Package 'GENEAsphere'

December 13, 2017

Type Package

Title Visualisation of Raw or Segmented Accelerometer Data

Version 1.4

Date 2017-11-24

Author Mango Solutions [aut, cre],
Joss Langford [aut, cre],
Charles Sweetland [aut, cre],
Activinsights Ltd [cph]

Maintainer Charles Sweetland <charles@Sweetland-solutions.co.uk>

Description Creates visualisations in two and three dimensions of simulated data based on detected segments or raw accelerometer data.

License GPL-3 | file LICENSE

LazyData yes

Depends ggplot2, rgl, MASS, misc3d, GENEAread

Suggests knitr, rmarkdown

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2017-12-13 18:09:09 UTC

R topics documented:

---

plotAccData *plotAccData*

Description

Creates a plot of the Acc Data given a resolution.

Usage

```
plotAccData(x, what = c("sd", "mean", "temperature", "light", "voltage"),
  draw = TRUE, resolution = 200, ...)
```

Arguments

x	should be an Acc Data object.
what	What variable to plot against time. Options are: 1. sd: Standard of movement given the resolutions 2. mean: Mean of movementgiven the resolutions 3. temperature 4. light 5. voltage
draw	if FALSE, plot a whole new plot. Otherwise, superimpose on to existing plot.
resolution	Resolution of plot to create.
...	resolution of underlying grid.

Details

From the raw data to create a representation of light, temperature and MAGSA (Mean Absolute Gravity Substituted Acceleration).

Creates a line plot at a certain resolution from the GENEAread AccData objects available.

Examples

```
## AccData = read.bin(datafile) # where data file is a GENEActiv .bin file.
## saveRDS(AccData , "AccData.rds")
x = readRDS(system.file("extdata", "AccData.rds", package = "GENEAsphere"))
plot.AccData(x, what = ("sd"))
plot.AccData(x, what = ("sd"))
plot.AccData(x, what = ("mean"))
plot.AccData(x, what = ("temperature"))
plot.AccData(x, what = ("light"))
plot.AccData(x, what = ("voltage"))
```

plotSegmentEllipse *Plot an ellipse representation of a segment 'confidence interval'*

Description

Create an ellipse representing a 'confidence interval' of a given segment and plot a projected representation.

Usage

```
plotSegmentEllipse(segmentationCSV, plotRows, projection = "aitoff",
  col = "red", singlePlot = TRUE, confidenceLevel = 0.05,
  alpha = thresholds, wrap = FALSE, greyGrid = FALSE)
```

Arguments

segmentationCSV	The file path to the csv file created from the segmentation process containing all features.
plotRows	The rows from the csv file to be used to simulate plotting data.
projection	The type of projection to be used. Can be any of those used by `mapproject` in the package `mapproj`.
col	A vector of character strings to indicate the colour of each ellipse. If the number of segments is greater than the length of the colours vector then colurs will be repeated.
singlePlot	(logical) Indicates wether all rows should be added to one plot.
confidenceLevel	The alpha value for the confidence interval. A value of 0.05 corresponds to a 95% confidence interval.
alpha	The alpha level to use for plotting colours.
wrap	(logical) Indicating whether segments should be wrapped around the sphere or cropped. By default ellipses are cropped.
greyGrid	(logical) Should the plot be created with a white background and grey grid or a grey background with white grid (default).

Details

This function uses the mean and standard deviation estimates for elevation and rotation of a segment to determine a confidence interval for each direction. This is then used to generate an ellipse representing the two dimensional confidence region. This ellipse is plotted onto a projected representation of the sphere. Required columns are:

1. UpDown.median
2. UpDown.mad
3. Degrees.median
4. Degrees.mad

Value

There is no return to the console. As a side effect a graphic is created.

