

Package ‘GGIR’

January 27, 2015

Type Package

Title Raw Accelerometer Data Analysis

Version 1.1-4

Date 2014-11-06

Maintainer Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

Description A tool to process and analyse data collected with wearable raw acceleration sensors. The package has been developed and tested for binary data from GENEActiv and GENEActiv devices and .csv-export data from Actigraph devices. These devices are currently widely used in research on human daily physical activity

License LGPL (>= 2.0, < 3)

Suggests MASS, signal, zoo, mmap, bitops, matlab, GENEActiv

Depends stats, utils, R (>= 2.10)

Author Vincent T van Hees [aut, cre],
Zhou Fang [ctb],
Jing Hua Zhao [ctb],
Severine Sabia [ctb]

NeedsCompilation no

Repository CRAN

Date/Publication 2014-11-06 15:20:44

R topics documented:

GGIR-package	2
data.calibrate	4
data.getmeta	5
data.inspectfile	5
g.analyse	6
g.binread	10
g.calibrate	11
g.getmeta	12
g.impute	15

g.inspectfile	16
g.part1	17
g.part2	19
g.plot	21
g.shell.GGIR	22

Index	26
--------------	-----------

GGIR-package	<i>a package to process multi-day raw accelerometer data</i>
--------------	--

Description

GGIR is an R-package to process multi-day raw accelerometer data. It was developed in the context of research on study human daily physical activity with wearable tri-axial acceleration sensors. The term raw accelerometry refers to data being expressed in m/s² or gravitational acceleration as opposed to the previous generation accelerometers which stored only processed summary measures.

The package has been developed with and for the accelerometer brands Genea and Geneactive. Additionally, it should work for csv data from Geneactiv and Actigraph accelerometer data. Although, I have tested this less thoroughly compared with the binary data formats from Genea and Geneactiv.

Note for Actigraph users: please do not export timestamps to the csv-file as this causes memory issues. To cope with the absence of timestamps the code will re-calculate timestamps from the sample frequency and the start time and date as presented in the file header.

Function [g.inspectfile](#) assesses to which monitor brand the files belongs and extracts the file header; function [g.calibrate](#) helps to investigate calibration error based on free-living data and propose correction factors; function [g.getmeta](#) extracts the signal features and other meta-data; [g.impute](#) takes that information, identifies unreliable signal sections (e.g. monitor not worn or signal clips near its extreme) and replaces these sections by imputed values; and finally [g.analyse](#) takes the output from all the functions, runs a basic descriptive analysis and then summarises the output both per measurement and per day of measurement.

To enhance the feasibility of using these individual functions I am providing a couple of shell functions to ease implementing the above functions in study data by less experience R-users. Here, the main shell function is [g.shell.GGIR](#) and allows for automating the full analysis of a dataset including all necessary calls to the functions above. Function [g.shell.GGIR](#) relies on functions [g.part1](#) and [g.part2](#), also part of this package. In summary, the user is expected to specify the location of the accelerometer data and the desired output folder. Next, data is loaded and pre-processed with [g.getmeta](#) and [g.calibrate](#). Next, the output is converted to a conveniently portable .RData-format away from the R workspace. Next, these .RData files are used as input for [g.part2](#).

Note that [g.part1](#) generates a folder structure to help the user keep track of various output files. The reason why [g.part1](#) and [g.part2](#) are not merged as one generic shell function is because [g.part1](#) takes much longer to run and involves only minor decisions of interest to the movement scientist. Function [g.part2](#) on the other hand is relatively fast and comes with all the decisions that directly impact on the variables that are of interest to the movement scientist. Therefore, the user may want to

run `g.part1` overnight or on a computing cluster, while `g.part2` can then be the main playing ground for the movement scientist. So, function `g.shell.GGIR` basically is the central point for operating both `g.part1` and `g.part2` and most users should not really need `g.part1` or `g.part2`.

If you want to use this package for a different data format (e.g. from a different accelerometer brand) then please provide me with: the R-code to read the data and example files for testing purposes.

Details

Package:	GGIR
Type:	Package
Version:	1.1-4
Date:	2014-11-06
License:	LGPL (>= 2.0, < 3)
Discussion group:	https://groups.google.com/forum/#!forum/rpackageggir

Author(s)

- Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>
- Zhou Fang co-developed function [g.calibrate](#)
- Jing Hua Zhao <jinghua.zhao@mrc-epid.cam.ac.uk> co-developed function [g.binread](#)
- Severine Sabia tested and provided feedback on various functions

References

- van Hees VT, Gorzelniak L, Dean Leon EC, Eder M, Pias M, et al. (2013) Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE 8(4): e61691. doi:10.1371/journal.pone.0061691
- van Hees VT, Fang Z, Langford J, Assah F, Mohammad A, da Silva IC, Trenell MI, White T, Wareham NJ, Brage S. Auto-calibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. J Appl Physiol (1985). 2014 Aug 7

Examples

```
## Not run:
#inspect file:
I = g.inspectfile(datafile)

#autocalibration:
C = g.calibrate(datafile)

#get meta-data:
M = g.getmeta(datafile)
```

```
## End(Not run)

data(data.getmeta)
data(data.inspectfile)
data(data.calibrate)

#impute meta-data:
IMP = g.impute(M = data.getmeta, I = data.inspectfile)

#analyse and produce summary:
A = g.analyse(I = data.inspectfile, C = data.calibrate, M = data.getmeta, IMP)

#plot data
g.plot(IMP, M = data.getmeta, I = data.inspectfile, durplot=4)
```

data.calibrate *Example output from g.calibrate*

Description

C is example output from g.calibrate

Usage

```
data(data.calibrate)
```

Format

The format is: chr "data.calibrate"

