FARCI: User Manual

Version 1.0.0 (April 13, 2020)

Authors: Saber Meamardoost and Rudiyanto Gunawan

Department of Chemical and Biological Engineering

University at Buffalo - SUNY

Contact email: sabermea@buffalo.edu and rgunawan@buffalo.edu

Table of contents

1 Overview

The user manual is written for the MATLAB distribution of FARCI (Fast and Robust Connectome Inference).

FARCI provides an efficient implementation of neuronal connectome inference from Calcium imaging neuronal activity data. FARCI outputs a partial correlation matrix, which is depicted as a circular graph, representing the neuronal connectome. When neuronal spatial information is available, the neuronal connectome also accounts for the position of neurons.

2 System requirements

This distribution of FARCI has been developed and tested in on MATLAB 2017b and 2018b.

3 FARCI package

FARCI package contains the following files and folders:

1. `FARCI_USER_MANUAL.doc` (this file)
2. `License.txt` modified BSD license for FARCI
3. `MAIN.m` FARCI main script, which can be adapted to run FARCI on a specific dataset
4. The folder `subfunctions` containing the following key subroutines and other subroutines:
    a. The folder `deconvolution` containing Suite2P and OASIS packages required for deconvolution of calcium imaging data
    b. `sp_deconv_inputs.m` : takes imaging and fluorescence decay rates as inputs
    c. `import_data.m` : imports calcium imaging data
    d. `sp_deconv.m` : for deconvolving the raw calcium imaging data and infer spikes
    e. `sp_denoise.m` : for denoising inferred spikes
    f. `sp_smooth.m` : smoothens the spikes
    g. `partial_corr.m` : for computing partial correlation statistics
    h. `connectome_inf.m` : infers network by thresholding partial correlation coefficients
    i. `circ_graph.m` : plots circular of the inferred neuronal connectome
    j. `graph_w_positions.m` : plots the inferred neuronal connectome based on actual spatial arrangement of neurons (if distance/position information is available)

5. The folder `examples` containing calcium imaging datasets along with distance/position data.

4 Examples

In the following, we describe the main steps of FARCI applied to publicly available calcium imaging datasets obtained from Neural Connectomics Challenge[1]. For each dataset, ONLY the most important results are reported. Please refer to the code `MAIN.m`for an example MATLAB script running FARCI analysis.

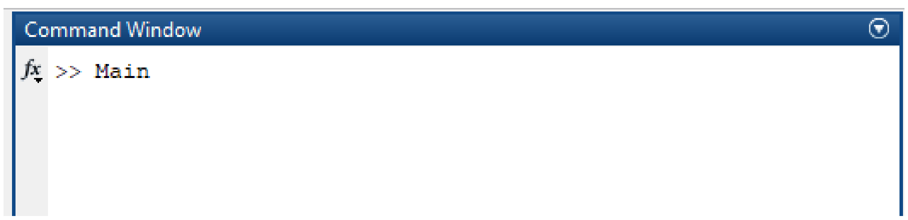
4.1 Example 1. Single dataset (small1)

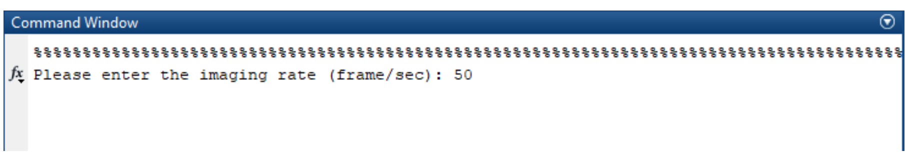
Here, we analyze a single dataset case using "small1" dataset provided in the Neural Connectomics Challenge that contains the calcium imaging data for 100 neurons and their positions in 2D.

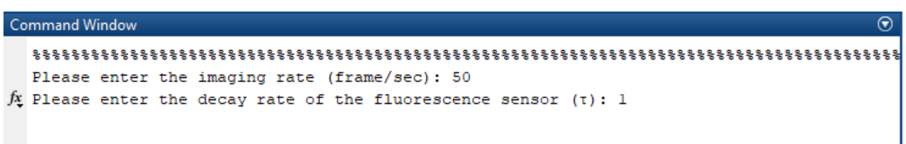
4.1.1 Deconvolution Parameters

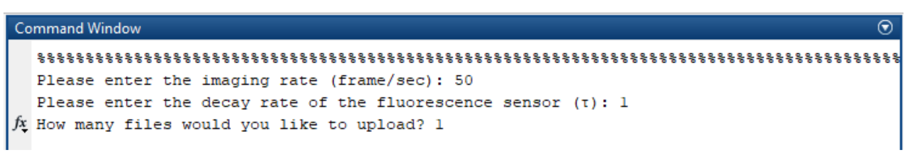
From FARCI folder, we run the `Main.m` script file. First, the user will be asked to enter deconvolution parameters, specifically imaging rate and fluorescence decay rate.

The following are screenshots from running FARCI in MATLAB.





