FARCI: User Manual

Version 1.0.0 (April 13, 2020)

Authors: Saber Meamardoost and Rudiyanto Gunawan

Chemical and Biological Engineering Department, University at Buffalo, SUNY

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

Table of contents

1 Overview

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

FARCI provides a user-friendly implementation of connectome inference workflow from calcium imaging data. FARCI outputs pairwise association matrix, inferred network matrix, circular graph of the neuronal connectome and, when spatial information is available, neuronal connectome based on actual position of neurons.

For detailed information about FARCI, please refer to the following manuscript.

Meamardoost, S., Gunawan R., **FARCI: Fast and Robust Connectome Inference**, bioRxiv, 2020 (link).

2 System requirements

This distribution of FARCI is written and developed in MATLAB [1].

FARCI has been successfully tested on MATLAB 2017b and 2018b.

3 FARCI package

FARCI package contains the following files and folders:

1. This `FARCI_USER_MANUAL.doc` file.
2. `License.txt` modified BSD license for FARCI
3. `MAIN.m` FARCI main script (use this script to run FARCI on your own dataset)
4. The folder `subfunctions` containing the following main 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` : deconvolves the raw calcium imaging data and infer spikes
    e. `sp_denoise.m` : denoises the inferred spikes
    f. `sp_smooth.m` : smoothens the spikes
    g. `partial_corr.m` : computes 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)

[1] https://www.mathworks.com

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[2]. For each dataset, ONLY the most important results are reported. Please refer to the file `MAIN.m` for an example MATLAB script of FARCI implementation.

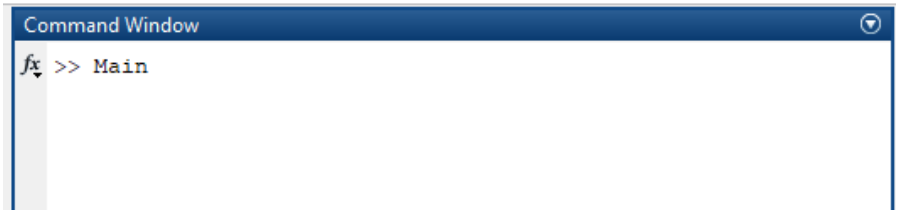
4.1 Example 1. Small1 dataset

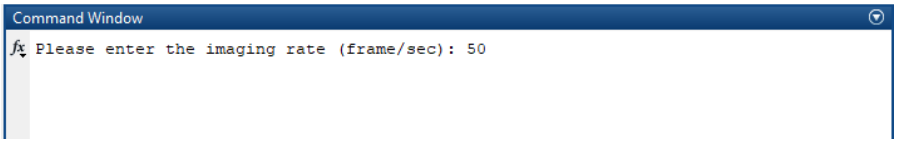
Here we analyze small1 dataset provided in the Neural Connectomics Challenge that contains calcium imaging data for 100 neurons along with their positions in 2D.

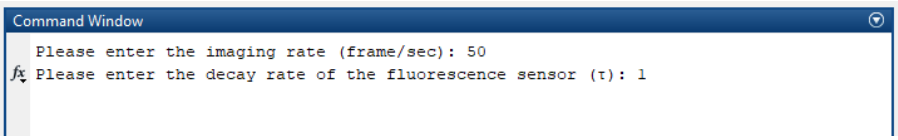
4.1.1 Deconvolution Parameters

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.