Source

The data was collected on one individual for testing purposes

Examples

```
data(data.calibrate)
```

data.getmeta

Example output from g.getmeta

Description

C is example output from g.getmeta

Usage

data(data.getmeta)

Format

The format is: chr "data.getmeta"

Source

The data was collected on one individual for testing purposes

Examples

data(data.getmeta)

data.inspectfile

Example output from g.inspectfile

Description

C is example output from g.inspectfile

Usage

data(data.inspectfile)

Format

The format is: chr "data.inspectfile"

Source

The data was collected on one individual for testing purposes

Examples

data(data.inspectfile)

g.analyse

function to analyse meta-data generated by [g.getmeta](#) and [g.impute](#)

Description

Analyses the output from other functions within the packages to generate a basic descriptive summary for each accelerometer data file. Analyses include: Average acceleration per day, per measurement, L5M5 analyses (assessment of the five hours with lowest acceleration and with highest acceleration). Further, the traditionally popular variable MVPA is automatically extracted in six variants: without bout criteria in combination with epoch length = 'ws3' seconds, 1 minute, and 5 minutes, and for bout durations 1 minute, 5 minutes or 10 minutes in combination with a 5 sec epoch.

Usage

```
g.analyse(I, C, M, IMP, qllevels = c(), qwindow = c(0, 24), quantiletype = 7,
L5M5window = c(0, 24), M5L5res = 10, includedaycrit = 16, ilevels = c(),
winhr = 5, idloc = 1,snloc=1,mvpathreshold = c(),boutcrit=c())
```

Arguments

I	the output from function g.inspectfile
C	the output from function g.calibrate
M	the output from function g.getmeta
IMP	the output from function g.impute
qllevels	array of percentiles for which value needs to be extracted. These need to be expressed as a fraction of 1, e.g. c(0.1, 0.5, 0.75). There is no limit to the number of percentiles. If left empty then percentiles will not be extracted. Distribution will be derived from short epoch metric data, see g.getmeta .
qwindow	start and end time, in 24 hour clock hours, over which distribution in metric values need to be extracted. Value = c(0,24) will consider all 24 hours.
quantiletype	type of quantile function to use (default recommended). For details, see quantile function in STATS package
L5M5window	start and end time, in 24 hour clock hours, over which L5M5 needs to be calculated. The calculation is done based on the average day
M5L5res	resolution of L5 and M5 analysis in minutes (default: 10 minutes)
includedaycrit	minimum required number of valid hours in day specific analysis (NOTE: there is no minimum required number of hours per day in the summary of an entire measurement, every available hour is used to make the best possible inference on average metric value per week)
ilevels	Levels for acceleration value frequency distribution in mg, e.g. c(0,100,200) There is no constricton to the number of levels.
winhr	window size in hours of L5 and M5 analysis (dedault = 5 hours)

idloc	If value = 1 (default) the code assumes that ID number is stored in the obvious header field. If value = 2 the code uses the character string preceding the character '_' in the filename as the ID number
snloc	If value = 1 (default) the code assumes that device serial number is stored in the obvious header field. If value = 2 the code uses the character string between the first and second character '_' in the filename as the serial number
mvpthreshold	Threshold for MVPA estimation. Threshold needs to be based on metric ENMO. This can be a single number or an array of numbers, e.g. c(100,120). In the later case the code will estimate MVPA separately for each threshold. If this variable is left blank c() then MVPA is not estimated
boutcriter	The variable boutcriter is a number between 0 and 1 and defines what fraction of a bout needs to be above the mvpthreshold

Details

The value summary is a dataframe and comes with the following variables:

- ID Participant id extracted from file header
- device_sn Device serial number extracted from file header
- dodylocation Body location extracted from file header
- filename Name of the binary file
- start_time Timestamp when experiment started
- startday Name of day when experiment started
- samplefreq Sample frequency (Hz)
- device Name of the device brand, e.g. Geneactiv
- clipping_score Fraction of 15 minute windows per file for which the acceleration in one of the three axis was close to the maximum for at least 80 percent of the time. This should be 0
- meas_dur_dys Measurement duration (days)
- complete_24hrcycle Fraction of 15 minute windows per 24 hours for which valid data is available at any day of the measurement
- meas_dur_def_proto_day Measurement duration (days) minus the hours that are ignored at the beginning and end of the measurement motivated by protocol design
- wear_dur_def_proto_day Measurement duration according to protocol (days) minus invalid time periods
- calib_err Estimated based on all non-movement periods in the measurement after applying the autocalibration
- calib_status Summary statement about the status of the calibration error minimisation
- ENMO ENMO is the main summary measure of acceleration. The value presented is the average ENMO over all the available data normalised per 24 hour cycles, with invalid data imputed by the average at similar timepoints on different days of the week. In addition to ENMO it is possible to extract other acceleration metrics (i.e. BFEN, HFEN, HFENplus)
- pX_ENMO_mg_0-24h This variable represents the Xth percentile in the distribution of short epoch acceleration values of the average day within the time interval as specified.

