NeuroART: Real-time analysis of neuronal population activity during calcium imaging for informed closed-loop experiments

System Requirements:

- Install Anaconda 4.6 or greater (if you are planning to use the Cite-on module for cell detection)
- MATLAB 2019b or greater
- MATLAB toolboxes: Image Processing Toolbox, Statistics and Machine Learning Toolbox
- (Optional) NVIDIA GPU with CUDA 10.2

Instructions for installing the cite-on module

Install all required dependencies by creating a new environment with the provided "env.yml"

- Open anaconda prompt from the folder "Cite-on_installation"
- Run the command: ***conda env create -f env.yml***

Then, build cython dependencies: ***python setup.py build_ext –inplace***

After completing the installation, make sure to add the following to system path variables
(control panel → system and security → system → advanced system settings → environment variables)

e.g. C:\Users\dulara\anaconda3\Library\bin\conda.bat
C:\Users\dulara\anaconda3\Scripts\conda.exe
C:\Users\dulara\anaconda3\

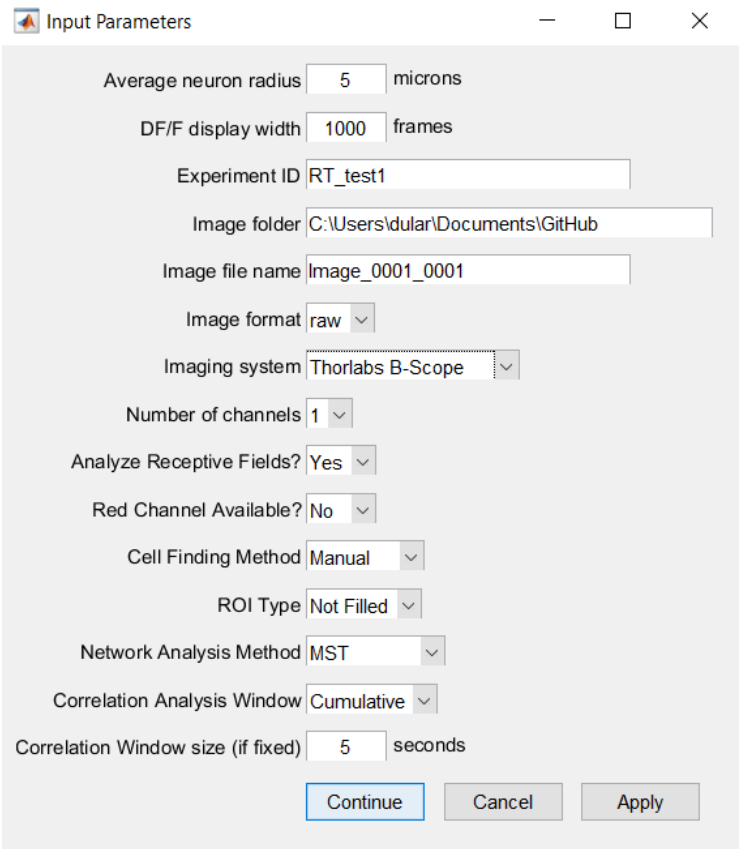
***Replace the anaconda installation path in the example above*

Step by step instructions to run the MATLAB application.

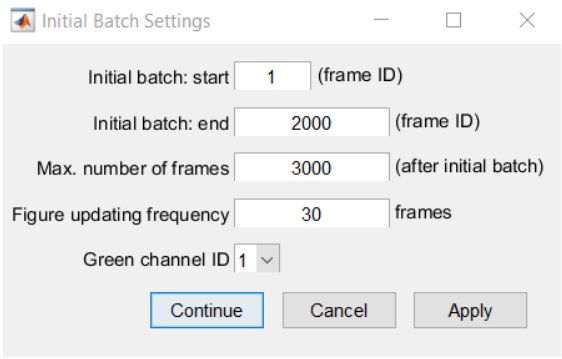
1) Download the folder containing all the MATLAB scripts/functions and open that folder in MATALB.
2) Run the main script, "main_complete_RealTime.m".
3) The first input dialog will appear on the screen as shown below. Provide all the required input parameters:
   Average neuron radius in microns
   DF/F display width: The number of frames of which you would like to observe the ΔF/F traces at a given time. (i.e., The length of the ΔF/F traces to be displayed in the app window). *Please make sure that this number does not exceed the number of frames you are planning collect as the initial batch for identifying cell locations.*
   Experiment ID: Output variables will be saved as a .mat file under this name.
   Image folder: The folder where the image frames will be saved in real-time.
   Image file name: The name of the image file (This can be either a RAW file or a TIF stack). It is not necessary to have this file while providing input parameters. The app will wait until this file appears in the folder specified as the "Image folder".
   Image format: RAW/TIF
   Imaging system: The name of the imaging system. If all the images are already available and you only need to analyze data, please select "Offline Mode".

