

Function Documentation

LabScripts

MATLAB

Compiled by
John Tyson-Carr
5th March 2019

Contents

Plotting_And_Statistics	4
RMANOVA (v1.0)	4
postHocTesting_fullAnalysis (v1.0)	4
createRectangleCoords (v1.0)	5
createSurfaceCube (v1.0)	5
removeImageEmptySpace (v1.0)	5
expandPolygon (v1.0)	6
linearRegression (v1.0)	6
polygonOverlapList (v1.0)	6
scatterLSLine3D (v1.0)	7
minimiseFigWhiteSpace (v1.0)	7
Data_Config	8
stimuliRandomisation (v1.0)	8
findArrayClusters (v1.0)	8
findPeaks (v1.0)	8
nDigitString (v1.0)	9
linearInterp_1D (v1.0)	9
nanUnique (v1.0)	9
normaliseData (v1.0)	10
Pix2VisualAngle (v1.0)	10
roundTo (v1.0)	11
EEG	12
clusterCovar (v1.0)	12
extractEpochNumberInSets (v1.0)	13
combineEvt (v1.0)	13
createEvtFormat (v1.0)	14
evtTriggerExtract (v1.0)	14
extractEpochEvents (v1.0)	15
importEventsToEEGLab (v1.0)	15
inputMissingEvent (v1.0)	15
mergeDatasetsInFolders (v1.0)	16
mergeSubjectComponents (v1.0)	16
MNIPointLoc (v1.0)	17
plotElectrodes (v1.0)	17
plotSigElectrodesOverTime (v1.0)	18

plotHydrocel129 (v1.0)	18
readEvt_6_1 (v1.0)	19
removeEvtAtf (v1.0)	19
rloc128 (v1.0)	19
saveEvt_6_1 (v1.0)	20
sigElectrodes (v1.0)	20
STUDY_subjectClusters (v1.0)	20
Cogent	21
vasScale (v3.0)	21
fixMousePos (v1.0)	22
organiseCoordsForCogent (v1.0)	22
wait (v1.0)	22
cog_FixationCross(v1.0)	23
cog_ImageLoad (v1.0)	23
cog_InsertText (v1.0)	24
Other	25
captureScreen (v1.0)	25
excel_kill (v1.0)	25
ListSubFolders (v1.0)	25
restorePath (v1.0)	25
testImageSimilarity (v1.0)	26

Plotting_And_Statistics

RMANOVA (v1.0)	
This function will carry out a repeated measures ANOVA on a table of data.	
Required Inputs	Description
<i>Data</i>	Data must be a {nSubjects x nVariable} table.
<i>FactorNames</i>	A cell containing the names of each of the factors. For example, factors of {'GENDER' 'AGE'}.
<i>FactorLevels</i>	A cell, the same length as 'FactorNames', but containing the levels for each of the factors. For example, factors of 'GENDER' and 'AGE' would contain: {'MALE'; 'FEMALE'} {'A18-24'; 'A25-30'; 'A30PLUS'}}
<i>LevelIndices</i>	A cell, the same length as 'FactorNames', indicating the factor and level that each variable in 'Data' belongs to.
Optional Inputs	Description
<i>SaveOutput</i>	Name of file to save data to (.txt format). (DEFAULT: [])
Outputs	Description
<i>ANOVATable</i>	A table containing the main effects and interactions between all factors.

postHocTesting_fullAnalysis (v1.0)	
Carries out statistics using the EEGLab statcond function. This function takes a cell array of data and carries out (permutation-based) ANOVAs and all possible post-hoc tests.	
Required Inputs	Description
<i>statData</i>	This is a cell array of data in the same format that is required by the statCond function.
Optional Inputs	Description
<i>Permutation</i>	Whether to carry out permutation analysis, or just regular ANOVA / T-Tests. (DEFAULT: 1)
<i>nPerm</i>	Number of permutation for "Permutation". (DEFAULT: 5000)
<i>RowCondition</i>	Name of conditions in each row. (DEFAULT: {'R01' 'R02' ... 'RXX'})
<i>ColCondition</i>	Name of conditions in each column. (DEFAULT: {'C01' 'C02' ... 'CXX'})
<i>IndividualLevels</i>	Cell array of condition names corresponding to the size of statData. (DEFAULT: {'R01C01' 'R01C02' ... 'RXXCXX'})
<i>ANOVATitle</i>	Title of ANOVA. (DEFAULT: 'ANOVA Results')
<i>PrintToFile</i>	File ID if printing to file. (DEFAULT: [])
<i>PlotData</i>	Plot the data into bar graph. (DEFAULT: 1)
<i>PlotData_Effect</i>	What effect to plot (1 = ROW; 2 = COL; 3 = INT). (DEFAULT: 3)
<i>PlotData_LineType</i>	Whether to plot 'SE' or 'SD'. (DEFAULT: 'SE')
<i>CILineWidth</i>	Width of confidence interval line. (DEFAULT: 3).
<i>CIOneWay</i>	Whether to plot confidence intervals in both directions, or just in the direction of the data. (DEFAULT: 1)

