Nadir Bilici

nadir.bilici@uphs.upenn.edu

562-331-5968

July-August 2016

Run Commands

NB\_runAnalysis\_IO

NB\_runAnalysis\_IO\_oneTrial

NB\_runAnalysis\_PP

NB\_runAnalysis\_SE

NB\_runAnalysis\_SO

Loading Data

NB\_loadall\_NLX (windows)

NB\_loadall\_NLX\_v3 (mac osx)

Preparing Data

NB\_prepdataCSC\_32chan

Creating Data Blocks

NB\_prep\_CSDdata\_IO

NB\_prep\_CSDdata\_IO\_oneTrial

NB\_prep\_CSDdata\_PP

NB\_prep\_CSDdata\_Stim

Plotting Data

NB\_plot32

NB\_plot32\_IO

NB\_plot32\_IO\_oneTrial

NB\_plot32\_PP\_slopes

NB\_plot32\_stim\_rateHistogram

NB\_saveFigures

Data Browsing

NB\_plot32\_browse

NB\_plot32\_response\_picker

Plot Helper Functions

NB\_plot\_adjust\_electrical

NB\_plot\_adjust\_electrical\_subplots

NB\_plot\_adjust\_optical

NB\_plot\_adjust\_optical\_subplots

NB\_plot\_adjust\_quick

NB\_plot\_findLatency

NB\_plot\_findLatency\_subplots

Calculate Functions

NB\_PP\_calculateSlopeRatio

Single Unit Functions

NB\_cellExtraction

NB\_makeEvents

NB\_spike\_fileLoad

NB\_spike\_plot\_psth

NB\_spike\_plotCellShape

NB\_spike\_saveCellShapes

File Organization/Management

NB\_prepAnalysis(numRecordings)

1. NB\_createFolders
2. NB\_sortFiles
3. NB\_batchRenameFiles

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% COMMON ERRORS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

1. PATH: Make sure you are in the right folder. Your folder has to contain the CSC/EV/TT files.
2. PLATFORM: When loading data, make sure you are using the right load functions. Mac commands have the suffix \_v3. Windows commands do not.
3. LEFTOVERS: Type ‘clear’ to clear the workspace in between each set of data. Lingering data can cause errors with calculations and plotting

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RUN COMMANDS %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Use these commands to load, prepare, and plot data

**NB\_runAnalysis\_IO**: used for IO experiments with two trials

**NB\_runAnalysis\_IO\_oneTrial**: used for IO experiments with only one trial

**NB\_runAnalysis\_PP**: used for paired pulse experiments

**NB\_runAnalysis\_SE**: used for biphasic electrical stim experiments

**NB\_runAnalysis\_SO**: used for optical stim experiments

**NB\_runAnalysis\_spike**: used to create a PSTH and raster for spikes

These commands are self-sufficient to create plots. Usually, I will use the run command based on the appropriate experiment, and then use plot helper functions to adjust axes and make the graph look pretty.

To dig deeper about how the data comes from the Neuralynx, how we process the data, and how to modify/adjust plots, please read below.

%%%%%%%%%%%%%%%%%%%%%%%%%%% LOADING DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%

OSX commands have the \_v3 suffix

WIN commands do not have the \_v3 suffix

If you are running into errors with data loading, make sure you are using the correct commands for your platform (OSX or WIN)

**NB\_loadall\_NLX** (windows)

**NB\_loadall\_NLX\_v3** (mac osx)

These functions load tetrode (TT), continuously sampled channel (CSC), and event (EV) data into the MATLAB workspace.

The importTT, importCSCs, and importEVs commands call on Nlx2Mat functions obtained from Neuralynx: <http://neuralynx.com/research_software/file_converters_and_utilities/>

%%%%%%%%%%%%%%%%%%%%%%%%%%% PREPARING DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%

**NB\_prepdataCSC\_32chan**

**NB\_importCSCWave**: reshapes CSC 3D matrix data into a 2D matrix

When we use Cheetah to record samples, each sample contains 512 points which is why CSC\_Sample data is imported as a 512x#### matrix. This function reshapes the 3D matrix into a data set that represents one continuous wave.