Number of channels: For single channel data, select 1. Otherwise, select the number of channels available in the acquired images.
Analyzing Receptive Fields? Select "Yes" if you are planning to perform tuning analysis
Red Channel Available? Select "Yes" if you are using red channel (static channel) for cell detection
Cell Finding Method: Four options available for cell finding: Manual, CaImAn, Cite-On, and from file (previously saved coordinates)
ROI Type: By default, this app performs calcium image analysis on donut shaped (not filled) regions of interest (ROIs) around each of the identified cells (neurons). If your cells have filled ROIs, select "Filled"
Network Analysis Method: During the real-time analysis loop, the app will rank the cells according to their degree in the functional network, which is derived based on the correlation between each pair of ΔF/F traces. Currently, we derive this network based on two algorithms, minimum spanning tree method ("MST") and the correlation threshold method ("Correlations"). Select the method you prefer.
Correlation Analysis Window: By default, correlations are calculated for a cumulative window (consider all the frames acquired until the latest frame). However, if you want to only consider short term correlations between the ΔF/F traces, select "Fixed". Otherwise, select "Cumulative" to continue the analysis on a cumulative window.
Correlation Window Size: If you selected "Fixed" above, please specify the preferred window size (in seconds).

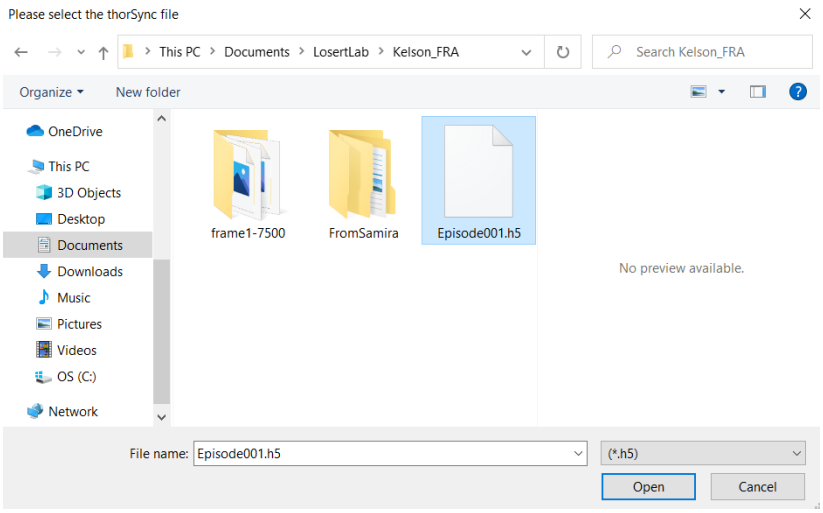
After inputting all the parameters, click "Continue".