Outputs	Description
<i>postHocStruct</i>	Structure containing information regarding ANOVA and post-hoc T-Tests.
<i>figHandle</i>	Handle for plotting data.

createRectangleCoords (v1.0)	
Given an origin, width and height, this will produce the coordinates for the resulting rectangle.	
Required Inputs	Description
<i>origin</i>	Rectangle origin.
<i>w</i>	Rectangle width.
<i>h</i>	Rectangle height.
Optional Inputs	Description
Outputs	Description
<i>coords</i>	Four points for rectangle corners.

createSurfaceCube (v1.0)	
Given an origin and size of the vertices of the cube, this will compute coordinates across the surface of the cube.	
Required Inputs	Description
<i>centre</i>	Cube origin.
<i>size</i>	Cube vertices size.
<i>distBetweenPoints</i>	Distance between each point on surface.
Optional Inputs	Description
Outputs	Description
<i>coordsSurface</i>	Coordinates of points across cube surface.

removeImageEmptySpace (v1.0)	
Takes a MATLAB variable containing an image in RGB form and removes empty space surrounding the image. This is quite computationally demanding, but it helps quite a lot when visualising images in figures.	
Required Inputs	Description
<i>imageVar</i>	RGB image (see 'imread').
Optional Inputs	Description
Outputs	Description

--

expandPolygon (v1.0)	
This will expand a polygon by a given factor whilst maintaining the centre point.	
Required Inputs	Description
<i>x</i>	X Coordinates for the polygon to be expanded.
<i>y</i>	Y Coordinates for the polygon to be expanded.
<i>factor</i>	The factor by which to expand the polygon (where 1 = same size).
Optional Inputs	Description
Outputs	Description
<i>Xexp</i>	Expanded X Coordinates.
<i>Yexp</i>	Expanded Y Coordinates.

linearRegression (v1.0)	
Carries out linear regression with a single predictor.	
Required Inputs	Description
<i>response</i>	Response variable.
<i>predictor</i>	Predictor variable.
Optional Inputs	Description
Outputs	Description
<i>R</i>	R value for regression.
<i>Rsq</i>	R squared value for regression.

polygonOverlapList (v1.0)	
This will return an index of overlapping polygons. For example, given a list of coordinates (CELL), will sequentially compare coordinates with one another and return all overlapping polygons. Requires Mapping Toolbox.	
Required Inputs	Description
<i>X</i>	Cell containing list of X coordinates.
<i>Y</i>	Cell containing list of Y coordinates.
Optional Inputs	Description
<i>drawPlot</i>	Draw the polygons. (DEFAULT: [])
Outputs	Description
<i>Overlap</i>	Indices indicating the location of any two overlapping polygons.

scatterLSLine3D (v1.0)	
Plot scatter plot in 3D with least-squares line.	
Required Inputs	Description
<i>XYZ</i>	[X Y Z] data to plot.
Optional Inputs	Description
<i>Colour</i>	Line and marker colour.
<i>MarkerSize</i>	Marker size.
Outputs	Description
<i>figHandle</i>	Figure handle.

minimiseFigWhiteSpace (v1.0)	
Minimise the white space on the current axis for saving.	
Required Inputs	Description
Optional Inputs	Description
Outputs	Description

Data_Config

stimuliRandomisation (v1.0)	
<p>Will randomise stimuli. Can take a table, character or numeric array as input. You can also define a seed number if you want to control the random number generator to produce predictable random sequences, or shuffle the random number generator to get different sequences each time.</p> <p>Some versions of MATLAB do not contain the "randperm" function that this script depends on, so it will only work on newer versions.</p>	
Required Inputs	Description
<i>stimList</i>	An array (table, numeric, character).
Optional Inputs	Description
<i>seed</i>	What seed to use for the random number generator. If you need predictable sequences each time this script is used, then input the same seed number. If different sequences are needed, use the "shuffle" parameter. (DEFAULT: 'shuffle')
Outputs	Description
<i>stimListRand</i>	The randomised stimList.
<i>originalIndex</i>	The randomly generated array used for randomisation.

findArrayClusters (v1.0)	
<p>Will take an array of N length and will extract clusters of values that overlap by a pre-defined amount.</p>	
Required Inputs	Description
<i>data</i>	Data array.
<i>overlap</i>	Allowed overlap to be included in cluster.
<i>minSize</i>	Minimum number of data points in cluster.
Optional Inputs	Description
Outputs	Description
<i>clusters</i>	Clusters of data.

findPeaks (v1.0)	
<p>Will find peaks within a given array of Nx1 / 1xN. NOTE THAT NO LOGIC IS GIVEN FOR WHAT SHOULD HAPPEN WHEN A PEAK IS EQUAL FOR TWO CONSECUTIVE TIME POINTS (YET).</p>	
Required Inputs	Description
<i>data</i>	Nx1 or 1xN array of data.

<i>overlap</i>	The script will find peaks and if multiple peaks are found within the latency (Peak Latency +/- overlap), it will find the largest peak and remove the others.
<i>threshold</i>	This will only find peaks that occur above a certain percentage of maximum power. Since the data is normalised between 0 and 1 within this script, the threshold should be between 0 and 1.
<i>norm</i>	Whether to normalise data first between 0 and 1.
Optional Inputs	Description
Outputs	Description
<i>peakLatencies</i>	The latencies at which the data peaks given a certain overlap and threshold.

nDigitString (v1.0)	
Converts a number to a character array of letters with leading zeros to equalise character length. For example, can convert the number 90 to a character array of "0090", or 127 to "0127". Does not work with negative numbers.	
Required Inputs	Description
<i>iteration</i>	Number to convert.
<i>lengthOfString</i>	Length of string to produce. Can also be an array of 1xN / Nx1.
Optional Inputs	Description
Outputs	Description
<i>string</i>	String of lengthOfString length.

linearInterp_1D (v1.0)	
Will take an array of N length and resize it to a length of finalSize. This is done by 1D interpolation.	
Required Inputs	Description
<i>data</i>	Array of N length.
<i>finalSize</i>	Final length of array.
Optional Inputs	Description
Outputs	Description
<i>interpData</i>	Interpolated data.

nanUnique (v1.0)	
Extract unique numbers in an array that has NaN values. The MATLAB built-in function 'unique' will return the number of instances that NaN was found.	
Required Inputs	Description

<i>x</i>	Data array.
Optional Inputs	Description
Outputs	Description
<i>y</i>	Unique values.

normaliseData (v1.0)	
Normalise an array between X and Y.	
Required Inputs	Description
<i>data</i>	Data array.
<i>x</i>	Lower bound.
<i>y</i>	Upper bound.
Optional Inputs	Description
Outputs	Description
<i>normalised</i>	Normalised data.

Pix2VisualAngle (v1.0)	
<p>Given the start and end point of two gaze positions, this function calculates the degrees of visual angle between the two points, and the direction in terms of angle.</p> <p>Optionally, you can input the size of the monitor and the resolution. This will mean you can accurately calculate the size of a single pixel. This will be assumed to be 0.0264583333 if these values are not given.</p>	
Required Inputs	Description
<i>start</i>	Start point of gaze [x y]. Can be a list of points.
<i>finish</i>	End point of gaze [x y]. Can be a list of points.
<i>viewingDistance</i>	Distance from monitor.
Optional Inputs	Description
<i>Plot</i>	Plot the start and end point. (DEFAULT: 0)
<i>Resolution</i>	Screen resolution. (DEFAULT: [])
<i>MonitorSize</i>	Screen size. (DEFAULT: [])
Outputs	Description
<i>amplitude</i>	Visual degrees between two points.
<i>direction</i>	Direction between two points.
<i>imageVarNew</i>	RGB image with no empty space around edges.

roundTo (v1.0)	
Round to the nearest defined amount. For example, round to nearest 0.25. Also has the ability to round up or down with optional argument input.	
Required Inputs	Description
<i>val</i>	Value to round.
<i>increment</i>	Increment to round to.
Optional Inputs	Description
<i>Direction</i>	<div>'up' > Round up to increment.</div> <div>'down' > Round down to increment.</div> <div>'nearest' > Round to nearest increment.</div>
Outputs	Description
<i>newVal</i>	Rounded value.

EEG

clusterCovar (v1.0)

Carries out regression analysis over a specified time interval for clustered independent components. Since multiple components can be contributed by a single subject, components from the same subject are first merged (via summation). This function accepts multiple predictors and utilises the `fitlm()` function.

The function outputs the data into a variable, as well as plots the data across the latency interval with corresponding Adjusted R² Values and P-Values. Note that these values are taken from the highest order of predictors. For example, inputting 3 predictors will produce the following formula:

$$y \sim x_1 + x_2 + x_3 + x_1:x_2 + x_1:x_3 + x_2:x_3 + x_1:x_2:x_3$$

In this scenario, we will plot data in relation to "x1:x2:x3".

Required Inputs	Description
<i>STUDY</i>	EEGLab STUDY structure containing clustered data.
<i>ALLEEG</i>	EEGLab ALLEEG structure.
<i>behaviouralData</i>	behavioural data. This must be in the format of a {1 x nPredictor} cell array of tables, each table being a {nSub x nCond} array. The titles of the tables must correspond to variables within the STUDY.condition variable.
<i>clusters</i>	
Optional Inputs	Description
<i>Latency</i>	Latency over which to carry out regression. (DEFAULT: [])
<i>AverageConditions</i>	Names of conditions in which you may want to average across. This should be a {1 x nAverage} cell array, with each column containing its own cell array of variable names. These variable names must correspond to names in the behaviouralData and the STUDY.condition variable. (DEFAULT: [])
<i>Regression</i>	Whether to carry out regression or correlation. (DEFAULT: 1)
<i>AverageOverLatency</i>	Whether you want to average over the latency input. (DEFAULT: 0)
<i>FDR</i>	False-discovery rate correction (BH-FDR). (DEFAULT: 0)
<i>PredictorNames</i>	Name of predictor variables. (DEFAULT: {})
<i>ResponseName</i>	Name of response variable. (DEFAULT: {})
<i>ForceStudyCond</i>	Force the STUDY.condition variable to be identical to the behaviouralData condition names as indicated by the behaviouralData table VariableNames. USE THIS WITH CAUTION, THIS ASSUMES THAT THE ORDER OF THE VARIABLES IN THE BEHAVIOURAL DATA TABLE CORRESPOND PERFECTLY TO THE STUDY CONDITIONS IN TERMS OF ORDERING. (DEFAULT: 0)
<i>PValYAxis</i>	The range of P-Values to visualize when plotting P-Values over latencies. (DEFAULT: 0.05)
Outputs	Description
<i>OUTPUT</i>	Data extracted from each of the regression analyses taken place, along with the corresponding predictor and response data. The full linear model for each analysis is included.

extractEpochNumberInSets (v1.0)

This function will take a directory that has a list of folders, each corresponding to a subject. Each folder should contain a number of set files corresponding to several conditions. It will load up the set files, and extract the number of epochs each set file contains and return a table.

If you also have other variables encoded for each event within the EEG.event structure, you can extract the mean of these variables for each set file.

If you have several events for each trial, and want to extract the mean of a across trials and not just across all epochs, this function can also extract a single value for each trial if the column indicating trial number is given.

Required Inputs	Description
<i>folder</i>	Folder containing several directories corresponding to subjects.
Optional Inputs	Description
<i>TrialNumColumn</i>	Name of column that indicates trial number. (DEFAULT: [])
<i>Conditions</i>	Name of conditions in set files. (DEFAULT: [])
<i>TrialAverage</i>	Name of columns that we want to obtain a trial average of. (DEFAULT: [])
<i>TrialAverage_ConditionAverage</i>	Cell array, corresponding to the number of variables in the "TrialAverage", with each cell array containing a cell array indicating what condition indices to average over. For example, {[1 2] [3] [4]} {[1] [2] [3] [4]}. (DEFAULT: []).
<i>AllEventAverage</i>	Name of columns that we want to obtain an epoch average of. (DEFAULT: [])
<i>AllEventAverage_ConditionAverage</i>	Cell array, corresponding to the number of variables in the "AllEventAverage", with each cell array containing a cell array indicating what condition indices to average over. For example, {[1 2] [3] [4]} {[1] [2] [3] [4]}. (DEFAULT: []).
<i>SaveAndClose</i>	Whether we want to save the plots to a folder and then close them, or just keep them open. Must be a directory. (DEFAULT: [])
<i>PlotERP</i>	Whether we want to plot the ERP for each condition. (DEFAULT: 1)
<i>PlotEpoch</i>	Epoch to plot. (DEFAULT: 1:size(EEG.data,2))
<i>PlotConditions</i>	Whether to plot conditions for ERPs or just the grand average. (DEFAULT: 0).
<i>ERPDataSave</i>	Where we want to save ERP data. (DEFAULT: [])
<i>EpochInformationSave</i>	Where we want to save Epoch Info. (DEFAULT: [])
<i>ERPOnly</i>	Plot ERP Only. (DEFAULT: 0)
<i>PlotElectrodes</i>	Plot specific electrodes. (DEFAULT: [])
Outputs	Description
<i>epochN</i>	Table indicating number of epochs in each condition.
<i>trialN</i>	Table indicating number of trials in each condition.
<i>trialAverage</i>	Trial average means.
<i>eventAverage</i>	Event average means.
<i>ERPData</i>	ERP Data.

combineEvt (v1.0)

Will combine at least two event files into a single .evt file. Event files must be .evt format exported from BESA, or at least in the same format.	
Required Inputs	Description
<i>saveLoc</i>	Save file for combined event file.
Optional Inputs	Description
<i>varargin</i>	Will take any number of input arguments listing filenames of event files that you wish to combine.
Outputs	Description

createEvtFormat (v1.0)	
Create a variable with the same format of .evt file.	
Required Inputs	Description
<i>timeStamps</i>	Time of events.
<i>code</i>	Code column (usually 1).
<i>triggers</i>	Triggers corresponding to events.
<i>multiply</i>	Whether to multiply the timestamps by 1,000,000 to return the values from seconds to microseconds that BESA produces.
Optional Inputs	Description
Outputs	Description

evtTriggerExtract (v1.0)	
This will extract a single trigger from a .EVT file. It will read the event file and extract all triggers defined by 'trigger', and then extract the trigger(s) from that list defined by 'index'.	
Required Inputs	Description
<i>evtFile</i>	BESA Event File (.evt) from which to extract event(s).
<i>trigger</i>	List of doubles indicating ID number of trigger to extract from event file.
<i>index</i>	When multiple events with same trigger ID are present, index refers to what trigger should be extracted, e.g. 1 = first trigger N, 'end' = last event in list of events, 'all' will return all events matching the triggers.
Optional Inputs	Description
<i>ExtractDIN</i>	Whether to extract the DIN line, rather than the exported trigger. Likely to cause a crash with BESA versions prior to 6.1. (DEFAULT: 0)
Outputs	Description

<i>TS</i>	A list of timestamps for the events that match 'trigger' and 'index'.
<i>CODE</i>	A list of codes for the events that match 'trigger' and 'index'.
<i>TD</i>	A list of trigger IDs for the events that match 'trigger' and 'index'.

extractEpochEvents (v1.0)

When working with epoched data, the EEG.event structure contains multiple repeats of events, since it repeats events that occur repeatedly across multiple epochs.

Required Inputs	Description
<i>EEG</i>	EEG data structure from EEGLab (Must be epoched).
Optional Inputs	Description
Outputs	Description
<i>events</i>	Indices of events that correspond to each epoch.

importEventsToEEGLab (v1.0)

Will import .evt file into EEG.event structure. BESA .evt files are in microseconds, whereas EEGLab takes milliseconds, so it will divide the timestamps by 1000. This function can also encode a value for each event that corresponds to that event. For example, if fixations are encoded, then the saccade amplitude can be encoded into a column in the EEG.event structure. This data needs to be an array in MATLAB and be the same size as the .evt file.

Required Inputs	Description
<i>EEG</i>	EEG data structure from EEG Lab.
<i>evtFile</i>	Event file containing events.
Optional Inputs	Description
<i>Include</i>	Trigger numbers to include.
<i>Exclude</i>	Trigger numbers to exclude.
<i>TriggerInfo</i>	Nested cells containing trigger number and the subsequent variable names for the EEG.event structure. (DEFAULT: []). Example: {{12,'type','fixation','TrialType','Small'} ... {22,'type','fixation','TrialType','Large'}};
<i>{RegressorName,RegressorData}</i>	Any other regressors to input as separate column can be input. To do so, name the column with the parameter and the subsequent variable should be the same size as the events file.
Outputs	Description
<i>EEG</i>	EEG data structure from EEG Lab.

inputMissingEvent (v1.0)

Will input a single event into a .evt file. The event must be in microseconds, for example, 1 second = 1,000,000. This is since .evt files default to microseconds.

Required Inputs	Description
<i>evtFile</i>	Event file to input event.
<i>latency</i>	Latency of event
<i>condition</i>	Trigger number for event.
Optional Inputs	Description
Outputs	Description

mergeDatasetsInFolders (v1.0)

This will take a directory with N folders in it. Each of these folders should correspond to a subject, with each subject folder containing a set file for each condition. This will iterate through each folder and merge the set files into new set files depending on the input. This is good if you want to create set files that are merged across conditions.

Required Inputs	Description
<i>folder</i>	Folder containing subject folders.
<i>fileAppendice</i>	Cell array containing a cell array for each new condition, containing the names of the set files that we wish to merge.
<i>fileAppendiceNew</i>	Cell array of conditions corresponding to the length of fileAppendice, indicating the names of the new conditions.
<i>saveDir</i>	Directory in which new set files will be saved. A new folder for each subject will be made. If this is the same as the folder input, it will save them in the same folder.
Optional Inputs	Description
Outputs	Description

mergeSubjectComponents (v1.0)

Clustered data can contain multiple components from the same subject. However, in order to do statistics, we need to merge the components from the same subject in a cluster. In line with the EEGLab manual, all components from the same subject will be summated to produce a single component for each cluster, although this technically isn't an IC anymore.

Required Inputs	Description
<i>STUDY</i>	EEGLab STUDY data structure with clustered data.
<i>ALLEEG</i>	EEGLab ALLEEG data structure.
<i>cluster</i>	Cluster for which we want to merge components.
Optional Inputs	Description
Outputs	Description
<i>data</i>	Merged data for desired cluster.
<i>STUDY</i>	EEGLab STUDY data structure.

ALLEEG	EEGLab ALLEEG data structure.
--------	-------------------------------

MNIPointLoc (v1.0)	
This will take an MNI coordinate (XYZ) and return the region that it most likely belongs to. This uses an online database and function 'cuixuFindStructure' produced Xu Cui (2007). This function can return the nearest Brodmann area, or nearest GrayMatter, or all regions within a specified cube of NxNxN mm.	
Required Inputs	Description
<i>MNI</i>	MNI coordinate to identify (XYZ).
Optional Inputs	Description
<i>DBFile</i>	The path to the 'TDdatabase.mat' file. Will default to the current dir (DEFAULT: []).
<i>CubeSpace</i>	Whether to extract a cube of regions. (DEFAULT: 0)
<i>CubeSpaceSize</i>	Size of cube (mm). (DEFAULT: 5)
<i>CubeSpaceDist</i>	Distance between points. (DEFAULT: 0.2)
<i>NearestGrayMatter</i>	Whether to extract nearest gray matter. (DEFAULT: 0)
<i>NearestBrodmann</i>	Whether to extract nearest Brodmann area. (DEFAULT: 0)
<i>Search_SizeIteration</i>	When searching through regions for either the nearest Brodmann or Gray Matter, this defines how much to extend the search area by on each iteration. Note that, in order to save computing time, a cube will be created of coordinates and only the surface coordinates will be localised. Each iteration, the cube will increase in size. (DEFAULT: 1)
<i>Search_Dist</i>	Distance between coordinates when searching. (DEFAULT: 0.2)
<i>Search_StopThreshold</i>	Size of cube when we should stop searching. (DEFAULT: 20)
Outputs	Description
<i>COORDSPACE</i>	Structure with information on source localisation.
<i>sourceError</i>	Whether an error was found during search.

plotElectrodes (v1.0)	
Will plot specified electrode numbers on a topographic map using the 'topoplot' function.	
Required Inputs	Description
<i>electrodeNumbers</i>	Numbers of the electrodes to plot on the scalp map.
Optional Inputs	Description
<i>electrodeParam</i>	Parameters that define the appearance of the electrode markers. This parameter takes the numbers of the electrodes (electrodeNumbers), the shape (e.g., s = square), the colour of the markers and the size. (DEFAULT: {electrodeNumbers,'s','red',50}).
<i>electrodeLocations</i>	Electrode locations variable. Importing an electrode location file into EEGLab will give you the variable that you require. However, I put this as an optional input since the "rloc128.m" function produces the same variable for "egihydrocel_129" electrode locations file. (DEFAULT: rloc128)

<i>markerSize</i>	Size of the marker in the plot.
Outputs	Description

plotSigElectrodesOverTime (v1.0)

This will take a [nCond x nSub x nElectrodes x nTimePoints] array and plot significant differences between conditions over time. You can either investigate all differences over the time course, or can specify a latency to investigate differences. You can also average across a specified latency, and save the plots to file.

At the moment, this will likely not work if:

- 1) Data not sampled at 1000 Hz.
- 2) Any of the parameters nElec, nTime, nSub, nCond are equal.
- 3) Electrode locations are different to that of the EGI_HYDROCEL_129.

Required Inputs	Description
<i>Data</i>	Data to plot.
<i>nElec</i>	How many electrodes.
<i>nTime</i>	How many timepoints.
<i>nSub</i>	How many subjects.
<i>nCond</i>	How many conditions.
<i>startLat</i>	When the baseline interval begins.
<i>latencyAverage</i>	How many time points do we include in a single time bin.
Optional Inputs	Description
<i>topoPerFig</i>	How many topographic maps to include in each figure. (DEFAULT: 20)
<i>startPlotLatency</i>	Start point to begin plotting. (DEFAULT: [])
<i>endPlotLatency</i>	End point to begin plotting. (DEFAULT: [])
<i>savePlots</i>	Directory to save plots in. (DEFAULT: [])
<i>sigPVals</i>	When plotting significant differences across the scalp, markers will increase in size for increasingly significant differences. This parameter defines the three P-Values that indicate marker size. (DEFAULT: [0.05 0.01 0.001])
<i>oneMap</i>	Whether to plot single map over specified latency. (DEFAULT: [])
<i>oneMapConds</i>	If plotting oneMap, then the conditions will need to be named, corresponding to the order within the data input. (DEFAULT: [])
Outputs	Description

plotHydrocel129 (v1.0)

Plots the EGI 129 channel hydrocel sensor net with labels.

Required Inputs	Description
Optional Inputs	Description
Outputs	Description

readEvt_6_1 (v1.0)	
Read .evt file exported from BESA. This works with BESA 6.1, since this BESA version produces a 4th column (DIN).	
Author: Andrej Stancak	
Required Inputs	Description
<i>Ename</i>	Event file name.
Optional Inputs	Description
Outputs	Description
<i>E</i>	Matlab array of events.

removeEvtAtf (v1.0)	
Takes an .evt file containing a list of events and recodes event triggers as "999" if it overlaps with the latency of any of the artefacts given in a second event file. This overlap is dependent on the epoch which is given in the format [-100 399], i.e. milliseconds.	
Required Inputs	Description
<i>evt</i>	Event file containing events to be recoded.
<i>evtAtf</i>	Event file containing artefact latencies. Artefact onset and offset must have codes of [21 22] respectively.
<i>epoch</i>	The epoch of the event. If the artefact occurs within the epoch of an event, it is excluded.
Optional Inputs	Description
<i>saveLoc</i>	Save location. If not given, the new .evt file will be put into same location as "evt".
<i>artiMult</i>	Whether to multiply the original trigger by artiMult, or whether to simply recode as 999. (DEFAULT: [])
Outputs	Description
<i>finalSaveLoc</i>	Final save location of event file.

rloc128 (v1.0)	
Will load up the .sfp and .elp for egi_hydrocel_129 sensor net and produce a channel locations variable.	
Author: Andrej Stancak	
Required Inputs	Description
Optional Inputs	Description
<i>elpFile</i>	Location of .elp file.

<i>sfpFile</i>	Location of .sfp file.
Outputs	Description
<i>elpFile</i>	Electrode locations variable.

saveEvt_6_1 (v1.0)

Will save a matrix of values to a .evt file. Matrix must be in .evt format already.

Author: Andrej Stancak

Required Inputs	Description
<i>events</i>	Contains a matrix matching the .evt format to be saved.
<i>saveLoc</i>	The location where new event file is to be saved.
Optional Inputs	Description
Outputs	Description

sigElectrodes (v1.0)

Given an array of P-Values corresponding to electrodes, this will plot the significant P-Values and increase the size of the marker based on the sigSizes input.