Some math if you are curious: Each ‘Sample’ at 32000 sample frequency is separated by 16000 microseconds. Each sample contains 512 points. So, each point is separated by 31.25 microseconds. Each second is 1,000,000 microseconds – which means each second is represented by 62.5 samples. So, in CSC\_Sample data, the first second of data captured is represented by column 1 – column 62 at row 256.

**NB\_timeVector3**: creates a timeVector containing timestamps. The length of timeVector should be equal to the length of CSC\_Wave. Each index of timeVector represents the timestamp of the index of CSC\_Wave’s sample data

Now, all of the data is stored in various vectors. You have 32 CSC\_Wave vectors and a timeVector. We need to put this all into a singular data block to efficiently manipulate and plot data.

%%%%%%%%%%%%%%%%%%%%%%% CREATING DATA BLOCKS %%%%%%%%%%%%%%%%%%%%%%%%%

If you need to adjust aspects of the plot like plot length or how much time you want to see before stim, do that in these functions with the ‘ms’ and ‘offsetms’ variables.

**NB\_prep\_CSDdata\_Stim**

First, this function aggregates all of the CSC\_Wave data into CSD\_DataBlock(channel, CSC samples, stim)

Now, we have a CSD\_DataBlock that contains CSC samples for each stim trial across the 32 channels.

The data is averaged across all stim trials to produce the 2D matrix, CSD\_Data.

**NB\_prep\_CSDdata\_IO**

IO experiments are traditionally run twice

First, this function matches the first and second set of stim timestamps (i.e. if there are 10 stims per trial, it matches stim 1 with 11; 2, 12; 3,13…. So, matchedStimOnTimestamps now has the equivalent IO trials next to each other.

Then, using the matched timestamp data, we create CSD\_DataBlock(channel, CSC samples, stim) to aggregate the data into one matrix.

So, CSD\_DataBlock(1,:,1) and CSD\_DataBlock(1,:,2) have the samples for stim 1 and 11 for channel 1.

CSD\_DataBlock\_IO takes each pair of trials and averages the data, so CSD\_DataBlock(1,:,1) and CSD\_DataBlock(1,:,2) now become CSD\_DataBlock\_IO(1,:,1).

**NB\_prep\_CSDdata\_IO\_oneTrial**

If the IO experiment is only run once, use this function to prepare CSD\_DataBlock\_IO.

**NB\_prep\_CSDdata\_PP**

This function is used to prepare the CSD\_DataBlock for paired pulse experiments. Because PP experiments have two stims in short secession. This function only looks at the first stim of each paired pulse to create the CSD\_DataBlock

%%%%%%%%%%%%%%%%%%%%%%%%% PLOTTING DATA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

**NB\_plot32**

Use for optical and electrical Stim

Plots 32 channels from CSD\_Data, which contains averaged sample data across all stims

**NB\_plot32\_IO**

Use for Input/Output experiments. Creates two plots

(1): Plots each IO step for targetChannel (I usually chose CSC32 because it was at the bottom of the probe and had the best responses). Top line is for step 1. Bottom line is for step 10. These are all for the same CSC channel, not for all 32 channels.

(2): Calculates the max amplitude for each IO step between msStartTime and msEndTime for targetChannel (I set this for channel 32). The X axis represents the 10 different current steps

**NB\_plot32\_IO\_oneTrial**

Same as NB\_plot32\_IO but for one IO experiment

**NB\_plot32\_PP\_slopes**

Use for Paired Pulse experiments

Plots data similar to NB\_plot32. Automatically calculates minimum and maximum slopes for the first and second stims (based on timestamps in the code) and divides slope2/slope 1. Data is stored as Ratio\_Up and Ratio\_Down

**NB\_PP\_calculateSlopeRatio** uses a manual input of beginning and end timepoints to calculate the slope ratio between stim2 vs. stim1. Data saved as ratioSlopes (see CALCULATE FUNCTIONS)

**NB\_plot32\_stim\_rateHistogram**

Use this function after using NB\_plot32\_response\_picker to plot the positive and negative responses along with a rate histogram. Asterisks represent positive responses and dots represent negative responses across the different trials.