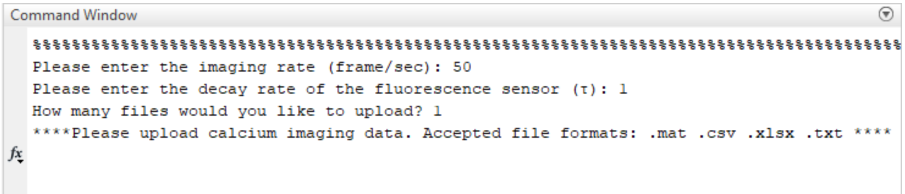


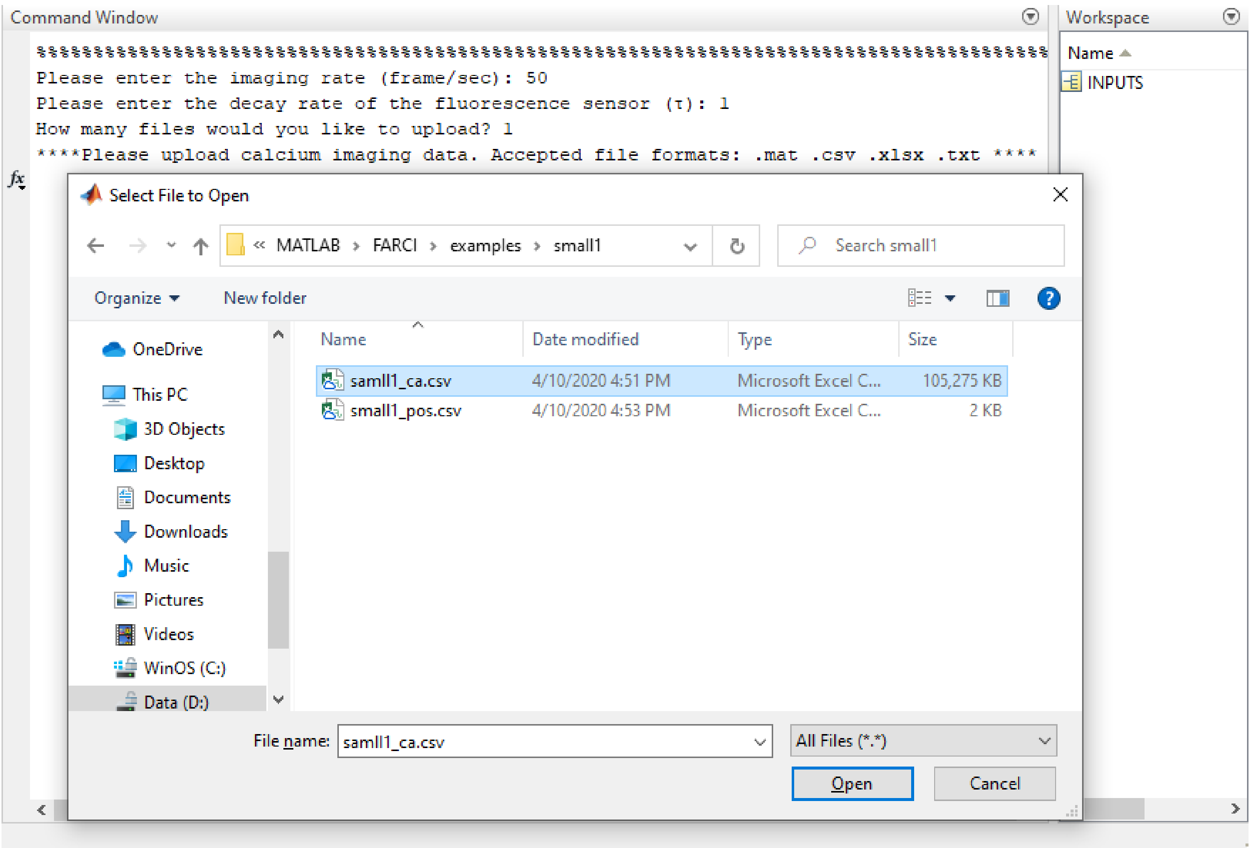


4.1.2 Data Import

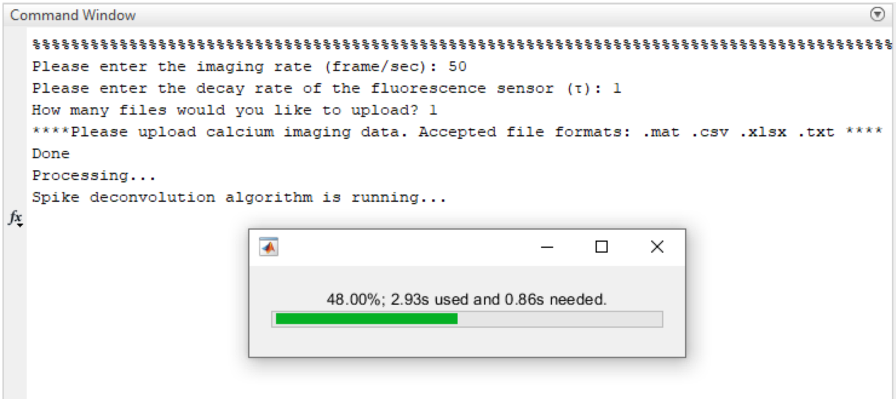
Next, we import small1 calcium imaging dataset (available in the subfolder examples/small1).

[1] http://connectomics.chalearn.org/



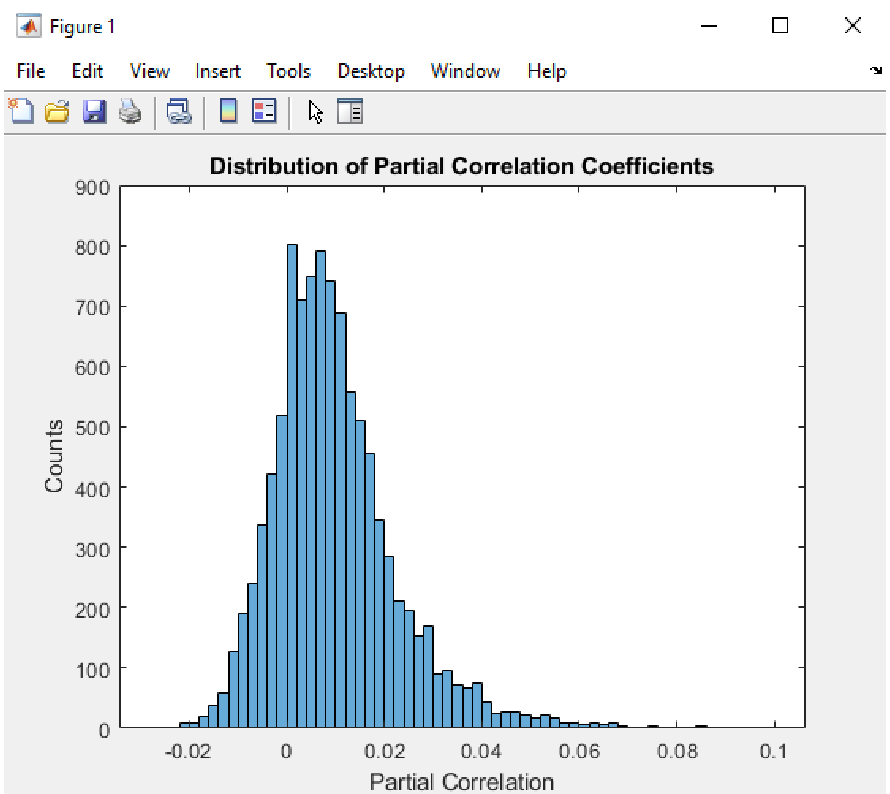


Once the data is imported, the deconvolution algorithm will run and spike data will be generated. Two preprocessing steps will be applied in order to denoise and smooth the inferred spikes from calcium imaging data.



4.1.3 Distribution of Partial Correlation Coefficients

After obtaining and processing the spikes, the partial correlation coefficients is calculated using the matrix inversion method. The first figure that is generated is the distribution of partial correlation coefficients.



The partial correlation statistics indicate the strength of association between each pair of neurons. Here, it is important to select a threshold for partial correlations so that only strong associations are considered as true edges.

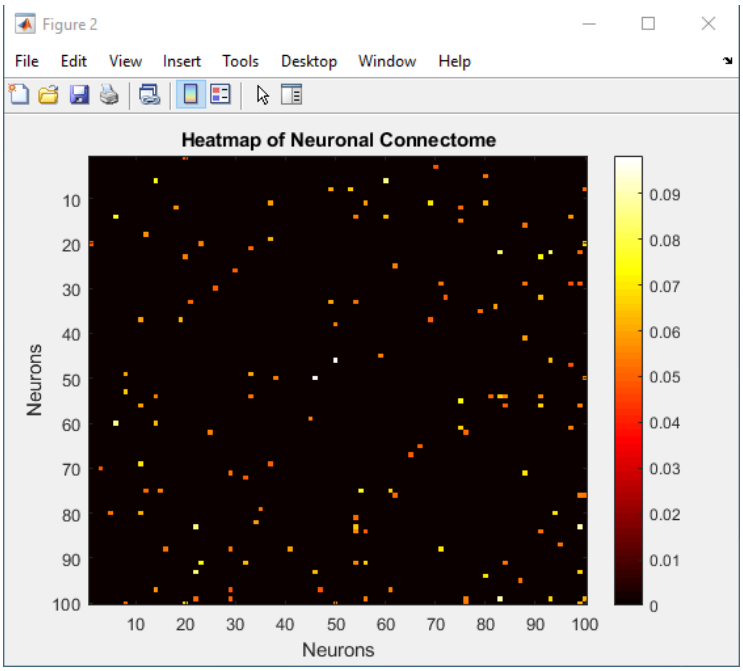
4.1.4 Thresholding the Partial Correlations

The distribution of partial correlations provides useful information about how weak vs strong associations can be distinguished. Furthermore, it enables the user to decide what level of thresholding is appropriate to remove the noise from the network. Here, the thresholding is defined as $Threshold = \mu + \alpha \times \sigma$, where μ and σ are the mean and standard deviation of partial correlations, respectively. The parameter is a cut-off parameter – the higher the α value, the sparser is the inferred connectome. As α is multiplied by σ, we recommend using a value between 1 to 5. Here, we use a moderate threshold ($\alpha = 3$).