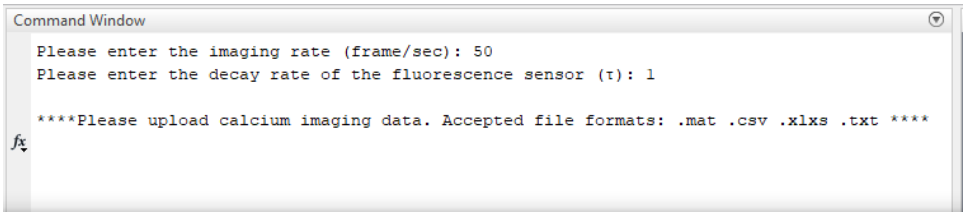


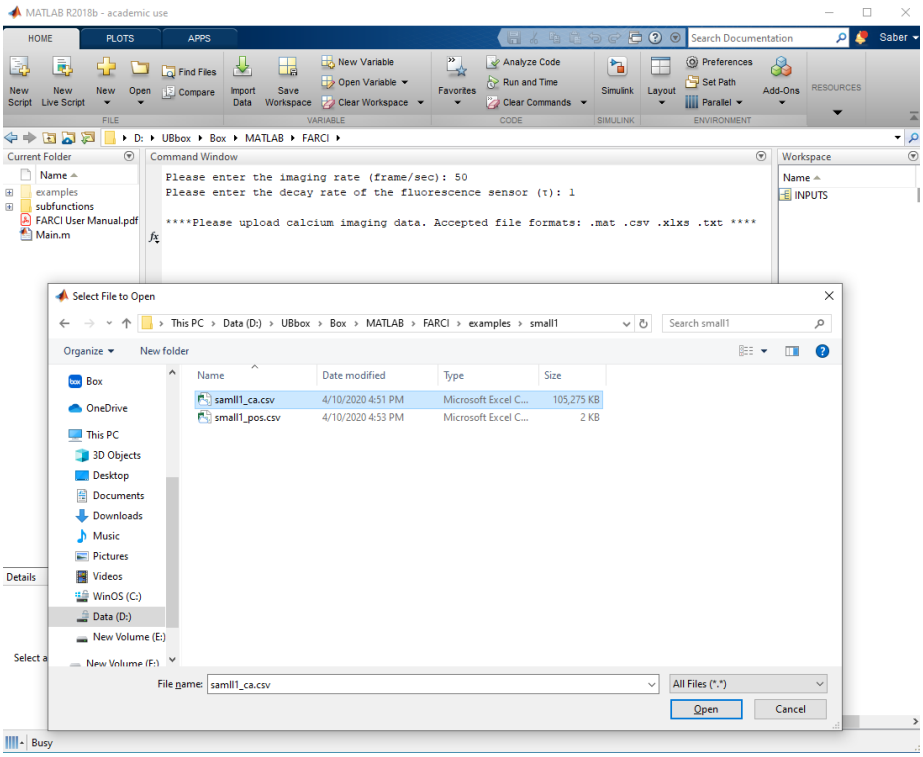


4.1.2 Data Import

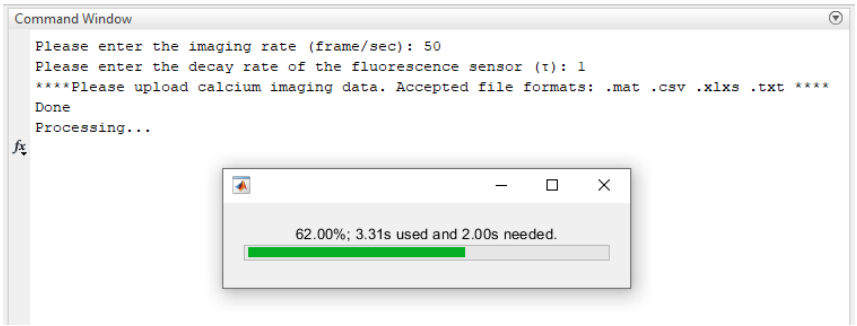
Next, we import small1 calcium imaging dataset (available in the subfolder examples/small1).

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



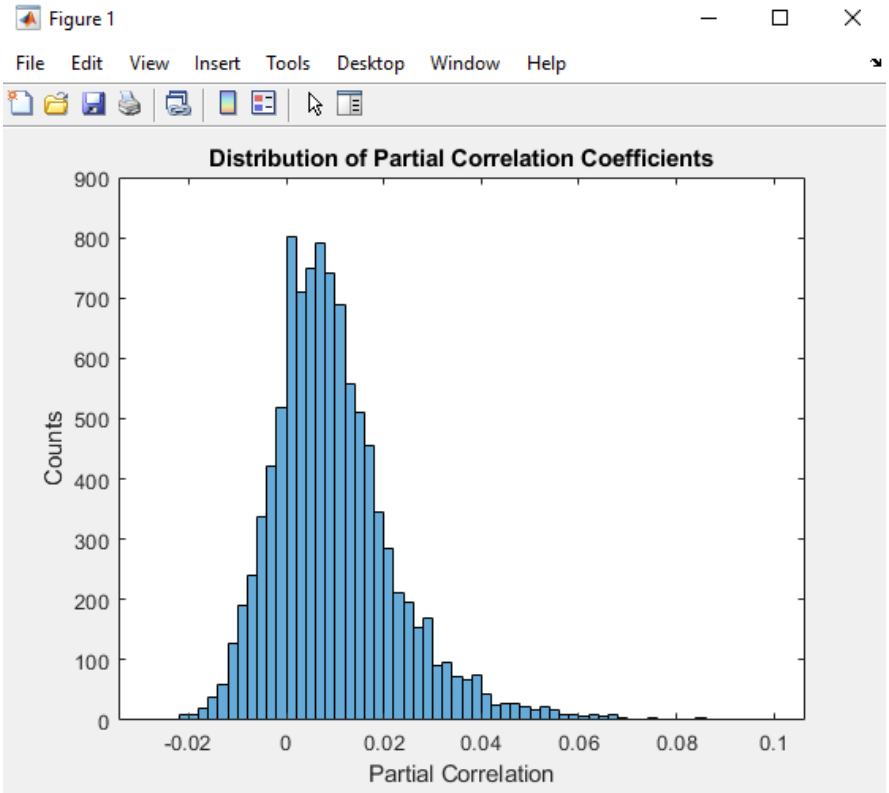


Once the data is imported, the deconvolution algorithm will run and spike data will be generated. Two preprocessing steps will be applied in order to denoise and smooth the inferred spikes from calcium imaging data.



