Package: agcounts (via r-universe)

September 11, 2024

Type Package

Title Calculate 'ActiGraph' Counts from Accelerometer Data

Version 0.6.9

Description Calculate 'ActiGraph' counts from the X, Y, and Z axes of a triaxial accelerometer. This work was inspired by Neishabouri et al. who published the article ``Quantification of Acceleration as Activity Counts in 'ActiGraph' Wearables" on February 24, 2022. The link to the article (<https://pubmed.ncbi.nlm.nih.gov/35831446>) and 'python' implementation of this code (<https://github.com/actigraph/agcounts>).

Depends R (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Imports data.table, gsignal, lubridate, magrittr, Rcpp, GGIR, stats, utils, zoo, reticulate, dplyr, stringr, ggplot2, reactable, shiny, bslib, read.gt3x, DBI, RSQLite

Suggests parallel, devtools, foreach, testthat (>= 3.0.0), shinytest2, covr

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo

URL https://github.com/bhelsel/agcounts

BugReports https://github.com/bhelsel/agcounts/issues

Repository https://bhelsel.r-universe.dev

RemoteUrl https://github.com/bhelsel/agcounts

RemoteRef HEAD

RemoteSha 64f02e5425803646e27d01389a6084267d284e4d

Contents

`agcalibrate` *Calibrate acceleration data*

Description

This function uses a C++ implementation of the GGIR ‘g.calibrate‘ function.

Usage

```
agcalibrate(
  raw,
  verbose = FALSE,
  tz = "UTC",
  imputeTimeGaps = FALSE,
  spherecrit = 0.3,
  sdcriter = 0.013,
  minloadcrit = 168L,
  debug = FALSE,
  ...
)
```

Arguments

`raw`	data frame of raw acceleration data obtained from
`verbose`	Print the progress of the calibration for the raw data, Default: FALSE
`tz`	the desired timezone, Default: `UTC`
`imputeTimeGaps`	Imputes gaps in the raw acceleration data, Default: FALSE
`spherecrit`	The minimum required acceleration value (in g) on both sides of 0 g for each axis. Used to judge whether the sphere is sufficiently populated
`sdcriter`	Criteria to define non-wear time, defined as the estimated noise measured in the raw accelerometer data.
`minloadcrit`	The minimum number of hours the code needs to read for the autocalibration procedure to be effective (only sensitive to multitudes of 12 hrs, other values will be ceiled)
`debug`	print out diagnostic information for C++ code
`...`	Additional arguments to pass into the agread function

Details

This function uses a C++ implementation of the GGIR ‘g.calibrate‘ function to return calibrated raw acceleration data.

Value

Returns the calibrated raw acceleration data

See Also

force_tz

Examples

```
path <- system.file("extdata/example.gt3x", package = "agcounts")
data <- read.gt3x::read.gt3x(path, asDataFrame = TRUE)
data <- agcalibrate(raw = data)
```

---

agcounts *agcounts: R Package for Extracting Actigraphy Counts from Accelerometer Data.*

Description

This R Package reads the X, Y, and Z axes in a GT3X accelerometer file and converts it to Actigraphy counts. This work was inspired by Neishabouri et al. who published the article "Quantification of Acceleration as Activity Counts in ActiGraph Wearables on February 24, 2022. The link to the article and Python implementation of this code https://github.com/actigraph/agcounts.

agcounts functions

get_counts

calculate_counts

agShinyDeployApp

agread *Read in raw acceleration data*

Description

This function reads in raw acceleration data with the pygt3x Python package, the read.gt3x R package with GGIR autocalibration, or the read.gt3x R package.

Usage

```
agread(
  path,
  parser = c("pygt3x", "GGIR", "read.gt3x"),
  tz = "UTC",
  verbose = FALSE,
  ...
)
```

Arguments

path	Path name to the GT3X file or the dataset with columns time, X, Y, and Z axis
parser	The parser to use when reading in the data. Parser values include pygt3x, GGIR, and read.gt3x options.
tz	the desired timezone, Default: UTC
verbose	Print the read method, Default: FALSE.
...	Additional arguments to pass into the agread function

Details

This function reads in raw acceleration data with the pygt3x Python package, the read.gt3x R package with GGIR autocalibration, or the read.gt3x R package.

Value

Returns the raw acceleration data

See Also

g.calibrate read.gt3x

Examples

```
agread(system.file("extdata/example.gt3x", package = "agcounts"), parser = "pygt3x")
```

agShinyDeployApp *agShinyDeployApp*

Description

This function deploys the agcounts Shiny app.

Usage

```
agShinyDeployApp(...)
```

Arguments

...	arguments passed to bs_theme

Details

This function deploys the agcounts Shiny app for data visualization and exploration. It also provides an opportunity to compare ActiGraph counts generated from the agcounts package with those from ActiGraph's .agd files.

Value

Deploys a Shiny app on localhost. No data or values are returned.

See Also

fluidPage, titlePanel, reexports, shinyApp bs_theme

---

calculate_counts *calculate_counts*

Description

Calculate ActiGraph activity counts from raw acceleration data by passing in a data frame with a time stamp, X, Y, and Z axis. This function allows the ability to work with the raw data from other files, but the data frame needs to have "start_time" and "stop_time" attributes. This is different from the get_counts function because it reads a raw data frame rather than a path name to a GT3X file.

Usage

```
calculate_counts(raw, epoch, lfe_select = FALSE, tz = "UTC", verbose = FALSE)
```

Arguments

raw	data frame of raw acceleration data obtained from read.gt3x
epoch	The epoch length for which the counts should be summed.
lfe_select	Apply the Actigraph Low Frequency Extension filter, Default: FALSE
tz	the desired timezone, Default: UTC
verbose	Print the progress of the Actigraph raw data conversion to counts, Default: FALSE.

Value

Returns a data.frame containing the ActiGraph count values

Examples

```
f <- system.file("extdata/example.gt3x", package = "agcounts")
d <- read.gt3x::read.gt3x(f, asDataFrame = TRUE, imputeZeroes = TRUE)
calculate_counts(d, 60)
```

---

get_counts *get_counts*

Description

Main function to extract counts from the Actigraph GT3X Files.

Usage

```
get_counts(
  path,
  epoch,
  lfe_select = FALSE,
  write.file = FALSE,
  return.data = TRUE,
  verbose = FALSE,
  tz = "UTC",
  parser = c("pygt3x", "GGIR", "read.gt3x"),
  ...
)
```

Arguments

path	Full path name to the GT3X File
epoch	The epoch length for which the counts should be summed.
lfe_select	Apply the Actigraph Low Frequency Extension filter, Default: FALSE
write.file	Export a CSV file of the counts, Default: FALSE

return.data	Return the data frame to the R Global Environment, Default: TRUE
verbose	Print the progress of the Actigraph raw data conversion to counts, Default: FALSE.
tz	the desired timezone, Default: UTC
parser	The parser to use when reading in the data. Parser values include pygt3x, GGIR, and read.gt3x options.
...	arguments passed to fwrite

Details

Main function to extract counts from the Actigraph GT3X Files.

Value

Writes a CSV file if write.file is TRUE and returns a data.frame if return.data is TRUE

See Also

read.gt3x

Examples

```
get_counts(
 path = system.file("extdata/example.gt3x", package = "agcounts"),
 epoch = 60, lfe_select = FALSE,
 write.file = FALSE, return.data = TRUE
 )
```

Index