Package 'Sojourn'

May 7, 2021

Type Package

Title Apply Sojourn Methods for Processing ActiGraph Accelerometer Data

Version 1.1.0

Depends R (>= 3.1.0)

Description Provides a simple way for utilizing Sojourn methods for accelerometer processing, as detailed in Lyden K, Keadle S, Staudenmayer J, & Freedson P (2014) <doi:10.1249/MSS.0b013e3182a42a2d>, Ellingson LD, Schwabacher IJ, Kim Y, Welk GJ, & Cook DB (2016) <doi:10.1249/MSS.0000000000000915>, and Hibbing PR, Ellingson LD, Dixon PM, & Welk GJ (2018) <doi:10.1249/MSS.0000000000001486>.

License GPL-3

Encoding UTF-8

LazyData true

Imports dplyr (>= 0.7), magrittr (>= 1.5), lubridate (>= 1.7.4), nnet (>= 7.3), PAutilities (>= 0.2.0), rlang (>= 0.2), stats, svDialogs (>= 1.0), utils, zoo (>= 1.8)

RoxygenNote 7.1.1

URL https://github.com/paulhibbing/Sojourn

BugReports https://github.com/paulhibbing/Sojourn/issues

Suggests data.table, testthat

NeedsCompilation no

Author Paul R. Hibbing [aut, cre], Kate Lyden [aut], Isaac J. Schwabacher [aut]

Maintainer Paul R. Hibbing <paulhibbing@gmail.com>

Repository CRAN

Date/Publication 2021-05-07 20:10:02 UTC

R topics documented:

apply_youth_sojourn . . . 2
compute.bouts.info . . . 3
enhance_actigraph . . . 4
example_data . . . 4
get_youth_sojourns . . . 5
input_demographic . . . 6
read_AP . . . 7
SIP_ag . . . 7
SIP_ap . . . 8
Sojourn . . . 9
sojourn_3x_SIP . . . 9
soj_1x_original . . . 10
soj_3x_original . . . 11

Index **12**

`apply_youth_sojourn` *Apply the youth Sojourn method*

Description

Function for using the youth Sojourn method developed by Hibbing et al. (2018)

Usage

```
apply_youth_sojourn(
  AG,
  vm = c("Vector.Magnitude", "ENMO"),
  Site = c("Hip", "Wrist"),
  demo_interactive = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

`AG`	a data frame of monitor and demographic data
`vm`	the variable to use for processing, either `"Vector.Magnitude"` (for activity counts) or `"ENMO"` (for raw acceleration)
`Site`	the wear location of the monitor, either `"Hip"` or `"Wrist"`
`demo_interactive`	logical. Input demographics interactively if missing variables are identified during format checking?
`verbose`	logical. Print processing updates to the console?
`...`	Further arguments passed to `youth_name_test`

Value

The original data frame, plus additional predictions made by the Sojourn method

Note

The functions `AGread::read_AG_counts` and `AGread::read_AG_raw` are recommended for assembling the monitor-specific portion of the `AG` data frame.

Examples

```
data(example_data, package = "Sojourn")

  results_youth_soj <- apply_youth_sojourn(
    AG = example_data,
    vm = "Vector.Magnitude",
    Site = "Hip"
  )
  utils::head(results_youth_soj)
```

---

compute.bouts.info *Summarize outcomes from data processed using the Sojourn method*

Description

A function to summarize predictions made by the original Sojourn method of Lyden et al. (2014).

Usage

```
compute.bouts.info(est.mets, units = c("secs", "mins"))
```

Arguments

`est.mets`	numeric vector of predicted metabolic equivalents
`units`	time units associated with each row of data

Value

a data frame summarizing the predictions made by the Sojourn method.

Examples

```
data(example_data, package = "Sojourn")
example_data <- soj_3x_original(
  example_data$axis1,
  example_data$axis2,
  example_data$axis3,
  example_data$Vector.Magnitude
```

```
)

compute.bouts.info(example_data$METs)
```

`enhance_actigraph` *Combine ActiGraph and activPAL data*

Description

Merge data streams for separate monitors in the SIP method

Usage

```
enhance_actigraph(ag, ap, verbose = FALSE)
```

Arguments

`ag`	ActiGraph data
`ap`	activPAL data
`verbose`	logical. Print information to console?

Examples