4.1.3 Distribution of Partial Correlation Coefficients

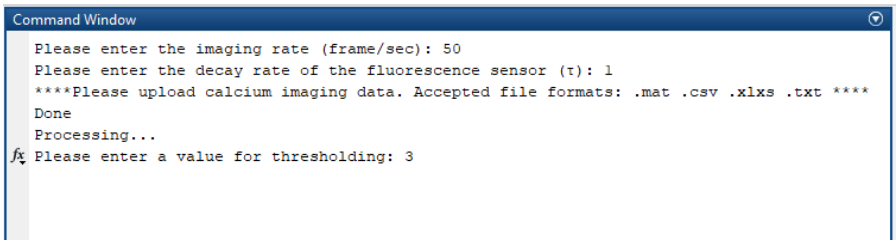
After obtaining and processing the spikes, partial correlation coefficients will be calculated using matrix inversion method. The first figure is the distribution of partial correlation coefficients.



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.

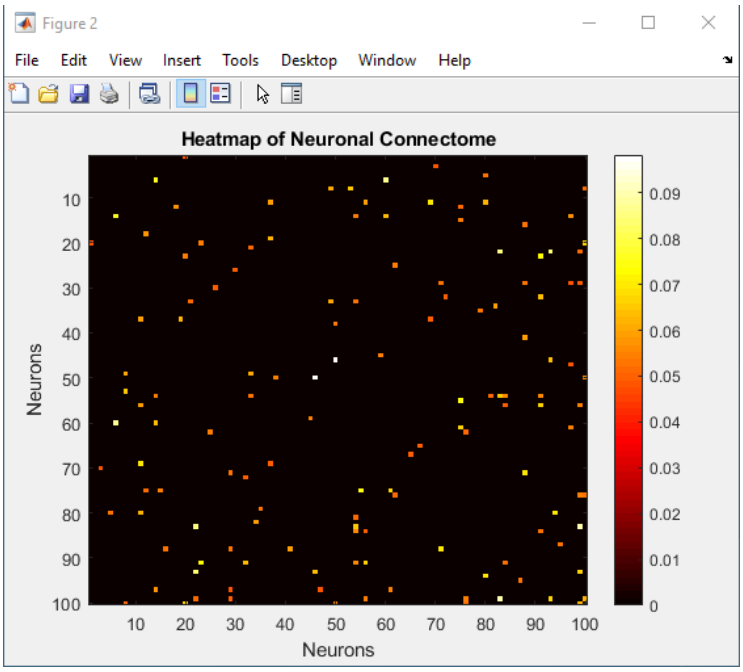
4.1.4 Thresholding the Partial Correlations

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. 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 moderate threshold ($\alpha = 3$) to ensure the resulting network is sparse enough so that distinct edges can be visualized.

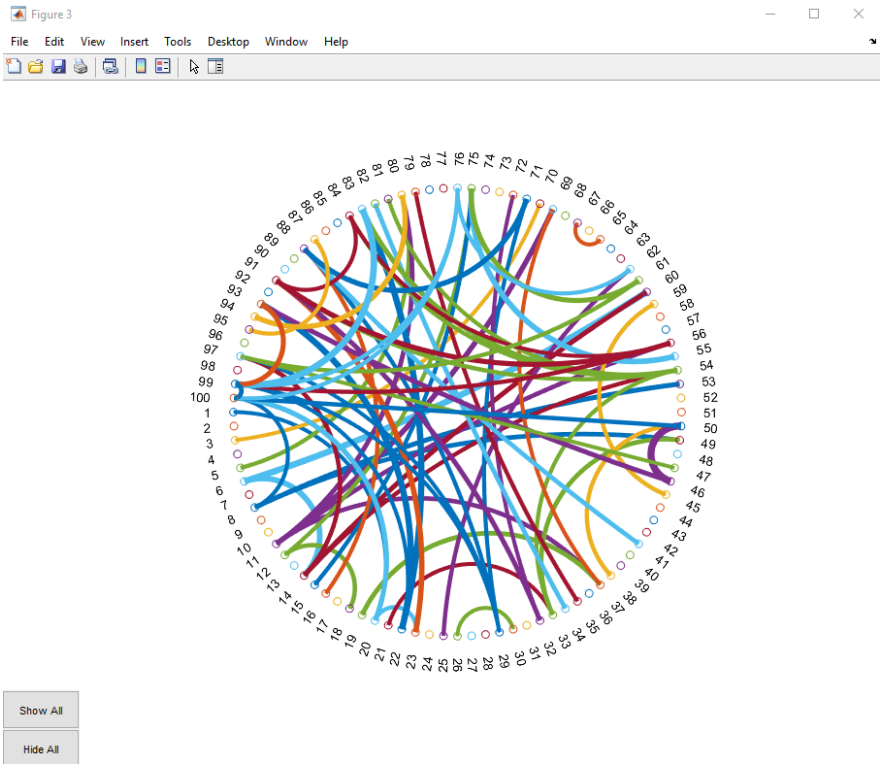


4.1.5 Neuronal Connectome Inference

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.