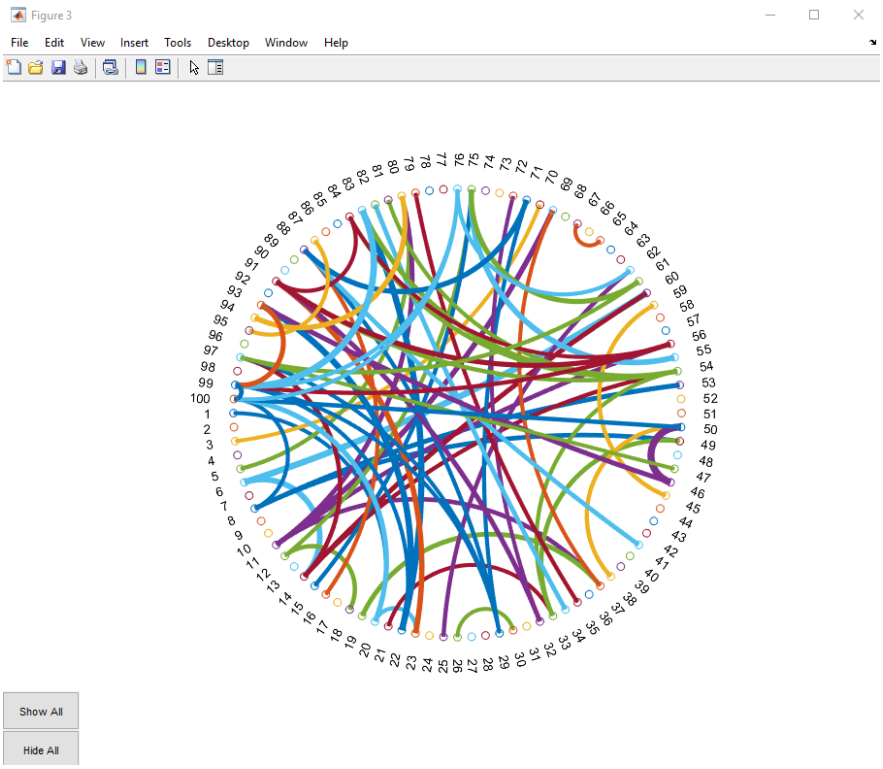
```
Command Window
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Please enter the imaging rate (frame/sec): 50
Please enter the decay rate of the fluorescence sensor (τ): 1
How many files would you like to upload? 1
****Please upload calcium imaging data. Accepted file formats: .mat .csv .xlsx .txt ****
Done
Processing...
Spike deconvolution algorithm is running...
To set a threshold, please enter a value for α:
fx (Note that, the threshold is defined as: Threshold = μ + α * σ): 3
```

4.1.5 Neuronal Connectome Inference

Once a desired threshold is chosen, the weak partial correlation coefficients are removed, and the neuronal connectome will be constructed based on the remaining coefficients. The final connectome is depicted in two different figures; the first figure is a heatmap of the partial correlations, representing the pairwise connections among the neurons. The non-zero elements of the partial correlation matrix give the edges of the connectome graph, and the partial correlation magnitudes indicate the strength of the connections.

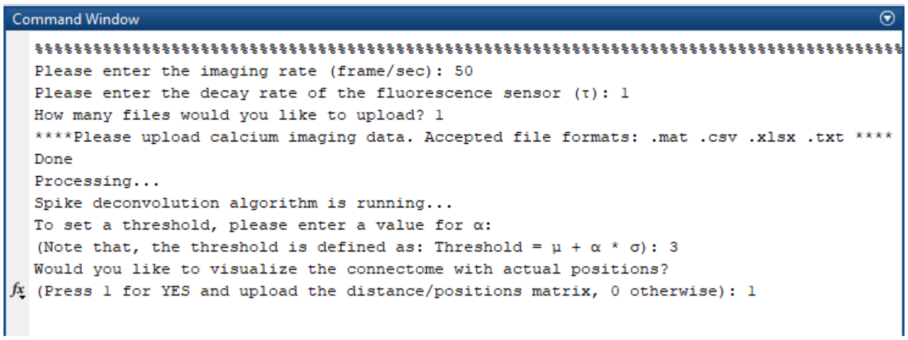


The second figure depicts a circular graph, where each neuron is shown by its user-provided index in the input data. In order to visualize distinct edges, different colors are used and the thickness of the edges is proportional to the strength of the connections.