- L5hr_ENMO_mg_0-24 Starting time in hours of the least active five* hours within the time interval as specified (* window size is modifiable in g.getmeta)
- L5_ENMO_mg Average acceleration over L5
- M5hr_ENMO_mg_0-24 Starting time in hours of the most active five* hours in the day within the time interval as specified (* window size is modifiable in g.getmeta)
- M5_ENMO_mg_0-24 Average acceleration over M5
- Accelerationa 1am-6am value of ENMO (mg) Average acceleration between 1am and 6am
- N valid WEdays Number of valid weekend days
- N valid WDdays Number of valid week days
- AD_... The variable ... was calculated per day and then averaged over all the available days
- WE_... The variable ... was calculated per day and then averaged over weekend days only
- WD_... The variable ... was calculated per day and then averaged over week days only
- WWE_... The variable ... was calculated per day and then averaged over weekend days. Double weekend days are averaged This is only relevant for experiments that last for more than seven days
- WWD_... The variable ... was calculated per day and then averaged over week days. Double weekend days were averaged. This is only relevant for experiments that last for more than seven days)
- ..._MVPA_E5S_B1M80_T100 MVPA calculated based on 5 second epoch setting bout duration 1 Minute and inclusion criterion of more than 80 percent. This is only done for metric ENMO at the moment, and only if mvpathreshold is not left blank
- ..._mean_ENMO... ENMO or other metric was first calculated per day and then average according to AD, WD, WWE, WWD
- data exclusion strategy A log of the decision made when calling g.impute: value=1 mean ignore specific hours; value=2 mean ignore all data before the first midnight and after the last midnight
- n hours ignored at the start of the measurement (if strategy = 1) A log of the decision made when calling g.impute
- n hours ignored at the end of the measurement (if strategy = 1) A log of the decision made when calling g.impute
- n days of measurement after which data is ignored (if strategy = 1) A log of the decision made when calling g.impute

The value dayssummary is a dataframe and comes with the following variables:

- ID Participant id extracted from file header
- filename File name
- calender_date Calendar data
- bodylocation Body location (if known)
- N valid hours Number of hours with valid data
- N hours Number of hours of measurement

- Day of the week Day of the week
- Day of measurement Day number relative to start of the measurement
- L5_ENMO_mg_0-24h Magnitude of average acceleration during the least active five hours calculated with metric ENMO. Within the time window as specified
- L5hr_ENMO_mg_0-24h Starting hour of L5 on a scale from 0 to 24, where 14.5 means 14:30. Within the time window as specified
- M5_ENMO_mg_0-24h Magnitude of average acceleration during the most active five hours calculated with metric ENMO. Within the time window as specified
- M5hr_ENMO_mg_0-24h Starting hour of M5 on a scale from 0 to 24, where 14.5 means 14:30. Within the time window as specified
- mean_ENMO_mg_1-6am Mean acceleration between 1am and 6am
- mean_ENMO_mg_24hr Mean acceleration over 24 hour period
- pX_ENMO_mg_0-24h Percentile in the short epoch distribution (ws3) with invalid data imputed. Within the time window as specified
- [A,B)_ENMO_mg_0-24h Time spent in minutes between (and including) acceleration value A in mg and (excluding) acceleration value B in mg. This is only done for metric ENMO at the moment, and only done if ilevels is not left blank
- MVPA_E5S_B1M80_T100 MVPA calculated based on 5 second epoch setting bout duration 1 Minute and inclusion criterion of more than 80 percent. This is only done for metric ENMO at the moment, and only if mvpathreshold is not left blank

Value

summary	summary for the file that was analysed (see details)
daysummary	summary per day for the file that was analysed (see details)

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

Examples

```
## Not run:
#inspect file:
I = g.inspectfile(datafile)

#autocalibration:
C = g.calibrate(datafile, use.temp = TRUE, spherecrit = 0.3, minloadcrit = 72,

#get meta-data:
M = g.getmeta(datafile, desiredtz = "Europe/London", window sizes = c(5, 900, 3600),
daylimit = FALSE, offset = c(0, 0, 0), scale = c(1, 1, 1), tempoffset = c(0, 0, 0),
do.bfen = FALSE, do.enmo = TRUE, do.angle = FALSE, do.lfenmo = FALSE, do.en = FALSE,
do.hfen = FALSE, do.hfenplus = FALSE, lb = 0.2, hb = 15, n = 4)

## End(Not run)
```

```

data(data.getmeta)
data(data.inspectfile)
data(data.calibrate)

#impute meta-data:
IMP = g.impute(M = data.getmeta, I = data.inspectfile)

#analyse and produce summary:
A = g.analyse(I = data.inspectfile, C = data.calibrate, M = data.getmeta, IMP)

```

g.binread	<i>function to read binary files as produced by the accelerometer named 'Genea', not to be confused with the 'GENEActiv' (see package GENEAREad for this)</i>
-----------	---

Description

For reading the binary data as collected with a Genea accelerometer (Unilever Discover, UK). For reading GENEActive binary data, see package GENEAREad.