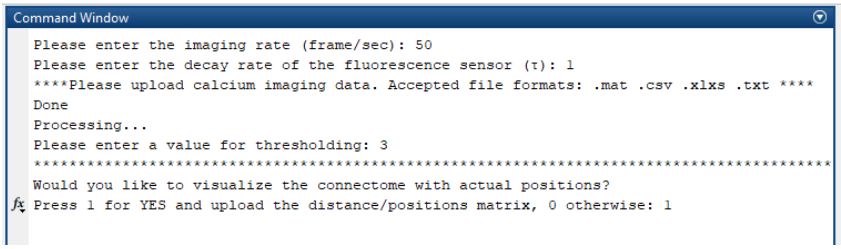


The second figure is a circular graph where each neuron is shown by its index and a small circle. In order to visualize distinct edges, different colors are used and the thickness of the edges is proportional to the strength of the connections.

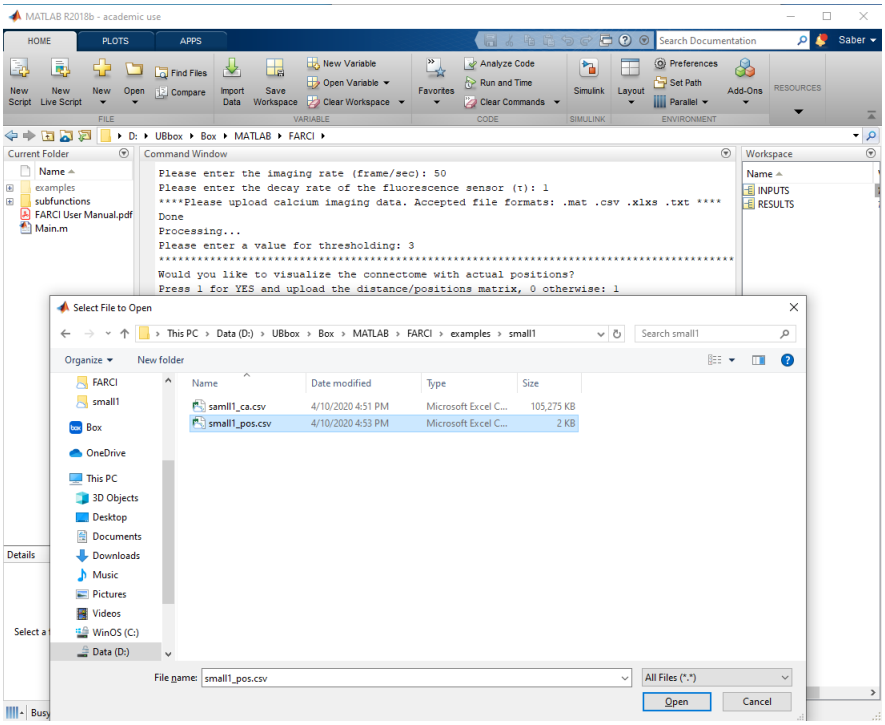


4.1.6 Neuronal Connectome with Actual Positions

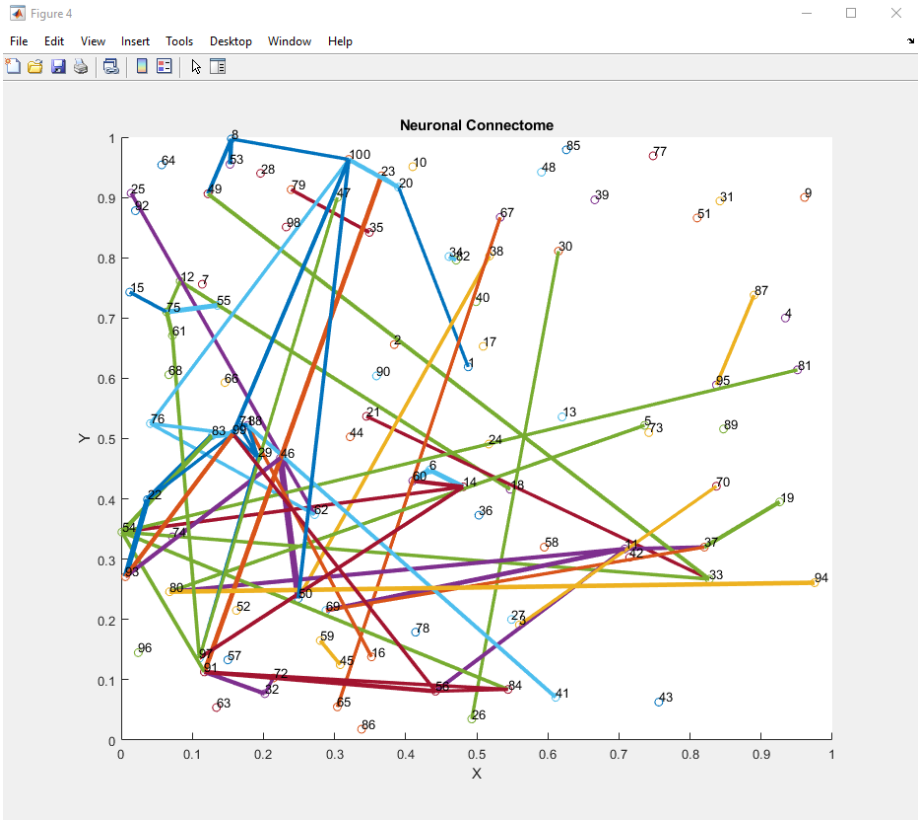
One additional network visualization figure can be generated when neuron positions data is available. This visualization is possible based on distance matrix or actual neuron positions in 2D.



