ANEIS hands-on

In vitro MEA data analysis: population dynamics characterization

Adam Armada-Moreira and Alessio Di Clemente

Multi-electrode arrays (MEA) have become a standard laboratory tool to study population-wide electrophysiology. However, the amount of data generated by this technique requires a specialized pipeline to extract relevant biological meaning.

In this hands-on, we will go over said pipeline, including spike detection, data visualization, extracting overall population-wide parameters, population burst analysis, and spatial analysis of activity. The different steps will be implemented in MATLAB, where the students will have the chance to create their own functions based on the theoretical rationale behind each step, or study already built functions that will be provided.

Goals: To expose the students to the logic behind MEA data analysis, so they can independently implement this pipeline with their data. Familiarize students with programming environments and code organization. Understanding what biological information can be extracted from each analysis.

Pre-requisites

- Install MATLAB, including the following toolboxes:
    - Signal Processing Toolbox
    - Statistics and Machine Learning Toolbox
- MATLAB Add-ons:
    - matlab-toml by George Kaplan
    - peaks2 by Kristupas Tikuišis
- Download hands-on materials

Session 1

Spike detection

To perform spike detection on MEA data, four main steps are needed:

- **Bandpassing the raw data**

While some MEA systems acquire only action potential (AP) data already filtered, other systems offer the capability to record both AP and local field potentials (LFP). In order to analyze and extract information from these data, the two signals must be separated. This can be done by bandpassing the raw data, since it is known that APs "are around" 1000 Hz while LFP occupy low frequency bands (below 100 Hz). Thus, to isolate AP, it is common practive to use a bandpass filter (meaning it keeps only the information happening inside the chosen frequency range) between 300 and 3000 Hz.

- **Determining the baseline**

Most methods for spike detection rely on threshold crossing. This threshold needs to be previously defined and will determine what are noisy signal fluctuations and what are significant changes (i.e., action potentials). As every step in data analysis, there are a myriad of methods to determine this threshold. One of the most robust methods is based on the estimation of baseline noise,[1] where the threshold is defined as a X times the baseline noise (X is generally between 3 - 5, depending on data quality and signal to noise ratio):

$$threshold = N \times \frac{median(|signal|)}{0.6745}$$

- **Peak detection**

Inside each block of points crossing the threshold, we find the peak and save the timestamp.

- **Spike time export**

As a final step, all the spike times need to be exported for further analysis. As a rule, there are some key parameters and information that need to be outputted:

- Spike times (normally it’s more robust to export in msec)
- Electrode ID for each detected spike
- Sampling rate of the recording
- Total duration of the recording (again, be mindful of the units used!)
- (optional) Polarity of the detected spike (since we can detect both positive and negative spikes, it can be useful to have this information for further analysis)

In our lab, we use an in-house develop toolbox written in Julia to do spike detection. It is divided in two components: **SpiQ** (https://github.com/mgiugliano/SpiQ) which contains the main scripts that can be used to access the core functions that do the work. In the repository you can find all the information on installation, usage, and output file format that it uses. **SpQEphysTools.jl** (https://github.com/mgiugliano/SpQEphysTools.jl) which is available as a Julia package directly from the Julia registry and is the backbone of the spike detection analysis.

Do it yourself!

Using the provided **loadData**() function, import the dataset into MATLAB and build a raster plot (where every action potential is represented as a dot, with time in the x-axis and electrode number in the y-axis).

Identifying inactive channels

Since neurons plated on MEAs form random networks, it is to be expected that their distribution across the electrodes is not uniform. This means that, for some cultures, we will find "inactive" electrodes, i.e., locations where neurons are not present and therefore no spikes are detected. As a rule of thumb, to be considered active, an electrode must register a firing rate > 0.02 Hz (which is an approximation of at least 1 spike per minute).

Do it yourself!

Find the number of active electrodes

Spike time histogram

Other than a raster plot, a common way to represent population-wide activity is the Spike Time Histogram (STH). To calculate this, the spike time data is binned in small intervals, where the population-wide firing rate is calculated. This allows us to have more quantifiable information about the network activity throughout the recording. Creating the STH involves an informed parameter and method choice:

- Bin size: The small intervals in which the spike times will be divided needs to be actively chosen. In general, this parameter is set in the range 1-10 msec, depending on the information you want to extract from the data.