Examples

```
## Not run:
segmentationCSV = system.file("data", "SegData.csv", package = "GENEAsphere")
plotRows = c(1:1)
plotSegmentEllipse(segmentationCSV, plotRows, projection = "aitoff",
  col = "red", singlePlot = TRUE, confidenceLevel = 0.05,
  alpha = thresholds, wrap = FALSE, greyGrid = FALSE)
## End(Not run)
```

---

plotSegmentFlat *Plot a flat representation*

Description

Create a flat representation of the spherical data.

Usage

```
plotSegmentFlat(segmentationCSV, plotRows, col = c("white", heat.colors(5,
  alpha = c(0.3, 0.2, 0.1, 0.05, 0.03))), singlePlot = TRUE, nsims = 1000)
```

Arguments

segmentationCSV	The file path to the csv file created from the segmentation process containing all features.
plotRows	The rows from the csv file to be used to simulate plotting data.
col	A vector of colours.
singlePlot	(logical) Indicates wether all rows should be added to one plot.
nsims	The number of simulated values to plot. (Default = 1000 however, if your computer has little RAM reduce this) @details This function takes the features from the segmentation procedure and uses them to simulate data for elevation and rotation. This data is then plot on a flat representation of the sphere. Required columns are: 1. UpDown.median 2. UpDown.mad 3. Degrees.median 4. Degrees.mad

Value

There is no return to the console. As a side effect a graphic is created.

Examples

```
segmentationCSV = system.file("data", "SegData.csv", package = "GENEAsphere")
plotRows = c(1:5)
plotSegmentFlat(segmentationCSV, plotRows,
                col = c("white",heat.colors(5, alpha = c(0.3, 0.2, 0.1, 0.05, 0.03))),
                singlePlot = TRUE, nsims= 1000)
```

`plotSegmentProjection` *Plot a projection representation*

Description

Create a projection representation of the spherical data

Usage

```
plotSegmentProjection(segmentationCSV, plotRows, projection = "aitoff",
  col = "red", singlePlot = TRUE, nsims = 1000)
```

Arguments

`segmentationCSV`	The file path to the csv file created from the segmentation process containing all features.
`plotRows`	The rows from the csv file to be used to simulate plotting data.
`projection`	The type of projection to be used. Can be any of those used by `mapproject` in the package `mapproj`.
`col`	A character string to indicate the colour of the heat mapping.
`singlePlot`	(logical) Indicates wether all rows should be added to one plot.
`nsims`	The number of simulated values to plot. (Default = 1000 however, if your computer has little RAM reduce this) @details This function takes the features from the segmentation procedure and uses them to simulate data for elevation and rotation. This data is then plotted on a projected representation of the sphere. Required columns are: 1. UpDown.median 2. UpDown.mad 3. Degrees.median 4. Degrees.mad

Value

There is no return to the console. As a side effect a graphic is created.

Examples

```
## Not run:
segmentationCSV = system.file("data", "SegData.csv", package = "GENEAsphere")
plotRows = c(1:1)
plotSegmentProjection(segmentationCSV, plotRows, projection = "aitoff",
                    col = "red", singlePlot = TRUE, nsims = 1000)
## End(Not run)
```

`plotSegmentSphere` *Create a spherical representation*

Description

From the segmentation features simulate data and create a spherical representation

Usage

```
plotSegmentSphere(segmentationCSV, plotRows, levels = c(0.9, 0.75, 0.5, 0.25,
  0.1), singlePlot = TRUE, col = heat.colors(5), alpha = c(0.03, 0.05,
  0.1, 0.2, 0.3), arrow = FALSE, nsims = 1000)
```

Arguments

`segmentationCSV`	The file path to the csv file created from the segmentation process containing all features.
`plotRows`	The rows from the csv file to be used to simulate plotting data.
`levels`	breakpoints for plotting density.
`singlePlot`	(logical) Indicates wether all rows should be added to one plot.
`col`	A vector of colours.
`alpha`	The range of alpha values to be used for plotting colours
`arrow`	(logical) Indicates whether an arrow for directionality should be added to the plot.
`nsims`	The number of simulated values to plot. (Default = 1000 however, if your computer has little RAM reduce this)

Details

This function takes the features from the segmentation procedure and uses them to simulate data for elevation and rotation. This data is then rotated to give the spherical representation which is ploted on the sphere. Required columns are:

1. UpDown.median
2. UpDown.mad
3. Degrees.median
4. Degrees.mad

Value

There is no return to the console. As a side effect an rgl graphic is created.