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 $(k, 3)$.



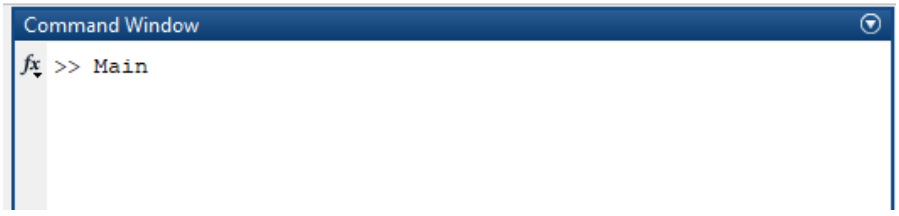
4.2 Example 2. Small2 dataset

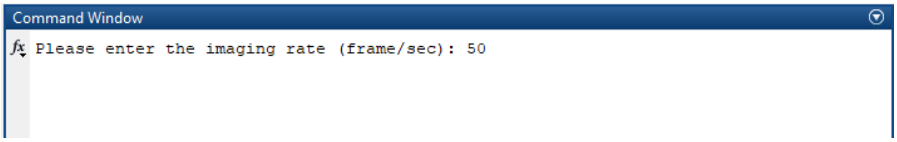
Here we analyze small2 dataset provided in the Neural Connectomics Challenge that contains calcium imaging data for 100 neurons along with their positions in 2D.

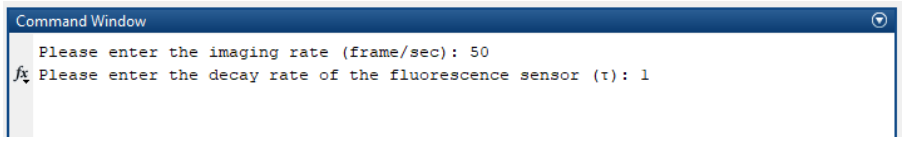
4.2.1 Deconvolution Parameters

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.