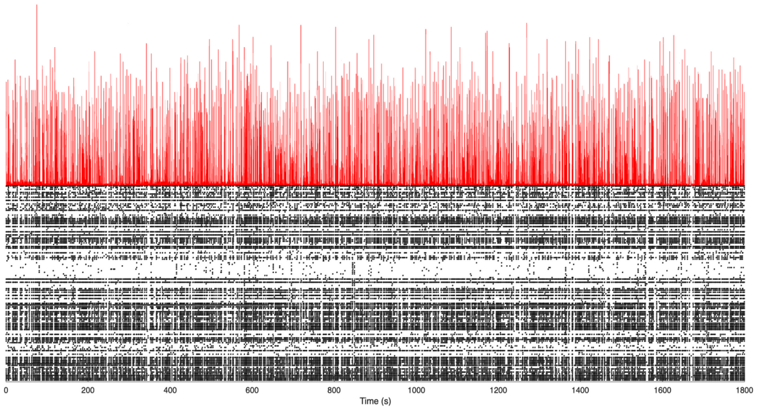
- Method: There are many ways to represent the STH. The simplest and more direct approach is to consider "spike counts", meaning how many spikes occur inside each bin. Next in complexity, there is "firing rate", in which the total number of spikes in each bin is divided by the duration of the bin, resulting in a representation of spikes/s or spikes/ms, depending on the units. Finally, it is also possible to normalize this firing rate, taking into account the total number of active electrodes, resulting in a "normalized firing rate".

Do it yourself!

Using the provided function **makePopSTH**(), try the different methods and plot their output.

(optional) Do it yourself!

Create a combined figure, containing the population STH stacked above the raster plot, similar to the following example:



Overall network parameters

With the spike time data we have so far, we can already calculate some some network-wide important parameters.

The first one is the **inter-spike interval (ISI)** - the time difference between sequential spikes detected in the same neuron or, in the case of standard MEAs where each electrode can record multi-unit activity, from a group of neurons captured in a single electrode. From all the calculated ISI, we can compute their distribution, the mean ISI, standard deviation, etc., which will give us an idea of the overall network connectivity and maturity.

Do it yourself!

Use the function **computeISI**() to get the inter-spike interval data. Alternately, try to write the code that computes the ISI yourself.

Plot the ISI distribution and compute the main statistics.

As a more detailed characterization of the ISI distribution, we can also identify the main peaks that account for the majority of the detected spikes. These regions and their relative intensity can provide us with a rough idea of population bursts and their prevalence, which we will explore in detail in Session 2.

Do it yourself!

Using the MATLAB built-in function **findpeaks**(), identify the most probable ISI times and estimate the fraction of spikes accounted for in those ranges (tip: use the peak half-height as border).

The second network parameter is the **population firing rate**. There are many methods to calculate this parameter:

- Directly, by dividing the total number of spikes by the recording duration;
- As a mean of each electrode's firing rate;
- Using the previously computed STH;
- As the inverse of the ISI.

While it is already informative to represent the population firing rate without any normalization, if we want to compare activity levels among different plated cultures, we need to account for the number of active electrodes. Thus, the firing rate should be reported as number of spikes per unit of time per active electrode.

Do it yourself!

Explore different methods to extract the population-wide FR, looking into possible differences and more detailed information each can give us.

Session 2

Population burst detection and basic characterization

Population bursts, also known as network bursts, are epochs of synchronous activity across several electrodes. They can be triggered by external stimulation but they also arise spontaneously in cultured neuronal networks. They originate from the presence of recurrent excitatory connections causing more and more excitatory neurons to be recruited into the ongoing burst. This mechanism alone would cause overall network activity to skyrocket. This does not happen because of two main mechanisms: inhibitory feedback connections and synaptic fatigue. The inhibitory drive increases with the level of activity due to the excitatory input to inhibitory neurons. This slows down network activity in a feedback loop. Moreover, synaptic fatigue will cause synapses to get depleted, further slowing down the overall activity. The interplay between these three main factors shapes the burst. As a consequence, by analyzing the population burst occurrences and dynamics, we can derive important information on the properties of the network.[2–4]

The first step in the process is, of course, identifying the population bursts. This can be done with different algorithms starting from the STH or the ISIs. In our lab, we apply a double threshold crossing strategy starting from the STH. Threshold are defined based on the number of active electrodes, the higher one to define the peaks, the lower one to identify start and end of the bursts.

Do it yourself!

Using the provided function **detectBursts**() identify and perform a basic characterization of population bursts.