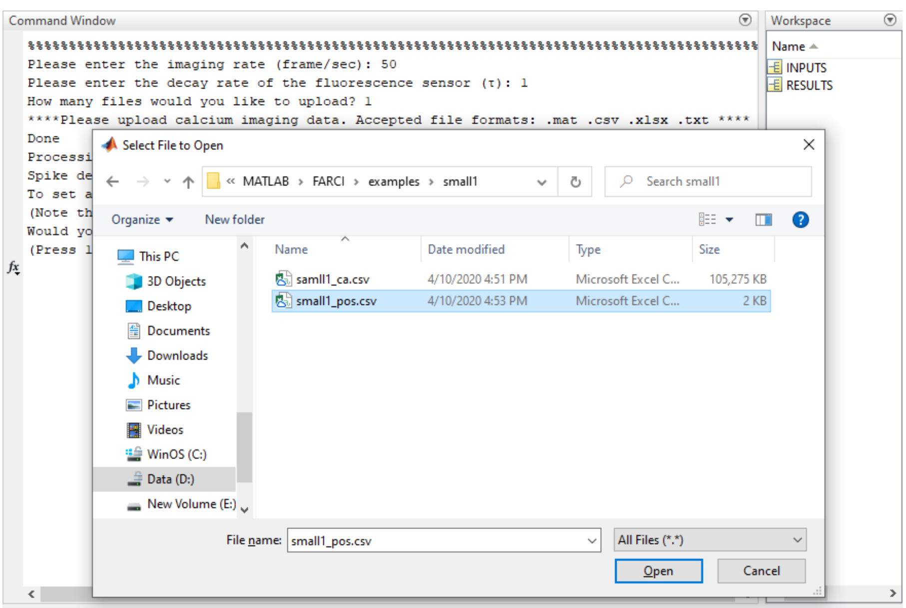


4.1.6 Neuronal Connectome with Actual Positions

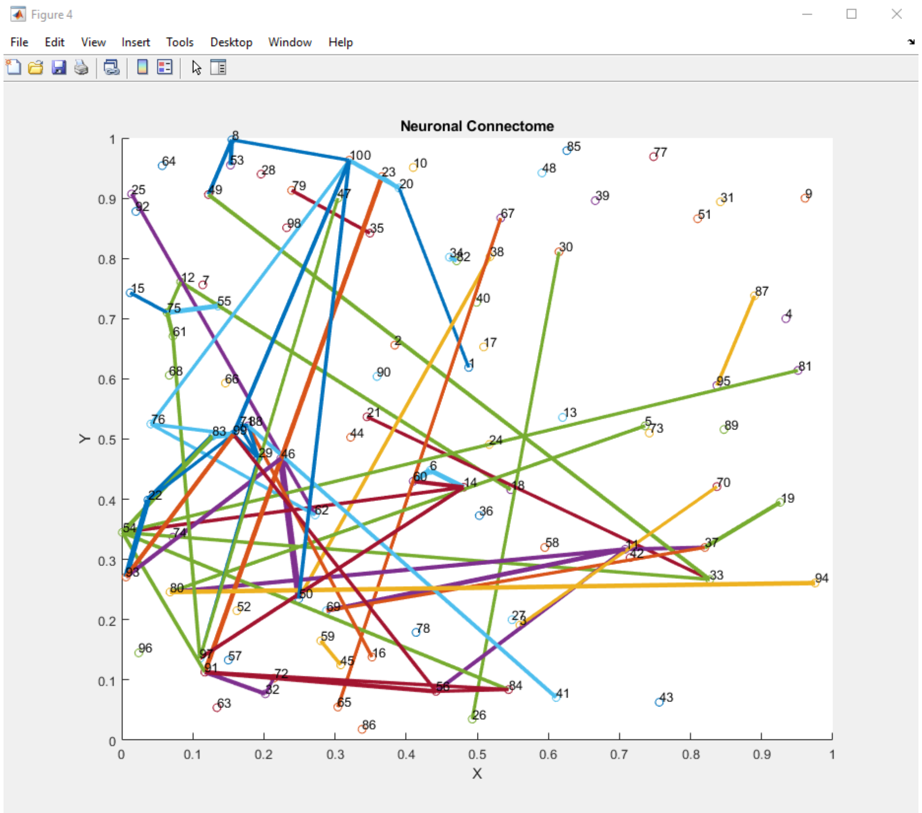
One additional network visualization figure can be generated when neuron positional data are available. The visualization is possible based on distance matrix or actual neuron positions in 2D.



If entered 1, the user will be asked to upload the position data either in form of an $(n \times n)$ pairwise distance matrix, where n is number of neurons, or an $(n \times 2)$ matrix with the x-y coordinates of the neurons.

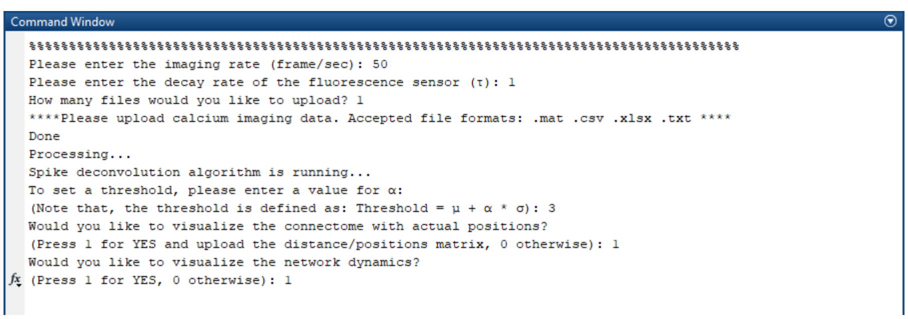


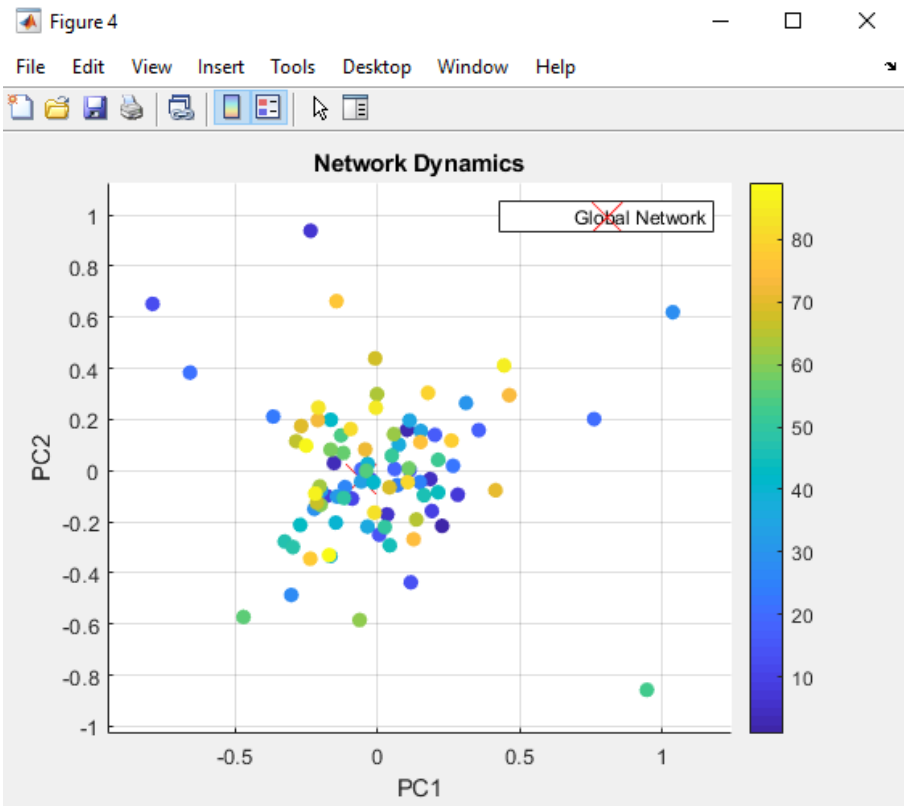
The connectome and the spatial positions of the neurons are depicted as a network graph, as shown in the following example figure. The edge colors are the same as those in the circular graph above, and the thickness of the edges again represents the strength of the connections (i.e. the magnitude of the partial correlations).



4.1.7 Neuronal Connectome Dynamics

Finally, FARCI can depict the neuronal dynamics by dividing the dataset into m subsets where $m = \left[\frac{t}{n \times 10}\right]$, t is number of time frames and n is number of neurons. The dynamics are then generated by projecting the subsets into lower dimension space using Principal Component Analysis (PCA).





The results of FARCI are stored in a data structure called RESULTS. This data structure contains 4 main variables: deconvolved spikes, partial correlation coefficients, the connectome matrix, and the edge matrix. The connectome matrix is an $(n \times n)$ adjacency matrix, where n is the number of neurons and each non-zero element (i, j), corresponds to an edge between neurons i and j, and the magnitude of the element represents the strength of the connection (i.e. the partial correlation coefficients). The edge matrix is a $(m \times 3)$ matrix where m is the total number of edges – the non-zero elements in the the connectome matrix. This matrix summarizes the network matrix where each row k indicates a connection between $(k, 1)$ and $(k, 2)$ with weight $(k, 3)$.

