Phase-Amplitude Coupling

cfc\_heatmap.m: Create co-modulogram of phase-amplitude coupling across frequency for phase and frequency for amplitude pairs

makedatafile.m: Create input for “modindex.m” function. This function lets you choose between two methods of extracting phase information from the lower frequency oscillation: The Hilbert transform, or the phase extrapolation method

makedatafile\_morlet.m: Create input for “modindex.m”, and extract lower frequency phase information with Morlet wavelets

modindex.m: Calculate the Kullback-Leibler distance between an observed phase-amplitude distribution and a uniform phase-amplitude distribution. This function calculates a modulation index (MI) value and normalized amplitude values across a range of phase bins

phase\_map.m: Create co-modulogram of normalized amplitude values across frequency for amplitude values and phase bins for a pre-determined frequency for phase

shuffle\_MI.m: Create a z-scored MI value relative to a bootstrapped distribution of phase-shuffled MI values

LFP\_triggered\_avg.m: Plot either the trough or peak-triggered LFP average for an oscillation in a given frequency band

LFP Filtering and Maintenance

skaggs\_filter\_var: Third-degree Butterworth filter. Filtering LFP is necessary before extracting phase and amplitude information using the Hilbert transform or phase extrapolation, but not necessary when using Morlet wavelets.

detrend\_lfp: De-trend LFP to get rid of low-frequency “drifting” artifacts

phase\_freq\_detect: Extrapolate phase information from a lower frequency oscillation by using phase interpolation. This function also determines the instantaneous per cycle frequency of a filtered oscillation. This function is called when phase extrapolation is selected for phase-amplitude coupling analyses

Single Unit Analysis

delay\_left\_right.m: Create raster and trial-averaged firing rate plots of single unit activity during start-box occupancy for prospective left and right-turn trials

entrainment.m: Calculate single unit phase-locking to a simultaneously recorded oscillation during start-box occupancy

lagged\_entrainment.m: Calculate single unit phase-locking across a range of LFP lags to assess directionality of entrainment

firing\_rate\_maze.m: Calculate single unit firing rate across start-box, stem, choice-point, reward-arm, and return-arm maze sections

phase\_locked\_MUA.m: Plot multi-unit activity (spiking of all simultaneously-recorded single units) with ongoing oscillatory activity to visualize population phase preferences

principle\_comp.m: Perform principal component analysis on stem-binned firing rate matrices for dual-task sessions (this is just something I had been messing around with – not really sure how it could be applied to our data yet)

pca\_trajectories.m: Create neural trajectories in principal component-derived state space for stem traversals during task performance. Calculate Euclidian distance metrics for quantifying deviations in population trajectories between tasks

peak\_waveform.m: Calculate mean peak waveform amplitude for clusters in a given session between pre- and post-session recording epochs (I used this script to calculate metrics of cluster stability during dual-task performance. This script could also be used to extract waveform information for cluster plotting, L-ratio calculations, etc.)

PETH\_delay.m: Peri-event time histogram and raster plot of single unit activity during start-box occupancy

plot\_all\_delay.m: Plot population activity during start-box occupancy (I was messing around with this script to look for evidence of “time cell”-like activity in PFC populations)

spk\_triggered\_avg.m: Plot the spike-triggered LFP average for a given neuron-LFP pair

stem\_left\_right: Same as “delay\_left\_right”, but for stem traversals. It would be cool to figure out a way to create raster plots for spatial bins instead of time bins

svm.m: Decodes which task is being performed during a single stem traversal by training a linear classifier with population firing rate activity during all other stem traversals. This script also tests classifier performance with ROC curves.

task\_fr: Compare firing rates across maze locations for dual-task performance.

rate\_map.m: Creates firing rate map for single units

Optogenetics

simple\_pulse.m: Pulses the laser in equally spaced intervals. Collects timestamp data for “laser on” and “laser off” periods

DNMP\_opto.m: Pulses the laser at specific sections of the maze during a pre-specified phase of the DNMP task. Once the phase of DNMP is selected, the laser continues to turn on and off during the specified phase for the duration of the session

DNMP\_opto\_interleaved.m: Same as “DNMP\_opto.m”, but the DNMP task phase at which the laser turns on and off is pseudorandomly selected from trial to trial. This script makes it possible to compare neural activity between different phases within the same recording session

Other

time\_spent.m: Calculate time spent at choice-point and stem velocity

theta\_only.m: Calculate theta-gamma coupling and phase coherence only during periods of high amplitude theta during start-box occupancy

morlet\_spectrogram.m: Plot spectrogram of continuously sampled data using Morlet wavelets

Functions that I Use Heavily, But Aren’t Mine

mtspectrumc.m: Calculate the power of a signal using the multi-taper method

coherency.m: Calculate phase coherence using the multi-taper method

cohgramc.m: Create a coherogram using the multi-taper method

GCspectral.m: Calculate a Granger causality index between LFP pairs across a range of frequencies

shadedErrorBar.m: Create shaded error bars for line graphs