Using the provided function **burstprofiling**() plot profiles of the bursts (i.e. their STH) aligned at their start and/or at their peak

Try to plot the STH highlighting burst starts, burst ends and burst peaks identified by **detectBursts**()

Extract population stats from the bursts table coming from **detectBursts**(), compare parameters between different recordings (e.g. burst duration, Inter Burst Intervals)

Session 3

Burst internal dynamics characterization

Some parameters (like burst duration, amplitude, time to peak…) can be immediately quantified once the bursts are identified. While these parameters can be informative, further analysis can allow us to get a deeper insight into network properties. In particular, the onset phase of the burst carries information about the efficiency of the recurrent excitation,[5,6] while intra-burst oscillations can offer insights into the interplay between excitatory and inhibitory compartments.[7–10]

Burst Onset Slope

The slope of the onset phase (i.e. how "steeply" the network activity raises) can be measured in absolute terms or in relative terms by normalizing each burst by its peak. While both ways are informative, the latter is better suited to compare different cultures.

Measuring the onset slope requires, first of all, to smooth the burst profiles to reduce the "noise".

> ***Do it yourself!***
>
> Using **smoothdata**() function try different strategies to smooth burst profiles (moving mean/median, local regression with lowest/loess) and plot the results
>
> Try to smooth burst profiles with **sgolayfilt**() function (allows more control over fitting parameters and employs Savitzky-Golay model)

Once we have the smoothed burst profiles, measuring the onset slope simply means taking the first order derivative of the onset phase (i.e. from beginning to peak).

> ***Do it yourself!***
>
> Using the provided function **getRiseSlopes**() extract onset slopes from each smoothed burst with and without normalization
>
> Extract maximum values of onset slopes for each burst and compare them between different recordings
>
> (optional) plot burst onset profile, derivative and max slope (this might be long and tedious, we can provide the code instead)

Intra-Burst Oscillation

As mentioned above, intra-burst oscillations in population firing frequency arise from the interplay between excitatory and inhibitory compartments. Quantifying this parameter means performing a spectral analysis on these oscillations to obtain a spectrogram and a dominant frequency of oscillation for each burst (another possible strategy is to perform the same analysis on the mean burst profiles but it requires larger datasets).

The first step into this kind of analysis is the extraction of the fast (i.e. ~10-200Hz) oscillations from the burst profile by removing the "slow" rise and decay. This can be done with different strategies, from filtering to subtraction of the smoothed profile from the original burst. We currently employ a surrogate spike-train jittering approach to obtain a copy of the burst containing only slow components and we isolate the fast ones by subtraction.

> ***Do it yourself!***
>
> Using the provided function **getFastandSlow**(), separate fast and slow components.
>
> Plot the original signal, slow, and fast components of each burst

Once fast oscillations have been isolated, we can apply short-time fast Fourier transform (STFT) to get spectrograms and power spectrum density (PSD) of each burst. Identifying the dominant frequency will then be just finding the max of the PSD.

> ***Do it yourself!***
>
> Using **spectrogram**() function, extract power x time window x frequency for one burst
>
> Extract PSD by averaging power for each frequency across time windows
>
> Write a loop to do the same for each burst in the recording
>
> Find dominant frequencies in different recordings and compare them

Session 4

MEA spatial analysis

So far, the performed analyses inform us about the levels of activity and synchronicity of the neuronal network. However, we can extract further information if we take into account the spatial organization of the MEA electrodes.

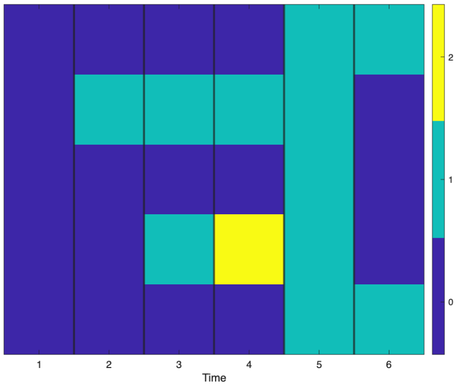
To start understanding the logic behind spatial analysis, we can plot the overall firing rate of each electrode taking into account their spatial coordinates.

> ***Do it yourself!***
>
> Use the provided function **get_electrode_xy**() to generate a spatial map of each electrode's mean FR. Plot the output frame-like matrix where each cell represents the XY coordinates of each electrode and is populated with the respective FR.

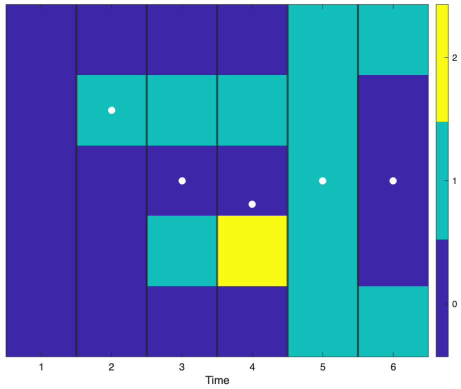
While this example considers only a static spatial analysis, we can apply the same principle to identify burst initiation locations and get a better insight into activity propagation within the network.

