Package 'meaRtools'

May 13, 2017

Type Package

Title R tools for MEA Analysis

Version 1.0.0

Date 2017-05-13

Author Quanli Wang, Sahar Gelfman, Diana Hall, Ryan Dhindsa

Maintainer Sahar Gelfman <sg3261@columbia.edu>

Description

Software tools for the characterization of neuronal networks as recorded on multi-electrode arrays.

Depends R (>= 3.2.2)

Imports lattice,tcltk,emdist,ggplot2 (>= 2.0.0), gridExtra,reshape2,plyr, gtools

License GPL(>=3)

Suggests knitr, rmarkdown

VignetteBuilder knitr

R topics documented:

`calc.burst.distributions`

calculate and plot burst featues distributions

Description

The function calculates normalized distributions of selected bursting features and plots distribution graphs of all treatments in a recording. The function also prints csv output in the /Analysis directory for downstream stats such as permutation test of treatment labels

Usage

```
calc.burst.distributions(s, minVals = 1, xlimit = 25, binsInSec = 5,
feature = "non", filterValuesByMin = 0, minValues = 0, perWell = 0, outputdir = getwd(),
min.electrodes=4, timeStamp="DATE_TIME")
```

Arguments

`s`	MEA data structure
`minVals`	minimum values number per electrode, electrodes with a smaller number of values than that are discarded
`xlimit`	max limit of values, for example: xlimit = 25 for IBI analysis means that IBIs longer than 25 seconds will not be part of distribution calculations
`binsInSec`	how many bins to cut each of the segments. For example: IBI analysis has 25 seconds as xlimit, to analyse in a 0.1 sec resolution binsInSec should be set to 10, for 1 sec resolution set binsInsec to 1

`feature` what feature to analyze, options are "IBI", "ISI, "nspikesInBurst", "duration", "spikesDensityInBurst"

`filterValuesByMin` should analysis disregard values with lower then filterValuesByMin number of values ? (0/1, default is 0). For example, if set to 1 for duration analysis, should analysis consider also bursts shorter than filterValuesByMin ?

`minValues` disregards values with lower then filterValuesByMin , only if filterValuesByMin set to 1

`perWell` should distribution analysis be performed by testing treatment differences on well level means (1) or electrode level means(0)

`outputdir` output directory

`min.electrodes` minimum electrodes for an active well

`timeStamp` time stamp for the output files

Details

Plot distributions calculates normalized distributions of bursting features. 'Normalized distribution' are a way to handle biases caused by noisy electrodes/wells. The function will calculate a normalized histogram (values 0-1) of each feature for each electrode. Next, it will average histogram values either per well and then average all wells per treatment, or directly per treatment. All comparisons between treatments will be then made by plotting the normalized histograms of each treatment and running a K-S test between them.

Note

Output is a made of: 1) Plots of all selected burst features distributions. 2) CSV files ending with _distributions.csv that harbor all electrodes per treatment for all the recordings loaded in the meaRtools pipeline for a specific MEA plate

Author(s)

Sahar Gelfman

Examples

```
# Load exapmple of recording Robject (MEA data structure)
data("S")
feature="IBI";
#calc.burst.distributions(S, minVals = 15, xlimit = 20, binsInSec = 5,
#feature = feature, perWell = 0, outputdir = "/Analysis")
```

`calc.burst.summary` *Calculate average and standard deviation of the bursting features.*

Description

The function calculates a summary of all the bursting features and returns a data.frame with those values.

Usage

```
calc.burst.summary(s, bursty.threshold = 1)
```

Arguments

`s`	MEA data structure
`bursty.threshold`	min number of bursts/minute to count as a bursty unit.

Value

A data frame with the following columns:

`channels`	electrode name
`spikes`	#spikes
`mean.freq`	firing rate (Hz)
`nbursts`	#bursts detected
`bursts.per.sec`	#bursts/second.matrix(nrow=0,ncol=1)
`bursts.per.min`	#bursts/min
`bursty`	is bursts.per.min >bursty.threshold (defaults to 1 burst/min)
`mean.dur`	mean burst duration
`sd.dur`	sd
`mean.spikes`	mean #spikes in a burst
`sd.spikes`	sd
`per.spikes.in.burst`	% of spikes in a burst
`per.spikes.out.burst`	% of spikes not in a burst
`mean.si`	mean Surprise Index (only for poisson .surprise measure)
`mean.isis`	mean ISI within a burst
`sd.mean.isis`	sd
`mean.IBIs`	mean IBI
`sd.IBIs`	sd
`cv.IBIs`	Coefficient of variation of IBI (= mean.IBI/sd.IBI)

Examples

```
# Load exapmple of recording Robject (MEA data structure)
data("S")
S$bs<-calc.burst.summary(S)
```

calculate.burst.features *Filter spikes and bursts in recording objects*

Description

Apply user defined filters on the spikes that were recorded and calculate spike features.

Usage

```
calculate.burst.features(s)
```

Arguments

s	A spikelist object returned from calling calculate.spike.features.

