Package ‘GCalcium’

March 6, 2019

**Type** Package

**Title** A Data Manipulation and Analysis Package for Calcium Indicator Data

**Version** 1.0.0

**Author** Andrew Tamalunas

**Maintainer** Andrew Tamalunas <atamalu@g.clemson.edu>

**Description** Provides shortcuts in extracting useful data points and summarizing waveform data. It is optimized for speed to work efficiently with large data sets so you can get to the analysis phase more quickly. It also utilizes a user-friendly format for use by both beginners and seasoned R users.

**Depends** R (>= 2.10)

**Imports** stats, graphics, caTools, dplyr

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**RoxygenNote** 6.1.0

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-03-06 16:53:27 UTC

### R topics documented:

averaged_trials .................................................. 2
avg_curve_slope .................................................. 3
between_trial_change ........................................... 3
centered_AUC .................................................... 4
consecutive_trial_change ........................................ 5
find_peaks ........................................................ 5
```r

table

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>averaged_trials</td>
<td>Averages signals across trials</td>
</tr>
</tbody>
</table>

**Description**

'averaged_trials' averages values over each time point, across the specified trials.

**Usage**

```r
averaged_trials(Dataframe, Trials)
```

**Arguments**

- **Dataframe**: a GCaMP-format data frame or matrix.
- **Trials**: numbers of trials to be averaged across.

**Value**

a data frame with observation times and averaged values.

**Examples**

```r
## Format data frame
df.new <- format_data(GCaMP)

## Plot the average fluorescence signal across trials 1-5
df.1thru5 <- averaged_trials(df.new, 1:5)
plot(x = df.1thru5$Time, df.1thru5$Values)
```
**avg_curve_slope**

Get average curve slopes

**Description**

‘avg_curve_slope’ finds the average curve slopes for a trial using inflect_points_df and lm

**Usage**

avg_curve_slope(Dataframe, Trial)

**Arguments**

- **Dataframe**
  a GCaMP-format data frame or matrix
- **Trial**
  a single trial number

**Value**

vector of average rate of change for each curve

**Note**

curves of a single value will result in the average slope being labeled as NA

**Examples**

```r
df.new <- format_data(data = GCaMP)
avg_curve_slope(Dataframe = df.new, Trial = 1)
```

**between_trial_change**

Compare mean activity between trials

**Description**

‘between_trial_change’ finds the difference in means during same time range between sets of trials.

**Usage**

between_trial_change(Dataframe, TrialRange1, TrialRange2, Time.period = c(min(Dataframe[1]), max(Dataframe[1])))

**Arguments**

- **Dataframe**
  a GCaMP-format data frame or matrix
- **TrialRange1**
  range of trial numbers to be compared to second set of trials
- **TrialRange2**
  range of trial numbers to be compared to first set of trials
- **Time.period**
  range of time to be compared between sets of trials
centered_AUC

Value

Number representing mean difference of trial set 2 and trial set 1

Examples

# How much did the mean value change 2s after epoch between trials 1-5 and trials 6-10?
df.new <- format_data(data = GCaMP)
between_trial_change(Dataframe = df.new, TrialRange1 = c(1, 5),
TrialRange2 = c(6, 10), Time.period = c(0, 2))

centered_AUC: Compute centered area under curve

Description

‘centered_AUC’ computes the area under the curve (AUC) after centering values using a specified function, such as mean or min. Computed using trapezoidal integration.

Usage

centered_AUC(Dataframe, Trial, FUN = mean)

Arguments

Dataframe: a GCaCaM-format data frame or matrix
Trial: a single trial number
FUN: a function to apply to each window

Value

Data frame of AUCs and curve number for each curve

Examples

# Get AUCs for trial 2, centered at the mean
df.new <- format_data(data = GCaMP)
centered_AUC(Dataframe = df.new, Trial = 2, FUN = mean)
consecutive_trial_change

Compare mean activity in consecutive trials

Description

‘consecutive_trial_change’ finds the change in means between consecutive trials.

Usage

consecutive_trial_change(Dataframe, Trials, Time.period)

Arguments

- Dataframe: a GCaMP-format data frame or matrix
- Trials: range of trial numbers to be compared
- Time.period: range of time to be compared

Value

Data frame with the "Mean.Change" column representing differences in means between trial n and trial n + 1 for the user-inputted range of trials.

Examples

```r
## How much did the mean value change after epoch between consecutive trials in trials 1-10?
df.new <- format_data(data = GCaMP)
consecutive_trial_change(Dataframe = df.new, Trials = c(1, 10), Time.period = c(0, 4))
```

find_peaks

Get peaks/valleys in waveform data

Description

‘find_peaks’ finds peaks or valleys in waveform by using inflection points, with filter of ’n’ increasing/decreasing points on both sides of each inflection point.

Usage

find_peaks(x, n.points = 3)

Arguments

- x: vector of numbers
- n.points: the number of decreasing (peaks) or increasing (valleys) data points on left and right of inflection point required to be considered a "peak". A positive number as an input finds peaks, and a negative number finds valleys.
Value

a numeric vector of indices

Examples

```r
### Format data frame
df.new <- format_data(GCaMP)

### How many peaks are there in trial 1 with 10 decreasing data points on each side?
peak.indices <- find_peaks(df.new$Trial1, n.points = 10)

### When do they occur?
data.frame(times = df.new$Time[peak.indices], vals = df.new$Trial1[peak.indices])
```

---

**format_data**  
Reformat and label data into GCaMP format

Description

‘format_data’ changes the format of data from wide time series to long format with labeled columns. If data is already in long format, the function simply re-labels the columns

Usage

`format_data(data)`

Arguments

data  
A data set with observation times in the first row or column, and observed values from trials in each following row or column.

Value

Data frame with labeled time and trial columns

Examples

