

PROJECT

Dynamics, functional connectivity and simulations grounded on experimental data

(GEPHI is a recommended software. Exercise 4 requires MATLAB/OCTAVE)

Exercise 1

You will get 5 datasets (package `SPONT_DATASETS.ZIP`) of spontaneous activity in neuronal cultures, i.e. neurons dissociated and grown on a glass substrate. Each dataset corresponds to a particular design, including homogeneous networks, spontaneously aggregated networks, and engineered networks.

The datasets contain the fluorescence traces of neurons as a function of time. The first column in the dataset is time, and the remaining columns is the fluorescence trace for each neuron. Although the spatial arrangement of the neurons is known by the experimentalist, this information is hidden to have more fun.

A) In this exercise, you should first plot the fluorescence traces and get a feeling of what information they provide. Then, you should create some sort of simple algorithm to detect the firing events. For instance, you can use a simple threshold method: a spike occurs at time t whenever the fluorescence trace crosses a preset value at this time.

B) At the end of the exercise you should construct the raster plot (train of spikes for each neuron as a function of time) for each dataset. The above inference method is known as ‘Schimdt trigger’. There are many others in the literature, such as ‘peeling’ or ‘foopsi’, which take into account the typical amplitude of a firing event as well as the fast rise and slow decay of the fluorescence signal, to extract the train of spikes. These methods are not easy at all to implement. You should be aware of the enormous complexity of inferring the precise spike trains from real, noisy data. Spike inference is still an open problem in neuroscience.

Exercise 2

Here you will use the generated raster plots to start characterizing the data.

A) For each dataset, you should compute the cross-correlation (CC) among all pairs of neurons. This provides a first approach to ‘functional connectivity’, procuring an undirected but weighted network, with the weight proportional to the correlation coefficient. Represent the connectivity matrices as a color map.

B) Now, you can use the function `modularity_dir` from the ‘brain connectivity toolbox’ (see e.g. <https://sites.google.com/site/bctnet/>) to order the matrix according to functional communities. You can also extract the community statistic Q . The larger the Q , the more segregated the system (higher number of communities).

C) Plot and classify the connectivity matrices according to Q .

D) Now look at the location of the neurons and the experimental details. Can you find any relation between the results of Q (or the functional connectivity) and the experimental details?

E) Optional: Study the ‘functional complexity’ (Zamora-López et al., Sci Rep 6, 38424, 2016) or the ‘dynamical richness’ (Yamamoto et al., Sci Adv 4, 4914, 2018) concepts. Use the computed cross-correlation values to extract the value of complexity Θ for the different sets.

Exercise 3

An interesting discussion in modern neuroscience is what functional connectivity inference algorithm is the best. Cross-correlation is fast and intuitive, but does not inform about the temporal directionality of interactions. Information-theoretic approaches are very powerful, but are less intuitive and need strong computer power.

Thus, when learning computational neuroscience, a thrilling exercise is to build your own functional connectivity algorithm. Here I propose to use the time delays concept. Time delays is appropriate when experimental data was acquired at high speed, for instance on the order of 100 images per second or higher in fluorescence imaging. Using time delays, one assumes that the propagation speed along axons is much faster than the integration time in the neurons. Thus, one could say that two neurons firing very close in time are directly connected. Or, more generally, the shorter the time delay among any two neuron pairs, the stronger their coupling.

A) Consider the data in the **CLUSTERED_CULTURE.ZIP** package. It contains a raster plot, the list of neurons (which are actually neuronal aggregates), an image of the culture, and a video. Plot the raster, compute the cross-correlation and generate a connectivity matrix for the functional connectivity. Order it according to communities. Compare what you see in the matrix with the video.

B) Develop by yourself a time delays approach. I recommend an approach as follows. First, identify the groups of coordinated activations, i.e. those neurons that fire together in a time window. Second, for each coordinated activation, compute the time delays among the participating neurons, and normalize it to convert time to functional weights. And, third, average the weights over all observed events of coordinated activity in which a neuron participated. Finally, build the new functional connectivity matrix.

C) Use the provided images to extract the structural network, understood as those pairs of neuronal aggregates that share a bundle of axons. Finally, compare structural and functional connectivity. Anything surprising? Can you extract any conclusions?

