User Guide for NeuronID

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

1. Overview

NeuronID is an open-source, automated toolkit for processing two-photon calcium imaging data, featuring motion correction, noise reduction, neuronal component segmentation, and signal extraction. This guide provides instructions for installing and running the software.

2. System Requirements & Installation

- **Required Software**: MATLAB R2023b or later (MathWorks, USA).
- **Installation**: No formal installation is required. Simply add the NeuronID directory and its subfolders to your MATLAB path.
- **Launching the Application**:
    - **GUI Mode (Recommended)**: Type CSE in the MATLAB command window, or directly run CSE.mlapp.
    - **Code Access**: To inspect or modify the source code, open CSE.mlapp in MATLAB's App Designer.

3. Data Preparation

We recommend organizing your data as follows for optimal compatibility:

1. Create a main folder for each subject (e.g., Mouse#1).
2. Inside this folder, create a subfolder named Raw to store all original two-photon imaging files (e.g., .tif stacks).
3. Ensure you have the following metadata for your imaging data:
    - **Pixel Dimensions** (width x height in pixels)
    - **Field of View** (physical size in micrometers, e.g., 512 µm x 512 µm)
    - **Imaging Frequency** (frame rate in Hz)

4. Operation Modes

NeuronID offers two primary analysis modes:

- **A. Single-Subject Analysis**
    1. Click Open Mouse File and select the subject's main folder (e.g., Mouse#1).
    2. Input the required parameters:
        - **Scan Scale Ratio**: (Physical Size per Pixel) = Field of View (µm) / Pixel Dimension.
        - **One F Duration (ms)**: Time to acquire one frame = 1000 / Imaging Frequency (Hz).
    3. Click Calculate to compute the calcium indicator's decay constant.
    4. Click One-Click Run (Mouse) for a fully automated analysis pipeline.
- **B. Batch Analysis (Multiple Subjects)**

1. Ensure all subject folders are within a single parent directory.
2. Prepare a mouse.xlsx file listing the information for all subjects (a template is provided in the Mice folder).
3. In the CSE.mlapp source code, modify the Path variable for the One-Click Run (Mice) button to point to your parent directory.
4. Click One-Click Run (Mice) to process all datasets automatically.

5. Step-by-Step Module Description

For users who wish to execute and validate each step manually, the workflow is broken down into the following modules:

- **5.1. Motion Correction**
  - **Function**: Corrects for rigid and non-rigid motion artifacts.
  - **Action**: Click the Motion Correction button.
  - **Algorithm**: Based on the NoRMCorre algorithm (https://github.com/flatironinstitute/NoRMCorre) with custom modifications.
  - **Output**: Corrected images are saved in the Registration folder.
  - **Note**: This step requires significant internal storage. Please ensure adequate disk space.
- **5.2. Noise Reduction**
  - **Function**: Reduces independent noise using a deep learning approach.
  - **Algorithm**: Based on the DeepInterpolation algorithm (https://github.com/AllenInstitute/deepinterpolation； https://github.com/MATLAB-Community-Toolboxes-at-INCF/DeepInterpolation-MATLAB).
  - **Workflow**:
    1. Normalize Parameters & Extract Key Frames: Prepares data for training.
    2. Train Personal Model: Performs transfer learning from a pre-trained model (default: 30 Hz). Adjust the Number of Epoch as needed.
    3. Evaluate Personal Model: Visualizes the training loss curve.
    4. Denoise Images: Applies the model to reduce noise.
  - **Quality Control**: Use SNR after Denoise and Evaluate Denoise to assess performance.
  - **Pre-trained Models**: Multiple models for different imaging frequencies are available in the AIModel folder. NeuronID will auto-select a compatible model if available.
  - **Output**: Denoised images are saved in the Denoise folder.
- **5.3. Segmentation of Neuronal Components**
  - **Function**: Identifies and segments individual neuronal somata.
  - **Workflow**:
    1. Initialization: Prepares the data for segmentation.
    2. Periodic Masks: Generates preliminary masks from temporal blocks (Default: Window Size=100, Step Size=20).

    3. Potential Location: Identifies and excludes neuropil-associated pixels.
    4. t-SNE Projection & Central Pixel: Clusters pixels by similarity and identifies key pixels (Default: Number of Center Pixels=9,000).
    5. Temporal Mask: Integrates information across time to generate a temporal mask.
    6. Mask Overlap & Combine Masks: Resolves overlapping regions.
    7. Spatial Mask: Produces the final somatic mask.
  - **Output**: All masks and intermediate results are saved in the Segment folder.
  - **Note**: This step is computationally intensive and requires sufficient internal storage.
- **5.4. Extraction of Neuronal Signal**
  - **Function**: Extracts and processes fluorescence signals from segmented neurons.
  - **Workflow**:
    1. Neuronal Fluorescence & Neuropil Fluorescence: Calculates raw signals.
    2. Correct Fluorescence: Removes neuropil contamination and corrects for photobleaching.
    3. Calcium Signals: Calculates ΔF/F to generate the calcium signal.
    4. Event Signals: Performs deconvolution to infer spiking activity, using the provided Decay Index.
  - **Output**: Extracted signals are saved in the Signal folder.

6. Utilities

- **Timeline Export**: Use View Frames and Export Timeline to export temporal metadata for each frame.
- **Mask Evaluation**: The Evaluate Mask button allows for quantitative comparison between different segmentation masks.

7. License, Copyright, and Citation

- **Copyright**: The overall architecture, integration code, and original modules of NeuronID are Copyright (c) Jikan Peng, Westlake University.
- **Third-Party Code**: This toolkit incorporates and modifies open-source components, including but not limited to:
  - **NoRMCorre**: Copyright (c) their respective authors. Source: https://github.com/flatironinstitute/NoRMCorre.
  - **DeepInterpolation**: Copyright (c) their respective authors. Source: (https://github.com/AllenInstitute/deepinterpolation； https://github.com/MATLAB-Community-Toolboxes-at-INCF/DeepInterpolation-MATLAB).
- **Use**: Use of NeuronID is permitted for non-commercial academic and research purposes only. For any commercial use, please contact the corresponding author.

- **Citation**: If you use NeuronID in your research, please cite our publication.