This is a distillation of notes on use of spyKING CIRCUS

Command-line Parameters

<https://spyking-circus.readthedocs.io/en/latest/code/parameters.html>

# Display the helpers

To know what are all the parameters of the software, just do:

>> spyking-circus -h

To know what are all the file formats supported by the software, just do:

>> spyking-circus help -i

To know more what are the parameter of a given file format X, just do

>> spyking-circus X -i

# Command line Parameters

The parameters to launch the program are:

-m or --method

What are the steps of the algorithm you would like to perform?

Defaults steps are:

1. filtering
2. whitening
3. clustering
4. fitting
5. merging

Note that filtering is performed only once, and if the code is relaunched on the same data, a flag in the parameter file will prevent the code to filter twice.

You can specify only a subset of steps by doing:

>> spyking-circus path/mydata.extension -m clustering,fitting

Note

The results of the merging step are still saved with a different extension compared to the full results of the algorithm. This is because we don’t claim that a full automation of the software can work out of the box for all dataset, areas, species, … So if you want to work from merged results, use the -e merged extension while converting/displaying results. But otherwise, just look to the ra results, without the merging step (see the devoted section documentation on Meta Merging), or even more (documentation on extra steps).

-c or --cpu

The number of CPU that will be used by the code. For example, just do:

>> spyking-circus path/mydata.extension -m clustering,fitting -c 10

-H or --hostfile

The CPUs used depends on your MPI configuration. If you wan to configure them, you must provide a specific hostfile and do:

>> spyking-circus path/mydata.extension -c 10 -H nodes.hosts

To know more about the host file, see the MPI section documentation on MPI

-b or --batch

The code can accept a text file with several commands that will be executed one after the other, in a batch mode. This is interesting for processing several datasets in a row. An example of such a text file commands.txt would simply be:

path/mydata1.extension -c 10

path/mydata2.extension -c 10 -m fitting

path/mydata3.extension -c 10 -m clustering,fitting,converting

Then simply launch the code by doing:

>> spyking-circus commands.txt -b

Warning

When processing files in a batch mode, be sure that the parameters file have been pre-generated. Otherwise, the code will hang asking you to generate them

-p or --preview

To be sure that data are properly loaded before filtering everything on site, the code will load only the first second of the data, computes thresholds, and show you an interactive GUI to visualize everything. Please see the documentation on Python GUI

Note

The preview mode does not modify the data file!

-r or --result

Launch an interactive GUI to show you, superimposed, the activity on your electrodes and the reconstruction provided by the software. This has to be used as a sanity check. Please see the documentation on Python GUI

-s or --second

If the preview mode is activated, by default, it will show the first 2 seconds of the data. But you can specify an offset, in second, with this extra parameter such that the preview mode will display the signal in [second, second+2]

-o or --output

If you want to generate synthetic benchmarks from a dataset that you have already sorted, this allows you, using the benchmarking mode, to produce a new file output based on what type of benchmarks you want to do (see type)

-t or --type

While generating synthetic datasets, you have to chose from one of those three possibilities: fitting, clustering, synchrony. To know more about what those benchmarks are, see the documentation on extra steps

Note

Benchmarks will be better integrated soon into an automatic test suite, use them at your own risks for now. To know more about the additional extra steps, documentation on extra steps

# Preview mode:

<https://spyking-circus.readthedocs.io/en/latest/GUI/python.html>

# Result GUI

In order to quickly visualize the results of the algorithm, and get a qualitative feeling of the reconstruction, you can see use a python GUI, similar to the previous one, showing the filtered traces superimposed with the reconstruction provided by the algorithm. To do so, simply do:

>> spyking-circus path/mydata.extension -r

spyKING CIRCUS: Configuration File (.params) Settings

## Configuration File

The code, when launched for the first time, generates a parameter file. The default template used for the parameter files is the one located in /home/user/spyking-circus/config.params. You can edit it in advance if you are always using the same setup.

