Mathematical Methods for Neural Science & Engineering

Coursework Two
Deadline: Fri 16th Dec, 3pm.

Each coursework consists of a set of "real" data analysis or mathematical modelling tasks, which you should be able to carry out with the aid of what you have learned from the lectures and problem classes.

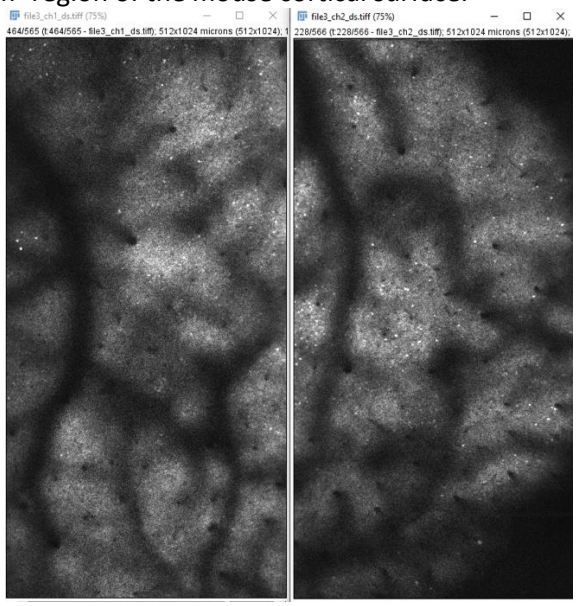
Your coursework should be created as a Jupyter Notebook. To submit this through Blackboard, first convert it to html. To do this, from the notebook itself run the following command:

```
! jupyter nbconvert --to html your_notebook_name.ipynb
```

Your coursework submission should include markup text describing your answers to the questions, and explanations, your code, and inline figures. Your figures should be of publication quality (i.e. of similar standard to those in a good journal paper) – label all axes, give units, provide legends where appropriate, etc. **Make sure that your figures are present and visible in your notebook before converting to html!**

You may collaborate, but you must write up your report **completely independently** (this will be checked), and reference all sources beyond the material on Blackboard.

In this coursework you will analyse data from the following paper :
C.-H. Yu, J. N. Stirman, Y. Yu, R. Hira and S.L. Smith, Diesel2p mesoscope with dual independent scan engines for flexible capture of dynamics in distributed neural circuitry, *Nature Communications* 12:6639, 2021.
We suggest that you familiarise yourself with the paper to understand where the data comes from, although there is no need to worry too much about the details of the optical technology described therein. The paper describes two-photon calcium imaging from the cortex of a mouse, with a novel instrument that allows collection of data from a large number of neurons simultaneously – thousands of neurons across a 5x5mm^2 region of the mouse cortical surface.



You have been provided with a number of datafiles, described below:

coord_file3_ch1.npy	x-y coordinates of neurons in channel 1
coord_file3_ch2.npy	x-y coordinates of neurons in channel 2 (adjacent)

DFF_file3_ch1.npy ΔF/F fluorescence time series for all neurons in channel 1
DFF_file3_ch2.npy ΔF/F fluorescence time series for all neurons in channel 2 (adjacent)
These two channels are arranged physically side by side as depicted above, and imaged simultaneously.

The file skel.py loads these in and does a few basic plots (which may help you answer Q1). After that, you are on your own…

You may make use of the networkx python toolbox. Where you are not calculating quantities in your own code, please ensure that you *describe* what the called routine is doing.

Coursework Two

1. Load the calcium imaging dataset provided in the Coursework 2 directory. Note: there are two channels which contain the left and right halves of the image neurons, and in the skeleton code I show loading both of them and merging them together; however it is fine to just use one of the channels (~half of the cells) if you wish to keep memory usage low. (a) Plot the physical locations of the cells in a 2D representation. (b) Plot the ΔF/F fluorescence time series for all cells as a matrix (Cells x Time) with colourscale indicating signal amplitude. (c) For 5 typical cells in which there are obvious calcium transients, plot the fluorescence time series in a separate plot with the 5 traces overlaid on a shared x axis but vertically offset.

2. [20 marks]

3. Calculate the matrix of Pearson correlation coefficients between the calcium fluorescence time series from each cell. Now plot the distribution of correlation coefficients.

[20 marks]

4. Form an undirected graph, with nodes representing each neuron, and edges a binary variable which is 1 if the correlation coefficient between the time series exceeds the x^{th} percentile of the distribution of correlation coefficients between different neurons (i.e. you are choosing the threshold to have (100 – *x*)% sparsity of connections). Choose *x* to be sufficiently high (i.e. the network sparsely connected) such that you can handle the number of edges on your computer, but sufficiently low to allow you to study the properties of the network. For example, if you were using the 2167 neurons in channel 1, a threshold at 98% would yield 93,000 edges (2167 * 2167 * 0.02). Do not allow self-edges. (a) Plot the adjacency matrix for the graph as an image, with white indicating no edge and black indicating an edge. (b) Also plot it using another graph visualisation method of your choice.

[20 marks]

5. What is the average degree $\overline{k}$ for the network described by the above adjacency matrix? What is the average clustering coefficient? What is the global clustering coefficient? Are these high compared to what you might expect for a random network? What is the network diameter D? (hint: is the network fully connected?) Would you describe this network as “Small World”?

[20 marks]

6. Plot the degree distribution for the network. Would you describe the network as “Scale-free”? Compare it to the degree distribution predicted by an Erdös-Renyi network of the same mean degree.

[20 marks]

Simon Schultz
02/12/2022