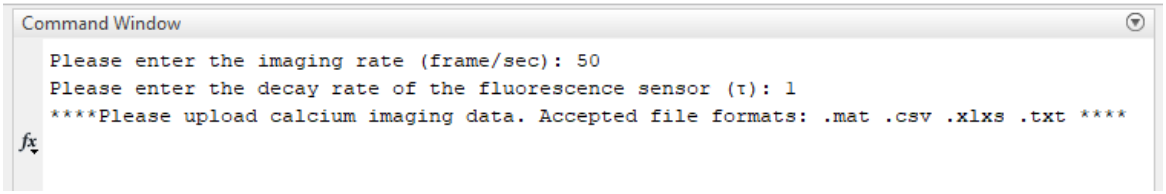


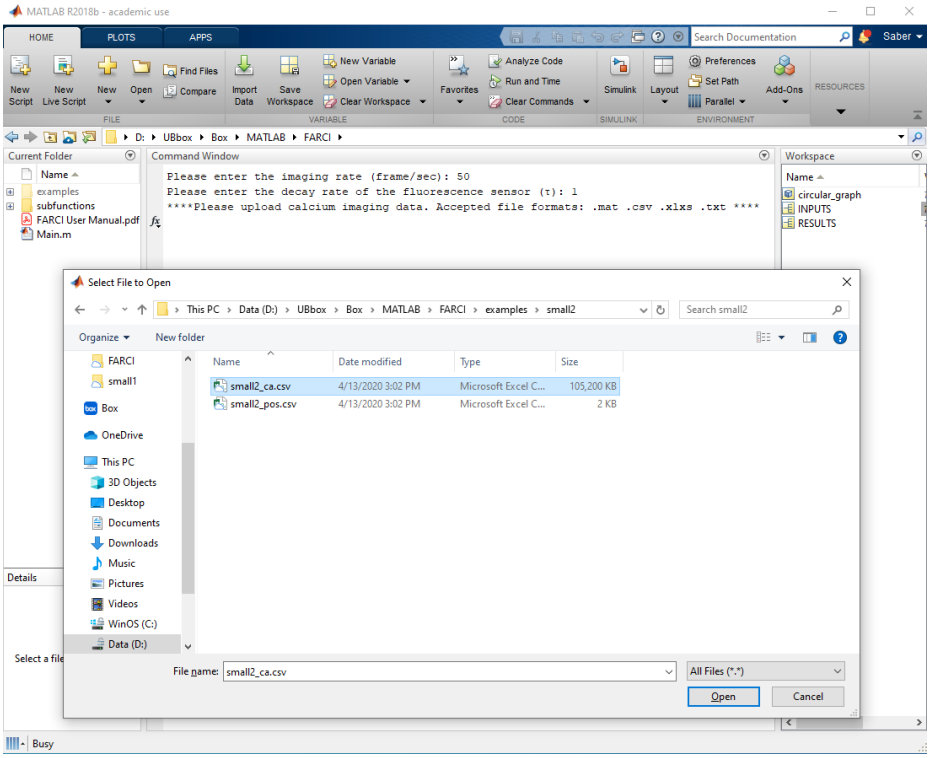




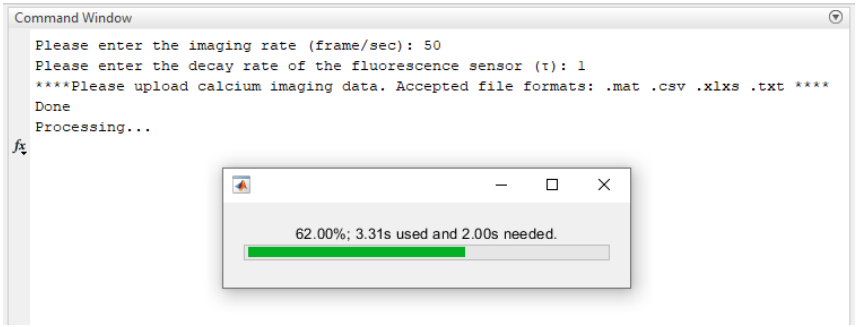
4.2.2 Data Import

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



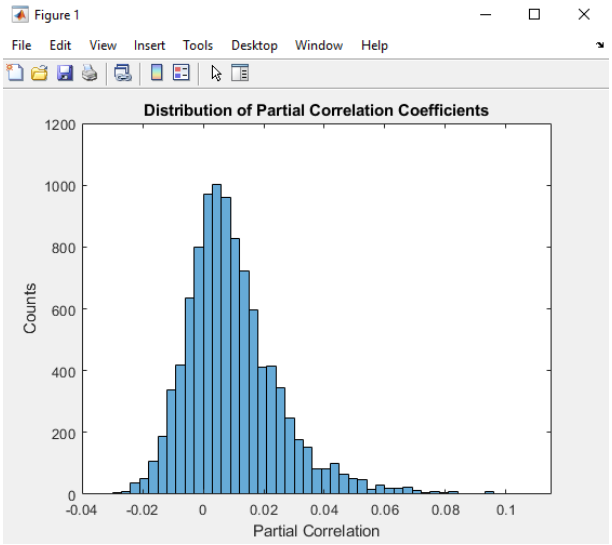


Once the data is imported, the deconvolution algorithm will run and spike data will be generated. Two preprocessing steps will be applied in order to denoise and smooth the inferred spikes from calcium imaging data.



4.2.3 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.

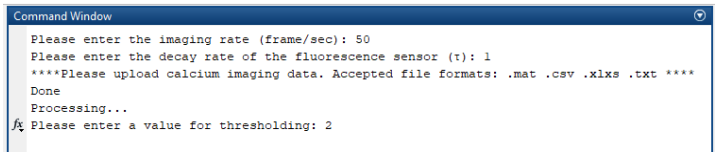


The partial correlation statistics indicate the strength of association between each pair of neurons. Here, it is important to select a threshold for partial correlations so that only strong associations are considered as true edges.

4.2.4 Thresholding the Partial Correlations

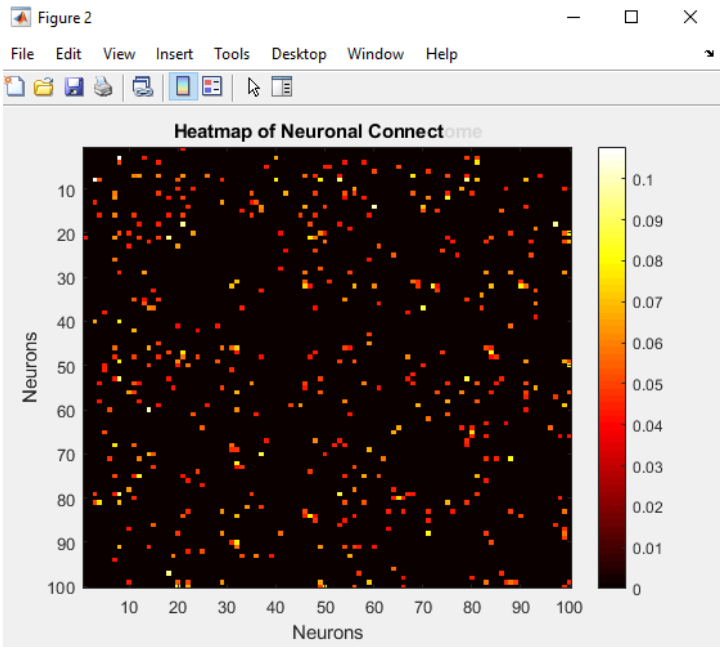
The distribution of partial correlations provides useful information about how weak vs strong associations can be distinguished. Furthermore, it enables the user to decide what level of thresholding is appropriate to remove the noise from the network. Here, the thresholding is defined as $Threshold = \mu + \alpha \times \sigma$, 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.

