Accelerometer data processing with GGIR

Vincent van Hees

April 23 2020

1 Introduction

1.1 What is GGIR?

GGIR is an R-package to process multi-day raw accelerometer data for physical activity and sleep research. The term ***raw*** in raw accelerometry refers to data being expressed in m/s^2 or gravitational acceleration as opposed to the previous generation accelerometers which stored data in accelerometer brand specific units. The signal processing includes automatic calibration, detection of sustained abnormally high values, detection of non-wear and calculation of average magnitude of dynamic acceleration based on a variety of metrics. Next, GGIR uses this information to describe the data per recording, per day of measurement, and (optionally) per segment of a day of measurement, including estimates of physical activity, inactivity and sleep. This vignette provides a general introduction to GGIR, additionally you can find a general introduction video and a mini-tutorial on YouTube. R package GGIR would not have been possible without the support of the contributors listed in the author list at GGIR, and their funders and employers.

1.2 Who has been using GGIR?

GGIR is increasingly being used by research groups acros the world. A non-exhaustive overview of academic publications related to GGIR can be found here.

1.3 How can I contribute to the GGIR development?

The development version of GGIR can be found on github, which is also where you will find guidance on how to contribute to the package development. Open source software may be for free, but the time investment to develop it is never for free. Therefore, we welcome contributions to GGIR in time investment, constructive feedback, financially, or otherwise.

1.4 How can I get service and support?

GGIR is open source software and does not come with service or support guarantees. As user-community you will have to help each other via the google group and GitHub issues as much as possible.

However, if you need dedicated support with the use of GGIR or need someone to adapt GGIR then Vincent van Hees is available as indpendent consultant, see also: https://www.movementdata.nl/.

2 Setting up your work

environment

2.1 Install R and RStudio

Download and install R

Download and install RStudio (optional, but recommended)

Download GGIR with its dependencies, you can do this with one command from the console command line:

```
install.packages("GGIR", dependencies = TRUE)
```

2.2 Prepare folder structure

1. GGIR works with the following accelerometer brands and formats:
    - GENEActiv .bin and .csv
    - ActiGraph .csv. Note for Actigraph users: In ActiLife you have the option to export data with timestamps. Please do not do this as this causes memory issues. To cope with the absense of timestamps GGIR will re-caculate timestamps from the sample frequency and the start time and date as presented in the file header.
    - Axivity .wav, .csv and .cwa
    - Genea (an accelerometer that is not available anymore, but which was used for some studies between 2007 and 2012) .bin and .csv
2. All accelerometer data that needs to be analysed should be stored in one folder, or subfolders of that folder.
3. Give the folder an appropriate name, preferable with a reference to the study or project it is related to rather than just 'data', because the name of this folder will be used later on as an identifier of the dataset.

2.3 GGIR shell function

GGIR comes with a large number of functions and optional settings (arguments) per functions. To ease interacting with GGIR there is one central function, named g.shell.GGIR, to talk to all the other functions. Being confronted with the amount of optional arguments might feel like stepping inside the cockpit of an airplane:

```
mode=c(1,2), f0=c(), f1=c(), selectdaysfile = c(), configfile=c(), overwrite =
FALSE, strategy = 1, maxdur = 7, do.cal = TRUE,  hrs.del.start = 0, hrs.del.end =
0, loglocation = c(), , acc.metric = "ENMO", storefolderstructure = FALSE,
windowsizes = c(5,900,3600), minloadcrit = 72, desiredtz = "Europe/London",
chunksize = 1, do.enmo = TRUE, do.lfenmo = FALSE, do.en = FALSE,  do.bfen = FALSE,
do.hfen = FALSE, do.hfenplus = FALSE, do.mad = FALSE, do.anglex = FALSE, do.angley
= FALSE, do.anglez = FALSE, do.roll_med_acc_x=FALSE, do.roll_med_acc_y=FALSE,
```

```
do.roll_med_acc_z=FALSE, do.dev_roll_med_acc_x=FALSE, do.dev_roll_med_acc_y=FALSE,
do.dev_roll_med_acc_z=FALSE, do.enmoa = FALSE, dynrange = c(), printsummary =
FALSE, includedaycrit = 16, M5L5res = 10, winhr = 5, qwindow = c(0,24), qlevels =
c(), ilevels = c(), mvpathreshold = 100, boutcriter = 0.8, ndayswindow = 7, idloc =
1, do.imp = TRUE, anglethreshold = 5, timethreshold = 5, ignorenonwear = TRUE,
colid=1, coln1=1, nnights=7, outliers.only=FALSE, excludefirstlast=FALSE,
excludefirstlast.part5=FALSE, criterror=3, includenightcrit=16,
relyonsleeplog=FALSE, sleeplogidnum=TRUE, def.noc.sleep=c(), do.visual=FALSE,
viewingwindow = 1, dofirstpage = TRUE, visualreport = FALSE, print.filename =
FALSE, backup.cal.coef = c(), bout.metric = 1, closedbout = FALSE,
IVIS_windowsize_minutes=60, IVIS_epochsize_seconds=30, constrain2range = TRUE,
do.part3.pdf = TRUE, boutcriter.in = 0.9, boutcriter.lig = 0.8, boutcriter.mvpa =
0.8, threshold.lig = 40, threshold.mod = 100,   threshold.vig = 400, timewindow =
c("MM","WW"), boutdur.mvpa = c(1,5,10), boutdur.in = c(10,20,30), boutdur.lig =
c(1,5,10), save_ms5rawlevels = FALSE,  mvpadur = c(1,5,10), epochvalues2csv =
FALSE, bout.metric = 1, window.summary.size = 10, dayborder = 0, iglevels = c()
```