Exercise 4 [REQUIRES MATLAB]

A fascinating aspect of biological, living neuronal networks is that they are grown on a physical space, for instance the glass substrate in a neuronal culture. Since axons are energetically costly, the probability of a neuron to connect with another one in a culture decreases with distance. As observed by Hernández-Navarro et al. (Phys Rev Lett 118, 208101, 2017), one can consider two major factors governing network structure and dynamics, namely the aggregation of the neurons and the correlation length.

A) Analyze the article to become aware that nearby neurons will easily connect to one another and form strong communities, and that these communities may become irrelevant if typical axonal length is similar to the culture’s size.

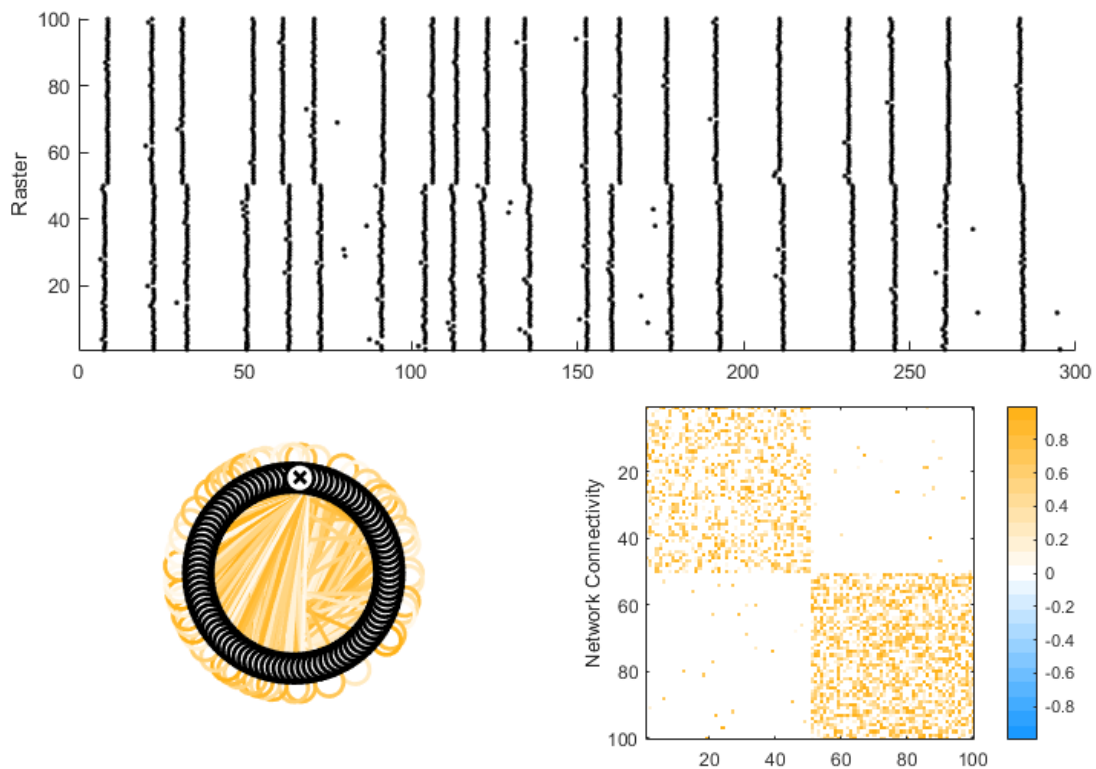
B) Consider $N=100$ neurons and place them on a square substrate. Use your imagination to design different levels of aggregation, but try to build a network in which high aggregation corresponds to about 10 neuronal aggregates. Finally, connect the neurons with a rule such that the shorter the distance among neurons, the higher the probability to be connected.

C) Play a little bit the software **SimLIFNet** and the examples provided in **Simul_MAIN**. The codes run in MATLAB/OCTAVE, and are provided in the package **SIMUL_NET.ZIP**. Be sure that you start with very few neurons (e.g. 10-20), and that you understand the importance of noise to drive activity.

D) Use the software **SimLIFNet** and **Simul_MAIN** (in MATLAB/OCTAVE) to simulate the dynamics of the network. Play with different aggregations and connectivity lengths to understand the impact of both actors in shaping a rich dynamic behavior.

E) Use cross-correlation, or any other desired approach, to compute the functional connectivity of the network and compare it with the structural one.

F) Optional: Play with the parameters of the simulation to be aware of the sensitivity of network dynamics to external drive and noise. Overall, think about the fact that small variations in dynamical traits have big changes in functional connectivity.



(Screen capture in MATLAB. It may differ in OCTAVE)