```r
df.new <- format_data(GCaMP)
```
**GCaMP**

**Pre-filtered GCaMP calcium activity waveforms**

**Description**

A dataset containing calcium activity and observed times during a behavioral paradigm pilot study; organized into 10 individual trials

**Usage**

data(GCaMP)

**Format**

A data frame with 11 rows and 814 columns

- **Time**: observed times of Trial.n values, in seconds
- **Trial.n**: values of calcium activity signals for trial n ...

**Source**

Dr. David Root, Maier-Watkins laboratory

---

**inflect_points**

*Find inflection points*

**Description**

‘inflect_points’ finds inflection points of waveform data

**Usage**

inflect_points(x)

**Arguments**

- **x**: A vector of numbers

**Value**

Inflection points of the input vector

**Examples**

def.new <- format_data(GCaMP)
inflect_points(def.new$Trial1)
inflect_points_df  Find inflection points and curve numbers

Description

‘inflect_points_df’ finds inflection points of activity waves, returning a summarized data frame.

Usage

inflect_points_df(Dataframe, Trial)

Arguments

Dataframe  a GCalcium-format data frame or matrix
Trial  a single trial number

Value

data frame with variables indicating the time, raw values, curve numbers, and inflection points corresponding to each data point of the input.

Examples

df.new <- format_data(GCaMP)
inflect_points_df(Dataframe = df.new, Trial = 1)

moving_window  Summarize data within windows of time

Description

‘moving_window’ summarizes data within windows of a certain length.

Usage

moving_window(Dataframe, Trial, Window.length, FUN = mean)

Arguments

Dataframe  a GCalcium-format data frame or matrix
Trial  a single trial number
Window.length  length of time each window encompasses
FUN  a function to apply to each window
Value

a data frame with start and stop times of each window, the chronological number of each window, and summarized values

Examples

```R
### Format data frame
df.new <- format_data(GCaMP)

### In trial 5, how does the average fluorescence change in 1 second time frames?
moving_window(Dataframe = df.new, Trial = 5, Window.length = 1, FUN = mean)
```

---

**perc_baseline**

*Find the percent change from baseline*

Description

‘perc_baseline’ calculates the percent change from a user-specified baseline period

Usage

`perc_baseline(Dataframe, Baseline.times, Baseline.frame = FALSE)`

Arguments

- `Dataframe` a GCaMP-format data frame or matrix
- `Baseline.times` range of time from Baseline.frame to compute the baseline value from
- `Baseline.frame` a GCaMP-format data frame or matrix containing the baseline period. If frame is not specified, Dataframe is automatically used

Value

a GCaMP-format data frame with values transformed to percent baseline

Examples

```R
### Format data frame
df.new <- format_data(GCaMP)

### Transform into percent baseline: relative to -3s to -1s before epoc
perc_baseline(Dataframe = df.new, Baseline.times = c(-3, -1))
```
### plot_trials

*Plot specified trials*

**Description**

'plot_trials' plots the user-inputted trials. This function automatically adjusts the x- and y- axis limits to fit all data.

**Usage**

```r
plot_trials(Dataframe, Trials, ...)
```

**Arguments**

- `Dataframe`: a GCaMP-format data frame or matrix
- `Trials`: set of trials to be plotted
- `...`: extra commands to be called to the blank base plot

**Value**

a single plot of user-specified trials

**Examples**

```r
### Format data frame
df.new <- format_data(GCaMP)

### Specify and plot trials
my.trials <- c(1, 2, 7, 8)
plot_trials(Dataframe = df.new, Trials = my.trials)
```

### within_trial_change

*Get mean activity change within a trial*

**Description**

'within_trial_change' finds the change in mean values between beginning and end of entered time range.

**Usage**

```r
within_trial_change(Dataframe, Trial, Beg.period = FALSE,
                     End.period = FALSE)
```
z_score

Arguments

- **Dataframe**: a GCaMP-format data frame or matrix
- **Trial**: a single trial number
- **Beg.period**: time frame of period 1 to be compared to period 2
- **End.period**: time frame of period 2 to be compared to period 1

Value

- vector with AUC for each curve in a trial

Examples

```r
# In trial 1, how much did the mean value change between -2s and 0s, and 0 and 2s?
df.new <- format_data(data = GCaMP)
within_trial_change(Dataframe = df.new, Trial = 1, Beg.period = c(-2, 0), End.period = c(0, 2))
```

---

**Description**

‘z_score’ transforms input values to z scores. Allows user input of mu and sigma values for comparing distributions.

**Usage**

```r
z_score(xvals, mu = FALSE, sigma = FALSE)
```

**Arguments**

- **xvals**: vector of numbers
- **mu**: the population mean
- **sigma**: the population standard deviation

**Value**

- a numeric vector of z scores
Examples

# Format data
df.new <- format_data(GCaMP)

# Split data
basevals <- df.new$Trial1[df.new$Time <= 0]
eventvals <- df.new$Trial1[df.new$Time > 0]

# Find baseline (pre-epoch) values
base.mu <- mean(basevals)
base.sigma <- sd(basevals)

# Compute values
z_score(x = eventvals, mu = base.mu, sigma = base.sigma)
Index

*Topic datasets
  GCaMP, 7

averaged_trials, 2
avg_curve_slope, 3
between_trial_change, 3
centered_AUC, 4
consecutive_trial_change, 5
find_peaks, 5
format_data, 6

GCaMP, 7

inflect_points, 7
inflect_points_df, 8

moving_window, 8

perc_baseline, 9
plot_trials, 10

within_trial_change, 10
z_score, 11