In this paragraph we will guide you through the main arguments relevant for 99% of research. First of all, it is important to understand that GGIR is structured in five parts and that g.shell.GGIR controls each of these parts:

- Part 1: Loads the data and stores derived features (aggregations) needed for the other parts. This is the time consuming part. Once this is done, parts 2-5 can be run (or re-run with different parameters in parts 2-5) relatively quickly.
- Part 2: Data quality analyses and low level description of signal features per day and per file. At this point a day is defined from midnight to midnight
- Part 3: Estimation of sustained inactivity and sleep periods, needed for input to Part 4 for sleep detection
- Part 4: o Labels the sustained inactive periods detected in Part 3 as sleep, or daytime sustained inactivity, per night and per file
- Part 5: Derives sleep and physical activity characteristics by re-using information derived in part 2, 3 and 4. Total time in intensity categories, the number of bouts, time spent in bouts and average acceleration (overall activity) is calculated.

By looking up the corresponding functions g.part1, g.part2, g.part3, g.part4, and g.part5 you can see what arguments are possible. All of these arguments are also accepted by g.shell.GGIR, because g.shell.GGIR is nothing more than a wrapper around those functions.

You will probably never need to think about most of the arguments listed above, because the default values are fine and a lot of arguments are only included to facilitate methodological studies where researchers want to have control over everything.

The bare minimum input needed for `g.shell.GGIR` is:

```r
library(GGIR)
g.shell.GGIR(datadir="C:/mystudy/mydata",
             outputdir="D:/myresults")
```

Argument `datadir` allows you to specify where you have stored your accelerometer data and `outputdir` allows you to specify where you would like the output of the analyses to be stored. If you copy paste the above code to a new R script (file ending with .R) and Source it in R(Studio) then the dataset will be processed and the output will be stored in the specified output directory.

Below we have highlighted the key arguments you may want to be aware of. We are not giving a detailed explanation, please see the package manual for that.

2.3.1 Key general arguments

- `mode` - which part of GGIR to run, GGIR is constructed in five parts.
- `overwrite` - whether to overwrite previously produced milestone output. Between each GGIR part, GGIR stores milestone output to ease re-running parts of the pipeline.
- `idloc` - tells GGIR where to find the participant ID (default: inside file header)
- `strategy` - informs GGIR how to consider the design of the experiment.
    - If `strategy` is set to value 1, then check out arguments `hrs.del.start` and `hrs.del.end`.
    - If `strategy` is set to value 3, then check out arguments `ndayswindow`.
- `maxdur` - maximum number of days you expect in a data file based on the study protocol.
- `desiredtz` - time zone of the experiment.
- `chunksize` - a way to tell GGIR to use less memomry, which can be useful on machines with limited memory.
- `includedaycrit` - tell GGIR how many hours of valid data per day (midnight-midnigth) is acceptable.
- `includenightcrit` - tell GGIR how many hours of a valid night (noon-noon) is acceptable.
- `qwindow` - argument to tell GGIR whether and how to segment the day for day-segment specific analysis.
- `mvpathreshold` and `boutcriter` - acceleration threshold and bout criteria used for calculating time spent in MVPA (only used in GGIR part2).
- `epochvalues2csv` - to export epoch level magnitude of acceleration to a csv files (in addition to already being stored as RData file)
- `dayborder` - to decide whether the edge of a day should be other than midnight.
- `iglevels` - argument related to intensity gradient method proposed by A. Rowlands.
- `do.report` - specify reports that need to be generated.
- `viewingwindow` and `visualreport` - to create a visual report, this only works when all five parts of GGIR have succesfully run.

2.3.2 Key arguments related to sleep analysis

- If you want the sleep analysis to be guided by a sleeplog (diary) then use `loglocation`, `colid`, `coln1`, and `nnights` to configure this.
- `ignorenonwear`
- If you want to create a visualisation of how sleep period time and sustained inactivity bouts match throughout a day then consider `do.visual`, `outliers.only`, and `criterror`.
- If you want to exclude the first and last night from the sleep analysis then used `excludefirstlast`.

- `def.noc.sleep` specifies how the sleep period time window should be estimated if no sleeplog is used.

Some more detail about using a sleeplog: Argument 'loglocation' is the location of the spreadsheet (csv) with sleep log information. The spreadsheet needs to have the following structure: one column for participant id, and then followed by alternatingly one column for onset time and one column for waking time (see example below). There can be multiple sleeplogs in the same spreadsheet. The first raw of the spreadsheet needs to be filled with column names, it does not matter what these column names are. Timestamps are to be stored without date as in hh:mm:ss. If onset corresponds to lights out or intention to fall asleep, then it is the end-users responsibility to account for this in the interpretation of the results.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	STNO	FGASLP2	FGAWK2	FGASLP3	FGAWK3	FGASLP4	FGAWK4	FGASLP5	FGAWK5	FGASLP6	FGAWK6	FGASLP7	FGAWK7	FGASLP8	FGAWK8	FGASLP9	FGAWK9
2	10002	21:40:00	5:35:00	23:30:00	6:45:00	22:20:00	6:30:00	22:45:00	7:45:00	22:30:00	6:20:00	22:20:00	6:30:00	22:20:00	6:10:00	22:30:00	6:40:0
3	10004	23:30:00	7:00:00	22:50:00	7:20:00	22:45:00	7:45:00	22:30:00	7:30:00	22:15:00	6:45:00	22:45:00	6:45:00	22:45:00	6:45:00	23:00:00	6:30:0
4	10005	22:15:00	6:45:00	23:00:00	6:45:00	22:30:00	8:00:00	20:00:00	7:00:00	21:00:00	6:30:00	22:00:00	6:45:00	22:00:00	6:45:00	22:00:00	7:15:0
5	10011	0:00:00	7:25:00	23:30:00	7:25:00	23:30:00	7:25:00	23:15:00	7:30:00	23:15:00	7:30:00	23:30:00	7:45:00	0:00:00	8:30:00	23:35:00	7:30:0
6	10013	23:00:00	6:23:00	22:50:00	6:30:00	22:50:00	6:45:00	22:50:00	6:20:00	23:00:00	6:40:00	22:50:00	6:30:00	22:55:00	6:30:00	23:00:00	6:30:0
7	10016	23:30:00	7:10:00	0:00:00	6:30:00	23:45:00	7:00:00	23:45:00	6:00:00	23:30:00	6:45:00	0:00:00	6:15:00	0:15:00	6:50:00	23:50:00	5:50:0
8	10017	23:00:00	8:00:00	23:00:00	8:15:00	22:30:00	9:00:00	0:00:00	9:00:00	20:22:00	8:00:00	22:30:00	8:15:00	22:15:00	8:15:00	23:30:00	8:00:0

2.3.3 Key arguments related to 24 hour time use

- `excludefirstlast.part5` - whether to ignore the last and first day.
- `bout.metric` - choose metric for calculating bouts (we recommend setting 4).
- Configure what percentage of a bout can be gaps in bouts `boutcriter.in`, `boutcriter.lig`, `boutcriter.mvpa`.
- Configure thresholds for acceleration levels (some may want to interpret this as intensity levels): `threshold.lig`, `threshold.mod`, `threshold.vig`.
- `timewindow` to specify whether days should be defined from midnight to midnight, from waking-up to waking-up, or both (all output is generated twice).
- Configure durations of bouts: `boutdur.mvpa`, `boutdur.in`, and `boutdur.lig`. Note that this can be a vector of multiple values indicating the minimum and maximum duration of subsequent bout types, e.g. 1-5 minutes MVPA, 5-10 minutes MVPA, and longer than 10 minutes MVPA.

2.3.4 Using published cut-points

Cut-points to estimate time spent in acceleration levels that are roughly liked to levels of energy metabolism have been proposed by:

- Esliger et al 2011: wrist and waist in adults.
- Schaefer et al 2014: wrist in 6-11 year old children.
- Roscoe et al 2017: wrist in 4-5 year old pre-school children.
- Phillips et al 2013: wrist and hip in 8-14 year olds.
- Vaha-Ypya et al 2015: hip in adults.
- Hildebrand et al 2014 and 2016: wrist and hip in 7-11 and 21-61 years old.
- If you are aware of any additional publications then let us know.

The first four publications make use of acceleration metrics that sum their values per epoch rather than average them per epoch like GGIR does. However, no need to worry: It is still possible to use those cut-points with a simple cut-point scaling trick and by selecting the right acceleration metric as detailed below.

Esliger 2011, Phillips 2013:

- In GGIR use metric ENMOa instead of ENMO with arguments do.enmoa = TRUE, do.enmo = FALSE, and acc.metric="ENMOa".
- `threshold.lig = (LightCutPointFromPaper/80) * 1000`
- `threshold.mod = (ModerateCutPointFromPaper/80) * 1000`
- `threshold.vig = (VigorousCutPointFromPaper/80) * 1000`
- `mvpathreshold = (ModerateCutPointFromPaper/80) * 1000`
- In the part2 results you will need the MVPA estimates that are related to ENMOa, not ENMO.
- In the part 5 results everything will be based on the new cut-points.

Roscoe 2017:

- In GGIR use metric ENMOa instead of ENMO with arguments do.enmoa = TRUE, do.enmo = FALSE, and acc.metric="ENMOa".
- `threshold.lig = (LightCutPointFromPaper/85.7) * 1000`
- `threshold.mod = (ModerateCutPointFromPaper/85.7) * 1000`
- `threshold.vig = (VigorousCutPointFromPaper/85.7) * 1000`
- `mvpathreshold = (ModerateCutPointFromPaper/85.7) * 1000`
- In the part2 results you will need the MVPA estimates that are related to ENMOa, not ENMO.
- In the part 5 results everything will be based on the new cut-points.

Schaeffer 2014:

- In GGIR use metric EN instead of ENMO with arguments do.en = TRUE, do.enmo = FALSE, and acc.metric="EN".
- Specify Schaeffer cut-points as:
- `threshold.lig = (LightCutPointFromPaper/75) * 1000`
- `threshold.mod = (ModerateCutPointFromPaper/75) * 1000`
- `threshold.vig = (VigorousCutPointFromPaper/75) * 1000`
- `mvpathreshold = (ModerateCutPointFromPaper/75) * 1000`- In the part2 results you will need the MVPA estimates that are related to EN, not ENMO.
- In the part 5 results everything will be based on the new cut-points.

Vaha-Ypya et al 2015:

- Use default setting do.mad= TRUE, acc.metric="MAD"
- Use the cut-points as provided by Vaha-Ypya directly. No need for scaling.

Hildebrand 2014 and Hildebrand 2016:

- Use default setting do.enmo= TRUE, acc.metric="ENMO"
- Use the cut-points as provided by Hildebrand directly. No need for scaling.

2.3.5 Example call

So, if you consider all the arguments above you me may end up with a call to g.shell.GGIR that could look as follows.

```
library(GGIR)
g.shell.GGIR(
             mode=c(1,2,3,4,5),
             datadir="C:/mystudy/mydata",
             outputdir="D:/myresults",
             do.report=c(2,4,5),
             #=====================
             # Part 2
             #=====================
             strategy = 1,
             hrs.del.start = 0,          hrs.del.end = 0,
             maxdur = 9,                 includedaycrit = 16,
             qwindow=c(0,24),
             mvpathreshold =c(100),
             bout.metric = 4,
             excludefirstlast = FALSE,
             includenightcrit = 16,
             #=====================
             # Part 3 + 4
             #=====================
             def.noc.sleep = 1,
             outliers.only = TRUE,
             criterror = 4,
             do.visual = TRUE,
             #=====================
             # Part 5
             #=====================
             threshold.lig = c(30), threshold.mod = c(100),  threshold.vig =
  c(400),
             boutcriter = 0.8,      boutcriter.in = 0.9,     boutcriter.lig =
  0.8,
             boutcriter.mvpa = 0.8, boutdur.in = c(1,10,30), boutdur.lig =
  c(1,10),
             boutdur.mvpa = c(1),
             #=====================
             # Visual report
             #=====================
             timewindow = c("WW"),
             visualreport=TRUE)
```

Once you have used g.shell.GGIR and the output directory (outputdir) will be filled with milestone data and results.

2.3.6 Configuration file

Function g.shell.GGIR stores all the explicitely entered argument values and default values for the argument that are not explicitely provided in a csv-file named config.csv stored in the root of the output folder. The config.csv file is accepted as input to g.shell.GGIR with argument `configfile` to replace the specification of all the arguments, except `datadir` and `outputdir`, see example below.

```
library(GGIR)
g.shell.GGIR(datadir="C:/mystudy/mydata",
             outputdir="D:/myresults", configfile = "D:/myconfigfiles
  /config.csv")
```

The practical value of this is that it eases the replication of analysis, because instead of having to share you R script, sharing your config.csv file will be sufficient. Further, the config.csv file contribute to the reproducibility of your data analysis.

Note 1: When combining a configuration file with explicitely provided argument values, the explicitely provided argument values will overrule the argument values in the configuration file. Note 2: The config.csv file in the root of the output folder will be overwritten every time you use g.shell.GGIR. So, if you would like to add annotations in the file, e.g. in the fourth column, then you will need to store it somewhere outside the output folder and explicitely point to it with `configfile` argument.

3 Time for action: How to run your analysis?

3.1 From the console

You can use

`source("pathtoscript/myshellscript.R")`

or use the Source button in RStudio if you use RStudio.

3.2 In a cluster

GGIR by default support multi-thread processing, which can be turned off byseting argument do.parallel = FALSE. If this is still not fast enough then I advise using a GGIR on a computing cluster. The way I did it on a Sun Grid Engine cluster is shown below, please note that some of these commands are specific to the computing cluster you are working on. For example, I had to update the bash scripts when working on PBS cluster, and the R package clustermq can be of value. Please consult your local cluster specilist to tailor this to your situation. In my case, I had three files for the SGE setting:

submit.sh

```
for i in {1..707}; do
    n=1
    s=$(($(($n * $[$i-1]))+1))
    e=$(($i * $n))
    qsub /home/nvhv/WORKING_DATA/bashscripts/run-mainscript.sh $s $e
done
```

run-mainscript.sh

```
#! /bin/bash
#$ -cwd -V
#$ -l h_vmem=12G
/usr/bin/R --vanilla --args f0=$1 f1=$2 < /home/nvhv/WORKING_DATA
  /test/myshellscript.R
```

myshellscript.R

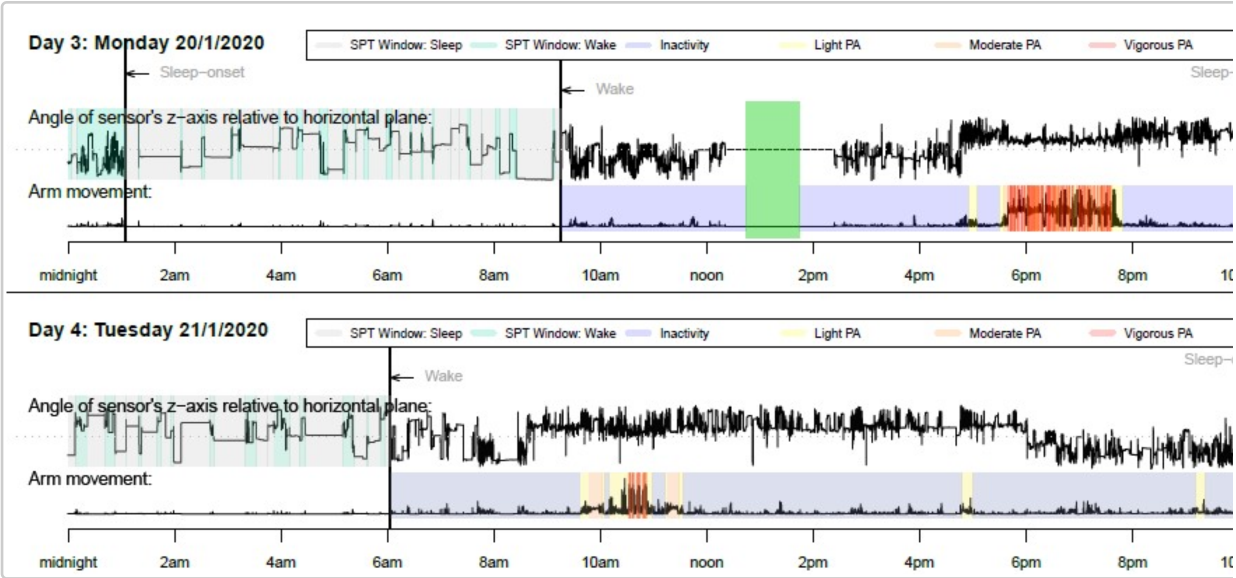
```
options(echo=TRUE)
args = commandArgs(TRUE)
if(length(args) > 0) {
  for (i in 1:length(args)) {
    eval(parse(text = args[[i]]))
  }
}
g.shell.GGIR(f0=f0,f1=f1,...)
```

You will need to update the ... in the last line with the arguments you used for g.shell.GGIR. Note that `f0=f0,f1=f1` is essential for this to work. The values of `f0` and `f1` are passed on from the bash script.