Link to configuration file docs:

<https://spyking-circus.readthedocs.io/en/latest/code/config.html>

# Configuration File Sections:

## DATA section:

For opto data, raw\_binary file format will be used

### Warning

This is the most important section, that will allow the code to properly load your data. If not properly filled, then results will be wrong. Note that depending on your file\_format, you may need to add here several parameters, such as sampling\_rate, data\_dtype, … They will be requested if they can not be infered from the header of your data structure. To check if data are properly loaded, consider using [the preview mode](https://spyking-circus.readthedocs.io/en/latest/GUI/python.html) before launching the whole algorithm

Parameters that are most likely to be changed:

Data section parameters for raw\_binary (some are general for all files, others are raw\_binary-specific):

file\_format = Can be raw\_binary, openephys, hdf5, ... See >> spyking-circus help -i for more info

stream\_mode = None None by default. Can be multi-files, or anything depending to the file format

mapping = Mapping of the electrode (.prb file)

suffix = Suffix to add to generated files

overwrite = True If you want to filter or remove artefacts on site. Data are duplicated otherwise

output\_dir = By default, generated data are in the same folder as the data.

parallel\_hdf5 = True Use the parallel HDF5 feature (if available)

RAW\_BINARY parameters (read/parallel write):

sampling\_rate = *<*type 'float'*>* *[\*\** mandatory *\*\*]*

data\_dtype = *<*type 'str'*>* *[\*\** mandatory *\*\*]* should be int16,uint16,float32,...

nb\_channels = <type 'int'> [\*\* mandatory \*\*] number of channels

data\_offset = *<*type 'int'*>* *[*default is 0*]* Optional, if a header with a fixed size is present

dtype\_offset = *<*type 'str'*>* *[*default is auto*]*

gain = *<*type 'int'*>* *[*default is 1*]* Optional, if you want a non unitary gain for the channels

.prb file information:

<https://spyking-circus.readthedocs.io/en/latest/code/probe.html>

## raw\_binary data format:

Link to file format docs:

<https://spyking-circus.readthedocs.io/en/latest/code/fileformat.html>

The simplest file format is the raw\_binary one. Suppose you have N channels

*𝑐*0,*𝑐*1,...,*𝑐𝑁*

And if you assume that *𝑐𝑖*(*𝑡*) is the value of channel *𝑐𝑖* at time t, then your datafile should be a raw file with values

*𝑐*0(0),*𝑐*1(0),...,*𝑐𝑁*(0),*𝑐*0(1),...,*𝑐𝑁*(1),...*𝑐𝑁*(*𝑇*)

This is simply the flatten version of your recordings matrix, with size N x T

Note: The values can be saved in your own format (int16, uint16, int8, float32). You simply need to specify that to the code

file\_format You must select a supported file format (see [What are the supported formats](https://spyking-circus.readthedocs.io/en/latest/code/fileformat.html)) or write your own wrapper (see [Write your own data format](https://spyking-circus.readthedocs.io/en/latest/advanced/datafile.html))

mapping This is the path to your probe mapping (see [How to design a probe file](https://spyking-circus.readthedocs.io/en/latest/code/probe.html))

stream\_mode If streams in you data (could be multi-files, or even in the same file) should be processed together (see [Using multi files](https://spyking-circus.readthedocs.io/en/latest/code/multifiles.html))

overwrite If True, data are overwritten during filtering, assuming the file format has write access. Otherwise, an external raw\_binary file will be created during the filtering step, if any.

output\_dir If you want all the file generated by SpyKING CIRCUS to be in a particular directory, instead of next to the raw data

parallel\_hdf5 Try to use the option for parallel write of HDF5. Need to be configured (see [how to install hdf5](https://spyking-circus.readthedocs.io/en/latest/introduction/hdf5.html))

## Detection

The detection section is:

radius = auto Radius [in um] (if auto, read from the prb file)

N\_t = 5 Width of the templates [in ms]

spike\_thresh = 6 Threshold for spike detection

peaks = negative Can be negative (default), positive or both

dead\_channels = If not empty or specified in the probe, a dictionary {channel\_group : [list\_of\_valid\_ids]}

Parameters that are most likely to be changed:

N\_t The temporal width of the templates. For in vitro data, 5ms seems a good value. For in vivo data, you should rather use 3 or even 2ms

radius The spatial width of the templates. By default, this value is read from the probe file. However, if you want to specify a larger or a smaller value [in um], you can do it here

spike\_thresh The threshold for spike detection. 6-7 are good values

peaks By default, the code detects only negative peaks, but you can search for positive peaks, or both

dead\_channels You can exclude dead channels either directly in the probe file, with the channels list, or with this dead\_channels parameter. To do so, you must enter a dictionary of the following form {channel\_group : [list\_of\_valid\_ids]}

## Filtering

The filtering section is:

cut\_off = 300, auto # Min and Max (auto=nyquist) cut off frequencies for the band pass butterworth filter [Hz]

filter = True # If True, then a low-pass filtering is performed

remove\_median = False # If True, median over all channels is substracted to each channels (movement artefacts)

common\_ground = # If you want to use a particular channel as a reference ground: should be a valid channel number

### Warning

The code performs the filtering of your data writing on the file itself. Therefore, you must have a copy of your raw data elsewhere. Note that as long as your keeping the parameter files, you can relaunch the code safely: the program will not filter multiple times the data, because of the flag filter\_done at the end of the configuration file.

Parameters that are most likely to be changed:

cut\_off The default value of 500Hz has been used in various recordings, but you can change it if needed. You can also specify the upper bound of the Butterworth filter

filter If your data are already filtered by a third program, turn that flag to False

remove\_median If you have some movement artefacts in your in vivo recording, and want to substract the median activity over all analysed channels from each channel individually

common\_ground If you want to use a particular channel as a reference, and subtract its activity from all others. Note that the activity on this particular channel will thus be null

## Triggers

The triggers section is:

trig\_file = External stimuli to be considered as putative artefacts [in trig units] (see documentation)

trig\_windows = The time windows of those external stimuli [in trig units]

trig\_unit = ms The unit in which times are expressed: can be ms or timestep

clean\_artefact = False If True, external artefacts induced by triggers will be suppressed from data

dead\_file = Portion of the signals that should be excluded from the analysis [in dead units]

dead\_unit = ms The unit in which times for dead regions are expressed: can be ms or timestep

ignore\_times = False If True, any spike in the dead regions will be ignored by the analysis

make\_plots = Generate sanity plots of the averaged artefacts [Nothing or None if no plots]

Parameters that are most likely to be changed:

trig\_file The path to the file where your artefact times and labels. See [how to deal with stimulation artefacts](https://spyking-circus.readthedocs.io/en/latest/code/artefacts.html)

trig\_windows The path to file where your artefact temporal windows. See [how to deal with stimulation artefacts](https://spyking-circus.readthedocs.io/en/latest/code/artefacts.html)

clean\_artefact If you want to remove any stimulation artefacts, defined in the previous files. See [how to deal with stimulation artefacts](https://spyking-circus.readthedocs.io/en/latest/code/artefacts.html)

make\_plots The default format to save the plots of the artefacts, one per artefact, showing all channels. You can set it to None if you do not want any

trig\_unit If you want times/duration in the trig\_file and trig\_windows to be in timestep or ms

dead\_file The path to the file where the dead portions of the recording, that should be excluded from the analysis, are specified. . See [how to deal with stimulation artefacts](https://spyking-circus.readthedocs.io/en/latest/code/artefacts.html)

dead\_unit If you want times/duration in the dead\_file to be in timestep or ms

ignore\_times If you want to remove any dead portions of the recording, defined in dead\_file. See [how to deal with stimulation artefacts](https://spyking-circus.readthedocs.io/en/latest/code/artefacts.html)

## Whitening

The whitening section is:

spatial = True Perform spatial whitening

max\_elts = 10000 Max number of events per electrode (should be compatible with nb\_elts)

nb\_elts = 0.8 Fraction of max\_elts that should be obtained per electrode [0-1]

output\_dim = 5 Can be in percent of variance explain, or num of dimensions for PCA on waveforms

Parameters that are most likely to be changed:

output\_dim If you want to save some memory usage, you can reduce the number of features kept to describe a waveform.