Value

Returns an 's' object containing all the spikes and bursts of all the loaded recording Robjects.

Author(s)

Diana Hall

calculate.isis *Calculate inter spike intervals*

Description

The function calculates all the interspikes interval between all spikes of each of the channels recorded.

Usage

```
calculate.isis(s)
```

Arguments

s	MEA data structure

Value

Returns the MEA data structure (S object in the example) with the following new lists:

S$isis	list of all isis for each channel
S$mean.isis	mean isis for each channel
S$sd.isis	sd of isis for each channel

Examples

```
data("S")
S <- calculate.isis(S)
```

calculate.network.bursts

Compute network bursts for a list of MEA recordings.

Description

For a list of MEA recordings, ususally from the same plate at different time point,This function detects and report network burst features at the well level.

Usage

```
calculate.network.bursts(s,Sigma, min_electrodes, local_region_min_nAE)
```

Arguments

`s`	A list of MEA recordings, typically from the same MEA plate at different time point.
`Sigma`	The window size used to generate network bursts.
`min_electrodes`	Minimum number of electrodes to call a network burst
`local_region_min_nAE`	Indicates if an adaptive threthold method should be used.

Value

Returns an object containing summary, nb.all, nb.features, result, and nb.features.merged.

`summary`	brief summary
`nb.all`	Each well has 3 data frames with nb times, one for each smoothing window
`nb.features`	a list containing a data-frame for each DIV analyzed
`result`	for each DIV analyzed, information on the DIV, times of nb for each well and each smoothing window
`nb.features.merged`	data frame with nb related features averaged across DIVs

Author(s)

Quanli Wang

References

Add reference to Yi-Fan Lu's paper when it is in press.

`calculate.network.spikes` *Compute the netwrok spikes statistics from spike lists.*

Description

Taken a spike list object for a set of electrodes, this function searches network spikes returns a list of all network spikes.

Usage

```
calculate.network.spikes(e, sur = 100,ns.N, ns.T)
```

Arguments

`e`	A spike list object for a set of electrodes.
`sur`	This parameter is related to the number of datapoints to be used in summmarizing mean network spikes, which will be only used for network spike diagnostics. The default value of 100 will usually be sufficient.
`ns.T`	global variable, time window of a network spike
`ns.N`	global variable, minimum number of coincident electrodes

Value

Returns a list of object, containing network spikes.

`wells`	A list of wells that network spikes were found and defined.
`ns.all`	A list of network spikes computed from the spike lists.
`well.layout`	The plate/well layout identified by the function.

References

Need to find the paper describe this method.

`calculate.spike.features` *Filter spikes and bursts in recording objects*

Description

Apply user defined filters on the spikes that were recorded and calculate spike features.

Usage

```
calculate.spike.features(RobjectFiles,  parameters)
```

Arguments

`RobjectFiles` A list of recording Robject files

`parameters` A list of parameters, see data("parameters")

Value

Returns an 's' object containing all the spikes and bursts of all the loaded recording Robjects.

Author(s)

Diana Hall

`dist.perm` *Burst distribution permutations*

Description

Perform two statistical tests to quantify difference between two burst probability distributions using burst probability distribution data. Performed test are the Maximum Distance between cumulative distributions and Earth Movers Distance between the original probability distributions.

Usage

```
dist.perm(datafile,np,type,kotype)
```

Arguments

`datafile` A _distributions.csv input file. Format as the output of calc.burst.distributions

`np` Number of permutations to perform

`type` Name of first genotype

`kotype` Name of second genotype

Value

A list containing results of two statistical tests for the input probability distributions data.

`data.EMD` Original value of EMD distance

`data.EMD` Original value of maximum distance

`perm.EMD` A permuted p.value of the EMD distance

`perm.p` A permuted p.value of the maximum distance

`outp` Maximum distances between genotypes for all permutations performed

`outEMD` Maximum Earth Movers Distance between genotypes for all permutations performed

`data.wt` Cumulative probabilities of the first genotype

`data.ko` Cumulative probabilities of the second genotype

`data.wt.Original` Probabilities distribution of the first genotype

`data.ko.Original` Probabilities distribution of the second genotype

References

See https://redmine.igm.cumc.columbia.edu/projects/mea/wiki for further details

Examples

```
#    result <- dist.perm(distributionFilePath,10000,"WT","KO")
```

---

filter.wells *Filter wells*

Description

Filter out wells for which the number of active electrodes is less than 4, at least 70 percent of the time

Usage

```
filter.wells(unfiltered.df, nae,min.electrodes = 4,
well.filter.maximum.DIV.inactive.ratio = 0.5)
```

Arguments

`unfiltered.df`	Dataframe generated by the spike.features() function
`nae`	A dataframe containing the number of active electrodes for the recording
`min.electrodes`	Minimum number of active electrode to consider a well for analysis
`well.filter.maximum.DIV.inactive.ratio`	The DIV inactive/active well ratio below which a well will be considered active for a set of DIVs

Value

A dataframe identical in format to the input, except that wells that do not meet the filtering criteria are removed.