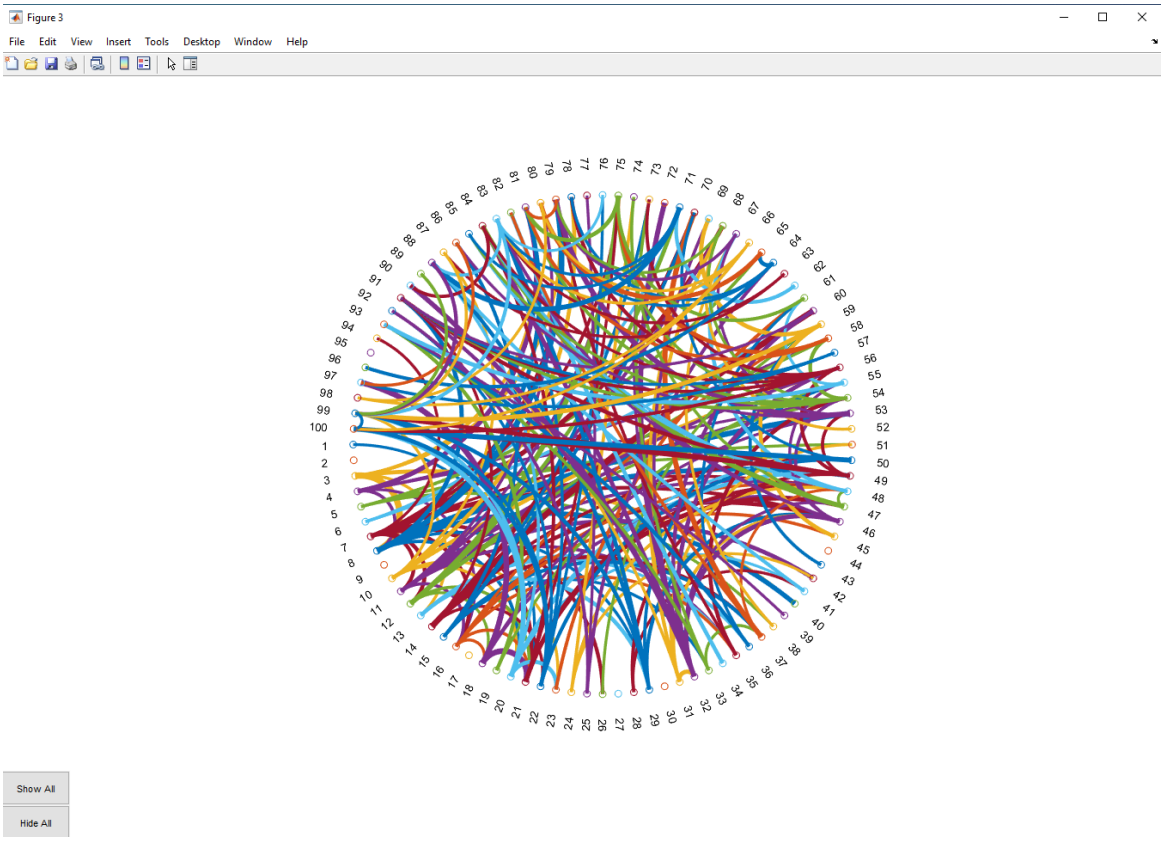


4.2.5 Neuronal Connectome Inference

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.

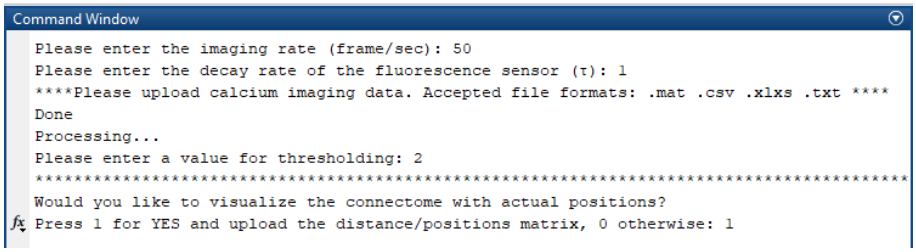


The second figure is a circular graph where each neuron is shown by its index and a small circle. In order to visualize distinct edges, different colors are used and the thickness of the edges is proportional to the strength of the connections.