Once this is all setup you will need to call `bash submit.sh` from the command line. ***Note: Please make sure that you process one GGIR part at the same time on a cluster, because each part assumes that preceding parts have been ran. You can make sure of this by always specifying argument mode to a single part of GGIR. Once the analysis stops update argument mode to the next part until all parts are done.*** The speed of the parallel processing is obviously dependent on the capacity of your computing cluster and the size of your dataset. We have been able to process all 4000 files from the Whitehall II Study in just a couple of hours.

4 Inspecting the results

- csv-spreadsheets with all the variables you need for physical activity, sleep and circadian rhythm research
- Pdfs with on each page a low resolution plot of the data per file and quality indicators
- R objects with milestone data
- Pdfs with a visual summary of the physical activity and sleep patterns as identified (see example below)



4.1 Output part 2

Part 2 generates the following output:

- part2_summary.csv: Person level summary (see below)
- part2_daysummary.csv: Day level summary (see below)
- QC/data_quality_report.csv: Overview of calibration results and whether or not a file was corrupt or too short to be procesed,
- QC/plots to check data quality 1.pdf: A pdf with visualisation of the acceleration time series in 15 minute resolution and with invalid data segments highlighted in colours (yellow: non-wear based on standard deviation threshold, brown: non-wear after extra filtering step (introduced in 2013), and purple: clipping)

4.1.1 Person level summary

Variable	Description
ID	Participant id
device_sn	Device serial number
bodylocation	Body location extracted from file header
filename	Name of the data file
start_time	Timestamp when recording started
startday	Day of the week on which recording started
samplefreq	Sample frequency (Hz)
device	Accelerometer brand, e.g. GENEACtiv
clipping_score	The Clipping score: Fraction of 15 minute windows per file for which the acceleration in one of the three axis was close to the

Variable	Description
	maximum for at least 80% of the time. This should be 0.
meas_dur_dys}	Measurement duration (days)
complete_24hcycle	Completeness score: Fraction of 15 minute windows per 24 hours for which no valid data is available at any day of the measurement.
meas_dur_def_proto_day	measurement duration according to protocol (days): Measurement duration (days) minus the hours that are ignored at the beginning and end of the measurement motived by protocol design
wear_dur_def_proto_day	wear duration duration according to protocol (days): So, if the protocol was seven days of measurement then wearing the accelerometer for 8 days and recording data for 8 days will still makethat the wear duration is 7 days
calib_err	Calibration error (static estimate) Estimated based on all 'non-movement' periods in the measurement after applying the autocalibration.
calib_status	Calibration status: Summary statement about the status of the calibration error minimisation
ENMO_fullRecordingMean	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 (diurnal balanced), with invalid data imputed by the average at similar time points on different days of the week. In addition to ENMO it is possible to extract other acceleration metrics (i.e. BFEN, HFEN, HFENplus). We emphasize that it is calculated over the full recording because the alternative is that a variable is only calculated overmeasurement days with sufficient valid hours of data.
ENMO	(only available if set to true in part1.R) 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 in part1.R (i.e. BFEN, HFEN, HFENplus) See also van Hees PLoSONE April 2013 for a detailed description

Variable	Description
	and comparison of these techniques.
pX_A_mg_0-24h_fullRecording	This variable represents the Xth percentile in the distribution of short epoch metric value A of the average day. The average day may not be ideal for describing the distirbution. Therefore, the code also extracts the following variable.
AD_pX_A_mg_0-24h	This variable represents the Xth percentile in the distribution of short epoch metric value A per day averaged across all days.
L5_A_mg_0-24	Average of metric A during the least active five* hours in the day that is the lowest rolling average value of metric A. (* window size is modifiable by argument winhr)
M5_A_mg_0-24	Average of metric A during the most active five* hours in the day that is the lowest rolling average value of metric A. (* window size is modifiable by argument winhr)
L5hr_A_mg_0-24	Starting time in hours and fractions of hours of L5_A_mg_0-24
M5hr_A_mg_0-24	Starting time in hours and fractions of hours of M5_A_mg_0-24
ig_gradient_ENMO_0-24hr_fullRecording	Intensity gradient calculated over the full recording.
1to6am_ENMO_mg	Average metric value ENMO between 1am and 6am
N valid WEdays	Number of valid weekend days
N valid WKdays	Number of valid week days
IS_interdailystability	inter daily stability
IV_intradailyvariability	intra daily variability
IVIS_windowsize_minutes	Sizes of the windows based on which IV and IS are calculated (note that this is modifiable)
IVIS_epochsize_seconds	size of the epochs based on which IV and IS are calculated (note that this is modifiable)
AD_...	All days (plain average of all available days, no weighting). The variable ... was calculated per day and then averaged over all the available days

Variable	Description
WE_...	Weekend days (plain average of all available days, no weighting). The variable ... was calculated per day and then averaged over weekend days only
WD_...	Week days (plain average of all available days, no weighting). The variable ... was calculated per day and then averaged over week days only
WWE_...	Weekend days (weighted average) 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_...	Week days (weighted average) 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)
WWD_MVPA_E5S_T100_ENMO	Time spent in moderate-to-vigorous based on 5 second epoch size and an ENMO metric threshold of 100
`WWE_MVPA_E5S_B1M80%_T100_ENMO`	Time spent in moderate-to-vigorous based on 5 second epoch size and an ENMO metric threshold of 100 based on a bout criteria of 100
`WE_[100,150)_mg_0-24h_ENMO`	Time spent between (and including) 100 mg and 150 (excluding 150 itself) between 0 and 24 hours (the full day) using metric ENMO data exclusion strategy (value=1, ignore specific hours; value=2, ignore all data before the first midnight and after the last midnight)
..._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
..._ENMO_mg...	ENMO or other metric was first calcualte 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 start of meas (if strategy=1)	number of hours ignored at the start of the measurement (if strategy = 1) A log of decision made in part2.R

Variable	Description
n hours ignored at end of meas (if strategy=1)	number of hours ignored at the end of the measurement (if strategy = 1). A log of decision made in part2.R
n hours ignored at end of meas (if strategy=1)	number of days of measurement after which all data is ignored (if strategy = 1) A log of decision made in part2.R
epoch size to which acceleration was averaged (seconds)	A log of decision made in part1.R
pdffilenumb	Indicator of in which pdf-file the plot was stored
pdfpagecount	Indicator of in which pdf-page the plot was stored

4.1.2 Day level summary

This is a non-exhaustive list, because most concepts have been explained in summary.csv

Variables	Description
ID	Participant id
filename	Name of the data file
calender_date	Timestamp and date on which measurement started
bodylocation	Location of the accelerometer as extracted from file header
N valid hours	Number of hours with valid data in the day
N hours	Number of hours of measurement in a day, which typically is 24, unless it is a day on which the clock changes (DST) resulting in 23 or 25 hours. The value can be less than 23 if the measurement started or ended this day
weekday	Name of weekday
measurement	Day of measurement Day number relative to start of the measurement
L5hr_ENMO_mg_0-24h	Hour on which L5 starts for these 24 hours (defined with metric ENMO)
L5_ENMO_mg_0-24h	Average acceleration for L5 (defined with metric ENMO)
`[A,B)_mg_0-24h_ENMO`	Time spent in minutes between (and including) acceleration value A in mg and (excluding) acceleration value B in mg based on metric ENMO

Variables	Description
ig_gradient_ENMO_0-24hr	Gradient from intensity gradient analysis (Rowlands et al 2018) based on metric ENMO for the time segment 0 to 24 hours
ig_intercept_ENMO_0-24hr	Intercept from intensity gradient analysis (Rowlands et al 2018) based on metric ENMO for the time segment 0 to 24 hours
ig_rsquared_ENMO_0-24hr	r squared from intensity gradient analysis (Rowlands et al 2018) based on metric ENMO for the time segment 0 to 24 hours

4.2 Output part 4

Part 4 generates the following output:

4.2.1 Night level summaries

- part4_nightsummary_sleep_cleaned.csv
- QC/part4_nightsummary_sleep_full.csv

The csv. files contain the variables as shown below.

Variable	Description
ID	Participant ID extracted from file
night	Number of the night in the recording
sleeponset	Detected onset of sleep expressed as hours since the midnight of the previous night.
wakeup	Detected waking time (after sleep period) expressed as hours since the midnight of the previous night.
SptDuration	Difference between onset and waking time.
sleepparam	Definition of sustained inactivity by accelerometer.
guider	guider used (see paragraph 'Waking-waking or 24 hour time-use analaysis').
guider_onset	Start of Sleep Period Time window derived from the guider.
guider_wake	End of Sleep Period Time window derived guider.
guider_SptDuration	Time SPT duration derived from guider_wake and guider_onset.

Variable	Description
error_onset	Difference between sleeponset and guider_onset
error_wake	Difference between wakeup and guider_wake
fraction_night_invalid	Fraction of the night (noon-noon or 6pm-6pm) for which the data was invalid, e.g. monitor not worn or no accelerometer measurement started/ended within the night.
SleepDurationInSpt	Total sleep duration, which equals the accumulated nocturnal sustained inactivity bouts within the Sleep Period Time.
duration_sib_wakinghours	Accumulated sustained inactivty bouts during the day. These are the periods we would label during the night as sleep, but during the day they form a subclass of inactivity, which may represent day time sleep or wakefulness while being motionless for a sustained period of time number_sib_sleepperiod} Number of noturnal sleep periods, with nocturnal referring to the Sleep Period Time window.
duration_sib_wakinghours_atleast15min	Same as duration_sib_wakinghours, but limited to SIBs that last at least 15 minutes.
number_sib_wakinghours	Number of sustained inactivity bouts during the day, with day referring to the time outside the Sleep Period Time window.
sleeponset_ts	sleeponset formatted as a timestamp
wakeup_ts	wakeup formatted as a timestamp
guider_onset_ts	guider_onset formatted as a timestamp
guider_wake_ts	guider_wake formatted as a timestamp
page	pdf page on which the visualisation can be found
daysleeper	If 0 then the person is a nightsleeper (sleep period did not overlap with noon) if value=1 then the person is a daysleeper (sleep period did overlap with noon)
weekday	Day of the week on which the night started
calendardate	Calendar date on which the night started
filename	Name of the accelerometer file
cleaningcode	see paragraph ‘Waking-waking or 24 hour time-use analaysis’).

Variable	Description
sleeplog_used	Whether a sleep log was used (TRUE/FALSE)
acc_available	Whether accelerometer data was available (TRUE/FALSE).