Required Inputs	Description
<i>E</i>	Electrode locations file. If empty, this defaults to rloc128.
<i>plotData</i>	An Nx1 array (N = number of electrodes) of P values showing significance at specific electrodes.
<i>sigSizes</i>	P-Values indicating what P-Values will have different size markers, e.g. [0.05 0.01 0.001].
Optional Inputs	Description
Outputs	Description

STUDY_subjectClusters (v1.0)

This will extract the subjects that belong to each cluster in a STUDY design. Data must have been clustered.

Required Inputs	Description
<i>STUDY</i>	EEGLab STUDY data structure.
<i>clusters</i>	Cluster(s) that you want to extract the subjects.
Optional Inputs	Description
Outputs	Description
<i>subjectCluster</i>	Subjects that belong to each cluster.

Cogent

vasScale (v3.0)	
Draw VAS in Cogent. Has ability to draw multiple scales in single call with varying orientations, as well as draw images to screen.	
Required Inputs	Description
<i>origin</i>	Origin of centres of scale. e.g. [0 300; 0 -300]
<i>w</i>	Width of scales in pixels. e.g. [500 300]
<i>h</i>	Height of scales in pixels. e.g. [50 50]
<i>Anchor</i>	Anchors for scales. e.g. { '0' '100' } { 'None' 'All' }
Optional Inputs	Description
<i>orientation</i>	Orientation of the scale. ("horizontal" or "vertical"; DEFAULT: "horizontal")
<i>type</i>	Whether you can drag the scale or only single clicks are registered. ("single", "continuous" or "box"; DEFAULT: "single")
<i>scaleColour</i>	Colour of the scale background. (DEFAULT: [1 1 1])
<i>fillColour</i>	Colour of scale fill. (DEFAULT: [1 0 0])
<i>frontSize</i>	Font size of anchors. (DEFAULT: 48)
<i>incrementLines</i>	Number of increment lines per scale. (DEFAULT: [0; 0; 0; ...])
<i>incrementSize</i>	Length of increment lines. (DEFAULT: 25)
<i>incrementColour</i>	Colour of increment lines. (DEFAULT: [scaleColour])
<i>incrementWidth</i>	Thickness of increment lines. (DEFAULT: 1)
<i>textColour</i>	Colour of anchors. (DEFAULT: [1 1 1])
<i>clickRegister</i>	Where to register clicking ("box" or "increment"; DEFAULT: "box") Current bug known for "increment" parameter wherein the scale will be highlighted beyond the scale box, but only if increment lock is off.
<i>drawOnly</i>	Whether to draw scales only and present, or to draw scales and allow selection. (1 or 0; DEFAULT: 0)
<i>incrementLock</i>	Whether to lock ratings to specific points. (0 or 1; DEFAULT: 0)
<i>incrementLockPoints</i>	How many points to allow rating to lock to. (DEFAULT: 10)
<i>forceChoice</i>	Whether to allow continuing without making a rating. (1 or 0; DEFAULT: 1)
<i>contBoxOrigin</i>	Origin of continue box. (DEFAULT: [550 -450])
<i>contBoxSize</i>	Size of continue box. (DEFAULT: [50 50])
<i>contBoxColour</i>	Colour of continue box. (DEFAULT: [0.5 0.5 0.5])
<i>scaleMaxTime</i>	Max amount of time to present scale for. (DEFAULT: [])
<i>mouseTrack</i>	Whether to track mouse during rating or not. Note that if you want to concatenate MOUSETRACK across several trials, MOUSETRACK will need to be initialized outside of this function, input into this variable as "MOUSETRACK", and output into the global workspace. This allows the same variable to be input into each instance of this function and each subsequent sample will be added to the bottom of the MOUSETRACK.xy structure. Otherwise, leave this parameter empty and a new MOUSETRACK structure will be produced for each instance of this function. (DEFAULT: []).
<i>mouseTrackHz</i>	Sampling rate at which to take mouse samples. (DEFAULT: 256)
<i>cogentSXY</i>	Sprite number to draw image onto screen each time scale is selected as well as X and Y location. Format should be Nx3 array with [SPRITE X Y] on each row for each image. (DEFAULT: [])
<i>cogentImageAlign</i>	Position to which images should be aligned to. (DEFAULT: {'c' 'c'})

<i>fixMousePosition</i>	Position as to which the mouse position should be fixed whilst the scale is being presented. (DEFAULT: [])
<i>fixMousePositionDuration</i>	How long the mouse position should be fixed for. Note that this parameter must be input if the mouse movement is to be limited for a set duration. Otherwise, the mouse position will just be moved to the location. (DEFAULT: [])
Outputs	Description
<i>ratings</i>	The rating for each scale.
<i>MOUSETRACK</i>	The mouse tracking structure containing data of mouse locations.
<i>flipTime</i>	The time (using cogstd) at which the flip command was issued.
<i>boxSync</i>	If scale type 'box' is used, this will report the anchor that is selected for each scale.