```
data(SIP_ag, package = "Sojourn")
data(SIP_ap, package = "Sojourn")
combined_data <- enhance_actigraph(SIP_ag, SIP_ap)
utils::head(combined_data)
```

`example_data` *Sample data for exploring original and youth Sojourn methods*

Description

Sample data for exploring original and youth Sojourn methods

Usage

```
example_data
```

Format

A data frame with 79989 rows and 11 variables:

id An example ID for the data set

Sex An example sex for the data set

Age An example age (in years) for the data set

BMI An example body mass index (in kg/m^2) for the data set

Timestamp POSIX-formatted variable giving the timestamp for each observation

axis1 Activity counts from the first axis

axis2 Activity counts from the second axis

axis3 Activity counts from the third axis

Vector.Magnitude Vector magnitude of activity counts (sqrt(sum(axis1^2, axis2^2, axis3^2)))

steps Predicted steps taken

incline Inclinometer status (0 = off, 1 = lying, 2 = sitting, 3 = standing)

`get_youth_sojourns` *Label Sojourns in a data stream according to the youth-specific algorithm*

Description

Identify Sojourns using the algorithm of Hibbing et al. (2018)

Usage

```
get_youth_sojourns(
  vm,
  short = 30,
  Output = c("Counts", "Raw"),
  Site = c("Hip", "Wrist"),
  epoch = 1,
  difference = 15,
  threshold = 100,
  verbose = FALSE
)
```

Arguments

`vm`	vector of triaxial accelerometer values, either the vector magnitude for activity counts, or the Euclidian Norm Minus One for raw acceleration
`short`	numeric scalar. Shortest allowable duration for a Sojourn. Should be detected automatically from the internally-stored grid search values for the attachment site (hip or wrist) and data type (counts or raw)

Output	the data type (counts or raw)
Site	the attachment site (hip or wrist)
epoch	the epoch length, in seconds
difference	the difference parameter
threshold	the threshold parameter
verbose	logical. Print processing updates to the console?

Value

A data frame (with nrow equal to length(vm)) that gives sojourn labels and durations

Examples

```
data(example_data, package = "Sojourn")
get_youth_sojourns(example_data$Vector.Magnitude,
  Output = "Counts", Site = "Hip")
```

---

input_demographic *Interactively input demographic information*

Description

Interactively input demographic information

Usage

```
input_demographic(...)
```

Arguments

...	Further arguments passed to svDialogs functions

Value

A data frame containing an ID, attachment site of the monitor (hip or wrist), and the participant's sex, age, and BMI

Examples

```
if (interactive()) {
  input_demographic()
}
```

read_AP *Read an activPAL events file*

Description

Read an activPAL events file

Usage

```
read_AP(filename, tz = "UTC")
```

Arguments

filename	character. Path to the file
tz	character. The timezone to use

Value

Data frame reflecting the data contained in `filename`.

Note

There must be a corresponding `.def` file located in the same directory as `filename`

Examples

```
ap_file <- system.file(
"extdata/sampledata_Events.csv",
package = "Sojourn"
)
if (isTRUE(requireNamespace("data.table"))) {
  ap_data <- read_AP(ap_file)
  utils::head(ap_data)
}
```

SIP_ag *ActiGraph sample data for exploring Sojourns Including Posture (SIP) method.*

Description

ActiGraph sample data for exploring Sojourns Including Posture (SIP) method.

Usage

```
SIP_ag
```

Format

A data frame with 12257 rows and 5 variables:

counts Activity counts from the first axis

axis2 Activity counts from the second axis

axis3 Activity counts from the third axis

vm Vector magnitude of activity counts

Time POSIX-formatted variable giving the timestamp for each observation

---

`SIP_ap` *activPAL sample data for exploring Sojourns Including Posture (SIP) method*

Description

activPAL sample data for exploring Sojourns Including Posture (SIP) method

Usage