Examples

```
segmentationCSV = system.file("data", "SegData.csv", package = "GENEAsphere")
plotRows = c(1:1)
plotSegmentSphere(segmentationCSV, plotRows, levels = c(0.9, 0.75, 0.5, 0.25, 0.1),
                  singlePlot = TRUE, col = heat.colors(5),
                  alpha = c(0.03, 0.05, 0.1, 0.2, 0.3), arrow = FALSE, nsims = 1000)
```

---

plotSphere *Plot an 3D sequence with sedentary sphere*

Description

From the output of the read.bin function in GENEAread to simulate data and create a spherical representation.

Usage

```
plotSphere(x, start = 0, end = 1, length = NULL, time.format = "auto",
  density = FALSE, col, alpha, arrow = TRUE, levels, add = FALSE, ...)
```

Arguments

x	The AccData input to be plotted
start	start time to enter in the format 0 to 1 or "dd hh:mm:ss"
end	end time to enter in the format 0 to 1 or "dd hh:mm:ss"
length	Length of interval.
time.format	Data extraction via get.intervals
density	Whether to plot a 3d density plot.
col	Colours to use for lines or density plot
alpha	Vector of transparencies to user for density plot
arrow	To display a place holder arrow to establish directionality
levels	Breakpoints for plotting of isospheres for density. Follows the formulation of 0.9 == respective isosphere contains the highest probability 10 of the population, according to kernel density estimate.
add	If draw, superimpose on to existing plot. Else add to a new plot.
...	Arguements that will be passed to the

Details

Takes the raw data output of the GENEActiv as AccData to plot points on the sedentary sphere.

Value

There is no return to the console. As a side effect an rgl graphic is created.

Examples

```
## Not run:
x = readRDS(system.file("extdata", "AccData.rds", package = "GENEAsphere"))
plotSphere(x)
## End(Not run)
```

`plotTL` *Produces a line plot of Temperature and Light.*

Description

3 line plot of Temperature and Light.

Usage

```
plotTL(AccData, start = NULL, end = NULL, length = NULL,
  resolution = 100)
```

Arguments

`AccData`	object to plot, can be matrix or AccData object.
`start`	Start of Data to plot. Passed to get.intervals, see codeget.intervals
`end`	End of Data to plot.
`length`	Length of interval
`resolution`	Resolution to plot the data given the data's frequency.

Details

From the raw data to create a representation of Light and Temperature.

Creates a temperature and ligth plot with 2 distinct axis from epoched data. The epoch is dependent on the resolution.

Examples

```
x = readRDS(system.file("extdata", "AccData.rds", package = "GENEAsphere"))
plotTL(x)
```

`positionals` *Positionals*

Description

Creates a positionals plot of the GENEAcitv .bin data.

Usage

```
positionals(AccData, start = 0, end = 1, length = NULL,
  max.points = 1e+06, ...)
```

Arguments

`AccData`	object to plot, can be matrix or accdata object
`start`	time at which to start (Default set at 0).
`end`	time at which to end (Default set at 1).
`length`	Length of interval.
`max.points`	maximum number of data points to plot. Data will be downsampled to achieve this.
`...`	Arguements passed to codeget.intervals

Details

From the raw data to create a representation of arm elevation and wrist rotation.

Value

There is no return to the console. As a side effect an rgl graphic is created.

Examples

```
x = readRDS(system.file("extdata", "AccData.rds", package = "GENEAsphere"))
positionals(x)
```

Index