fixMousePos (v1.0)	
Fix mouse position for set amount of time.	
Required Inputs	Description
<i>Duration</i>	Seconds to fix mouse for.
<i>x</i>	X coordinate to fix.
<i>y</i>	Y coordinate to fix.
Optional Inputs	Description
Outputs	Description

organiseCoordsForCogent (v1.0)	
Organise coordinates for rectangle comprised of 4 points into clockwise order, with first row being top left point.	
Required Inputs	Description
<i>x</i>	X coordinates of four points of rectangle.
<i>y</i>	Y coordinates of four points of rectangle.
Optional Inputs	Description
Outputs	Description
<i>coordsNew</i>	4x2 array of coordinates, in clockwise order with top left coordinate being first row.

wait (v1.0)	
Wait specified amount of time in seconds. Utilises "cogstd".	
Author: Andrej Stancak	
Required Inputs	Description

<i>deltatime</i>	Time to wait in seconds.
Optional Inputs	Description
Outputs	Description

cog_FixationCross(v1.0)

Plot fixation cross on screen. This function may not require any parameters, as it will automatically detect the information required. It will use monitor size to automatically configure size, and default to the centre of the screen. If more than one monitor is detected, monitor parameter will need to be given.

Required Inputs	Description
Optional Inputs	Description
<i>x</i>	Origin of cross on x-axis. (DEFAULT: 0)
<i>y</i>	Origin of cross on y-axis. (DEFAULT: 0)
<i>monitor</i>	What monitor to display on. (DEFAULT: [])
<i>width</i>	Width of cross. (DEFAULT: Monitor Width / 4)
<i>height</i>	Height of cross. (DEFAULT: Monitor Width / 4)
<i>colour</i>	Colour of cross (RGB). (DEFAULT: [1 1 1])
<i>lineWidth</i>	Width of line. (DEFAULT: 2)
Outputs	Description

cog_ImageLoad (v1.0)

Present image on Cogent window. You can give this function either:

- 1) Image variable read into MATLAB, for example, using imread().
- 2) Character array indicating location of image file.
- 3) Cell array of character arrays indicating location of all image files to be read in.

Required Inputs	Description
<i>sprite</i>	Sprite ID(s) to draw images to.
<i>image</i>	Image variable, character array for image file or cell array of image file locations.
Optional Inputs	Description
<i>resizeX</i>	Width of image in pixels we wish to resize. (DEFAULT: [])
<i>resizeY</i>	Height of image in pixels we wish to resize. (DEFAULT: [])
<i>xPos</i>	X Position of image. (DEFAULT: 0)
<i>yPos</i>	Y Position of image. (DEFAULT: 0)
<i>draw</i>	Whether to flip screen. (DEFAULT: 0)
<i>loadBMP</i>	Whether to use 'cgloadbmp' function instead of loading up an array. (DEFAULT: 0)
Outputs	Description

cog_InsertText (v1.0)	
This will take a single character array and present it on the Cogent window. To do this, it will first split the text into lines depending on a pre-defined wrap width. This wrap width is in number of characters rather than pixels.	
Required Inputs	Description
<i>text</i>	Text to draw.
Optional Inputs	Description
<i>x</i>	X Position for text. (DEFAULT: 0)
<i>y</i>	Y Position for text. (DEFAULT: 0)
<i>font</i>	Text font. (DEFAULT: 'Arial')
<i>fontSize</i>	Text font size. (DEFAULT: 32)
<i>colour</i>	Text colour. (DEFAULT: [1 1 1])
<i>wrapWidth</i>	Wrap width for lines. (DEFAULT: 65)
<i>alignment</i>	Text alignment/ (DEFAULT: {'c' 'c'})
Outputs	Description

Other

captureScreen (v1.0)	
Take screenshot of all screens and store in variable.	
Required Inputs	Description
Optional Inputs	Description
<i>View</i>	Whether to view screenshot in figure. If you want to view it, input view. If not, leave function input parameters empty.
Outputs	Description
<i>IM</i>	Cell array, with length corresponding to number of screens. Each array corresponds to a screenshot of each screen in RGB, with the size corresponding to the resolution of the screen.

excel_kill (v1.0)	
When reading in excel files to MATLAB, excel files can sometimes become non-openable outside of MATLAB because of being open in MATLAB, despite MATLAB being cleared. This will terminate all open excel applications.	
CAUTION: WILL TERMINATE EXCEL APPLICATIONS.	
Required Inputs	Description
Optional Inputs	Description
Outputs	Description

ListSubFolders (v1.0)	
List subfolders in directory.	
Required Inputs	Description
<i>folder</i>	Main directory to look for subfolders.
Optional Inputs	Description
Outputs	Description
<i>subFolds</i>	Subfolders in main folder.

restorePath (v1.0)	
Restore default MATLAB path. Sometimes, the path becomes corrupted and you might be getting warnings when MATLAB starts up, this may fix it.	
Required Inputs	Description

Optional Inputs	Description
Outputs	Description

testImageSimilarity (v1.0)

Given a directory of images, this will scan through all possible pairs of images and test for a high level of similarity. This comes in handy if you are gathering stimuli from multiple sources and images become mixed up.

Required Inputs	Description
<i>directory</i>	Directory containing images.
Optional Inputs	Description
Outputs	Description

(v1.0)	
Required Inputs	Description
Optional Inputs	Description
Outputs	Description