**NB\_saveFigures**

Saves .m (MATLAB), .eps (Illustrator), and .png (Image) figures titled ‘CSCPlot’

%%%%%%%%%%%%%%%%%%%%%%%%%%%% DATA BROWSING %%%%%%%%%%%%%%%%%%%%%%%%%%%

**NB\_plot32\_browse**

Use for optical/electrical stim experiments. Plots 32 channels for each stim. Click Enter to quickly browse through each stim and gauge response

**NB\_plot32\_response\_picker**

Use for optical/electrical stim experiments in which response is not optimal. Allows user to differentiate positive (type 1 and Enter) and negative responses (type 2 and Enter). Saves data into CSD\_DataBlock and CSD\_Data\_Negative.

%%%%%%%%%%%%%%%%%%%%%%%%% PLOT HELPER FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%

**NB\_plot\_adjust\_electrical**

Adjust the offset and y-axis values after creating a plot with electrical stim, IO, or PP. Use 1/2 to increase/decrease offset values. 3/4 to increase/decrease y-axis values. 5 to reset axes through manual input. 0 to save figures.

**NB\_plot\_adjust\_electrical\_subplots**

Use for electrical stim experiments in which you have subplots (by plotting with NB\_plot32\_stim\_rateHistogram)

**NB\_plot\_adjust\_optical**

Adjust the offset values after creating a plot with electrical stim, IO, or PP. Use 1/2 to increase/decrease offset values. 0 to save figures. The reason this is different that the adjust\_electrical function is that optical stim does not have the electrical stim artifact at time = 0. So, we can tell MATLAB to make ‘axis tight’ to always get rid of whitespace on the top or bottom of the graph.

**NB\_plot\_adjust\_optical\_subplots**

Same as NB\_plot\_adjust\_electrical\_subplots – use for optical experiments when you have subplots

**NB\_plot\_adjust\_quick**

Quickly set the y-axes for a plot using user input. Click twice on the graph; first to set the maximum Y axis value and secondly to set the minimum Y axis value

**NB\_plot\_findLatency**

Run this function and then click on graph to calculate latency at a certain point.

**NB\_plot\_findLatency\_subplots**

Same as NB\_plot\_findLatency. Use this function when you have subplots like with NB\_plot32\_stim\_rateHistogram

%%%%%%%%%%%%%%%%%%%%%%%%%% CALCULATE FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%%

**NB\_PP\_calculateSlopeRatio**

After plotting the paired pulse graph, this function generates an additional graph showing the slope of targetChannel (I use CSC32)

The function asks for user input to mark the minimum and maximum time window for the first stimulation. Then, the function calculates the time window for the second stim.

Using these time windows, the slope within the time windows for the first and second stims are calculated. Slope 2 is divided by slope 1 to produce ratioSlopes.

%%%%%%%%%%%%%%%%%%%%%%% SINGLE UNIT FUNCTIONS %%%%%%%%%%%%%%%%%%%%%%%%

These functions were used for a single channel tungsten electrode. Tetrode data was clustered using SpikeSort3D and then saved to a TT1\_cells.ntt file.

**NB\_spike\_fileLoad**

Uses **Nlx2MatSpike** and **Nlx2MatEV** functions to pull out spike and event data from TT1\_cells.ntt and Events.nev

**NB\_spike\_cellExtraction**

Extracts spike timestamp data from TT1\_Timestamps

**NB\_spike\_makeEvents**

Extracts event timestamps from EV\_Timestamps. Creates a matrix for stimOnTimestamps

**NB\_spike\_plot\_psth**

Plots a peri-stimulus time histogram and raster plot for spikes within the inter-stim window

This produces the PSTH for an individual cell. Adjust the name of the cell ‘Cell\_1\_X’ by replacing all instances of it in the code to create PSTHs for the other cells.

**NB\_spike\_plotCellShape**

Plots the shape of all clusters

**NB\_spike\_saveCellShapes**

Saves figure of cell shapes