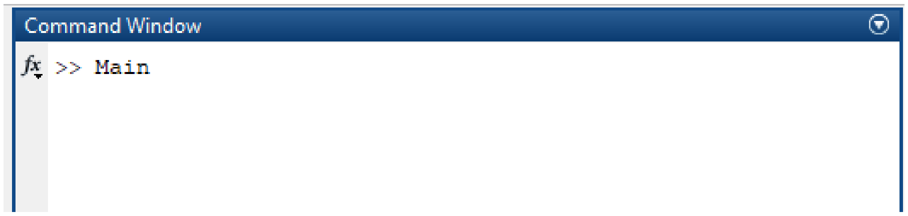
4.2 Example 2. Multiple datasets (longitudinal data)

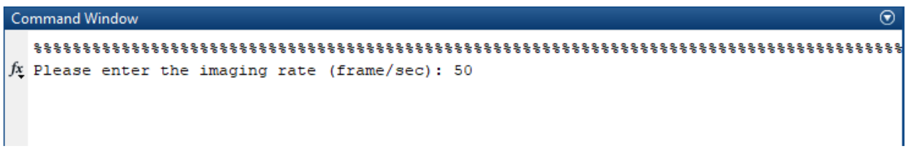
Here we analyze a case of multiple datasets where the data are collected at different time points or conditions. For simplicity, we use two files for "small1" and "small2" datasets assuming they are collected at two different time points. The datasets can be loaded separately (as csv, mat, etc.) or as a single .mat file that contains all the datasets stacked in a cell array. FARCI runs the connectome inference algorithm for different datasets and generates all the plots for each connectome. Eventually, the connectome dynamics will be visualized in a single plot which can be useful in understanding the temporal evolution of the connectome.

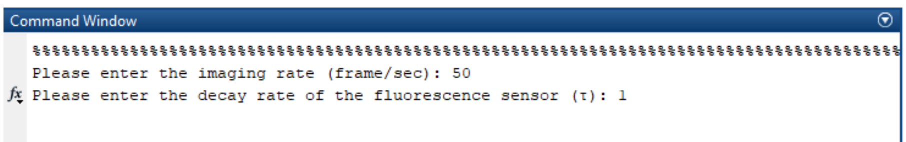
4.2.1 Deconvolution Parameters

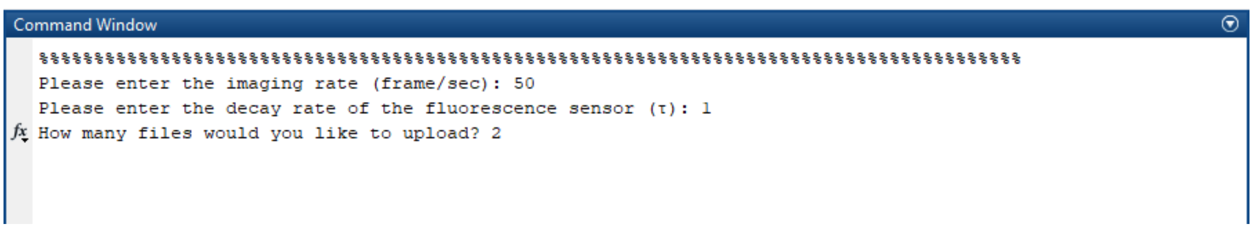
We begin with changing the current directory in MATLAB to the FARCI folder. Then, we run the `Main.m` script file in the main folder of FARCI. First, the user will be asked to enter deconvolution parameters: imaging rate and fluorescence decay rate.

The following are screenshots from running FARCI in MATLAB.