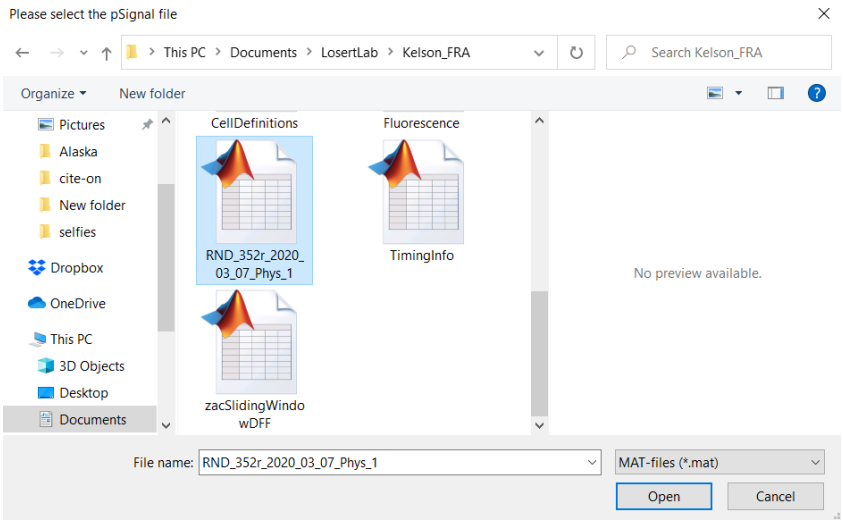


4) Next, specify the initial batch of frames (IDs of starting and ending frames) for cell identification. Make sure that the **length of the initial batch** is greater than or equal to the **DF/F display width**. Also, provide the maximum number of frames that will be collected during the experiment. This will be used as an upper bound for the total number of frames. Therefore, just provide an approximate number for this field.
Figure updating frequency: How frequently the figures/plots need to be updated in the GUI. (e.g., every 30 frames)
Green channel ID: For single channel data this field will be set to 1. For multi-channel data, specify which channel is the green channel. (e.g., Consider a two-channel dataset where the green channel is the second one. Then, green channel ID = 2)



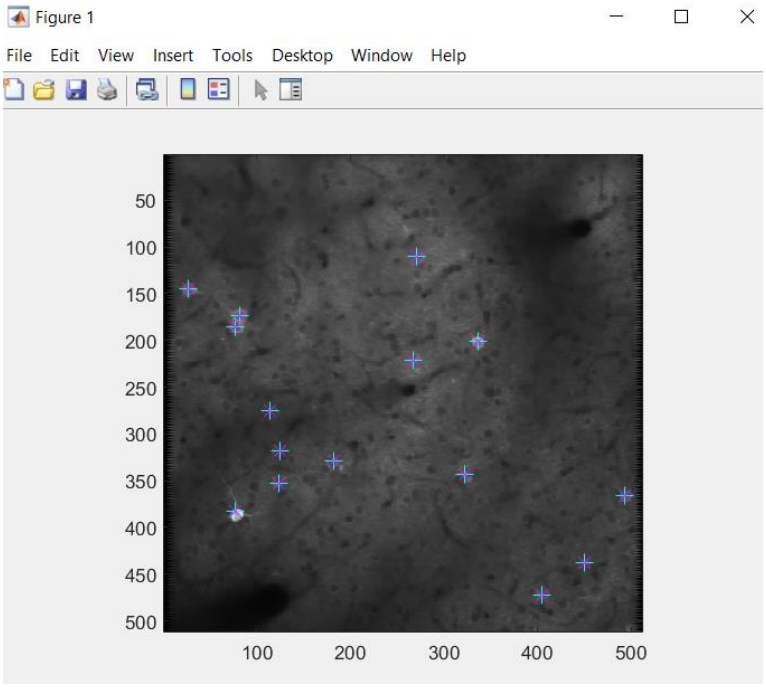
5) If you selected "Yes" for analyzing receptive fields, you will be prompted to locate the "thorSync" file and the "pSignal" file as shown below. (Only for Thorlabs B-Scope)





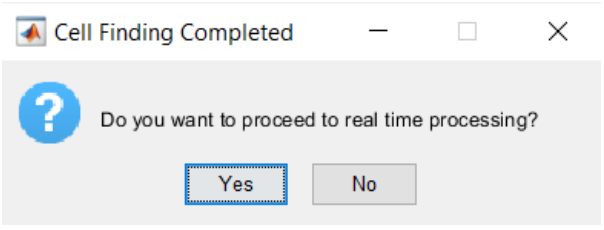
6) Make sure that the image acquisition has started. Wait until the app recognizes the image files and the initial batch of frames. If you opted for using the red (static) channel for cell identification, you will be prompted to provide the directory path and the file identifier for red channel data.

