Two\_photon\_imaging\_V2

**Analysis-** contains subfolders relating to all analyses

* cellBased
  + CO
    - *checkCellROICOContourOverlap*: Classifies cell ROIs as either cytochrome oxidase patch or interpatch cells
  + Cone
    - *calculateConeInput*: Calculate cone input metrics for existing monkey orientation/cone experimental data
  + Correlation
    - *gatherCorrMetrics*: Gathers precalculated correlation metrics for individual animas across multiple recordings
    - *getCorrMetricCellPair*: Gets correlation info for specific cell pair numbers from a single recording, used in *plotCorrelationPairResponses*
    - *pairwiseNoiseCorrelations*: Calculates prestimulus ‘noise correlations’ for individual experiments, used by *pairwiseNoiseCorrelationsWrapper*
    - *pairwiseNoiseCorrelationsWrapper*: wrapper function which runs *pairwiseNoiseCorrelations* on multiple recordings
    - *pairwiseStimulusCorrelations*: Calculates ‘stimulus response correlations’ for individual experiments, used by *pairwiseStimulusCorrelationsWrapper*
    - *pairwiseStimulusCorrelationsWrapper*: wrapper function which runs *pairwiseStimulusCorrelations* on multiple recordings
  + Dendrite
    - *checkDendriteCellOverlap*: Creates image stack which overlays two recordings to match up dendrites and cell ROIs
    - *createPatchROIOverlayImage*:Ccreates flattened overlay image of ROIs on greyscale average image, called by *checkDendriteCellOverlap*
  + dualChannel
    - *checkDualChannelExpression*: checks whether cell ROIs are visible in both recorded channels. Used in *checkDualChannelExpression\_wrapper*
    - *checkDualChannelExpression\_wrapper*: Wrapper for running the *checkDualChannelExpression* on multiple recordings
    - *countOverlapROIsFromFIJI*: Counts number of dual channel cells based on ROI color
  + orientation
    - *calculateOSI*: Calculates a bunch of orientation selectivity indices for a single cell, called by *calculateOSIPopulation*
    - *calculateOSIPopulation*: runs orientation tuning calculations on all cells in a recording, called by *calculateOSIPopulation\_wrapper*
    - *calculateOSIPopulation\_wrapper*: batch runs multiple recordings data for orientation tuning metrics
    - *compareOrientationFitsRMSE*: Compares orientation tuning fits for different cone color conditions to see if they maintain orientation tuning across cone types
    - *convertPixelOrientationSelectivityToCell*: Converts pixel orientation selectivity maps to full cell ROI shapes
    - *gatherOrientationMetrics*: Gathers already calculated orientation selectivity info from the subfolders
    - *gatherPopulationOSI*: Gathers already calculated orientation selectivity info from the subfolders
    - *getSingleGausFitPlotSigma*: Creates single gaussian fit for monkey 0-180 orientation tuning and plots sigma value map
    - *testOrientationFitsPerBlock*: Fits orientation tuning curves on individual blocks for monkey data
* dispersion
  + - *dispersionAnalysis*: Run dispersion analysis and recording day file for single wavelength
    - *getDispersionIntensity*: Calculates average intensity per pixel over depth for a single dispersion run
    - *plotDispersionData*: Plots intensity/mW for all dispersion runs for a single wavelength
* fixes
  + - *convertExperimentStructure2ClassObject*: Converts all experimentStructures to matlab objects REDUDANT
    - *fixDroppedTrialExperimentStructure*: Will fix a last dropped trial in a processed experimentStructure file. Will only fix window average fields
    - *fixXYPostionError*: Fixes error found in XY location decoding REDUDANT
* pixelwise
  + - *createStimVSPrestimSTDDiffGPU*: Creates difference pixelwise image for stim vs prestim for a recording
    - *pixelwiseOrientationSelectivity*: Creates pixelwise orientation preference and selectivity maps for a recording
    - *pixelwiseOrientationSelectivityBatch*: Batch runs *pixelwiseOrientationSelectivity* for a recording day file
* utilities
  + - *grabRecordingLocationsXYZ*: Grabs XYZ positions for all recordings in a recording day file
    - *mean\_vector\_direction\_magnitude*: Creates mean vector angle and size for pixel orientation selectivity maps
* vascular
  + - *lineAnalysisForBloodVessels*: Produces line intensity plot vessels for different orientation types (used in F32 application)
    - *plotLineDynamicsForF32*: Creates the plot for displaying line dynamics

**Calibrations-** contains subfolders relating to screen and stimulus computer event calibration

* DAQ
  + - *PrairieVoltageInfo.m*: Output file from *readEventFileSetupV2* containing all the information for each event voltage level (BACKUP, not used)
    - *readEventFileSetupV2*: Reads voltage file from *testDAQOutSignal* run, may need some fiddling to get to work properly
    - *testDAQOutSignal*: Runs voltage level step output to use for event signal calibration on the stimulus computer
* monitor
  + - *LCDmonitorCalibrations.m*: Cone capture level based on the LCD monitor calibration

**Dev-** contains development folder with various scrap functions

**Doc-** contains setup documents

**Extraction-** contains subfolders related to calcium trace extraction

* Batch
  + - *runCaAnalysisWrapper*: Runs full calcium imaging extraction analysis on experiment day file
* CaExtraction
  + - *baselinePercentileFilter*: Creates baseline percentile filter for dF/F subtraction
    - *CaAnalysis*: Runs full calcium imaging extraction analysis on single recording file
    - *CaExtraction*: Enacts main calcium imaging extraction
    - *calculateNeuropilRoiRadius*: Calculates neuropil ROI radius used for the neuropil-signal extraction
    - *createImageJROIsFromLabeledROI*: Creates ImageJ ROIs based on labelled ROI image
    - *createLabeledROIFromImageJPixels*: Creates a labeled ROI from the ROIs defined in ImageJ's ROI Manager
    - *estimateNeuropil*: Estimates neuropil signal by using Dipoppa et al (Carandini lab) method
    - *generateNeuropilROIs*: Generates neuropil ROIs based on already chosen cell ROIs
* ROIExtraction
  + - *chooseROIs*: Used to choose ROIs for cells for recordings, called by *CaAnalysis* and *runCaAnalysisWrapper*
    - *chooseROIsCallbck*: Choose ROI callback for the data viewing app
    - *chooseROIWrapper*: Batch wrapper for choosing ROIs
* traceProcessing
  + - *estimate\_percentile\_level*: Estimates percentile level for baseline calcium trace calculation
    - *kde*: Kernel density estimation which is used to create baseline filtered trace
    - *splitDFIntoConditions*: Splits dF/F trace into the various subcomponents and prestim and stim time periods

**MIJI-** functions related to ImageJ in matlab

* + - *createCellFOVLimit*: Allows user to create subfield to limit cells for analysis
    - *intializeMIJ*: Starts up ImageJ in matlab
    - *loadImagingData2FIJI*: Loads in raw movie with motion correction for display in imageJ/FIJI

**neuralNetTraining-** functions related to creating neural net to automate cell ROI creation

* + - *copyDataForNeuralNetTraining*: Copies data for library use for cell ROI neural net training
    - *createUnet2P*: Builds U-net within matlab to start training
    - *preprocessImages4UNet*: Creates ROI masks used in training the neural net and expands the data
    - *testAccuracyTrainROINet*: Tests the accuracy of the trained ROI extraction net
    - *unetLayersV2*: Architecture of U-net

**plotting-** functions related to plot creation and figures

* + - *compareOrientationTuningFits*: Plots different orientation tuning curve fits to compare different methods
    - *compareOrientationTuningFitsLS*: Plot LS orientation fits on monkey data for different cone/color conditions
    - *plotCellIndexMap*: Plots and saves RGB image of index preference for a variety of indices
    - *plotCellOrienationPref*: Creates a RGB image of maximal response for orientation pref for every cell based on the DF/F signal
    - *plotConeInputRatioMaps*: Plots and saves RGB image of index preference for L vs M and S vs LM ratios
    - *plotConeResponseRatioMaps*: Plots and saves RGB image of index preference for L vs M and S vs LM response ratios
    - *plotCorrCoeffDistance*: Plots correlation coeffs across distance for populations of data for both non PV and PV cell types in mouse for stimulus or noise data types
    - *plotCorrCoeffDistanceHist*: Plot histogram of number of pairs for different cell type interactions across distance
    - *plotCorrelationPairResponses*: Plots average condition responses, stimulus and noise correlations for specified pairs of cells ( used for mouse data)
    - *plotDendriteProfilesPerCnd*: Allows user to choose dendrite ROIs and then calcium trace extraction. Then plots average responses per condition
    - *plotFitsPerTrialCell*: Plot orientation fits to each block of orientation tuning/cone condition data for an idea of how well fits work
    - *plotLineProfilesPerCnd*: Allows user to choose line to plot orientation dynamics (used for vascular dynamics)
    - *plotOnOffResponseMaps*: Plots an image categorizing cells as ON, OFF or ON/OFF responsive
    - *plotOrientationContrastOSIDepth*: Plot orientation selectivity index across depth and contrast condition for mouse orientation/contrast along multiple z locations.
    - *plotOrientationShifts*: Plots preferred orientation shifts between different cone condition types
    - *plotOrientationShiftsCO*: Plots preferred orientation shifts between different cone condition types for cytochrome oxidase patch vs interpatches
    - *plotOrientationTuningDendrite*: Plotting function used in *plotDendriteProfilesPerCnd* to create dendrite condition response plots
    - *plotOrientationTuningPerCell*: Plots condition average trial responses for specified cells
    - *plotRandomROIFPerCnd*: Allows a user to choose new ROIs to examine their condition average response.
    - *plotSigmaFromSingleGausFit*: Plots sigma value for each recording across all cells
    - *plotStitchImage2P*: Not used anymore
    - *plotZScoreCellMaps*: Creates image of cells colored by zscore
    - *polarPlotOrientation*: Creates polar plots of orientation preference for each cone/secondary condition

**preprocessing-** functions related to data preprocessing

* batch
  + - *prepData*: Completes data preprocessing on a raw data folder for two photon imaging
    - *prepDataBatch*: Completes data preprocessing on a recording day folder for two photon imaging
    - *prepDataBatchWithOrientationPixelPref*: Completes data preprocessing on a recording day folder for two photon imaging. Also completes pixelwise orientation tuning images
* imageHandling
  + - *createChannelOverlapImage*: Function to create average based on the lowest n number of pixels in the time domain
    - *createChannelOverlapImageWrapper*: Runs through all the processed files in subfolders and calls the overlap image function
    - *createStimSTDAverage*: Function to create STD sum images for prestim and stim times
    - *createStimSTDAverageGPU*: Function to create STD sum images for prestim and stim times on GPU instead of CPU
    - *createSummaryImages*: Creates and saves a variety of summary images from the experimental data from the *prepData* pipeline
    - *prepDataZStack*: Prepares z-stack image runs
    - *prepImagingData*: Prepares imaging data
* metaDataExtraction
  + - *alignEvents2Frames*: Converts event time stamps to nears imaging frame index
    - *checkConsistencyPrairie2PTB*: Checks consistency between stimulus computer events and recorded events in PraireView
    - *extractFieldFromStruct*: Extracts information from structures derived from priaire xml2struct and places it into cell array
    - *extractFromStruct*: Extracts information from structures derived from xml2struct and places it into cell array (only works for single field)
    - *findEvents*: Finds events with decoded event array during analysis pipeline
    - *PrairieVoltageInfo.mat*: Output file from *readEventFileSetupV2* containing all the information for each event voltage level
    - *prepImagingMetaData*: Loads in imaging meta data from the prairie xml file
    - *prepTrialData*: Reads in trial data into experimentStructure and aligns events to frames
    - *readEventFilePrairie*: Reads prairie event file and decodes analogue signal into discrete levels
    - *readEventFilePrairieV2*: Secondary version which is still under development
    - *readScanXMLSettings*: Function to read in galvo scan settings from the raw XML struture
    - *readVoltageFile*: Reads in voltage recording file very quickly
    - *stringEvent2Num*: Converts string events into event number code
    - *xml2struct*: Converts xml to matlab structure
* motionCorrection
  + - *bandpassFermiFilter*: Performs a bandpass fermi-filter at the desired wavelength cutoff. Used in motion correction pipeline
    - *bandpassFermiFilterGPU*: Performs a bandpass fermi-filter at the desired wavelength cutoff on GPU. Used in motion correction pipeline
    - *imageRegistration*: Performs motion correction on image set
    - *shiftImageStack*: Shifts image stack based on previously calculated motion correction vectors
    - *subMicronInPlaneAlignment*: Performs sub-pixel image registration using ManuelGuizar-Sicairos's DFT-based algorithm
    - *subMicronInPlaneAlignmentGPU*: Performs sub-pixel image registration using ManuelGuizar-Sicairos's DFT-based algorithm on GPU

**presentationScripts-** function related to movie creation and displaying images in FIJI

* + - *createMovieFromPrairieFiles*: Creates .avi file from raw prairie recording directory.
    - *displayAllOrientationImages*: Opens and displays all orientation summary images as a FIJI/ImageJ stack so that you can view them easier
    - *displayStackOverlaysFIJI*: Display summary images from different 2P recording runs and overlays their line ROIs. Allows user to check for overlap of dendrites etc.

**PTB\_Experiments-** subfolders involved in stimulus presentation

* calibration
  + - *PTB\_TestColorSquareWave*: Scrap code used to test out colors and gamma correction with psychtoolbox
    - *PTBOrientationColorValuesMonkey*: Holds the cone capture levels for the LCD monitor for the monkey color experiments. Calculated based of *LCDmonitorCalibrations*
    - *PTBOrientationColorValuesMonkeyV2*: Holds the cone capture levels for the LCD monitor for the monkey color experiments. Calculated based of *LCDmonitorCalibrations*. Cut down version to four conditions
    - *PTBSineTest*: Test to display sine gratings at random locations in terms of degrees of degrees of visual angle
    - *PTBTestColors*: Displays defined colors as fullscreen
    - *PTBTestScreenSync*: Displays black & white screen for defined on & off times to see how the screen stim and send events into Prairie sync up
* Experiments
  + - *PTB\_DotMotion*: Runs a random dot motion stimulus for general activation of the visual cortex
    - *PTB\_FlickerStim*: Runs a variety of screen flicker stimulus type, used for stimulating while doing z stack series
    - *PTB\_Melanopsin\_Mnky*: Visual stimulus script with uses Paul's sheeple arduino LED stimulator to flash blue and red light stimulus. Sends events out to the Bruker Prairie 2P computer via Measurement Computing USB-1208FS
    - *PTB\_Melanopsin\_Mnky\_Flash*: Visual stimulus script with uses Paul's sheeple arduino LED stimulator to flash blue light stimulus. Used to test light evoked response during brain injection surgeries
    - *PTB\_Melanopsin\_Mnky\_Plexon*: Visual stimulus script with uses Paul's sheeple arduino LED stimulator to flash blue and red light stimulus. Sends events out to the Plexon system with indivdual bit events (stim ON, stim OFF) via Measurement Computing USB-1208FS
    - *PTB\_RFMappingByHand*: Course manual mapping program, does not output any events but plays moving gabor continously.
    - *PTB\_ZStackFlicker*: Flickers orientation gratings for general activation of visual cortex
    - *PTBContrastVSOri*: Visual stimulus script which presents moving sinusoidal gratings of different orientations and contrasts
    - *PTBOrientation*: Visual stimulus script which presents moving sinusoidal gratings of different orientations
    - *PTBOrientationIntensityMnky*: Experiment which displays moving or static square wave black and white gratings of different intensity (contrast) levels
    - *PTBOrientationSFMnky*: Experiment which displays moving sine gratings of different orientations and spatial frequencies, optimized for monkey
    - *PTBOrientationSFMouse*: Experiment which displays moving sine gratings of different orientations and spatial frequencies, optimized for mouse
    - *PTBOrientationWColorMatchBW\_Mnky*: Experiment which displays moving or static monochrome square wave and specified cone isolating gratings which are luminance matched
    - *PTBOrientationWColorMnky*: Experiment which displays moving or static square wave cone isolating gratings

**Utilities-** functions used in the rest of the scripts

General Pipeline Flow

* prepDataBatchWithOrientationPixelPref(['D:\Data\2P\_Data\Raw\Mouse\gCamp6s\AAV1\_V1\_M1\20211208\']);
* runCaAnalysisWrapper('D:\Data\2P\_Data\Processed\Mouse\gCamp6s\AAV1\_V1\_M1\20211208\');
* calculateOSIPopulation\_wrapper('D:\Data\2P\_Data\Processed\Mouse\gCamp6s\AAV1\_V1\_M1\20211208\');
* plotOrientationTuningPerCell('D:\Data\2P\_Data\Processed\Mouse\gCamp6s\AAV1\_V1\_M1\20211208\',[24 69 99 53]);