```
SIP_ap
```

Format

A data frame with 12257 rows and 5 variables:

Time POSIX-formatted variable giving the timestamp for each observation

DataCount Integer value giving the index of the sample from which the row of data is drawn

Interval Duration (in seconds) of the interval from one data point to the next

ActivityCode Integer giving the posture activity classification: 0 is sedentary, 1 is standing, and 2 is stepping

CumulativeStepCount Integer giving the cumulative step count from the start of the file to the current data point.

ActivityScore Numeric giving MET-hours

`Sojourn` *Sojourn: Apply Sojourn Methods for Processing ActiGraph Accelerometer Data*

Description

The Sojourn package provides a convenient way to apply the family of Sojourn methods published in previous works including Lyden K, Keadle S, Staudenmayer J, & Freedson P (2014) <doi:10.1249/MSS.0b013e3182a42a2d>, Ellingson LD, Schwabacher IJ, Kim Y, Welk GJ, & Cook DB (2016) <doi:10.1249/MSS.0000000000000915>, and Hibbing PR, Ellingson LD, Dixon PM, & Welk GJ (2018) <doi:10.1249/MSS.0000000000001486>.

Details

It is meant for use with data from ActiGraph monitors and (in the case of Sojourns Including Posture, by Ellingson et al. (2016)) activPAL monitors. File reading is not included in the functionality of the Sojourn package. For help with that preliminary step, users are directed to the packages AGread (for ActiGraph files) and activpalProcessing (for activPAL files).

`sojourn_3x_SIP` *Triaxial Sojourn method for the* R*hrefhttps://pubmed.ncbi.nlm.nih.gov/27015380/SIP method*

Description

Triaxial Sojourn method for the SIP method

Usage

```
sojourn_3x_SIP(ag, short = 30)
```

Arguments

`ag`	combined ActiGraph and activPAL data on which to identify transitions and make predictions
`short`	the minimum duration of a qualifying Sojourn

Value

A data frame of processed data using the SIP method

Examples

```
data(SIP_ag, package = "Sojourn")
data(SIP_ap, package = "Sojourn")
data <- Sojourn::enhance_actigraph(SIP_ag, SIP_ap)
utils::head(sojourn_3x_SIP(data))
```

`soj_1x_original` *Invoke the original uni-axial Sojourn method*

Description

Calls the uni-axial Sojourn method from Lyden et al. (2014).

Usage

```
soj_1x_original(
  counts,
  perc.cut = 0.05,
  perc.cut.2 = 0.12,
  perc.cut.3 = 0.55,
  too.short = 10,
  sit.cut = 90,
  long.soj = 120
)
```

Arguments

`counts`	numeric vector of vertical axis counts
`perc.cut`	decision tree threshold 1
`perc.cut.2`	decision tree threshold 2
`perc.cut.3`	decision tree threshold 3
`too.short`	minimum length of one Sojourn
`sit.cut`	cut-point for determining sitting
`long.soj`	maximum length of one Sojourn

Value

a data frame of processed data

Examples

```
data(example_data, package = "Sojourn")
results_1x <- soj_1x_original(example_data$axis1)
utils::head(results_1x)
```

`soj_3x_original` *Invoke the original triaxial Sojourn method*

Description

Calls the triaxial Sojourn method from Lyden et al. (2014).

Usage

```
soj_3x_original(
  counts,
  counts.2,
  counts.3,
  vect.mag,
  short = 30,
  verbose = FALSE
)
```

Arguments

`counts`	numeric vector of activity counts from the first axis
`counts.2`	numeric vector of activity counts from the second axis
`counts.3`	numeric vector of activity counts from the third axis
`vect.mag`	vector magnitude of the activity counts
`short`	minimum length of one Sojourn
`verbose`	logical. Print updates to console?

Value

a data frame of processed data

Examples

```
data(example_data, package = "Sojourn")
results_3x <- soj_3x_original(
  example_data$axis1,
  example_data$axis2,
  example_data$axis3,
  example_data$Vector.Magnitude
)

utils::head(results_3x)
```

Index

∗ **datasets**
example_data, 4
SIP_ag, 7
SIP_ap, 8

apply_youth_sojourn, 2

compute.bouts.info, 3

enhance_actigraph, 4
example_data, 4

get_youth_sojourns, 5

input_demographic, 6

read_AP, 7

SIP_ag, 7
SIP_ap, 8
soj_1x_original, 10
soj_3x_original, 11
Sojourn, 9
sojourn_3x_SIP, 9

youth_name_test, *2*