7) Cell identification will be performed using the method you selected at step 3.
   - **CaImAn:** If you selected this option, the app will use the cell identification routine of the "CaImAn" software package to find the coordinates of the cell centers. (Based on the mean image of the initial batch of frames)
   - **From File:** If you already have the coordinates of all the cells in the field of view (FOV), select this option and select the .mat file which contains these coordinates. This .mat file must have a variable named "ptsIdx" which has three columns: ID, y-coordinates, and x-coordinates of the cells, respectively. This option is useful when you are running multiple image acquisitions on the same FOV where it is not necessary to repeat the cell finding procedure.
   - **Manual:** Manually select the cells which are visible in the mean image. Press the "Enter" key once cell clicking is complete. Press the "delete" key if you want to undo the previous click.

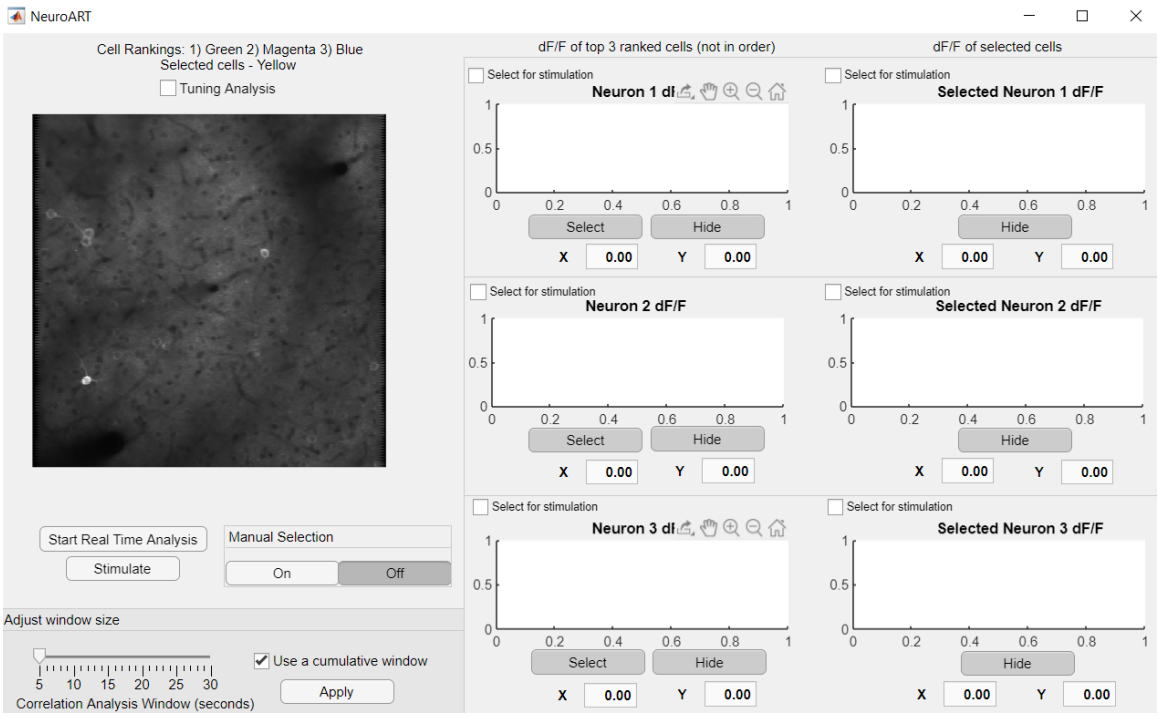


- **Cite-on:** Uses a pretrained Convolutional Neural Network (CNN) for cell identification. Before using this python library for cell identification, you must install the cite-on python library and setup an anaconda environment with all the required modules.

