

# Exercise K

## Spike sorting

The goal of the exercise is to learn how to separate signals from several neurons recorded by an extracellular single-tip or multipoint electrode. It is proposed to use for spike sorting public-domain “WaveClus” shown to be superior in most cases at least for single-channel records. WaveClus is based on wavelet decomposition. Thus, it is especially suitable for sorting time-dependent signals. A more general method of object sorting based on principle components is realized in KlustaKwik. It is the most cited method up to now, but the recent study (Wild2012(spike sorting comparison).pdf) has shown that WaveClus outperforms it at least in case of single-channel records and high or moderate noise levels.

Taking into account noted above conclusions, we decided to try WaveClus first.

In the current exercise you have to sort spikes of neurons obtained by a 16-channel electrode with linear distribution of recording pads along the shank with 50  $\mu\text{m}$  span. Thus, a neuron typically generated spikes visible in two or three nearest channels. This can help separate these signals. The sampling rate was 25 kHz and duration of the given episode is 1 minute. The Matlab file size is about 85 MB. Signals were recorded from the robust nucleus of arcopallium in the anesthetized zebra finch. This nucleus is a part of a vocal pathway carrying signals from the brain to the syrinx and respiration-controlling muscles.

The recommended sequence of actions is the following:

1. Download WaveClus from Basics of Instrumentation web site. Unzip it and place its path including subdirectories in the Matlab “Path” (in menu “File”).
2. Load synthesized dataset that comes with WaveClus (in directory Sample\_data, file test.mat) and sort spikes there to become familiar with the program. Read wave\_clus\_intro.pdf in case of questions.
3. Load one channel with the largest spikes (#8) from the given dataset RA\_16ch\_25kHz\_1min.mat. Sort spikes in this channel. Do not forget to change sampling rate. Increase the maximal allowed spike size, as one neuron gives especially large spikes. Note that WaveClus assumes that spikes go up, and in our dataset they go down. Thus, invert the dataset before importing. Try to find parameters that give nice spike separation.
4. Select four sequential channels that have the largest spikes. Import them in WaveClus. To do this follow description on page 3 of ‘wave\_clus\_intro.pdf’. Try to sort spikes. How many neurons can you separate? Unfortunately, the data import description in ‘wave\_clus\_intro.pdf’ is too brief. To import multichannel data in WaveClus you have to do two things: **1.** Create a set of Matlab files, one file per channel, in which a channel will be saved in variable ‘data’. I.e., if you have 16 channels, 16 files should be created. **2.** Create one text file in which names of all created Matlab files will be listed one file per line. This text file should have name ‘polytrode1.txt’. These two operations are coded in Matlab script ‘data\_convert.m’ that you can download from our website to save your time. Then you can detect spikes running ‘Get\_spikes\_pol(1)’, and cluster spikes running ‘Do\_clustering\_pol(1)’. However, most probably clustering results will be unsatisfactory. Thus, you have to load the generated file ‘times\_polytrode1.mat’ in WaveClus using the menu option ASCII (pre-clustered). There you can see the results of clustering and can adjust parameters of sorting.
5. If performance of your computer is sufficient, try to sort spikes in the whole dataset, perhaps, excluding the first three channels that don’t have visible spikes. How many neurons can be separated? Please save all nice pictures that you get.

If you have time, you can try to sort spikes in the same dataset by KlustaKwik. To do this, download the file KlustaKwikExample.zip and unpack it. Copy the original data file RA\_16ch\_25kHz\_1min.mat to the unpacked directory with the scripts. To detect spikes run GetSpikeShapes.m. To extract features from these shapes for sorting run GetFeatures.m. To sort spikes run SortSpikes.m. The last script calls KlustaKwik executable. Scripts plot intermediate results that may be interesting to see. An addition interesting feature of these scripts is that they use GPU for filtering if possible. This accelerates this operation 10-50 times against CPU filtering.