Examples

```
#data("S")
#data("parameters")
#s<-list(); s[[1]]<-S
# spike.features<-IGM.aggregate.features(s, feat.type="spike", parameters )
# nae = spike.features$nAE
  # filtered.spike.features = lapply(spike.features, function(x) filter.wells(x, nae))
```

`generate.raster.plot` *generate.raster.plot*

Description

Creates a pdf raster plot of selected user selected well from an 's' object. Options include verticle lines showing network spike times, vertical bars showing bursts as well as # showing count of spikes in burst and network spikes.

Usage

```
generate.raster.plot(RobjectFile = NULL, outputdir = NULL, well.for.raster
                 = NULL, interval.for.raster = NULL, show.bursts = F,
                 show.burst.number = F, show.networkspikes = F,
                 show.ns.number = F,  show.nb=F, window.size=NULL)
```

Arguments

`RobjectFile` Default value is NULL, in which case tcltk pop-up file chooser will prompt user to select an 's' object. Otherwise, provide a full path to to a .RData 's' object that contains burst and network data.

`outputdir` A directory (character string in quotes) where pdf is to be saved. Default is NULL, in which case the plot will be saved in the directory RobjectFile location.

`well.for.raster` A well name, character string, from plate. e.g. well.for.raster="A3". Default is NULL, in which case first well in plate will appear in plot.

`interval.for.raster` A vector of min and max time (s) for raster marks. e.g. interval.for.raster=c(30,60) Default is NULL, in which case the whole recording interval will be used.

`show.bursts` A boolean value sets whether bursts are indicated by red horizontal line (TRUE/FALSE) e.g. show.bursts=FALSE Default=FALSE

`show.burst.number` A boolean value sets whether # spikes/bursts are indicated (TRUE/FALSE). show.bursts must be set to true in order that show.burst.number=T e.g. show.burst.number=FALSE Default=FALSE

`show.networkspikes` A boolean value sets whether network spikes are indicated by green vertical line (TRUE/FALSE) e.g. show.networkspikes=FALSE Default=FALSE

`show.ns.number` A boolean value sets whether # electrodes in network spikes are indicated (TRUE/FALSE) e.g. show.ns.number=FALSE Default=FALSE

`show.nb` A boolean value sets whether network bursts should be indicated in raster by orange horizontal lines (TRUE/FALSE) e.g. show.ns.number=FALSE Default=FALSE

`window.size` A numeric value indicating which of the three smoothing sizes available in the R-object should be used in network burst identification e.g. show.ns.number=10Default=NULL

Value

A pdf raster plot will be displayed in system viewer.

Author(s)

Diana Hall

Examples

```
##generate.raster.plot(RobjectFile=NULL,
#                       well.for.raster=NULL,
#                       interval.for.raster=NULL,
#                       show.bursts=F,
#                       show.burst.number=F,
#                       show.networkspikes=F,
#                       show.ns.number=F,
#                       show.nb=F,
#                       window.size=NULL )
```

`get.burst.info` *get burst feature information*

Description

The function returns a list of values of a burst feature for a desired channel

Usage

```
get.burst.info(allb, index)
```

Arguments

`allb`	The bursting features matrix of a channel (located in recording object - S object in example: S$allb[[channel number]]
`index`	Name of the requested burting feature. Can be "beg","end", "IBI","len", "durn", "mean.isis" or "SI"".

Value

List of values of the requested feature (index) for the desired channel.

Examples

```
data("S")
S$allb[[1]]
```

get.data *get.data*

Description

pop up file chooser with caption. Also, sets directory of analysis output.

Usage

```
get.data(caption = "")
```

Arguments

`caption`	text to display in pop-up file chooser to prompt user to select appropriate file. Default is no caption.

Value

Creates 2 directories:

'Analysis' directory in the parent directory of user selected file.

'R_objects' a subdirectory of 'Analysis'

Examples

```
## get.data(caption="Please select a spike-list file for analysis")
```

get.experimental.log.file *get.experimental.log.file*

Description

Extract data from experimental log file: a csv file with columns for well, treatment, dose, size and units.

Usage

```
get.experimental.log.file(file, masterChemFile = masterChemFile)
```

Arguments

`file`	spike-list csv file, one of the possible plate recording file formats available from Axion. Format: one spike and corresponding electrode name per row. See Axion biosystems website for details.
`masterChemFile`	A csv file containing the following columns: "Project", "Experiment.Date", "Plate.SN", "DIV", "Well", "Treatment", "Size", "Dose", and "Units". Empty wells must still be represented in file. If column is irrelavent to a given data set, then 'NA' or blank is sufficient. "Project" columm must match the first character string preceeding "_" in spike-list file name. e.g. exampleRecording_1012016_plate1_DIV1_spike_list.csv". Similarly, "Experiment.Data" and "Plate.SN" must match second and third character strings as separated by "_" in spike-list file name. "DIV" column does not need to be matched.

Value

list containing character vector of experimental log information.

`well`	well name e.g. "A4""
`treatment`	treatment on well e.g. 'WT'
`size`	size information of chemical treatment
`dose`	dose information for treatment
`units`	units of dosage e.g. uL/g

References

See http://www.axionbiosystems.com/products/software/ for details on spike-list csv file format

Examples

```
##masterChemFile<-paste0( system.file(package = "meaRtools"),
#"/data",
#"/exampleRecording_1012016_plate1_expLog.csv" )

##spike.list.file<-paste0( system.file(package = "meaRtools"),
#"/data",
#"/exampleRecording_1012016_plate1_DIV1_spike_list.csv" )

##plate.data<-getxperimental.log.file( file=spike.list.file, masterChemFile = masterChemFile )
```

`get.file.basename` *get.file.basename*

Description

Retreives the first 4 character strings separated by '_' from from a file path to a .RData object.

Usage

```
get.file.basename(filename)
```

Arguments

`filename`	a file name or full file path. filename must have file extension '.RData'.

Details

filename must have file extension '.RData'.

Value

Returns the first 4 character strings separated by '_' from from a file path to a .RData object.

Author(s)

Diana Hall

Examples

```
data("S") # load data
get.file.basename(S$file)
```

get.num.AE *get.num.AE*

Description

Adds a field to a ’s’ spike object ’nAE’ that lists for each well the # of active electrodes (electodes firing > 5spike/minute).

Usage

```
get.num.AE(s2)
```

Arguments

`s2`	an ’s’ object containing spike trains, channel names, etc.

Value

returns ’nAE’ field in ’s’ which is a vector of # of active electrodes (electodes firing > 5 spikes/minute). Each vector entry is named by the well to which the data corresponds.

Author(s)

Diana Hall

Examples

```
data("S") # load data
b<-get.num.AE(S)
b$nAE
```

get.project.plate.name *get.project.plate.name*

Description

returns the first portion of file .RData spike object named according to convention of Project name, experiment date (MMDDYYYY format) and plate serial number separated by a ’_’ as in "exampleRecording_1012016_plate1_DIV1_spike_list.csv" in data package directory.

Usage

```
get.project.plate.name(file)
```

Arguments

`file`	a full file path or file name

Value

Returns a character string of the project name, experiment date and plate serial number in a .RData file path. see example.

Examples

```
data("S") # load data
get.project.plate.name(S$file)
```

---

get.wt *Get WT*

Description

Extracts all treatments/genotypes and allows user to choose single treatment as wild type/reference for downstream analyses

Usage

```
get.wt(s)
```

Arguments

`s`	MEA dataframe structure

Value

A string corresponding to the user's choice

Examples

```
data("S")
s<-list()
s[[1]]<-S
##wt <- get.wt(s)
```

has.network.spikes *A utility function to check if network spikes are detected.*

Description

For an returned object from calcualte.network.spikes, this function provides a utility checking if it contains any network spikes from any well.

Usage

```
has.network.spikes(nspikes)
```

Arguments

nspikes	The network spike object returned by calling calcualte.network.spikes.

Value

Return a boolean value indicating if any network spikes are found from the network spikes object.

IGM.aggregate.features *Aggregate Feature Data*

Description

Takes data from S object (MEA data structure) and makes a list of dataframes. Each dataframe corresponds to one feature, containing values for each well across each DIV of recording

Usage

```
IGM.aggregate.features(s, feat.type, parameters)
```

Arguments

s	MEA data structure
feat.type	Type of features (e.g. "spikes", "ns", "bursts")
parameters	A list of parameters, see data("parameters")

Value

A list of dataframes for a given set of features

Examples

```
data("S")
data("parameters")
s<-list()
s[[1]]<-S
spike.features = suppressWarnings( IGM.aggregate.features(s, "spike", parameters))
ns.features = suppressWarnings( IGM.aggregate.features(s, "ns", parameters) )
burst.features = suppressWarnings( IGM.aggregate.features(s, "burst", parameters) )
```

`IGM.compute.mean.firingrate.by.well` *mean.firingrate.by.well*

Description

Creates a data frame with columns for well firing rate, mean electrode firing rate, well name and DIV. See details for computations.

Usage

```
IGM.compute.mean.firingrate.by.well(s)
```

Arguments

`s`	Well firing rate= total spikes per well/recording time.

Details

Well firing rate= total spikes per well/recording time. Electrode level firing rate= average across all electrodes in a well(total spikes on electrode/recording time)

Examples

```
data("S")

res<-IGM.compute.mean.firingrate.by.well(S)
res[1:4,]
```

`IGM.plot.active.wells.network.spikes` *plot.active.wells.network.spikes*

Description

Plots related to network spike for each well with network spikes in format of users choosing.

Usage

```
IGM.plot.active.wells.network.spikes(nspikes)
```

Arguments

`nspikes`	list of attributes related to network spikes: wells, plate layout and network spike information for each well. See calculate.network.spikes for further details.

