Fiber Photometry Data Analysis

The Bonsai Workflow Template from Neurophotometrics.

Link to Bonsai Textbook: <https://static1.squarespace.com/static/60ff345fca665d50e1adc805/t/65366fc82f93fd1b67b65a5b/1698066378544/BonsaiGuide_20231023.pdf>

A screenshot of a computer

Description automatically generated

1

2

3

1. These nodes process the data and write the data file. The FP3002 node allows for Bonsai to connect to the FP system. The photometry data node allows for visualization of the data. The photometry writer writes the raw data file.
   1. **Output**: SubjectID.csv
   2. A screenshot of a table

      Description automatically generated
   3. Frame Counter:
   4. System Timestamp = time since the system was turned on in seconds.
   5. Computer Timestamp = time of day according to the computer on milliseconds.
   6. LedState = this tells you which LED is on for each frame.
      1. 7 = all LEDs are on
      2. 1 = 415 channel
      3. 2 = 470 channel
2. The node KeyDown reacts to a key pressed on the computer. The computer timestamp is a group of nodes which records the timestamp when the specified key is pressed. CsvWriter writes a csv with the computer timestamp when the key was pressed.
   1. The expanded group of nodes named computer timestampA diagram of a company

      Description automatically generated
   2. This sequence appears three times in the workflow. The key ‘S’ is to record the start time after a 15-minute photobleaching period. The key ‘A’ is to record the pre and post injection timestamps. The key ‘H’ is to record the handling periods where the key is pressed before picking up the mouse and again after placing the mouse back down.
      1. **Outputs**: SubjectIDStart.csv, SubjectIDInjection.csv, SubjectIDHandle.csv
      2. A close-up of a number

         Description automatically generated
      3. This is an example of the output, Item1 is the key pressed and Item2 is the timestamp of when the key is pressed.
3. Connect a camera to Bonsai using the Camera Capture nose and process a video output using the VideoWriter node.
   1. The sequence for writing a document with a csv containing the frame and computer timestamps.A diagram of a company

      Description automatically generated
   2. Outputs: SubjectID.avi and SubjectIDVideoFrames.csv

Transforming the data files before plotting and calculating the zdFF scores

Reading the Bonsai Outputs

* The script defines a cohort folder and a subject folder which is used to locate the data and analyze multiple subjects in a row within the cohort folder. The outputs (SubjectIDStart.csv, SubjectIDInjection.csv and SubjectIDHandle.csv) are also read as well for each subject using the subject folder list.
* Template file path: '/location on computer/cohort folder/subject folders/bonsai outputs’
* Examples:
  + cohort\_folder:
    - '/Users/samdhanani/Desktop/Cohort\_G’
  + csv\_file\_path:
    - '/Users/samdhanani/Desktop/Cohort\_G/218\_1/218\_1.csv'
  + handle\_file\_path:
    - '/Users/samdhanani/Desktop/Cohort\_G/218\_1/218\_1Handle.csv'
* A new folder called “transformed files’ is created for the changes made to the bonsai output csv files. The original files will remain untouched in the subject folder.
* The variables x and sampling\_rate\_hz are also set in this portion of the code where x is the number of minutes you want to pull out before and after the injection timestamps. The variable sampling\_rate\_hz is the frame rate the data was recorded in.

SubjectID\_photobleach\_removed.csv

* This code takes the timestamp from SubjectIDStart.csv and filters out rows in the SubjectID.csv with a timestamp smaller than the start timestamp. This is all based on the computer timestamp, not the system time stamp.
* It then saves this cropped data frame to a new data frame and then saves the data to a new csv file.

SubjectID\_photobleach\_handle\_removed.csv

* First, the start and end timestamps of the handling data are extracted to an empty list called “exclusion\_ranges”. This is done using a for loop that goes through each row in the handle data frame and stores the timestamps as start and end tuples in exclusion\_ranges. In the handle data frame, the first row becomes the start time and the second becomes the end time and so on until there are no more rows.
* Next, the list of exclusion ranges is used to filter the photobleach\_handle\_removed data frame where a for loop goes through the list. The for loop filters the data frame by keeping rows with a computer timestamp less than the start time and greater than the end time and then checks again for another tuple in the list exclusion ranges.
* Finally, the last filtering step removes instances of consecutive led states and this affects data processing downstream. A for loop goes through the photobleach\_handle\_removed data frame and checks for consecutive led states by checking if two rows next to each other have an equal value in the led state column. If there are two consecutive led states of “1” then the first row is removed. If there are two consecutive led states of “2” then the second row is removed.
* An arbitrary x axis is created for the reference led state and the signal led state to plot the datapoints as the removal of the handling data creates gaps of data on the x axis.

