APR 15, 2024

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.8epv5r1b4g1b/v1

Protocol Citation: Sasha Burwell 2024. In vivo Ephys: Spike Sorting Protocol. **protocols.io** https://dx.doi.org/10.17504/protocols.io.8epv5r1b4g1b/v1

License: This is an open access protocol distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited

Protocol status: Working

Created: Apr 02, 2024

Last Modified: Apr 15, 2024

In vivo Ephys: Spike Sorting Protocol

In 2 collections

Sasha Burwell[1]

[1]Duke University

ASAP Collaborative Research Network

Sasha Burwell
Duke University

ABSTRACT

This protocol details the spike sorting procedures used on the collected electrode recording data.

PROTOCOL integer ID: 98229

Keywords: ASAPCRN

Spike Sorting Protocol

1 For more information on setting up and using Spyking Circus, see: https://spyking-circus.readthedocs.io/en/latest/code/index.html .

1.1 Make a copy of the continuous.dat data file from your recording into a new “sorted_data” folder.

> **Note**
>
> Important note: Make sure to place a **copy** the continuous.dat file in this folder, and not the original, as the file will change as a result of spike sorting and you want to preserve the original raw data. This means you will also need to make a new copy the raw continuous.dat file if you need to redo the spike sorting analysis for any reason.

2 Add a .params file with the same name as the .dat file to the folder (i.e., continuous.params). The continuous.params used for these experiments:

```
[data]
file_format    = raw_binary
data_offset    = 0
mapping        = path\open_ephys_16prb.prb     # Mapping of the electrode

#see prb file below
suffix         =           # Suffix to add to generated files
data_dtype     = int16     # Type of the data
dtype_offset   = auto      # Padding for data (if auto: int16 is 0)
skip_artefact  = False     # Skip part of signals with large fluctuations
sampling_rate  = 30000     # Sampling rate of the data [Hz]
stationary     = True      # Should be False for long recordings
radius         = auto      # Radius [in um] (if auto, read from the prb
file)
alignment      = True      # Realign the waveforms by oversampling
global_tmp     = True
multi-files    = False     # If several files mydata_0,1,..,n.dat should be
processed together

[detection]
spike_thresh   = 4
N_t            = 2 or 3    # Width of the templates [in ms]. We used both 2
and 3 in different sortings to extract all the templates included.
peaks          = positive  # can be positive, negative, or both

[filtering]
cut_off        = 250       # Cut off freq for the butterworth filter [Hz]
filter         = True      # If True, then a low-pass filtering is performed

[whitening]
chunk_size     = 60        # Size of the data chunks [in s]
safety_time    = 1         # Temporal zone around which templates are
isolated [in ms]
temporal       = False     # Perform temporal whitening
spatial        = True      # Perform spatial whitening
max_elts       = 10000     # Max number of events per electrode
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

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

```
safety_space   = True      # If True, we exclude spikes in the vicinity of a
selected spikes
safety_time    = auto      # Temporal zone around which templates are
isolated [in ms]
max_elts       = 15000     # 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]
nclus_min      = 0.0075    # Min number of elements in a cluster (given in
percentage)
max_clusters   = 20        # Maximal number of clusters for every electrodes
nb_repeats     = 3         # Number of passes used for the clustering
smart_search   = 0         # Parameter for the smart search [0-1]. The
higher, the more strict
sim_same_elec  = 0.75      # Distance within clusters under which they are
re-merged
cc_merge       = 1         # If CC between two templates is higher, they are
merged
noise_thr      = 0.8       # Minimal amplitudes are such than
amp*min(templates) < noise_thr*threshold
make_plots     = png       # Generate sanity plots of the clustering
remove_mixture = True      # At the end of the clustering, we remove
mixtures
of templates

[fitting]
chunk          = 1         # Size of chunks used during fitting [in second]
gpu_only       = False     # Use GPU for computation of b's AND fitting
amp_limits     = (0.3, 5)  # Amplitudes for the templates during spike
detection
amp_auto       = True      # True if amplitudes are adjusted automatically
for every templates
refractory     = 0         # Refractory period, in ms [0 is None]
max_chunk      = inf       # Fit only up to max_chunk

[merging]
cc_overlap     = 0.5       # Only templates with CC higher than cc_overlap
may be merged
cc_bin         = 2         # Bin size for computing CC [in ms]

[extracting]
safety_time    = 3         # Temporal zone around which spikes are isolated
[in ms]
max_elts       = 1000      # Max number of collected events per templates
output_dim     = 5         # Percentage of variance explained while
```

```
performing PCA
cc_merge       = 1         # If CC between two templates is higher, they are
merged
noise_thr      = 0.8       # Minimal amplitudes are such than
amp*min(templates) < noise_thr*threshold

[noedits]
filter_done    = False     # Automatically edited – make sure this is False
before running the sorter
```

3

The .params file needs to have correct path to a .prb file (the mapping of the electrode you are using). We recorded with bundles of 16 individual electrodes, and used the following .prb file:

```
total_nb_channels = 16
radius            = 80
channel_groups = {
    1: {
        'channels': list(range(16)),
        'graph' : [],
        'geometry': {
                0: (0, 300),
                1: (100, 300),
                2: (200, 300),
                3: (300, 300),
                4: (0, 200),
                5: (100, 200),
                6: (200, 200),
                7: (300, 200),
                8: (0, 100),
                9: (100, 100),
                10: (200, 100),
                11: (300, 100),
                12: (0, 0),
                13: (100, 0),
                14: (200, 0),
                15: (300, 0),
        }
    }
}
```

4 Launch the Spyking Circus GUI from the anaconda terminal.

5 Run the sorting in the Main Algorithm tab.

5.1 Browse → select continuous.dat file.

5.2 Increase CPUs as desired, then click “Run”.

6 When the next window pops up, click “Suggest pairs”, “Select pairs”, then “Merge and finalize”.

7 To view results: View Results tab → Browse → select the analyzed continuous.dat file.

7.1 Options:

1. Matlab GUI
2. Extension = “merged”

7.2 Click “Run”.

7.3 See https://spyking-circus.readthedocs.io/en/latest/GUI/sorting.html for detailed documentation on what each panel of the GUI does.

8 Sorting:

8.1 Set bin size to 1, click “Big ISI”, and update the scale so you can see whole waveforms (zooming out usually works).

8.2 Go through all templates.

8.3 First pass: Kill a cell if it has above a 1% incidence of refractory period violations.

> **Note**
>
> RPV: % of spike pairs where your assumption about the refractory period are violated.

8.4 Second pass: merge templates that you think are the same cell, using the two template view.

- Criteria used: similar waveforms, cross correlation dropping to zero at zero, coefficient of similarity close to 1.
- Sometimes you have to merge more than two templates for one cell.
- Merge template with fewer spikes into the one with more spikes.

8.5 Third pass: kill units that don’t meet requirements (too sparse, not spiking consistently across the whole session, auto correlation doesn’t go to zero, RPV has gone above 1%).

9 Add suffix (ex, ‘v1’ for first attempt at sorting) and save.