## Clustering

The clustering section is:

extraction = median-raw # Can be either median-raw (default), median-pca, mean-pca, mean-raw, or quadratic

sub\_dim = 10 # Number of dimensions to keep for local PCA per electrode

max\_elts = 10000 # Max number of events per electrode (should be compatible with nb\_elts)

nb\_elts = 0.8 # Fraction of max\_elts that should be obtained per electrode [0-1]

nb\_repeats = 3 # Number of passes used for the clustering

make\_plots = # Generate sanity plots of the clustering

merging\_method = nd-bhatta # Method to perform local merges (distance, dip, folding, nd-folding, bhatta)

merging\_param = default # Merging parameter (see docs) (3 if distance, 0.5 if dip, 1e-9 if folding, 2 if bhatta)

sensitivity = 3 # The only parameter to control the cluster. The lower, the more sensitive

cc\_merge = 0.95 # If CC between two templates is higher, they are merged

dispersion = (5, 5) # Min and Max dispersion allowed for amplitudes [in MAD]

smart\_search = True # Parameter to activate the smart search mode

### Note

This is a key section, as bad clustering will implies bad results. However, the code is very robust to parameters changes.

Parameters that are most likely to be changed:

extraction The method to estimate the templates. Raw methods are slower, but more accurate, as data are read from the files. PCA methods are faster, but less accurate, and may lead to some distorted templates. Quadratic is slower, and should not be used.

max\_elts The number of elements that every electrode will try to collect, in order to perform the clustering

nb\_repeats The number of passes performed by the algorithm to refine the density landscape

smart\_search By default, the code will collect only a subset of spikes, randomly, on all electrodes. However, for long recordings, or if you have low thresholds, you may want to select them in a smarter manner, in order to avoid missing the large ones, under represented. If the smart search is activated, the code will first sample the distribution of amplitudes, on all channels, and then implement a rejection algorithm such that it will try to select spikes in order to make the distribution of amplitudes more uniform.

cc\_merge After local merging per electrode, this step will make sure that you do not have duplicates in your templates, that may have been spread on several electrodes. All templates with a correlation coefficient higher than that parameter are merged. Remember that the more you merge, the faster is the fit

merging\_method Several methods can be used to perform greedy local merges on each electrodes. Each of the method has a parameter, defined by merge\_param. This replaces former parameters sim\_same\_elec and dip\_threshold

dispersion The spread of the amplitudes allowed, for every templates, around the centroid.

make\_plots By default, the code generates sanity plots of the clustering, one per electrode.

## Fitting

The fitting section is:

amp\_limits = (0.3, 30) Amplitudes for the templates during spike detection

amp\_auto = True True if amplitudes are adjusted automatically for every templates

collect\_all = False If True, one garbage template per electrode is created, to store unfitted spikes

ratio\_thresh = 0.9 Ratio of the spike\_threshold used while fitting [0-1]. The lower the slower

Parameters that are most likely to be changed:

collect\_all If you want to also collect all the spike times at which no templates were fitted. This is particularly useful to debug the algorithm, and understand if something is wrong on a given channel

ratio\_thresh If you want to get more spikes for the low amplitudes templates, you can decrease this value. It will slow down the fitting procedure, but collect more spikes for the templates with an amplitude close to threshold

## Merging

The merging section is:

erase\_all = True If False, a prompt will ask you to remerge if merged has already been done

cc\_overlap = 0.85 Only templates with CC higher than cc\_overlap may be merged

cc\_bin = 2 Bin size for computing CC [in ms]

default\_lag = 5 Default length of the period to compute dip in the CC [ms]

auto\_mode = 0.75 Between 0 (aggressive) and 1 (no merging). If empty, GUI is launched

remove\_noise = False If True, meta merging will remove obvious noise templates (weak amplitudes)

noise\_limit = 0.75 Amplitude at which templates are classified as noise

sparsity\_limit = 0.75 Sparsity level (in percentage) for selecting templates as putative noise (in [0, 1])

time\_rpv = 5 Time [in ms] to consider for Refraction Period Violations (RPV) (0 to disable)

rpv\_threshold = 0.02 Percentage of RPV allowed while merging

merge\_drifts = True Try to automatically merge drifts, i.e. non overlapping spiking neurons

drift\_limit = 0.1 Distance for drifts. The higher, the more non-overlapping the activities should be