Usage

```
g.binread(binfile, start = 0, end = 0)
```

Arguments

binfile	filename (required)
start	start point for reading data, this can either be a timestamp "year-month-day hr:min:sec" or a page number (optional)
end	end point for reading data, this can either be a timestamp "year-month-day hr:min:sec" or a page number (optional)

Details

If only start is defined then g.binread will read all data beyond start until the end of the file is reached

Value

rawxyz	matrix with raw x, y, and, z acceleration values
header	file header
timestamps1	timestamps for rawxyz in seconds since 1970-01-01 00:00
timestamps2	timestamps for rawxyz in day time format
batt.voltage	matrix with battery voltage and corresponding timestamps

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk> Jing Hua Zhao <jinghua.zhao@mrc-epid.cam.ac.uk>

g.calibrate	<i>function to estimate calibration error and make recommendation for addressing it</i>
-------------	---

Description

Functions starts by identifying ten second windows of non-movement. Next, the average acceleration per axis per window is used to estimate calibration error (offset and scaling) per axis. The function provides recommended correction factors to address the calibration error and a summary of the callibration procedure.

Usage

```
g.calibrate(datafile, use.temp = TRUE, spherecrit = 0.3, minloadcrit = 72,
printsummary = FALSE, chunksize=c())
```

Arguments

datafile	name of binary file
use.temp	use temperature sensor data if available (Geneactive only)
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
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). After loading these hours only extra data is loaded if calibration error has not be reduced to under 0.01 g.
printsummary	if TRUE will print a summary when done
chunksize	number between 0.2 and 1 to specifiy the size of chunks to be loaded as a fraction of a 12 hour period, e.g. 0.5 equals 6 hour chunks. The default is 1 (12 hrs). For machines with less than 4Gb of RAM memory a value below 1 is recommended.

Value

scale	scaling correction values, e.g. c(1,1,1)
offset	offset correction values, e.g. c(0,0,0)
tempoffset	correction values related to temperature, e.g. c(0,0,0)
cal.error.start	absolute difference between Euclidean norm during all non-movement windows and 1 g before autocalibration

cal.error.end	absolute difference between Euclidean norm during all non-movement windows and 1 g after autocalibration
spheredata	average, standard deviation, Euclidean norm and temperature (if available) for all ten second non-movement windows as used for the autocalibration procedure
npoints	number of 10 second no-movement windows used to populate the sphere
nhoursused	number of hours of measurement data scanned to find the ten second time windows with no movement
meantempcal	mean temperature corresponding to the data as used for autocalibration. Only applies to data collected with GENEActiv monitor.

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk> Zhou Fang

References

- van Hees VT, Fang Z, Langford J, Assah F, Mohammad A, da Silva IC, Trenell MI, White T, Wareham NJ, Brage S. Auto-calibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. *J Appl Physiol* (1985). 2014 Aug 7

Examples

```
## Not run:
datafile = "C:/myfolder/testfile.bin"

#Apply autocalibration:
C = g.calibrate(datafile)
print(C$scale)
print(C$offset)

## End(Not run)
```

g.getmeta

function to extract meta-data (features) from data in binary file

Description

Reads a binary file in blocks, extracts various features and stores average feature value per short or long epoch. Acceleration and angle metrics are stored at short epoch length. The non-wear indication score, the clipping score, temperature (if available), light (if available), and Euclidean norm are stored at long epoch length. The function has been designed and thoroughly tested with binary files from GENEActiv and GENEActiv. Further, the function should be able to cope with csv-format data procuded by GENEActiv and Actigraph

Usage

```
g.getmeta(datafile, desiredtz = "Europe/London",
windowsizes = c(5, 900, 3600),daylimit = FALSE,
offset = c(0,0,0), scale = c(1,1,1),
tempoffset = c(0,0,0),do.bfen = FALSE, do.enmo = TRUE,
do.angle = FALSE, do.lfenmo = FALSE, do.en = FALSE,
do.hfen = FALSE, do.hfenplus = FALSE,
do.telindert2013=FALSE,do.anglex=FALSE,do.angley=FALSE,do.anglez=FALSE,
do.enmoa = FALSE,
lb = 0.2,hb = 15, n = 4,meantempcal=c(),chunksize=c())
```

Arguments

datafile	name of binary file
desiredtz	desired timezone: see also http://en.wikipedia.org/wiki/Zone.tab
windowsizes	c(Short epoch length in seconds, Long epoch length in seconds, Window size for assessing non-wear time in seconds)
daylimit	number of days to limit (roughly), if set to FALSE no daylimit will be applied
offset	offset correction value per axis, usage: value = scale(value,center = -offset, scale = 1/scale)
scale	scaling correction value per axis, usage: value = scale(value,center = -offset, scale = 1/scale)
tempoffset	temperature offset correction value per axis, usage: value = scale(value,center = -offset, scale = 1/scale) + scale(temperature, center = rep(averagetemperature,3), scale = 1/tempoffset)
do.bfen	if TRUE, calculate metric BFEN with band-pass filter configuration set by lb and hb
do.enmo	if TRUE (default), calculate metric ENMO with negative values rounded to zero
do.angle	if TRUE, calculate metric angle using only the y-axis
do.lfenmo	if TRUE, calculate metric LFENMO with low-pass filter configuration set by hb
do.en	if TRUE, calculate metric EN
do.hfen	if TRUE, calculate metric HFEN with low-pass filter configuration set by hb
do.hfenplus	if TRUE, calculate metric HFENplus with band-pass filter configuration set by lb and hb
do.telindert2013	if TRUE, calculate the 5 sec epoch Actiwatch count replication as described by te Lindert et al 2013 in the journal SLEEP (volume 36, issue 5, page 781)
do.anglex	if TRUE, calculate the angle of the x-axis relative to the horizontal plane (degrees) utilizing all three axes
do.angley	if TRUE, calculate the angle of the y-axis relative to the horizontal plane (degrees) utilizing all three axes
do.anglez	if TRUE, calculate the angle of the z-axis relative to the horizontal plane (degrees) utilizing all three axes

<code>do.enmoa</code>	if TRUE (default), calculate metric ENMOa which is equal to metric ENMO but with the absolute taken from the Euclidean norm minus one.
<code>lb</code>	lower boundary of the frequency filter (in Hertz)
<code>hb</code>	upper boundary of the frequency filter (in Hertz)
<code>n</code>	order of the frequency filter
<code>meantempcal</code>	mean temperature corresponding to the data as used for autocalibration. Only applies to data collected with GENEActiv monitor. If autocalibration is not done or if temperature was not available then leave blank (default)
<code>chunksize</code>	number between 0.2 and 1 to specify the size of chunks to be loaded as a fraction of a 24 hour period, e.g. 0.5 equals 12 hour chunks. The default is 1 (24 hrs). For machines with less than 4Gb of RAM memory a value below 1 is recommended.

Value

<code>metalong</code>	dataframe with long epoch meta-data: EN, non-wear score, clipping score, temperature
<code>metashort</code>	dataframe with short epoch meta-data: timestamp and metric
<code>tooshort</code>	indicator of whether file was too short for processing (TRUE or FALSE)
<code>corrupt</code>	indicator of whether file was considered corrupt (TRUE or FALSE)

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

References

van Hees VT, Gorzelniak L, Dean Leon EC, Eder M, Pias M, et al. (2013) Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE 8(4): e61691. doi:10.1371/journal.pone.0061691

Examples

```
## Not run:
datafile = "C:/myfolder/testfile.bin"

#Extract meta-data:
M = g.getmeta(datafile)

#Inspect first couple of rows of long epoch length meta data:
print(M$metalong[1:5,])

#Inspect first couple of rows of short epoch length meta data:
print(M$metashort[1:5,])

## End(Not run)
```

g.impute	<i>function to identify invalid periods in the meta-data as generated by g.getmeta and to impute these invalid periods with the average of similar timepoints on other days of the measurement</i>
----------	--

Description

Functions takes the output from [g.getmeta](#) and information about the study protocol to impute the data.

Usage

```
g.impute(M, I, strategy = 1, hrs.del.start = 0, hrs.del.end = 0, maxdur = 0,
ndayswindow = 7)
```

Arguments

M	output from g.getmeta
I	output from g.inspectfile
strategy	how to deal with knowledge about study protocol. value = 1 means select data based on hrs.del.start, hrs.del.end, and maxdur. Value = 2 makes that only the data between the first midnight and the last midnight is used for imputation. Value = 3 only selects the most active X days in the files. X is specified by argument ndayswindow
hrs.del.start	how many HOURS after start of experiment did wearing of monitor start?
hrs.del.end	how many HOURS before the end of the experiment did wearing of monitor definitely end?
maxdur	how many DAYS after start of experiment did experiment definitely stop? (set to zero if unknown = default)
ndayswindow	If strategy is set to 3 then this is the size of the window as a number of days

Value

metashort	imputed short epoch variables
rout	matrix to clarify when data was imputed for each long epoch time window and the reason for imputation. Value = 1 indicates imputation. Columns 1 = monitor non wear, column 2 = clipping, column 3 = additional nonwear, column 4 = protocol based exclusion and column5 = sum of column 1,2,3 and 4.
averageday	matrix with n columns for n metrics values and m rows for m short epoch time windows in an average 24 hours period

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

Examples

```
## Not run:
#inspect file:
I = g.inspectfile(datafile)

#autocalibration:
C = g.calibrate(datafile)

#get meta-data:
M = g.getmeta(datafile)

## End(Not run)

data(data.getmeta)
data(data.inspectfile)

#impute meta-data:
IMP = g.impute(M=data.getmeta, I=data.inspectfile)
```

g.inspectfile	<i>function to inspect binary accelerometer file for brand, sample frequency and header</i>
---------------	---

Description

Inspects binary accelerometer file for key information, including: monitor brand, sample frequency and file header

Usage

```
g.inspectfile(datafile)
```

Arguments

datafile	name of data file
----------	-------------------

Value

header	fileheader
monn	monitor name (genea, geneactive)
monc	monitor brand code (1 = genea; 2 = geneactive, 3 = actigraph)
dformn	data format (bin, csv)
dformc	data format code (1 = bin, 2 = csv)
sf	samplefrequency in Hertz
filename	filename

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

g.part1

function to load and pre-process acceleration files

Description

Calls function [g.getmeta](#) and [g.calibrate](#), and converts the output to .RData-format which will be the input for [g.part2](#). Here, the function generates a folder structure to keep track of various output files. The reason why these [g.part1](#) and [g.part2](#) are not merged as one generic shell function is because [g.part1](#) takes much longer to and involves only minor decisions of interest to the movement scientist. Function [g.part2](#) on the other hand is relatively fast and comes with all the decisions that directly impact on the variables that are of interest to the movement scientist. Therefore, the user may want to run [g.part1](#) overnight or on a computing cluster, while [g.part2](#) can then be the main playing ground for the movement scientist. Function [g.shell.GGIR](#) provides the main shell that allows for operating [g.part1](#) and [g.part2](#).

Usage

```
g.part1(datadir=c(),outputdir=c(),f0=1,f1=c(),windowsizes = c(5,900,3600),
        desiredtz = "Europe/London",chunksize=c(),studyname=c(),
        do.enmo = TRUE,do.angle = FALSE,
        do.lfenmo = FALSE,do.en = FALSE,
        do.bfen = FALSE,do.hfen=FALSE,
        do.hfenplus = FALSE,
        do.telindert2013=FALSE,do.anglex=FALSE,do.anglely=FALSE,
        do.anglez=FALSE,do.enmoa=FALSE,do.cal = TRUE,
        lb = 0.2, hb = 15, n = 4,
        use.temp=TRUE,spherecrit=0.3,minloadcrit=72,
        printsummary=FALSE,print.filename=FALSE)
```

Arguments

datadir	Directory where the accelerometer files are stored or list of accelerometer file-names and directories
outputdir	Directory where the output needs to be stored. Note that this function will attempt to create folders in this directory and uses those folder to organise output
f0	File index to start with (default = 1). Index refers to the filenames sorted in increasing order
f1	File index to finish with (defaults to number of files available)
windowsizes	see g.getmeta
desiredtz	see g.getmeta
chunksize	see g.getmeta

studyname	If the datadir is a folder then the study will be given the name of the data directory. If datadir is a list of filenames then the studyname will be used as name for the analysis
do.bfen	if TRUE, calculate metric BFEN with band-pass filter configuration set by lb and hb, see g.getmeta
do.enmo	if TRUE (default), calculate metric ENMO, see g.getmeta
do.angle	if TRUE, calculate metric angle using only the y-axis
do.lfenmo	if TRUE, calculate metric LFENMO with low-pass filter configuration set by hb, see g.getmeta
do.en	if TRUE, calculate metric EN, see g.getmeta
do.hfen	if TRUE, calculate metric HFEN with low-pass filter configuration set by hb, see g.getmeta
do.hfenplus	if TRUE, calculate metric HFENplus with band-pass filter configuration set by lb and hb, see g.getmeta
do.teLindert2013	if TRUE, calculate the 5 sec epoch Actiwatch count replication as described by te Lindert et al 2013 in the journal SLEEP (volume 36, issue 5, page 781)
do.anglex	if TRUE, calculate the angle of the x-axis relative to the horizontal plane (degrees) utilizing all three axes
do.angley	if TRUE, calculate the angle of the y-axis relative to the horizontal plane (degrees) utilizing all three axes
do.anglez	if TRUE, calculate the angle of the z-axis relative to the horizontal plane (degrees) utilizing all three axes
do.enmoa	if TRUE (default), calculate metric ENMOa which is equal to metric ENMO but with the absolute taken from the Euclidean norm minus one.
do.cal	Whether to apply auto-calibration or not, see g.calibrate . Default and recommended setting is TRUE
lb	lower boundary of the frequency filter (in Hertz)
hb	upper boundary of the frequency filter (in Hertz), see g.getmeta
n	order of the frequency filter, see g.getmeta
use.temp	see g.calibrate use temperature sensor data if available (Geneactive only)
spherecrit	see g.calibrate 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
minloadcrit	see g.calibrate 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). After loading these hours only extra data is loaded if calibration error has not be reduced to under 0.01 g.
printsummary	see g.calibrate if TRUE will print a summary when done
print.filename	Whether to print the filename before before analysing it (default is FALSE). Printing the filename can be useful to investigate problems (e.g. to verify that which file is being read).

Arguments

metadatadir	Directory where the output from g.part1 was stored
f0	File index to start with (default = 1). Index refers to the filenames sorted in increasing order
f1	File index to finish with (defaults to number of files available)
strategy	how to deal with knowledge about study protocol. value = 1 means select data based on hrs.del.start, hrs.del.end, and maxdur. Value = 2 makes that only the data between the first midnight and the last midnight is used for imputation, see g.impute
hrs.del.start	how many HOURS after start of experiment did wearing of monitor start?, see g.impute
hrs.del.end	how many HOURS before the end of the experiment did wearing of monitor definitely end?, see g.impute
maxdur	how many DAYS after start of experiment did experiment definitely stop? (set to zero if unknown = default), see g.impute
includedaycrit	minimum required number of valid hours in day specific analysis (NOTE: there is no minimum required number of hours per day in the summary of an entire measurement, every available hour is used to make the best possible inference on average metric value per week)
L5M5window	start and end time, in 24 hour clock hours, over which L5M5 needs to be calculated. The calculation is done based on the average day
M5L5res	resolution of L5 and M5 analysis in minutes (default: 10 minutes)
winhr	window size in hours of L5 and M5 analysis (default = 5 hours)
qwindow	start and end time, in 24 hour clock hours, over which distribution in metric values need to be extracted. Value = c(0,24) will consider all 24 hours.
qlevels	array of percentiles for which value needs to be extracted. These need to be expressed as a fraction of 1, e.g. c(0.1, 0.5, 0.75). There is no limit to the number of percentiles. If left empty then percentiles will not be extracted. Distribution will be derived from short epoch metric data, see g.getmeta .
ilevels	Levels for acceleration value frequency distribution in mg, e.g. c(0,100,200) There is no constriction to the number of levels.
mvpthreshold	Threshold for MVPA estimation. Threshold needs to be based on metric ENMO. This can be a single number or an array of numbers, e.g. c(100,120). In the later case the code will estimate MVPA separately for each threshold. If this variable is left blank c() then MVPA is not estimated
boutcriter	The variable boutcriter is a number between 0 and 1 and defines what fraction of a bout needs to be above the mvpthreshold
ndayswindow	If strategy is set to 3 then this is the size of the window as a number of days
idloc	If value = 1 (default) the code assumes that ID number is stored in the obvious header field. If value = 2 the code uses the character string preceding the character '_' in the filename as the ID number
do.imp	Whether to impute missing values (e.g. suspected of monitor non-wear) or not by g.impute . Default and recommended setting is TRUE

Value

The function provides no values, it only ensures that other functions are called and that their output is stored in the folder structure as created with [g.part1](#).

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

References

- van Hees VT, Gorzelniak L, Dean Leon EC, Eder M, Pias M, et al. (2013) Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE 8(4): e61691. doi:10.1371/journal.pone.0061691
- van Hees VT, Fang Z, Langford J, Assah F, Mohammad A, da Silva IC, Trenell MI, White T, Wareham NJ, Brage S. Auto-calibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. J Appl Physiol (1985). 2014 Aug 7

Examples

```
## Not run:  
metadatadir = "C:/myresults/output_mystudy"  
g.part2(metadatadir)  
  
## End(Not run)
```

`g.plot` *function to generate a plot for quality check purposes*

Description

Function takes meta-data as generated by [g.getmeta](#) and [g.impute](#) to create a visual representation of imputed time periods

Usage

```
g.plot(IMP, M, I, durplot)
```

Arguments

IMP	output from g.impute
M	output from g.getmeta
I	output from g.inspectfile
durplot	number of days to plot

Value

function only produces a plot, no values

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

Examples

```
## Not run:
#inspect file:
I = g.inspectfile(datafile)

#autocalibration:
C = g.calibrate(datafile)

#get meta-data:
M = g.getmeta(datafile)

## End(Not run)
data(data.getmeta)
data(data.inspectfile)

#impute meta-data:
IMP = g.impute(M = data.getmeta, I = data.inspectfile, strategy = 1,
hrs.del.start = 0, hrs.del.end = 0, maxdur = 0)

#plot data
g.plot(IMP, M = data.getmeta, I = data.inspectfile, durplot=4)
```

g.shell.GGIR

function shell for [g.part1](#) and [g.part2](#)

Description

Function calls [g.part1](#) and [g.part2](#)

Usage

```
g.shell.GGIR(mode=c(1,2),datadir=c(),outputdir=c(),studyname=c(),f0=1,f1=0,
windowsizes = c(5,900,3600), desiredtz = "Europe/London",do.enmo = TRUE,
do.angle = FALSE,do.lfenmo = FALSE,do.en = FALSE,do.bfen = FALSE,
do.hfen=FALSE,do.hfenplus = FALSE,
do.telindert2013=FALSE,do.anglex=FALSE,do.angley=FALSE,do.anglez=FALSE,
do.enmoa=FALSE,chunksize=1,do.cal=TRUE,strategy = 1,
hrs.del.start = 0.5,hrs.del.end = 0.5, maxdur = 7,includedaycrit = 16,
L5M5window = c(0,24), M5L5res = 10,winhr = 5,qwindow=c(0,24),
qllevels = c(0.1), ilevels = c(0,10),mvpathreshold = c(100),
```

```
boutcriter = 0.8,do.imp=TRUE,idloc=1,lb = 0.2, hb = 15, n = 4,
use.temp=TRUE,spherecrit=0.3,minloadcrit=72,printsummary=FALSE,ndayswindow=7,
print.filename=FALSE)
```

Arguments

mode	Specify whether g.part1 mode = 1, g.part2 mode =2, or both mode = c(1,2) need to be run. Default: mode = c(1,2)
datadir	Directory where the accelerometer files are stored or list of accelerometer filenames and directories.
outputdir	Directory where the output needs to be stored. Note that this function will attempt to create folders in this directory and uses those folder to organise output
studyname	If the datadir is a folder then the study will be given the name of the data directory. If datadir is a list of filenames then the studyname will be used as name for the analysis
f0	File index to start with (default = 1). Index refers to the filenames sorted in increasing order
f1	File index to finish with (defaults to number of files available)
window sizes	see g.getmeta
desiredtz	see g.getmeta
chunksize	see g.getmeta
do.bfen	if TRUE, calculate metric BFEN with band-pass filter configuration set by lb and hb, see g.getmeta
do.enmo	if TRUE (default), calculate metric ENMO, see g.getmeta
do.angle	if TRUE, calculate metric angle using only the y-axis, see g.getmeta
do.lfenmo	if TRUE, calculate metric LFENMO with low-pass filter configuration set by hb,see g.getmeta
do.en	if TRUE, calculate metric EN, see g.getmeta
do.hfen	if TRUE, calculate metric HFEN with low-pass filter configuration set by hb, see g.getmeta
do.hfenplus	if TRUE, calculate metric HFENplus with band-pass filter configuration set by lb and hb, see g.getmeta
do.telindert2013	if TRUE, calculate the 5 sec epoch Actiwatch count replication as described by te Lindert et al 2013 in the journal SLEEP (volume 36, issue 5, page 781)
do.anglex	if TRUE, calculate the angle of the x-axis relative to the horizontal plane (degrees) utilizing all three axes, see g.getmeta
do.angley	if TRUE, calculate the angle of the y-axis relative to the horizontal plane (degrees) utilizing all three axes, see g.getmeta
do.anglez	if TRUE, calculate the angle of the z-axis relative to the horizontal plane (degrees) utilizing all three axes, see g.getmeta
do.enmoa	if TRUE (default), calculate metric ENMOa which is equal to metric ENMO but with the absolute taken from the Euclidean norm minus one.