To do so, we will explore a spatial activity analysis method: **Center of Activity Trajectory** (CAT).[11] CAT gives us a sequence of Centers of Activity (CA) in time bins across a chosen time period. The CA is equivalent to a center of mass, where "mass" is the density of spikes across the spatial arrangement of the electrodes. For more detailed information on CAT analysis, check [12].

Let's consider an example where we have 5 vertically arranged electrodes, where spikes were detected in 6 time bins:



If we calculate the CA (or "center of mass") for each time bin, we get:



The CA is calculated by multiplying the STH by the spatial location of the electrodes and dividing the sum by the total number of spikes detected in that time bin:

```
STH_frame = [1 0 0 0 1]
spatialLocation = [1 2 3 4 5]
weightedFrame = STH_frame .* spatialLocation
weightedFrame = [1 0 0 0 5]
CA = sum(weightedFrame) / sum(STH_frame)
CA = 6 / 2
CA = 3
```

Do it yourself!

Using the provided function **computeCAT**(), apply this analysis to one detected burst and plot the trajectory in time.

Programmatically run and store in a variable the data of CAT for all bursts.

Using the analysis output, plot all the start and peak locations.

Identify the most common initiation regions, based on the initiation location density distribution (Tips: Use **ksdensity**() to generate the 2D density map. Use **multithresh**() and **peaks2**() to identify the most common regions)

Based on the CAT information, identify the main regions where population bursts initiate.

Separate the bursts according to their spatial origin and plot their parameters accordingly.

Apply the CAT analysis to burst decay.

Explore the script **extra_ActivityVideo**.

References

1. Quiroga, R. Q., Nadasdy, Z. & Ben-Shaul, Y. Unsupervised Spike Detection and Sorting with Wavelets and Superparamagnetic Clustering. Neural Computation 16, (2004).
2. Van Pelt, J., Corner, M. A., Wolters, P. S., Rutten, W. L. C. & Ramakers, G. J. A. Longterm stability and developmental changes in spontaneous network burst firing patterns in dissociated rat cerebral cortex cell cultures on multielectrode arrays. Neuroscience Letters 361, (2004).
3. Lonardoni, D. et al.. Recurrently connected and localized neuronal communities initiate coordinated spontaneous activity in neuronal networks. PLOS Computational Biology 13, (2017).
4. Van Pelt, J., Wolters, P. S., Corner, M. A., Rutten, W. L. C. & Ramakers, G. J. A. Long-Term Characterization of Firing Dynamics of Spontaneous Bursts in Cultured Neural Networks. IEEE Transactions on Biomedical Engineering 51, (2004).
5. Gigante, G., Deco, G., Marom, S. & Del Giudice, P. Network Events on Multiple Space and Time Scales in Cultured Neural Networks and in a Stochastic Rate Model. PLOS Computational Biology 11, (2015).
6. Rubin, J. E., Hayes, J. A., Mendenhall, J. L. & Del Negro, C. A. Calcium-activated nonspecific cation current and synaptic depression promote network-dependent burst oscillations. Proceedings of the National Academy of Sciences 106, (2009).
7. Atallah, B. V. & Scanziani, M. Instantaneous modulation of gamma oscillation frequency by balancing excitation with inhibition.. Neuron 62, 566–77 (2009).
8. Buzsáki, G. & Wang, X. J. Mechanisms of gamma oscillations.. Annu Rev Neurosci 35, 203–25 (2012).
9. Brunel, N. & Wang, X. J. What determines the frequency of fast network oscillations with irregular neural discharges? I. Synaptic dynamics and excitation-inhibition balance.. J Neurophysiol 90, 415–30 (2003).
10. Pulizzi, R. et al.. Brief wide-field photostimuli evoke and modulate oscillatory reverberating activity in cortical networks.. Sci Rep 6, 24701 (2016).
11. Chao, Z. C., Bakkum, D. J. & Potter, S. M. Region-specific network plasticity in simulated and living cortical networks: comparison of the center of activity trajectory (CAT) with other statistics. Journal of Neural Engineering 4, (2007).
12. Gandolfo, M., Maccione, A., Tedesco, M., Martinoia, S. & Berdondini, L. Tracking burst patterns in hippocampal cultures with high-density CMOS-MEAs. Journal of Neural Engineering 7, (2010).