4.2.2 Person level summaries

- part4_summary_sleep_cleaned.csv
- QC/part4_summary_sleep_full.csv

In the person level report the variabls are derivatives of the variables in the night level summary. Please fine below extra clarification on a few of the variable names for which the meaning may not be obvious:

Variable	Description
…_mn	mean across days
…_sd	standard deviation across days
…_AD	All days
…_WE	Weekend days
…_WD	Week days
sleeplog_used	Whether a sleeplog was available (TRUE) or not (FALSE)
sleep_efficiency	Accelerometer detrive sleep efficiency within the sleep period time calculatd as the ratio between acc_SleepDurationInSpt and acc_SptDuration (denominator). Only available at person level, because at night level the user can calculate this from existing variables.
n_nights_acc	Number of nights of accelerometer data
n_nights_sleeplog	Number of nights of sleeplog data.
n_WE_nights_complete	Number of weekend nights complete which means both accelerometer and estimate from guider.
n_WD_nights_complete	Number of weekday nights complete which means both accelerometer and estimate from guider.
n_WEnights_daysleeper	Number of weekend nights on which the person slept until after noon.
n_WDnights_daysleeper	Number of weekday nights on which the person slept until after noon.
duration_sib_wakinghour	Total duration of sustained inactivity bouts during the waking hours.

Variable	Description
number_sib_wakinghours	Number of sustained inactivity bouts during the waking hours.
average_dur_sib_wakinghours	Average duration of the sustained inactivity bouts during the day (outside the sleep period duration). Calculated as duration_sib_wakinghour divided by number_sib_wakinghours per day, after which the mean and standard deviation are calculated across days.

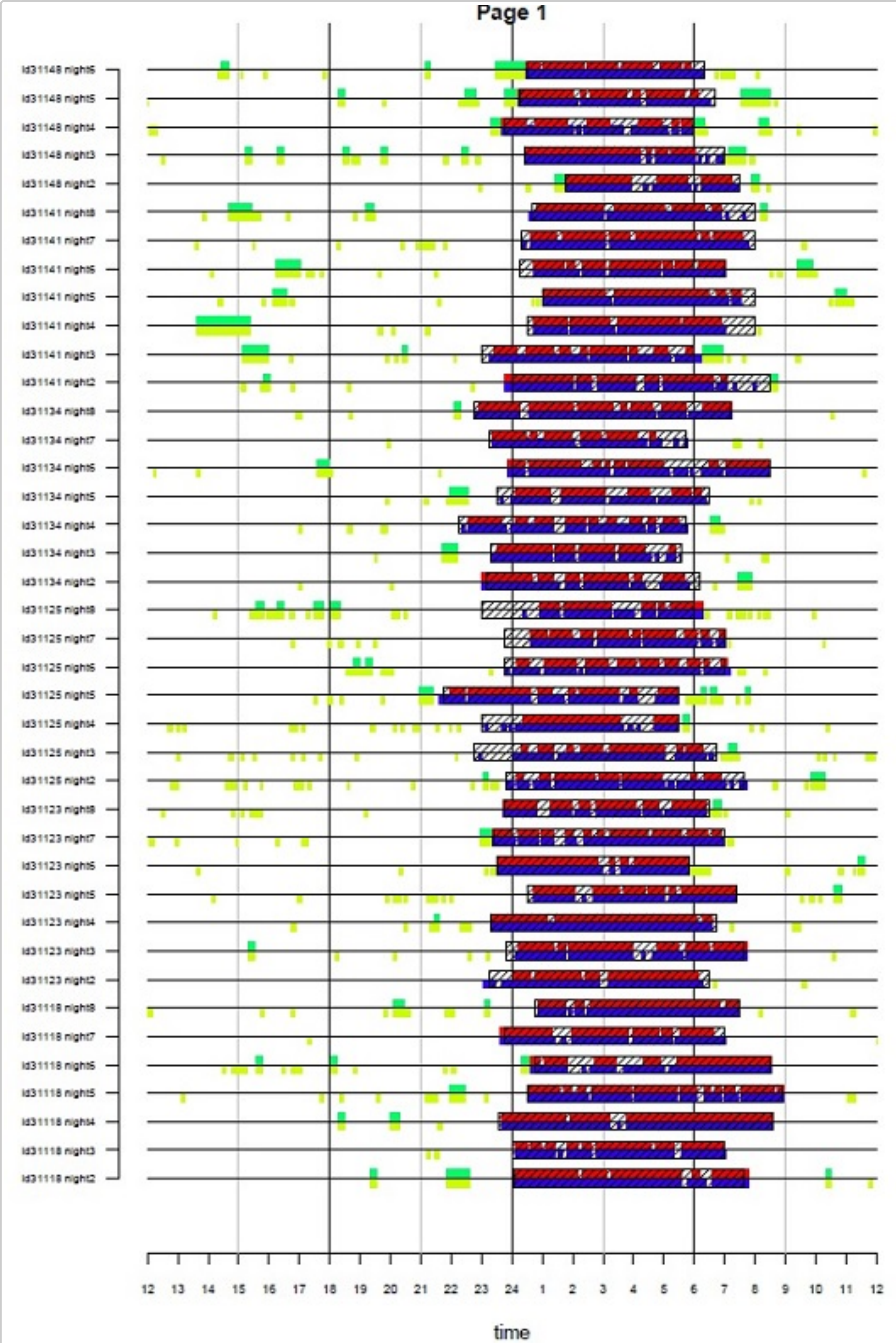
4.2.3 visualisation_sleep.pdf

Visualisation to support data quality checks: - visualisation_sleep.pdf (optional)

When input argument `do.visual` is set to TRUE GGIR can show the following visual comparison between the time window of being asleep (or in bed) according to the sleeplog and the detected sustained inactivity bouts according to the accelerometer data. This visualisation is stored in the results folder as visualisation_sleep.pdf.