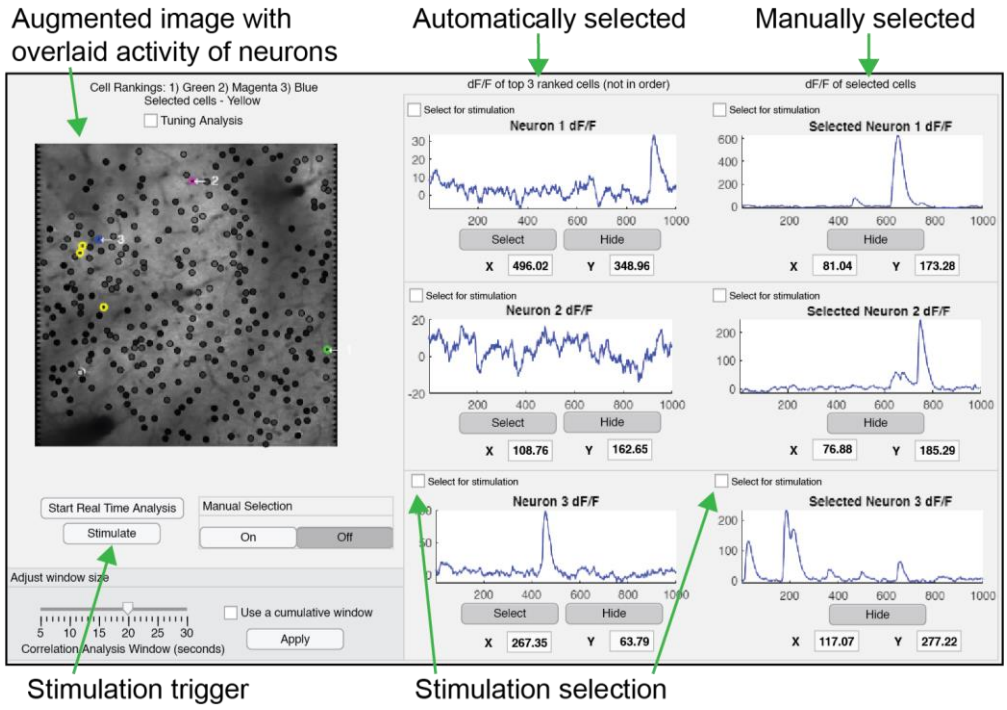
8) Wait until all the frames in the initial batch are motion corrected, denoised and ΔF/F traces are calculated for all the identified cells. After completion of these steps, you will be prompted to the dialog box shown below. If you want to continue analyzing the next incoming frames in real-time, click "Yes". This is the beginning of the real-time loop of the NeuroART pipeline.



9) After completing all the steps, the following window will appear. Select, “Start Real Time Analysis” to continue with the real-time analysis!



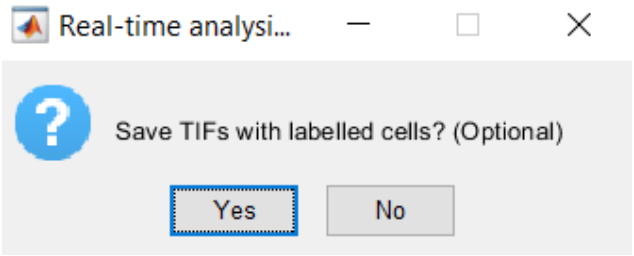
10) The figure below indicates different features of the NeuroART GUI, which can be useful during the real-time analysis.



Use the slider at the bottom left corner to adjust the correlation analysis window size while the images are being acquired.

Tuning analysis module can be turned on/off using the checkbox in the GUI.

11) After the completion of an experiment, all the frames can be saved with labels in the TIF file format. (This is optional)



12) After closing the app window, all the output variables can be found in a .mat file under the name provided as “Experiment ID”.

e.g., **DFFsmooth**: ΔF/F traces of all the identified cells
F: Fluorescence intensity of all the identified cells
cell_centroids: x and y coordinates of cells
cellRanks: Top 5 ranked cells recorded whenever the GUI is updated (e.g., every 30 frames)