Value

returns a multi-page plot.

See Also

calculate.network.spikes xyplot.network.spikes

Examples

```
data("S")
data('parameters')
nspikes <- calculate.network.spikes( S, parameters$sur ,parameters$ns.N, parameters$ns.T )
## pdf(file=NSPlotPath)
## xyplot.network.spikes(nspikes)
## plot.active.wells.network.spikes(nspikes)
## dev.off()
```

IGM.plot.mean.firingrate.by.eletrode.by.div

plot.mean.firingrate.by.eletrode.by.div

Description

Displays average firing rate by well and by electrode for each DIV available.

Usage

```
IGM.plot.mean.firingrate.by.eletrode.by.div(s)
```

Arguments

s	's' object. must be a list, with each DIV a different entry.

Examples

```
data("S")
s<-list()
s[[1]]<-S
#plot.mean.firingrate.by.eletrode.by.div(s)
```

IGM.plot.mean.firingrate.by.well.by.div

plot.mean.firingrate.by.well.by.div

Description

Displays average firing rate by well for each DIV available. First plot well rate in average Hz/electrode and second plot is Hz/total spikes well.

Usage

```
IGM.plot.mean.firingrate.by.well.by.div(s)
```

Arguments

`s`	's' object. must be a list, with each DIV a different entry.

Value

Plot is output, location and path to plot may be controlled by R's plotting apparatus e.g. 'pdf()'

Author(s)

Diana Hall

Examples

```
data("S")
s<-list()
s[[1]]<-S
#plot.mean.firingrate.by.well.by.div(s)
```

`IGM.plot.network.spikes` *Generic method for plotting network spikes.*

Description

The generic plotting function for network spikes.

Usage

```
IGM.plot.network.spikes(ns, ...)
```

Arguments

`ns`	The network spike object returned after running calculate.network.spikes.
`...`	Additional plotting options that is general to plot functions.

Value

None. network spikes related plots will be generated in current plotting device.

IGM.plot.plate.summary.for.bursts *Plot burst features*

Description

Plots all bursting features in a _burst_plot.pdf under the output directory.

Usage

```
IGM.plot.plate.summary.for.bursts(s, outputdir,parameters)
```

Arguments

s	MEA data structure
outputdir	Output directory
parameters	meaRtools basic parameter list

Details

The plot function will plot all the features calculated for the bursts in the recording. Those include: Mean Firing Rate by Plate (Hz), Mean firing rate, Mean Duration, Number of bursts by channel and well, Mean Inter Burst Interval, Mean ISI within bursts, Mean burst per minute, Mean spikes in a burst and % spikes in a burst. The function also calls calc.burst.distributions to calculate and plot burst feature distributions.

Value

A _burst_plot.pdf is printed under the output directory

Examples

```
data("S")
#IGM.plot.plate.summary.for.bursts(S,"/Analysis")
```

IGM.plot.plate.summary.for.spikes *plot.plate.summary.for.spikes*

Description

Diana needs to add document here.

Usage

```
IGM.plot.plate.summary.for.spikes(s, outputdir)
```

Arguments

`s`	's' .RData object. Each DIV must constitute one entry in list.
`outputdir`	directory path where plot will be saved to.

Value

Multiple page plot in pdf format containing data on which electrodes have recorded any spikes, ISI (inter-spike interval) histogram by plate and by well by electrode, log ISI histogram by plate and by electrode, average electrode firing rate by well, & binned electrode firing rate over recording duration.

Author(s)

Diana Hall

Examples

```
data("S")
s<-list()
s[[1]]<-S
## plot.plate.summary.for.spikes(s, outputdir="/Desktop")
```

`IGM.write.UI.to.log` *IGM.write.UI.to.log*

Description

Writes a named list e.g. parameters or file paths to a vector of specified file paths. Useful for documenting the parameters and files used in an analysis.

Usage

```
IGM.write.UI.to.log(files=NULL,parameterList, new.file=F )
```

Arguments

`files`	vector of full file paths to already created .txt files where parameterList content will be written. e.g. files=c('/Desktop/log1.txt', '/Desktop/output/log2.txt)
`parameterList`	A named list of containing pertinent information. An entry of parameterList can itself be a named list, in which case the resulting output file with write the names and contents of each sublist.
`new.file`	new.file=TRUE will overwrite any existing file, restarting the log in effect. Default: new.file=FALSE

Value

writes to .txt files specified.

Author(s)

Diana Hall

Examples

```
# IGM.write.UI.to.log(files='/Desktop/logfile.txt',parameterList, new.file=F )
```

IGM.xyplot.network.spikes — *xyplot for network spikes at the plate level.*

Description

xyplot for network spikes at the plate level. It will detech the plate layout and plot individual wells according to plate layout.

Usage

```
IGM.xyplot.network.spikes(nspikes)
```

Arguments

`nspikes`	The returned obect from calling summary.network.spikes.

Value

Return the handle of xyplot.