To know more about how those merges are performed and how to use this option, see [Automatic Merging](https://spyking-circus.readthedocs.io/en/latest/code/merging.html). Parameters that are most likely to be changed:

erase\_all If you want to always erase former merging, and skip the prompt

auto\_mode If your recording is stationary, you can try to perform a fully automated merging. By setting a positive value, you control the level of merging performed by the software. Values such as 0.75 should be a good start, but see see [Automatic Merging](https://spyking-circus.readthedocs.io/en/latest/code/merging.html) for more details. The lower, the more the merging will be aggressive.

remove\_noise If you want to automatically get rid of noise templates (very weak ones), just set this value to True.

noise\_limit normalized amplitude (with respect to the detection threshold) below which templates are considered as noise

sparsity\_limit To be considered as noisy templates, sparsity level that must be achieved by the templates. Internally, the code sets to 0 channels without any useful information. So the sparsity is the ratio between the number of channels with non-zero values divided by the number of channels that should have had a signal. Usually, noise tends to only be defined on few channels (if not only one)

time\_rpv When performing merges, the code wil check if the merged unit has a valid ISI without any RPV. If yes, then merge is performed, and otherwise this is avoided. This is the default time using to compute RPV. If you want to disable this feature, set this value to 0.

rpv\_threshold Percentage of RPV allowed while merging, you can increase it if you want to be less stringent.

drift\_limit To assess if a unit is drifting or not, we compute distances between the histograms of the spike times, for a given pair of cells, and assess how much do they overlap. For drifting units, they should not overlap by much, and the threshold can be set by this value. The higher, the more histograms should be distinct to be merged.

## Converting

The converting section is:

erase\_all = True # If False, a prompt will ask you to export if export has already been done

sparse\_export = True # If True, data for phy are exported in a sparse format. Need recent version of phy

export\_pcs = prompt # Can be prompt [default] or in none, all, some

export\_all = False # If True, unfitted spikes will be exported as the last Ne templates

Parameters that are most likely to be changed:

erase\_all If you want to always erase former export, and skip the prompt

sparse\_export If you have a large number of templates or a very high density probe, you should use the sparse format for phy

export\_pcs If you already know that you want to have all, some, or no PC and skip the prompt

export\_all If you used the collect\_all mode in the [fitting] section, you can export unfitted spike times to phy. In this case, the last N templates, if N is the number of electrodes, are the garbage collectors.

Parameters:

<https://spyking-circus.readthedocs.io/en/latest/code/parameters.html>

## spyKING CIRCUS

Working on spyking…

To enter anaconda:

$ conda activate circus

To run algorithm:

>> spyking-circus path/mydata.extension

Probe file:

<https://spyking-circus.readthedocs.io/en/latest/code/probe.html?highlight=probe>

Configuration file:

<https://spyking-circus.readthedocs.io/en/latest/code/config.html>

11 Aug 2020

## spyKING CIRCUS

<https://elifesciences.org/articles/34518>

>> spyking-circus <filename> -p

command to show inputs

to look at results, can use

>> spyking-circus <filename> -r

Use phy:

circus-gui-python <filename>

Probe sites seem ok

<https://spyking-circus.readthedocs.io/en/latest/advanced/files.html>

<https://spyking-circus.readthedocs.io/en/latest/issues/faq.html>

<https://www.jneurosci.org/content/32/43/14859.abstract>

<https://spyking-circus.readthedocs.io/en/latest/GUI/sorting.html>

<https://spyking-circus.readthedocs.io/en/latest/code/multifiles.html>

14 August 2020

## spyKING CIRCUS

Link to Paper: <https://elifesciences.org/articles/34518>

Notes on Algorithm

1. Spike detection (putative) via threshold crossing.
2. Isolate snippets (waveforms) associated with each spike time
3. Snippets organized into groups depending on physical position (recording site/electrode)

Spyking Circus Steps: