Documentation of the hippocampal neuron data processing pipeline

Contents

[Spike\_data\_to\_dataframe 3](#_Toc159074055)

[lfp\_data\_to\_dataframe.py 4](#_Toc159074056)

# Spike\_data\_to\_dataframe

Gets the sampling rate for one epoch and fits the spikes of all neurons of that epoch in bins. Creates in that way a numpy array with all spiking times for one epoch.

Contains different functions to convert the spiking times into dictionaries.

Can create a sorted pandas dataframe containing the extracted cellinfo.mat files for all animals in the *animals* – dictionary.

**make\_neuron­\_dataframe\_modified**

Parameters:

|  |  |
| --- | --- |
| animals | **Dictionary** containing the file paths to the folders for all animals. |

Loads the cellinfo.mat files for all animals in the dictionary. Converts and concatenates the files to one sorted dataframe. The index is a unique neuron key which can be used to access specific spiking data.

Returns:

|  |  |
| --- | --- |
| Cell information | **Sorted pandas dataframe** containing all information on all cells for all animals in all epochs. Index = unique neuron key |

**spike\_time\_idex\_association**

Parameters:

|  |  |
| --- | --- |
| neuron\_key | **Tuple**, unique for identifying a neuron  (animal name, day, epoch, tetrode number, neuron number) |
| animals | **Dictionary** |
| time\_function | **Function** to determine sampling rate, start and end time of recording. By default get\_sampling\_rate |

Using the time\_function, a recording index for the specific epoch is calculated and used as bins. The spiking data corresponding to neuron\_key is then loaded and fit over the bins. Returns a pandas Series indicating whether the neuron spiked at a given time.

Returns

|  |  |
| --- | --- |
| Spikes dataframe | **Pandas dataframe** indicating the spiking times of the neuron in an epoch |

**generate\_spike\_indicator\_dict**

Parameters:

|  |  |
| --- | --- |
| neuron\_key\_list | **List** containing unique neuron keys (**str**) that can be generated using *make\_neuron\_dataframe\_modified* |
| animals | **Dictionary** |

Unpacks the neuron keys and creates dictionary of all spiking dataframes for specified neurons (if possible), using spike\_time\_index\_association

|  |  |
| --- | --- |
| Spike\_indicator\_dict | **Dictionary** with spiking dataframes for given neuron\_key\_list.  Key = neuron\_key |

**time\_index­\_dict**

Parameters:

|  |  |
| --- | --- |
| State\_day\_epoch\_neuron\_key\_dcit | **Dictionary** containing keys for each neuron in each animal and epoch |
| Animals | **Dictionary** |

Creates time\_dict of the same structure as state\_day\_epoch\_neuron\_key\_dict. Then gets dictionary of time index of spike trains for each cell

Returns:

|  |  |
| --- | --- |
| Time\_dict | **Dictionary** containing index times of spike trains for each neuron in state\_day\_epoch\_neuron\_key\_dicts |

# lfp\_data\_to\_dataframe.py

Extracts and converts matplot files. Creates pandas dataframe containing information on all tetrodes, given animal and epoch. Loads data for each tetrode and indexes it with recording time. Can be necessary for getting the trial times if they differ among epochs.

**get\_sampling\_rate**

Parameters

|  |  |
| --- | --- |
| epoch\_key | **Tuple** containing Animal (named tuple containing short\_name and directory, key = short name), day (int) and epoch (int) -> identifies animal in certain epoch |
| animals | **Dictionary** containing filenames, key = short\_name |

Main function. Calls *make\_tetrode\_dataframe* to create dataframe with information on all tetrodes in the epoch. Iterates over the dataframe: creates a tetrode\_key and tries to create pandas series with LFP recordings, using *get\_LFP\_dataframe*. Returns the first successful pandas series.

Returns:

|  |  |
| --- | --- |
| LFP recording | LFP recording of one (1) tetrode, indexed by recording time. |

**make\_tetrode\_dataframe**

Parameters: epoch\_key, animals

Loads the \*\_tetinfo.mat file for information on the tetrodes. Converts the matplot file into a pandas Dataframe.

Returns:

|  |  |
| --- | --- |
| Tetrode dataframe | **Pandas dataframe** containing:  columns: numcells, area, depth, animal, day, epoch, tetrode\_number, tetrode\_id (constructed from the 4 preceding variables)  rows: for each tetrode in given episode  The dataframe is indexed by the variables: animal, day, epoch and tetrode number  -> allows the creation of *tetrode\_key* |

**get\_LFP\_dataframes**

Parameters:

|  |  |
| --- | --- |
| tetrode\_key | **tuple** with sufficient information to identify specific dataset of LFP-recording:  [animal short name (str), day (int), epoch (int), tetrode\_number (int) ] |
| animals | **Dictionary** |

Calls *get\_LFP\_filename\_modified* to get the EEG data of the animal specified in the tetrode\_key. Calls *reconstruct\_time* to calculate the recording time. Converts the EEG data into a pandas Series, using the recording times as index.

Returns:

|  |  |
| --- | --- |
| LFP data | **Pandas Series**, recording time being the index  Name e.g. “bon\_01\_01\_001” for tetrode\_key = [“bon”, 1, 1, 1] |