Author(s)

Quanli Wang

---

isi — *isi*

Description

calculates the isi (inter-spike interval) (s) between sucessive spikes in a input spike train.

Usage

```
isi(train)
```

Arguments

`train`	spike train: a set of non-decreasing timestamps (s)

Value

a vector of isis: first entry is ISI between first & second spike in input spike train and so forth. Total length is 1 less than input spike train.

Author(s)

Diana Hall

Examples

```
data("S") # load data
b<-isi(S$spikes[[1]])
S$spikes[[1]][1:4]
b[1:3]
```

load.spikelist *Load Robject File*

Description

Loads a previously saved Rdata file of a recording.

Usage

```
load.spikelist(spkDataFile)
```

Arguments

spkDataFile	MEA recording Rdata file

Value

loaded R object

Examples

```
#    s1 <- load.spikelist(dir to saved Rdata file)
```

mi.find.bursts *Find bursts*

Description

For one spike train, find the bursts using the maximum interval method.

Usage

```
mi.find.bursts(spikes,mi.par)
```

Arguments

`spikes`	A spike train of one channel, located in MEA data structure (example S$spikes[[1]]).
`mi.par`	A list of burst features: **beg.isi** Beginning inter spike interval **end.isi** Ending inter spike interval **min.ibi** Minimum inter burst interval to combine bursts **min.durn** Minimum duration to consider as burst **min.spikes** Minimum spikes to consider as burst

Value

Returns a matrix of burst information for a specific channel. Matrix columns are:

`beg`	the number of spike that is first in the burst
`end`	number of the last spike in the burst
`IBI`	time interval from previous burst
`durn`	durarion of burst in seconds
`mean.isis`	average inter spike interval within the burst
`SI`	surprise index, allways 1 for mi algorithm

Author(s)

Stephen Eglen

References

Eytan and Marom (2006) J Neuroscience.

Examples

```
data("S")
allb <- lapply(S$spikes, mi.find.bursts, S$parameters$mi.par )
```

NB.matrix.to.feature.dfs

Convert network burst data matrix to a list of data frames.

Description

Convert network burst data matrix to a list of dataframes. Each dataframe has rows representing wells while columns representing different timepoints(DIVs). The dataframe format alllows well level permutaiton based tests to be done much easier.

Usage

```
NB.matrix.to.feature.dfs(Matrix_and_feature_names)
```

Arguments

`Matrix_and_feature_names`
The data matrix return by calling function calculate.network.bursts.

Value

Returns a list of dataframes, ith each representing a feature matrix, with rows for wells and columns for different timepoints(DIVs).

Author(s)

Quanli Wang

See Also

`calculate.network.bursts`

`parameters` *A list of parameters with default values that user can customize.*

Description

A list of parameters with default values that user can customize.

Usage

```
data("parameters")
```

Format

The format is: List of 20 $ spike.csv : logi TRUE $ spike.plot : logi TRUE $ burst.csv : logi TRUE $ burst.plot : logi TRUE $ burst.type : chr "mi" $ s.min : num 5 $ ns.csv : logi TRUE $ ns.plot : logi TRUE $ elec.min.rate : num 0.0167 $ elec.max.rate : num 1000 $ well.min.rate : num 0 $ mi.par :List of 5 ..$ beg.isi : num 0.1 ..$ end.isi : num 0.25 ..$ min.ibi : num 0.8 ..$ min.durn : num 0.05 ..$ min.spikes: num 5 $ ns.T : num 0.01 $ ns.N : num 3 $ sur : num 100 $ burst.distribution.IBI :List of 7 ..$ perform : num 1 ..$ min.cases : num 15 ..$ x.lim : num 20 ..$ bins.in.seg : num 5 ..$ min.values : num 0 ..$ filter.by.min: num 0 ..$ per.well : num 0 $ burst.distribution.durn :List of 7 ..$ perform : num 1 ..$ min.cases : num 15 ..$ x.lim : num 18 ..$ bins.in.seg : num 10 ..$ min.values : num 0 ..$ filter.by.min: num 0 ..$ per.well : num 0 $ burst.distribution.ISI :List of 7 ..$ perform : num 1 ..$ min.cases : num 15 ..$ x.lim : num 0.5 ..$ bins.in.seg : num 100 ..$ min.values : num 0 ..$ filter.by.min: num 0 ..$ per.well : num 0 $ burst.distribution.nSpikes :List of 7 ..$ perform : num 1 ..$ min.cases : num 5 ..$ x.lim : num 200 ..$ bins.in.seg : num 1 ..$ min.values : num 0 ..$ filter.by.min: num 0 ..$ per.well : num 0 $ burst.distribution.spikeFreq:List of 7 ..$ perform : num 1 ..$ min.cases : num 15 ..$ x.lim : num 300 ..$ bins.in.seg : num 1 ..$ min.values : num 0 ..$ filter.by.min: num 0 ..$ per.well : num 0

Examples

```
data(parameters)
```

permute.features.and.plot *Write PDF*

Description

Generates a PDF containing plots and p-values for each feature. P-values are generating using Mann Whitney and permutation tests. This function requires that you create a list of dataframes for a given feature type (e.g. spikes) using the aggregate.data() function

Usage

```
permute.features.and.plot(s, wt, np, features.list, type, output.dir)
```

Arguments

s	MEA data structure
wt	The treatment that will act as the wildtype/reference for the Mann Whitney and Permutation tests
np	Number of permutations to be performed
features.list	A list of dataframes containing data for a given feature
type	Type of features contained in features.list (e.g. spikes, ns, or bursts)
output.dir	Directory where output files will be generated

Value

A PDF file containing the plots and p-values.

