**Guide for spike analysis**

**Data analysis pipeline**

**Copy data**

1. Copy the data folder from the recording computer to the server (Open Ephys\Open Ephys Data).

**Preprocess data for Kilosort**

1. Open the Open Ephys\Open\_Ephys\_Data\_Analysis forder in Matlab
2. In load\_command\_KiloSort\_A4.m, change the variables:

* experimentName (YYYY-MM-DD\_HH-MM-SS, according to the folder where the data was saved)
* sessionName (V1\_YYYYMMDD\_i, where i is the number of the experiment on that day(1, 2, 3 etc)). If the recording is not in V1, this part of the name can be exchanged accordingly. Run the first section ONLY – it will create new folders (‘data’, ‘matlab analysis’ and ‘kilosort analysis’) and move the data files into the folder called ‘data’ within the YYYY-MM-DD\_HH-MM-SS folder

1. In the 2nd section insert the new parameters:

* recordingDepth of each electrode shank (first shank first) as a negative number
* channelNo: total channels
* probe: select between '2x16\_E1', '1x16\_P1', '2x32\_H6'
* animal.name: ‘YYYYMMDD\_LV1’. Instead of LV1 there can be RV1 or what ever region is recorded. The date represents when the animal was injected
* animal-sex: ‘f’ or ‘m’
* animal.strain: ‘Gad2Cre’, ‘NexCre’, ‘PvCre’, ‘Bl6’ etc
* animal.virus: for ex ‘AAv9-mOp2A’
* chOffset = 0 for 16- and 64-channel probes and 16 for 32-channel probes
* conditionNames: adjust accroding to the stim IDs used in the recording
* trialDuration, preTrialTime, visStim, optStimInterval, visStimDuration (all in seconds) according to those used on Master8
* Run the section

1. Two figures and an information are displayed:

* the first shows the recording over time – are there any big variations over time?
* The 2nd shows median and STD of each channel
* Channels with an outlier STD (mean +/- 2\*STD) are printed in the Command Window. Do they match with the plot in the 2nd figure? – you might want to exclude them later

1. In the 3rd section, modify selCh if necessary (pick a channel with an average median and std from the displayed figure and add chOffset) and run the section
2. A new figure is displayed. It contains:

* the recorded signal (raw, high-pass @150 Hz or bandpass @0.6-6kHz) – first subtrial (it can be changed by pressing Next/ Prev. or entering a number and pressing Return). Look at the subtrials with an outlier STD or Max (see below)
* STD of the signals above and the absolute(Max) of the high-pass @150 Hz signal, across all subtrials
* Decide based on std and max which subtrials should be excluded. You want to exclude as few as possible (preferably none), but you also don’t want to have noisy trials.

1. In the 4th section type in the exclude variable the subtrials to be excluded. You can also leave this variable empty. Run the section. This will calculate which other subtrials need to be excluded, to ensure there are equal numbers of all conditions
2. Run the 5th section. This does the following:

* saves the **sessionInfo.mat** and **timeSeries.mat** structures in the ‘matlab analysis’ folder
* exports the file as a binary file for kilosort analysis to the ‘kilosort analysis’ folder
* adds the experiment to the allExp.mat structure

**Run Kilosort and Phy2 analysis**

1. Type kilosort (after having added its path (/local/ruxandra/Kilosort2-master/) upon first use ever)
2. Fill in the fields:

* copy the file binary .dat file name and delete/autofill the folder names. The folder names are the folders containing the .dat file
* Select probe layout and number of channels according to what silicone probe was used
* Sampling freq = 20000
* Time range (s) = 0 Inf
* min firing rate per channel = 0
* Threshold = 10 6
* Lambda = 10;
* AUC for splits = 0.9

1. Before clicking ‘Run All’, run the first lines in SpikeDataLoading\_openEphys\_KiloSort\_A1.m : Clear variables (clearvars -except experimentName sessionName), set the ks parameters (ks= get(gcf, 'UserData'); ks.ops.fshigh = 300; ks.ops.Th = [10 6])
2. Press ‘Run All’ in kilosort
3. Make sure there are no errors prompted by kilosort and wait until ‘Done’ is prompted. If there are errors, it is probably because no spikes are detected in the first trials of the recording. You can try to check which trials are creating problems by excluding them from the Time range (s) in Kilosort. For example, if each trial has 10 conditions of 11 second, type ‘110 Inf’ in Time range (s). If this still creates problems, try a multiple of that specific number, as if excluding 2 or 3 trials. If this solved the problem, you have to delete all files created by this analysis in the ‘matlab analysis’ and ‘kilosort analysis’ and redo the entire analysis till here, by adjusting the variable exclude in the 4th section in load\_command\_KiloSort\_A4.m to also contain the problematic subtrials.
4. Open the Anaconda Prompt (windows) or Terminal (Linux) in the ‘kilosort analysis’ folder
5. Type „activate phy2” (Windows) or „source activate phy2“ (Linux)
6. Type „phy template-gui params.py“.
7. Order by ContamPct. All **good** units should have a ContamPct < 20. Additionally some **mua** units might have a ContamPct < 20. They could be considered later good units if the waveform is similar to that of a good unit (gradient of amplitude across channels)
8. For now, mark all mua units with ContamPct > 20 as mua (Edit/Move best to mua or Alt+M). You can select a few of them and mark the simultaneously, to speed up the process, but don’t select **all** of them at once
9. In the SpikeDataLoading\_openEphys\_KiloSort\_A1.m script in Matlab run the 1st, 2nd and 3rd sections
10. Open the PlotPSTHandRaster\_openEphys\_KiloSort\_A3.m script in Matlab, Run the 1st and 2nd section to get the time courses of good units
11. If you want to display the time course of a specific mua, go to the 5th section of PlotPSTHandRaster\_openEphys\_KiloSort\_A3.m and modify the first 4 lines: comment out the first, uncomment the next 3 and modify unitIDs to include the MUA of interest. Run the section
12. Use the created figures (together with step 13 and 14) to help decide if a cluster is noise (heartbeat? Artefact? Spike count < 200?) or if two clusters are double counted (large bin in the middle of the CCG) or should be merged.
13. Select the first good unit and then press space. This will compare the good unit with the most similar unit. If their waveform, waveform location, ACGs and CCG, FiringRateView and traces look similar, they might be the same unit. You can compare waveforms by clicking in WaveformView / Toggle Waveform overlap. You can merge them with Edit/Merge or pressing G
14. If you think clusters contain noise, in Phy, go to CorrelogramView, Set window to 1000 and go through the clusters and detect heartbeat noise (6-8 Hz frequency, appears in all channels) or other types of noise (optoelectrical artefacts). Then reset window to 50 and Set bin to 0.2 (ms)
15. After comparing this unit with all the others, and if its timecourse displayed in Matlab doesn’t look like noise, classify this as a good unit Edit/Move best to good or Alt+G) and proceed to the next unit
16. After classifying all units, click File/Save in Phy and type close all in Matlab

**Run matlab analysis**

1. Run the 1st section in the SpikeDataLoading\_openEphys\_KiloSort\_A1.m script
2. The variable spikeClusterData.uniqueCodes opens – the first column represent unit IDs and the 2nd column represents channel numbers. Sometimes, the wrong channel number is assigned to a unit ID. Double-check at least for the good clusters the channel number (from Phy) with the number in the 2nd column; make sure the channel number in Phy is correct (compare to the waveform view), sometimes it is displayed wrong. If you find discrepancies, update the 2nd column in spikeClusterData.uniqueCodes accordingly
3. Run the other sections till the end of the script
4. In the PlotPSTHandRaster\_openEphys\_KiloSort\_A3.m script in Matlab, run the first 2 sections
5. In the displayed figures, select (click the radio button) the Good/SUA clusters with visual responses (VisEv), no visual response but optogenetic effect on spontaneous activity (Spont) or no response/effect (none)
6. Run sections 3 and 4
7. Run section 5 (Mua analysis) and select in each figure the Mua clusters with good visual response (VisEv) or no/poor visual response (none)
8. Run sections 6 and 7
9. the excel variable opens. Copy these numbers to the excel file containing data about the experiments (Elphys experiments conclusions\_new.ods, for example)
10. Run section 8 – it displays average of good units and of Mua
11. Open waveformAnalysis\_openEphys\_KiloSort\_A4m, run the first section and make sure the waveforms look like proper spikes and the amplitude distribution is uni-modal
12. The inverted spikes are displayed in the command window. These are spikes which look upside down. The script automatically inverts them back. Look in the Matlab figures and in Phy for these spikes. If you think they are just shifted in time and not inverted, modify line 102 (shiftCodes) to include the unit ID which you think is just shifted and not inverted. This waveform will be shifted according to line 115: waveformTimes = waveformTimes + 4/samplingRate; Run again section 1.
13. Run section 2 – it displays figures with various indices, Interspike interval histograms (50 ms and 200 ms) and FFT of the spike distribution (interneurons have often a higher power at 30-40 Hz)
14. Run section 3 – displays figures with waveforms, normalized waveforms and 2D plots of various feature pairs
15. Run section 4 – here the detected connections are displayed. If some connections are false, rerun the section after modifying the first two lines
16. Run section 5 – it saves all the data
17. For the orientation protocol, run the oriAnalysis.m or oriAnalysis\_gaussian.m script (might still be under development, as we didn’t proceed further with this analysis)

Which script creates which structure?

load\_command\_KiloSort\_A1.m : **sessionInfo.mat**, **timeSeries.mat**

SpikeDataLoading\_openEphys\_KiloSort\_A1.m : **spikeClusterData.mat**

PlotPSTHandRaster\_openEphys\_KiloSort\_A1.m : **clusterTimeSeries.mat**

waveformAnalysis\_openEphys\_KiloSort\_A1.m: **cellMetrics.mat**

Check the existence of the respective file after each running each script.