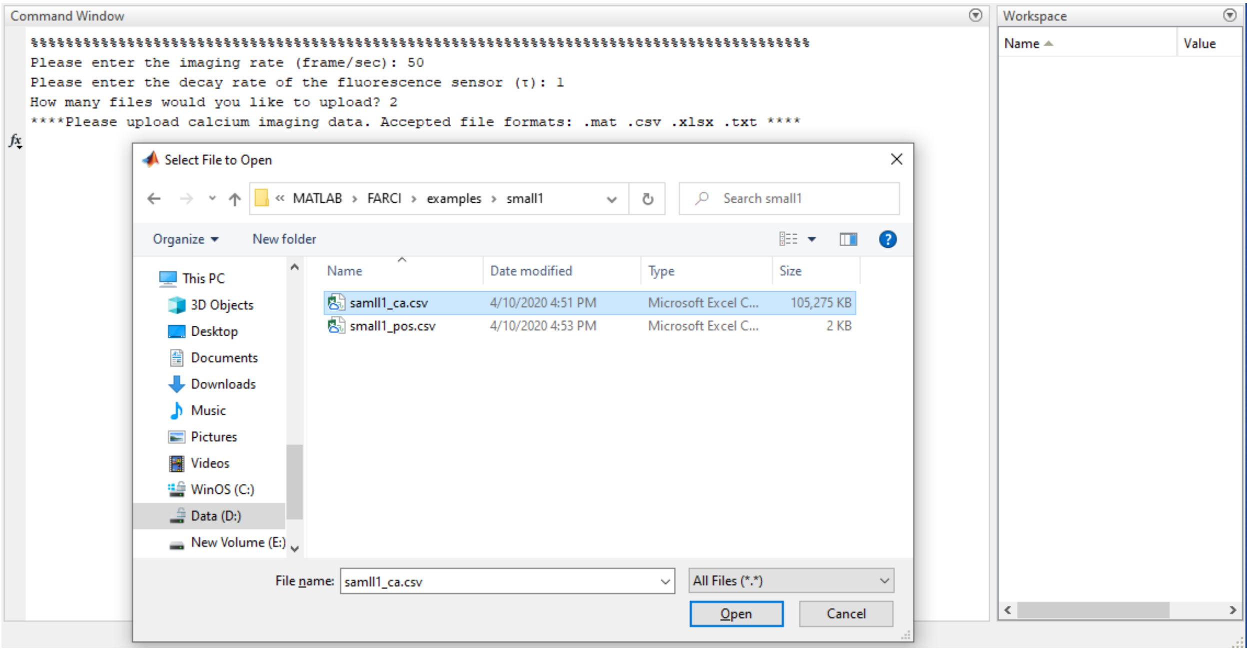


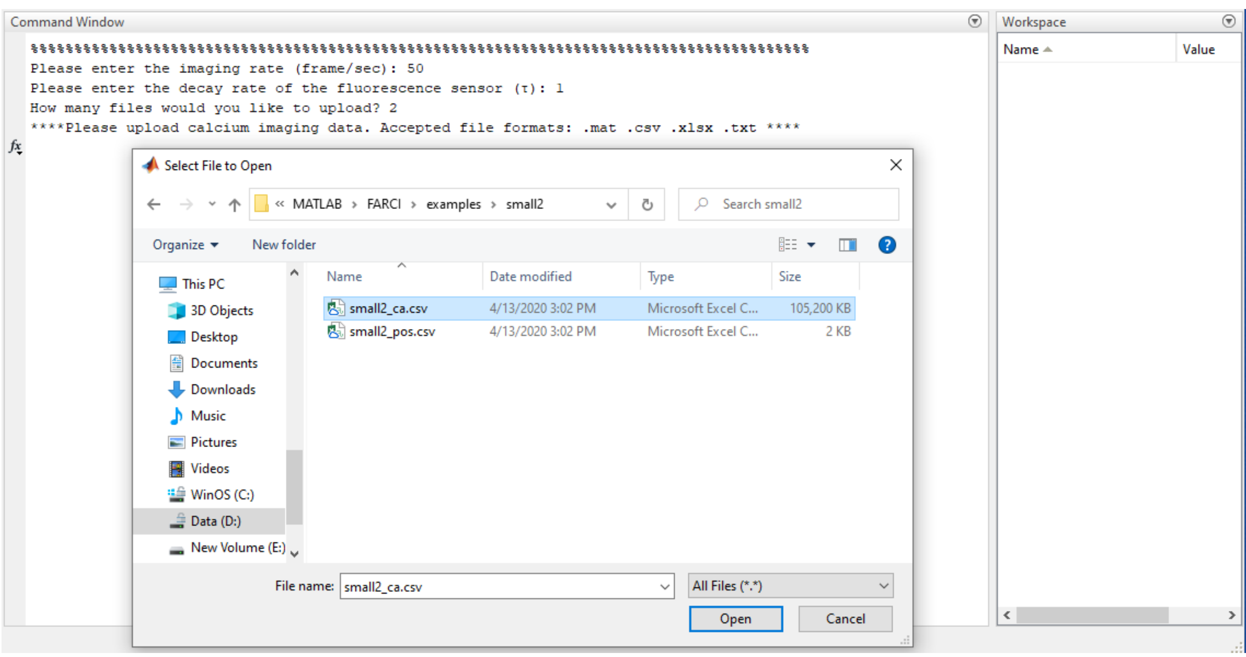




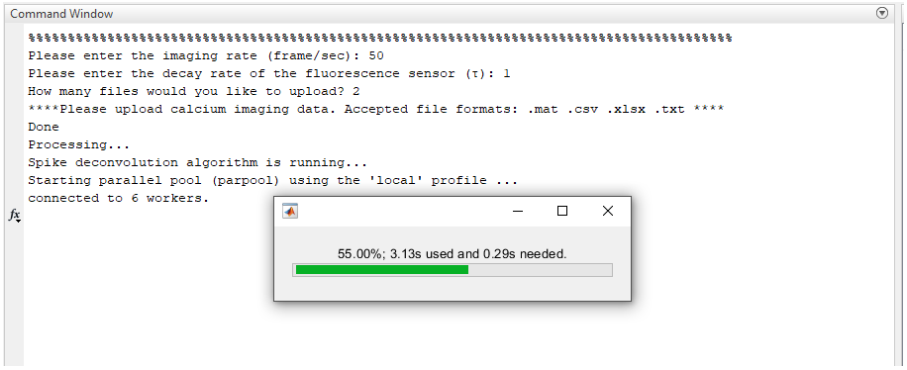
4.2.2 Data Import

Next, we import small2 calcium imaging dataset (available in the subfolder examples/small2).



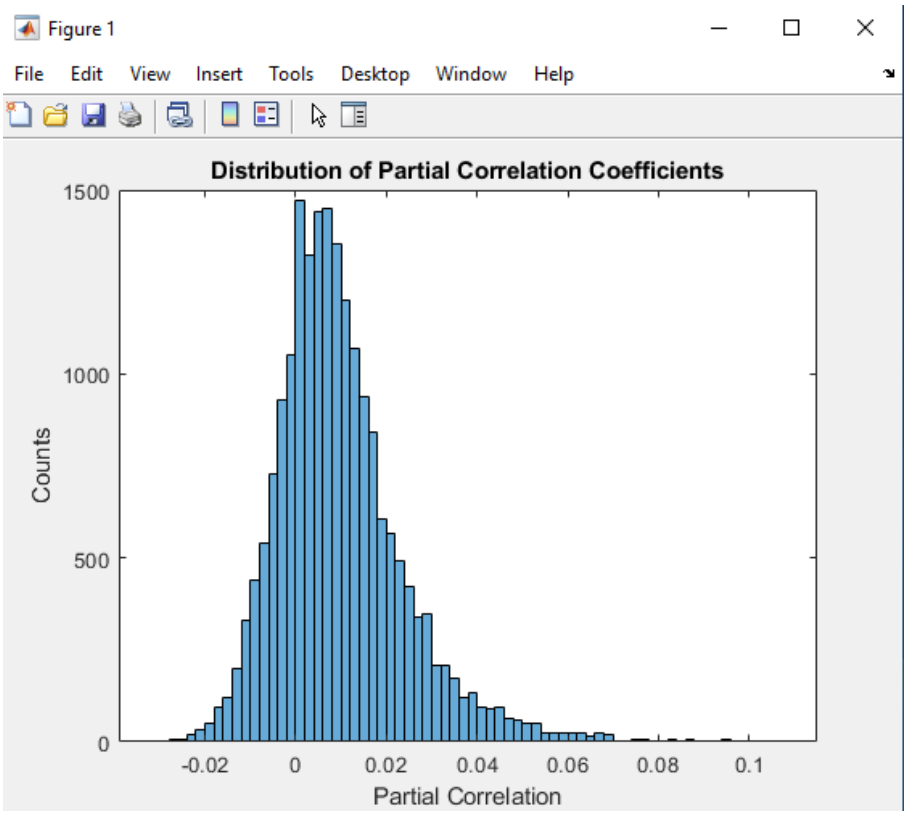


Once the data are imported, the deconvolution algorithm will run and spike data will be generated. Two preprocessing steps will be applied in order to denoise and smooth the inferred spikes from calcium imaging data.



4.2.3 Distribution of Partial Correlation Coefficients

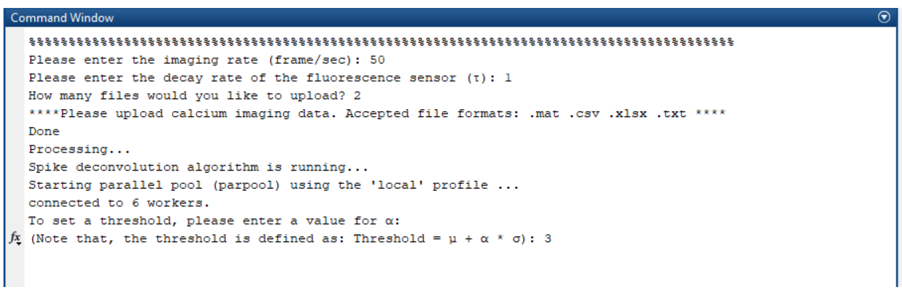
After obtaining and processing the spikes, partial correlation coefficients will be calculated using matrix inversion method. The first figure is the distribution of partial correlation coefficients.



The partial correlation statistics indicate the strength of association between each pair of neurons. Here, it is important to select a threshold for partial correlations so that only strong associations are considered as true edges.