Author(s)

Ryan Dhindsa

Examples

```
data("S")
#spike.features<-IGM.aggregate.features(S, feat.type="spike" )
#wt <- "untreated"
#output.dir = getwd()
#permute.features.and.plot(S, wt, np, spike.features, "spikes", output.dir)
```

read.spikelist *Axion convert spkList to Robject*

Description

Converts the Axion spikList file to a Rdata object and initializes it with all spike and plate info

Usage

```
read.spikelist(key, spkListFile, chem.info ,Robject.dir)
```

Arguments

key	base name of spkList file
spkListFile	The full spkList file name (including path)
chem.info	plate layout information list as loaded using function chem.info.2
Robject.dir	Directory of robject files

Value

save.file	Full path of the saved Robject data file

See Also

chem.info.2

Examples

```
##masterChemFile<-paste0( system.file(package = "meaRtools"),
#"/data",
#"/exampleRecording_1012016_plate1_expLog.csv" )

##spike.list.file<-paste0( system.file(package = "meaRtools"),
#"/data",
#"/exampleRecording_1012016_plate1_DIV1_spike_list.csv" )

##title<-strsplit(basename(spike.list.file), ".csv")[[1]][1]
# get plate chemical info for each file in the list

##plate.chem.info<-chem.info.2( file=spike.list.file, masterChemFile = masterChemFile )

##RobjectFileName<-read.spikelist(key=title,
#                                              spkListFile=plate.chem.info,
#                                         chem.info=plate.chem.info,Robject.dir="/")
```

remove.spikes *remove.spikes*

Description

removes all spikes and associated meta data from ’s’ spike object except those specified by ’ids’.

Usage

```
remove.spikes(s, ids)
```

Arguments

s	’s’ list object, needs to contain a ’spikes’ field with spike train
ids	Name or index of channel(s) to be kept, all other channels removed. either name of channel, e.g. "E5_12" or an vector of idices c(1,2) corresponding to channel index. If a negative index is given, then that channel and associated data will be removed.

Value

’s’ object.

See Also

construct.s

Examples

```
data("S") # load data
r<-remove.spikes(S, c(-1, -2))

S$channels[1:2] # original 's' object first 2 channels
r$channels[1:2] # first 2 channels have been removed

S$NCells # original count of channels
r$NCells # count of channels after 2 channels removed

S$nspikes # original spike count of first 2 channels
r$nspikes # spike count of first 2 channels after 2 channels removed

# OR keep only first 2 channels
t<-remove.spikes(S, c(1, 2))
t$channels
```

S *example 'S' object*

Description

An example 'S' list object containing multiple fields describing 1 minute recording on a 48 well plate.

Usage

```
data("S")
```

Format

channels electrode names

spikes a list of spike trains for each channel

nspikes # spikes for each channel

NCells total # electrodes

meanfiringrate mean firing rate by channel

file full path of file

layout electrodes grid positions for all electrodes on plate

rates list with average count and firing per time.interval (s) as well as plate average

rec.time 2 element vector of first and last spike time of recording

goodwells well names for all wells meeting minimum firing criteria

treatment treatments for each well

size chemical compound size for each treatment

units units of dose of treatment

dose dose of treatment

well well names

nAE # active electrodes (firing>5spikes/min)

cw wells that each channel belongs to

parameters A list of parameters, see data("parameters")

allb for each electrode, a matrix of burst related information

bs burst summary, a data frame containing burst endpoints by electrode

ns.all for each well, a list of network spike information

isis list of inter-spike interval (isi) (s) by channel

mean.isis list of average isi by channel

sd.isis list of standard deviation of isi by channel

well.stats data frame containing well level firing rate information

Details

Created by use of functions available in package.

Examples

```
data('S')
names(S)
```

si.find.bursts *Find bursts*

Description

For one spike train, find the bursts using the Poisson surprise method.

Usage

```
si.find.bursts(spikes,s.min,burst.isi.max)
```

Arguments

spikes	A spike train of one channel, located in MEA data structure (example S$spikes[[1]]).
s.min	A minimum value for the surprise index
burst.isi.max	ISI threshold used by the the surprise index algorithm.

