westlake University. Use of these codes is permitted for scientific research purposes; however, any commercial use is strictly prohibited.