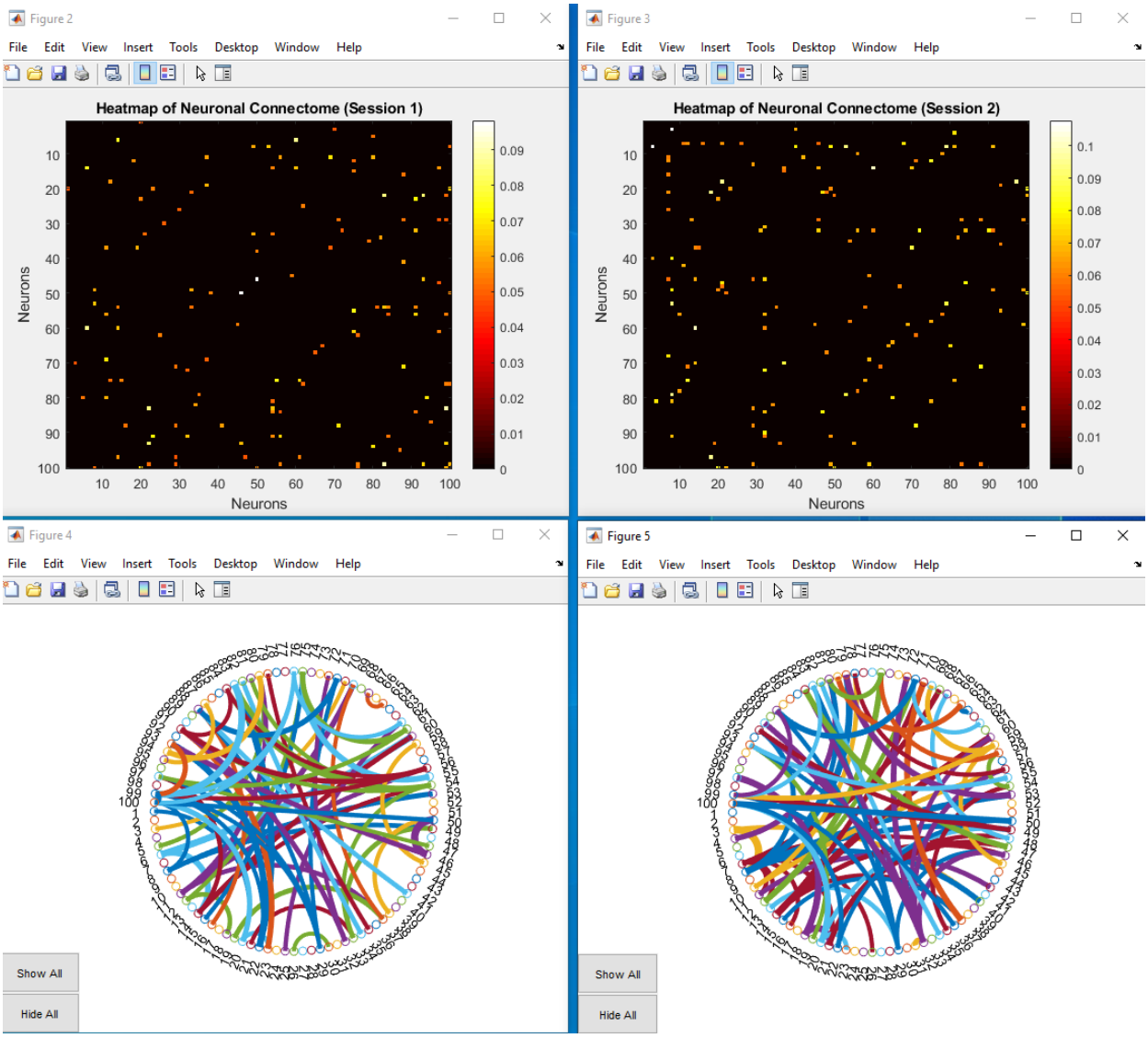
4.2.4 Thresholding the Partial Correlations

The distribution of partial correlations provides useful information about how weak vs strong associations can be distinguished. Furthermore, it enables the user to decide what level of thresholding is appropriate to remove the noise from the network. Here, the thresholding is defined as $Threshold = \mu + \alpha \times \sigma$, where μ and σ are the mean and standard deviation of partial correlations, respectively. In addition, α is a tuning coefficient that allows adjusting the sparsity of the network. As it is clear from the equation, higher alpha leads to a higher threshold and consequently a sparser network. We use an integer, usually 1-5, for the value of alpha, but the user can enter any arbitrary value. Here, we use a low threshold ($\alpha = 2$) to ensure the resulting network has a higher edge density compared to the previous example.



4.2.5 Neuronal Connectome Inference

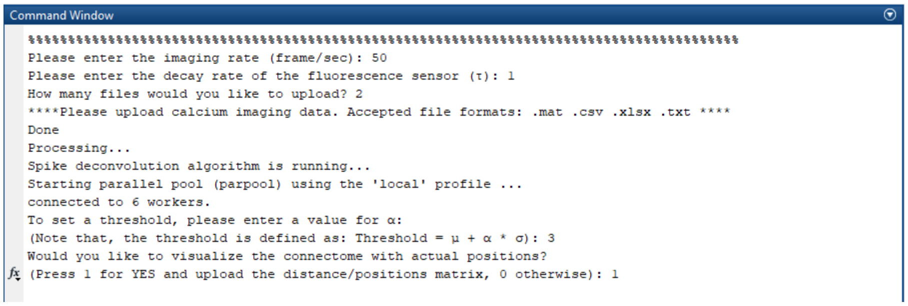
Once a desired threshold for the partial correlations is chosen, the weak connections will be removed and the network will be constructed based on the remaining connections. The final network will be generated automatically in two different figures; the first figure is a heatmap of pairwise connections between neurons. The edges are non-zero elements of the graph where the colors indicate the strength of the connections.



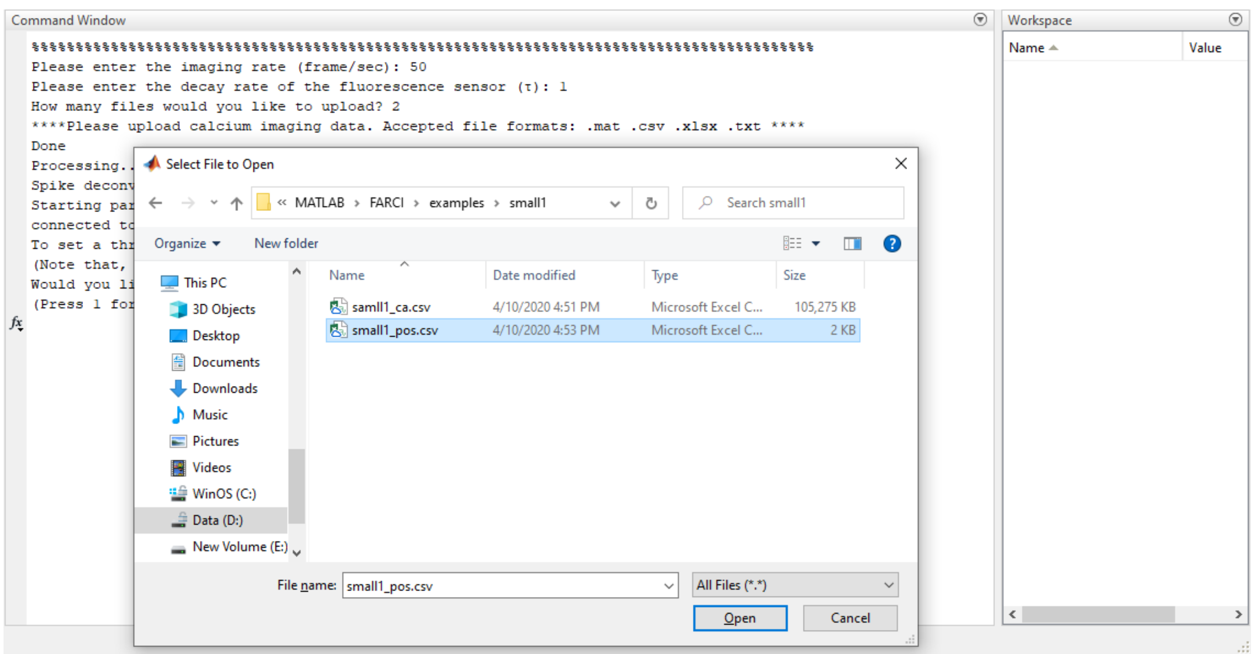
The second figure is a circular graph where each neuron are shown by its index and a small circle. In order to visualize distinct edges, different colors are used and the thickness of the edges is proportional to the strength of the connections.