Value

Returns a matrix of burst information for a specific channel. Matrix columns are:

beg	the number of spike that is first in the burst
end	number of the last spike in the burst
IBI	time interval from previous burst
durn	durarion of burst in seconds
mean.isis	average inter spike interval within the burst
SI	surprise index

Author(s)

Stephen Eglen

References

Eytan and Marom (2006) J Neuroscience.

Examples

```
data("S")
allb <- lapply(S$spikes, si.find.bursts, S$parameters$s.min )
```

`summarize.network.spikes` *Generate network spikes based features.*

Description

This function takes the returned object from calcualte.network.spikes function and parse and filter them using customed filters to regenerate features used by IGM MEA projects.

Usage

```
summarize.network.spikes(e, nspikes, ns.E, sur)
```

Arguments

`e`	A spike list object for a set of electrodes.
`nspikes`	The spike list object returned from calling calculate.network.spikes.
`ns.E`	Minumum number of spikes for each electrode within the network spike window. Most IGM MEA projects use a value of 2.
`sur`	This parameter is related to the number of datapoints to be used in summmarizing mean network spikes, which will be only used for network spike diagnostics. The default value of 100 will usually be sufficient.

Value

Returns a new spikes object with filtered and re-calcualted features.

Author(s)

Quanli Wang

`write.features.to.files` *Write feature data to an output file*

Description

Takes in list of dataframes (one per feature) from an MEA data structure that is produced by IGM.aggregate.features and writes output to Files. Each dataframe corresponds to one feature, containing values for each well across each DIV of recording

Usage

```
write.features.to.files(s, features.list, output.dir, type)
```

Arguments

`s`	MEA data structure
`features.list`	list of dataframes, one for each feature.
`output.dir`	Output directory)
`type`	Type of features (e.g. "spikes", "ns", "bursts")

Value

Write one csv per feature for the feature type requested.

Examples

```
data("S")
s<-list()
s[[1]]<-S
  spike.features = IGM.aggregate.features(s, "spike")

#  write.features.to.files(s, spike.features, analysis$output.dir, "spikes")
```

`write.network.spikes.to.csv`

Summarize and write netowrk spikes features into a csv file.

Description

Summarize and write netowrk spikes features into a csv file.

Usage

```
write.network.spikes.to.csv(s, nspikes, outputdir)
```

Arguments

`s`	A list of MEA recordings, typically from the same MEA plate at different time point.
`nspikes`	The spike list object returned from calling calculate.network.spikes.
`outputdir`	The user defined output directory while the cvs file to be writen. There should not have a file sperator at the end of the outputdir.

Value

None.

`write.plate.summary.for.bursts` *Prints bursting features*

Description

The function reads the MEA data structure and uses the 'allb' list built using mi.find.bursts. It then prints all bursting features summary per well and per channel in _bursts.csv and _well_bursts.csv

Usage

```
write.plate.summary.for.bursts(s, outputdir)
```

Arguments

`s`	MEA data structure
`outputdir`	Output directory

Value

Output file _bursts.csv holds all features generated for bursts per well and per channel:

`treatment`	the treatment/genotype based on the experimental log file plan
`well`	well number
`nAE`	number of active electrodes
`nAB`	number of electrodes with bursts
`duration`	total duration of bursts
`mean.dur`	mean duration of bursts
`mean.freq`	firing rate (Hz)
`nbursts`	number of bursts
`bursts.per.sec`	bursts/second.matrix(nrow=0,ncol=1)
`bursts.per.min`	bursts/min
`sd.dur`	sd of burst duration
`mean.freq.in.burst`	average frequency of spikes in a burst
`sd.freq.in.burst`	sd of frequency of spikes in a burst
`mean.spikes.in.burst`	mean number of spikes in a burst
`sd.spikes.in.burst`	sd of number of spikes in a burst
`total.spikes.in.burst`	total number of spikes in a bursts
`per.spikes.in.burst`	percent of spikes in a burst
`mean.ISIs`	mean ISI within a burst

`sd.ISIs`	sd ISI within a burst
`mean.IBIs`	mean IBI
`sd.IBIs`	sd of IBIs
`cv.IBIs`	Coefficient of variation of IBI (= mean.IBI/sd.IBI)
`file`	input recording file

Examples

```
data("S")
d<-dir.create(paste0(getwd(),"/Analysis") )
s<-list(); s[[1]]<-S
write.plate.summary.for.bursts(s, paste0(getwd() ) )
```

`write.plate.summary.for.spikes` *write.plate.summary.for.spikes*

Description

Produces csv output related to firing rate by DIV to directory of user specified output directory

Usage

```
write.plate.summary.for.spikes(s, outputdir)
```

Arguments

`s`	's' spike .RData object. Must be a list with one entry per DIV.
`outputdir`	

Value

One .csv file for each DIV is output and one additional file comprising all DIVs. Quantification of activity levels including total spike count, well and elctrode level firing rate, as well as ISI and standard deviation of applicable features.

Author(s)

Diana Hall

Examples

```
data("S")
s<-list()
s[[1]]<-S
## path<-system.file()
## write.plate.summary.for.spikes(s , path)
```

Index