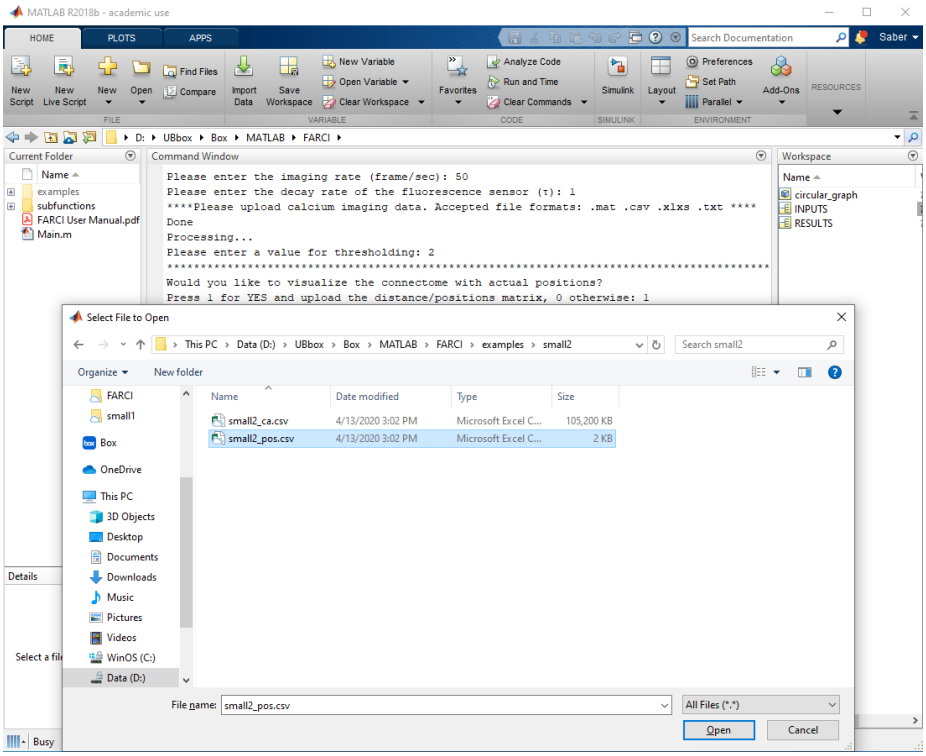


4.2.6 Neuronal Connectome with Actual Positions

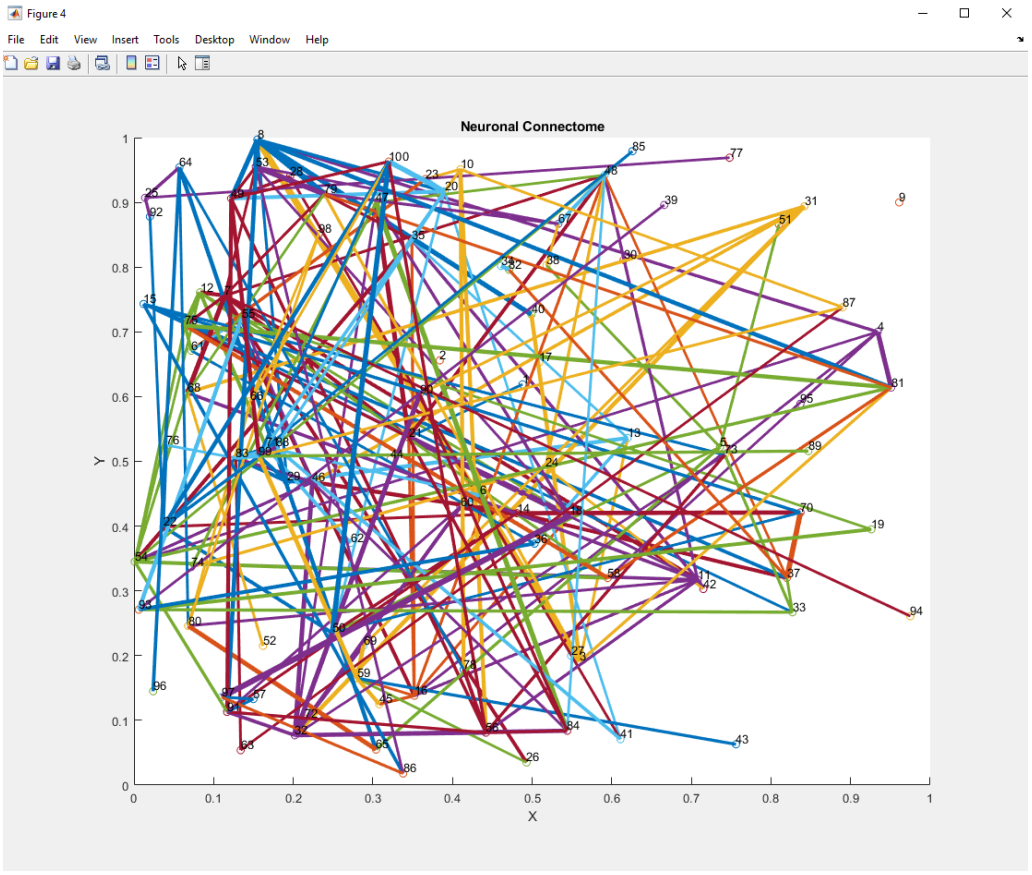
One additional network visualization figure can be generated when neuron positions data is available. This visualization is possible based on distance matrix or actual neuron positions in 2D.



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 ma trix where each row k indicates a connection between $(k, 1)$ and $(k, 2)$ with weight $(k, 3)$.



5 Questions and Comments

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