4.2.6 Neuronal Connectome with Actual Positions

One additional network visualization figure can be generated when neuron positions data is available. This visualization is possible based on distance matrix or actual neuron positions in 2D.

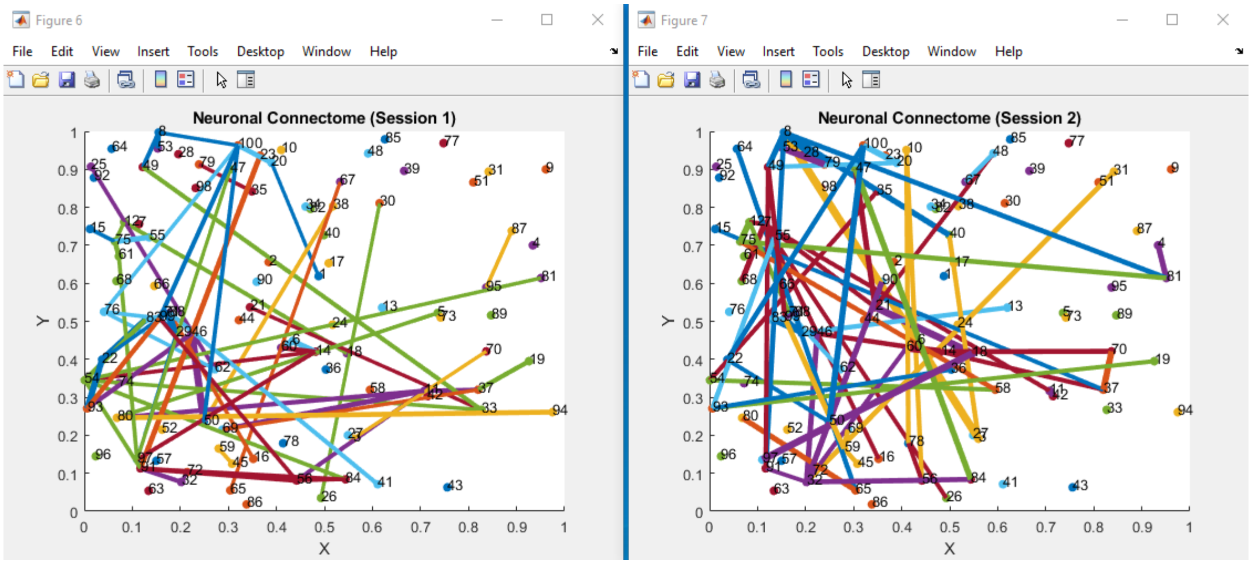


If entered 1, the user will be asked to upload the position data either in form of a pairwise distance matrix, a $(n \times n)$ matrix where n is number of neurons, or actual positions of neurons, a $(n \times 2)$ matrix where first and second columns represent positions in x and y coordinates, respectively.



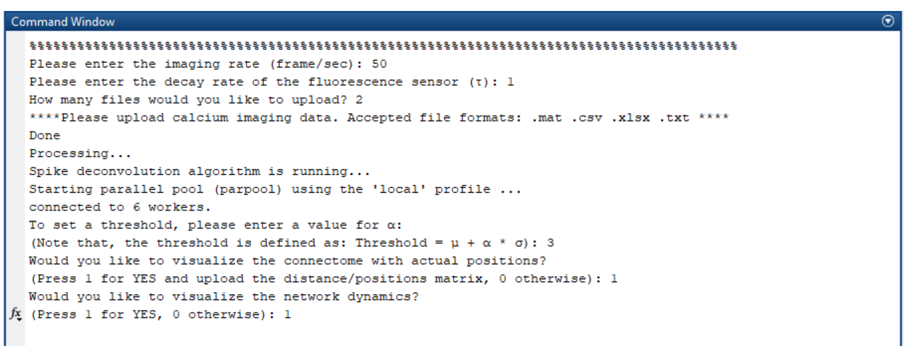
The network will be shown in a new figure, while the edge colors are conserved to be consistent with what was used in the circular graph and the thickness of the edges represents the strength of the connections. The final outputs will be stored in one data structure called RESULTS. This data structure contains 4 main variables: deconvolved spikes, partial correlation coefficients, and network and edges matrices. The network matrix is a $(n \times n)$ matrix, where n is the number of neurons and each non-zero element (i, j), corresponds to an edge between neurons i and j and the magnitude of this element represents the weight of the connection. On the other hand, edges matrix is a $(m \times 3)$

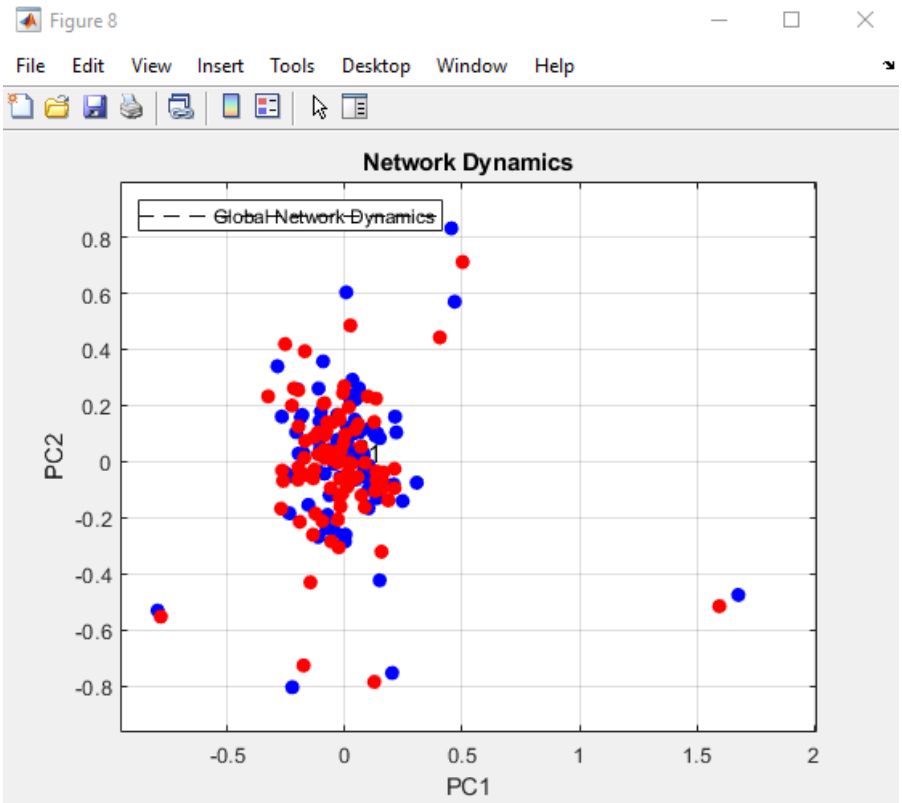
matrix where m is the total number of edges. This matrix summarizes the network matrix where each row k indicates a connection between $(k, 1)$ and $(k, 2)$ with weight $(k, 3)$.



4.2.7 Neuronal Connectome Dynamics

Finally, FARCI can depict the neuronal dynamics by dividing the dataset into m subsets where $m = [\frac{t}{n \times 10}]$, t is number of time frames and n is number of neurons. The dynamics are then generated by projecting the subsets into lower dimension space using Principal Component Analysis (PCA).





5 Questions and Comments

Please address any problem or comment to: sabermea@buffalo.edu or rgunawan@buffalo.edu.