SubjectID\_interpolated.csv

* This code interpolates the data so that for each time stamp there is data for the each led state. The code starts by deinterleaving the led states and uses numpy.interp to estimate new data points based on the existing data.
* The data format is changed so that the led states are no longer interleaved and an arbitrary x axis is created for the reference and signal led states to plot the datapoints as the removal of the handling data creates gaps of data on the x axis.

SubjectID\_all\_baseline.csv

SubjectID\_all\_amph.csv

SubjectID\_all\_full\_session.csv

* This code creates the csv files above from the photobleach\_removed\_handle\_df data (not interpolated). The code begins by reading the injection csv file and using the start and end time of the injection period to determine the pre-injection and post-injection data.
* The first timestamp in the injection csv file is used to determine the end of the pre-injection baseline period and the second timestamp in the injection csv file is used to determine the start of the post injection amph period which lasts until the end of the recording.
* The full session data is made up of datapoints before and after the injection start and end times, in other words the injection time is removed from the data.

SubjectID\_Xmin\_baseline.csv

SubjectID\_Xmin\_amph.csv

SubjectID\_all\_full\_session.csv

* The code starts by calculating how many rows to pull out before and after the injection time stamps based on the time the user wants and the frame rate.
* The variable num\_rows\_interval determines how many rows are going to be pulled out based on the rows\_per\_min equation and the user defined time defined as ‘x’.
* This is done for the pre and post injection times where the last row of the baseline data frame is identified and the variable num\_rows\_interval is subtracted from that.
* In the amph data frame the first row is identified and the variable num\_rows\_interval is added to that.
* Then the interval\_pre\_injection\_df and interval\_post\_injection\_df are combined to make the full\_session\_df.

SubjectID\_time\_subtracted\_values\_amph.csv

* This code is for quality control. It calculates how much time is being removed during the post\_injection time period by summing up all the time differences greater than the injection end time stamp in column ‘Item2’ in the SubjectIDHandle.csv.

Calculating the dFF values

Harris lab script

Peak Analysis

* cohort\_folder is defined as the location of the folder containing the subject folders.
* The variables suffix and time are defined so that the csv outputs are named based on the length of time being analyzed or the criteria being used in the find\_peaks function.
* The subject folders are defined and the variable time is used to select csv files with a specified time length as seen in files\_transform

The following file outputs will be extracted for full\_session, baseline, and amph datasets.

* A path is defined for PdfPages with the suffix ‘\_plots\_zdFF\_criteriaspecification.csv’.
* Then the average and standard deviation are calculated.
* The next section defines an upper and lower bound to filter out data based on the average +/- any number of standard deviations. This section can be commented out if this type of filtering is not wanted.
* The function find\_peaks is used to define a peak using the zdFF column from the zdFF.csv file, a height = 0, and a distance of 100ms. The timestamps and values of the peaks are defined by a variable to be used later in the code.
* The next section of code analyzes the peak frequencies, individual peak amplitudes, and average amplitudes.
* Next the AUC is calculated based on the peak values.
* Graphs of the peaks identified on the zdFF data graphed to visualize the data being analyzed. This uses the timestamps and values calculated earlier.
* The next section pulls out the individual amplitudes for each dataset using the peak measurements calculated.
* The last section of the code creates a csv file for every subject in the cohort folder with the following data: 'Average zdFF', 'Standard Deviation zdFF', 'Num Peaks','Peaks/Second', 'Average Peak Amplitude', 'AUC'
* Criteria being currently used:
  + 100ms
    - find\_peaks is defined using the zdFF column, a height minimum of zero, and a duration of 100ms
  + No criteria
    - find\_peaks is defined using the zdFF column, no height minimum, and no duration

Works Cited

Martianova, E., Aronson, S., Proulx, C. D. Multi-Fiber Photometry to Record Neural Activity

in Freely-Moving Animals. J. Vis. Exp. (152), e60278, doi:10.3791/60278 (2019).