Read Me - Event Detection

1 Goal of the script

This script detects all the events on the %DR/R traces of all the experiments performed for one sub-type of neurons (ie MDN, A1, MAN). The event detection is based on the first order derivative of the %DR/R. All the events detected are then aligned together by setting the event time point to 0s and storing a 20 seconds window of data around it. The average and 95% confidence interval of all the events traces are then computed and stored in png and eps format.

2 Setup of the Environment

This script was run on Python 2.7.10 and used the following libraries :

- Numpy (v 1.13.1)
- Matplotlib (v 1.3.1)
- cPickle (v 1.71)
- Seaborn (v 0.8.0)

The script runs on the versions mentioned next to the libraries. Click on the libraries names to access their documentation.

3 Folder Organisation

To run this script, the dictionary created at the end of script P4 from the ROI selection program is needed. This dictionary should be stored in the output folder created in script P1. To learn more about folder organisation, please refer to the Read Me file related to the ROI selection program.

4 Running the Script

Two different event detection scripts were written - one for A1 neurons (left or right) and the other one for Moonwalker Descending Neurons (MDN) and Moonwalker Ascending Neurons (MAN).

The A1-LOR-sns script is used for the detection of A1 neurons events. The user must set the following parameters :

- The boolean boolL that is used to run the script either for the right related events (if set to 0) or the left related events (if set to 1).
- The Percentile value can be changed (currently set to the threshold value that was selected for A1 neurons).
- « outFigDir » path should be set to the folder where the plots will be stored.

The MAN-MDN-sns script is used for the detection of MDN and MAN. The user must set the following parameters :

- value of A where 1 is related to MDN flies 1-2-3 and 2 is related to MAN.
- boolL value that is set to 1 for left neuron related events and set to 0 for right neuron related events.
- « outFigDir » path should be set to the folder where the plots will be stored.
- The Percentile value can be changed (currently set to the threshold value that was selected for MDN and MAN neurons).

The script can be run once the user set the parameters of his choice.

5 Detailed explanations of the script

One explanation will be given for both scripts as they are similar except for the complementary or non complementary event detection.
The dictionaries from each experiments are first opened one by one and the %DR/R and optic flow traces are extracted. The first order derivative is computed in getDeriv() and the derivatives of the traces are stored in two lists. The first one is the global list (one for left neuron and one for right neuron) that stores the derivative of the %DR/R in one list per experiment (it is therefore a list of x list where x represents the number of experiments that are being analyzed together). The second one is the DerivTotL list that stores in one unique list all the values from the derivatives from %DR/R left neuron trace. This second list will be used to calculate the percentile value used as the threshold. A respective list for right neuron is also created.
Once all the experiments have been opened and loaded, the 97.5 percentile value of the derivative for MDN and MAN (90th for A1) is calculated using Numpy library. The global list containing all the derivative values is then checked and the indexes of the derivative values crossing the threshold value are stored in IndxThreshTot list. This list is also a list of x lists where x represents the number of experiments that are being analyzed together (same organization than the global derivative list).
The eventIdxTot list is then computed for left and right by finding, for each event detected in the IndxThreshTot list, the nearest previous event where the derivative crossed the 0 value. This new list contains the indexes at which the first order derivative crosses the 0 values for every derivatives computed from all the experiments being analyzed. This new list does not contain events that happened in the first 10 seconds of each experiments. This is set by the « window » parameter and can be changed by the user - the window parameter is also used to set the time points that will be plotted.
A complementary event detection function was created for MAN and MDN (not in the A1 script). This function receives the left and right event total lists and compared the indexes position. For each events that are in the left event list but not in the right event list, a time window of indexes of 2 seconds around the event idx is created - meaning that a list with all the indexes from 1 second before to 1 second after the event time point is computed and then a check for interactions between this new list and the events detected in the right neuron traces is performed. If interactions exists, no event is added to the right event detected batch. However, if no events were detected in the right neuron %DR/R trace, then the left event idx is added to the right event batch. (the same analysis is performed for right neuron events that were not detected in the left neuron event batch). The time window of two seconds is defined by the

windowLARnotEqual parameter (set to 1 as it takes 1 second on each side of the event time point - before and after).
alignedDataToPlot() is then used to find and store the 20 seconds window of %DR/R and optic flow data around every events detected (10 seconds before and 10 seconds after the event time point that is set to 0 second in the final plot). If the events detected are happening at the end of the experiments (and the traces are not long enough to fill in the 10 seconds after the event detection) a NaN tail is added to the trace. Control traces were also created in this function. Every time an event was detected in one experiment, a random number was created and linked to the experiment number (- computationally it was stored in a list having the same index of the experiment list in the global stored list). All the random number were set as control event time point and were stored in the equivalent random lists.
The mean of all the detected events is computed in detectAverageVal() function. All data are then interpolated to the larger data set with interpolAllData() function. All the data sets are reduced to 500 data points per seconds in the diviseData() function - this parameter can be modified by changing the factor parameter.
Finally, 95% bootstrapped confidence intervals are computed using the Seaborn library and plotted in the figICSNSOneSubplot() function.