lb	lower boundary of the frequency filter (in Hertz), see g.getmeta
hb	upper boundary of the frequency filter (in Hertz), see g.getmeta
n	order of the frequency filter, see g.getmeta
use.temp	see g.calibrate use temperature sensor data if available (Geneactive only)
spherecrit	see g.calibrate 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
minloadcrit	see g.calibrate 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). After loading these hours only extra data is loaded if calibration error has not be reduced to under 0.01 g.
printsummary	see g.calibrate if TRUE will print a summary when done
strategy	how to deal with knowledge about study protocol. value = 1 means select data based on hrs.del.start, hrs.del.end, and maxdur. Value = 2 makes that only the data between the first midnight and the last midnight is used for imputation, see g.impute
hrs.del.start	how many HOURS after start of experiment did wearing of monitor start?, see g.impute
hrs.del.end	how many HOURS before the end of the experiment did wearing of monitor definitely end?, see g.impute
maxdur	how many DAYS after start of experiment did experiment definitely stop? (set to zero if unknown = default), see g.impute
includedaycrit	minimum required number of valid hours in day specific analysis (NOTE: there is no minimum required number of hours per day in the summary of an entire measurement, every available hour is used to make the best possible inference on average metric value per week)
L5M5window	start and end time, in 24 hour clock hours, over which L5M5 needs to be calculated. The calculation is done based on the average day
M5L5res	resolution of L5 and M5 analysis in minutes (default: 10 minutes)
winhr	window size in hours of L5 and M5 analysis (dedault = 5 hours)
qwindow	start and end time, in 24 hour clock hours, over which distribution in metric values need to be extracted. Value = c(0,24) will consider all 24 hours.
qlevels	array of percentiles for which value needs to be extracted. These need to be expressed as a fraction of 1, e.g. c(0.1, 0.5, 0.75). There is no limit to the number of percentiles. If left empty then percentiles will not be extracted. Distribution will be derived from short epoch metric data, see g.getmeta .
ilevels	Levels for acceleration value frequency distribution in mg, e.g. c(0,100,200) There is no constriction to the number of levels.
mvpthreshold	Threshold for MVPA estimation. Threshold needs to be based on metric ENMO. This can be a single number or an array of numbers, e.g. c(100,120). In the later case the code will estimate MVPA seperately for each threshold. If this variable is left blank c() then MVPA is not estimated
boutcriter	The variable boutcriter is a number between 0 and 1 and defines what fraction of a bout needs to be above the mvpthreshold

ndayswindow	If strategy is set to 3 then this is the size of the window as a number of days
idloc	If value = 1 (default) the code assumes that ID number is stored in the obvious header field. If value = 2 the code uses the character string preceding the character '_' in the filename as the ID number
do.cal	Whether to apply auto-calibration or not, see g.calibrate . Default and recommended setting is TRUE
do.imp	Whether to impute missing values (e.g. suspected of monitor non-wear) or not by g.impute . Default and recommended setting is TRUE
print.filename	Whether to print the filename before analysing it (default is FALSE). Printing the filename can be useful to investigate problems (e.g. to verify that which file is being read).

Value

The function provides no values, it only ensures that other functions are called and that their output is stored.

Author(s)

Vincent T van Hees <vincent.van-hees@newcastle.ac.uk>

References

- van Hees VT, Gorzelniak L, Dean Leon EC, Eder M, Pias M, et al. (2013) Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE 8(4): e61691. doi:10.1371/journal.pone.0061691
- van Hees VT, Fang Z, Langford J, Assah F, Mohammad A, da Silva IC, Trenell MI, White T, Wareham NJ, Brage S. Auto-calibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. J Appl Physiol (1985). 2014 Aug 7

Examples

```
## Not run:
datafile = "C:/myfolder/mydata"
outputdir = "C:/myresults"
g.shell.GGIR(mode=c(1,2),datadir=datadir,outputdir=outputdir)

## End(Not run)
```

Index

*Topic **datasets**

- data.calibrate, [4](#)
- data.getmeta, [5](#)
- data.inspectfile, [5](#)

- data.calibrate, [4](#)
- data.getmeta, [5](#)
- data.inspectfile, [5](#)

- g.analyse, [2](#), [6](#), [19](#)
- g.binread, [3](#), [10](#)
- g.calibrate, [2](#), [3](#), [6](#), [11](#), [17](#), [18](#), [24](#), [25](#)
- g.getmeta, [2](#), [6](#), [12](#), [15](#), [17](#), [18](#), [20](#), [21](#), [23](#), [24](#)
- g.impute, [2](#), [6](#), [15](#), [19–21](#), [24](#), [25](#)
- g.inspectfile, [2](#), [6](#), [15](#), [16](#), [21](#)
- g.part1, [2](#), [17](#), [17](#), [19–23](#)
- g.part2, [2](#), [17](#), [19](#), [22](#), [23](#)
- g.plot, [21](#)
- g.shell.GGIR, [2](#), [17](#), [22](#)
- GGIR (GGIR-package), [2](#)
- GGIR-package, [2](#)