Explanation of the image: Each line represents one night. Colors are used to dinstinguish definitions of sustianed inacitivty bouts (2 definitions in this case) and to indicate existence or absense of overlap with the sleeplog. When argument `outliers.only` is set to FALSE it will visualise all available nights in the dataset. If `outliers.only` is set to TRUE it will visualise only nights with a difference in onset or waking time between sleeplog and sustained inactivity bouts larger than the value of argument `criterror`.

This visualisation with outliers.only set to TRUE and critererror set to 4 was very powerful to identify entry errors in sleeplog data in van Hees et al PLoSONE 2015. We had over 25 thousand nights of data, and this visualisation allowed us to quickly zoom in on the most problematic nights to investigate possible mistakes in GGIR or mistakes in data entry.



Page 1
id31148 night6
id31148 night5
id31148 night4
id31148 night3
id31148 night2
id31141 night8
id31141 night7
id31141 night6
id31141 night5
id31141 night4
id31141 night3
id31141 night2
id31134 night8
id31134 night7
id31134 night6
id31134 night5
id31134 night4
id31134 night3
id31134 night2
id31125 night8
id31125 night7
id31125 night6
id31125 night5
id31125 night4
id31125 night3
id31125 night2
id31123 night8
id31123 night7
id31123 night6
id31123 night5
id31123 night4
id31123 night3
id31123 night2
id31118 night8
id31118 night7
id31118 night6
id31118 night5
id31118 night4
id31118 night3
id31118 night2
12 13 14 15 16 17 18 19 20 21 22 23 24 1 2 3 4 5 6 7 8 9 10 11 12
time

4.3 Output part 5

The output of part 5 is dependent on the parameter configuration, it will generate as many output files as there are unique combination of the three thresholds provide. For example, the output could be:

For example, the following files will be generated if the threshold configuration was 30 for light activity, 100 for moderate and 400 for vigorous activity: - part5_daysummary_MM_L30M100V400_T5A5.csv - part5_daysummary_WW_L30M100V400_T5A5.csv - part5_personsummary_MM_L30M100V400_T5A5.csv - part5_personsummary_WW_L30M100V400_T5A5.csv - file summary reports/Report_nameofdatafile.pdf

4.3.1 Day level summary

Variables	Description
sleeponset	onset of sleep expressed in hours since the midnight in the night preceding the night of interest, e.g. 26 is 2am.
wakeup	waking up time express in the same way as sleeponset.
sleeponset_ts	onset of sleep expressed as a timestamp hours:minutes:seconds
daysleeper	if 0 then the person woke up before noon, if 1 then the person woke up after noon
cleaningcode	See paragraph 'Sleep and full day time-use analysis in GGIR'.
dur_fulldaywindow_min	Length of the day window defined in minutes.
dur…	duration
acc…	(average) acceleration according to default metric specific by acc.metric
nightwak…	night waking (waking periods within the Sleep period time window)
SIB	sustained inactivity bouts, See paragraph 'Sleep and full day time-use analysis in GGIR'.
OIN	other inactivity (other than SIB).
Nblock	number of blocks
D10	bout lengths with 10 minutes and longer
T120	threshold of 120 mg and higher
dur_LIGB_D10T50_120	the time spent in bouts of light activity of at least 10 minutes

Variables	Description
WW	in filename refers to analyses based on the timewindow from waking to waking up
MM	in filename refers to analyses done on windows between midnight and midnight
INB	are bouts of time during which acceleration is below an acceleration threshold for at least X percent of the time.
LIG50_120	If there is no D in the variable name then it refers to unbouted time spent between those thresholds.
D1T100 and D10T100	If boutdur.mvpa holds two bout duration thresholds like 1 and 10 minutes when it is set to c(1,10) then you will get D1T100 and D10T100. In this case D1T100 effectively means bouts between 1 and 9.99 minutes, while D10T100 refers to bouts of at least 10 minutes.
TMOD and TVIG	Total moderate and total vigorous PA with all bouted and unbouted time contributing. Just MOD and VIG are only the unbouted time spent in those categories.
nightwak	night waking time
TSIB	total of all sustained inactivity bouts (regardless of whether they contributed or not to inactivity bouts (INB). Variable names with SIB in them are the sustained inactivity bouts that are not part of the inactivity bouts. Only TSIB shows the total time spent in all sustained inactivity bouts. I make this distinction, because sustained inactivity bouts, can be part of inactivity bouts and are then counted towards the inactivity bout (INB variables) and not towards the SIB category.
TOIN	Total other inactivity
TIN	Total inactivity = TOIN + TSIB = Total other inactivty + Total sustained inactivity.
dur_sleepperiod_min	Duration of Sleep Period Time window wihtin this day window.
dur_fulldaywindow_min	Duration this day window
dur_wakinghours_min	Duration of waking hours within this day window

Levels of behaviour from least active to most active are: Sleep or SIB, Other inactivity, Light, Moderate, Vigorous

Additionally they a grouped in:

- MVPA bouts (Moderate or Vigorous),
- Light bouts (just light),
- Inactivity bouts (Other inactivity or SIB),
- And the non-bouted time spent in each of the five categories (SIB, OIN, LIG, MOD,

VIG)
- Total of each category: TSIB, TOIN, TLIG, TMOD, TVIG

Example variable explanations: dur_day_OIN30_min is the time spent in minutes in other inactiivty during the day with a threshold of 30 mg not part of inactivity bouts, dur_day_MOD100_400_min is the time spent in moderate activity defined between 100 and 400mg but not part of an MVPA bout, dur_MVPA_D1T100_min is time spent in MVPA bouts defined as 100 mg or higher and lasting at least 1 minute and with no upper boundary if there is no other variable that starts with dur_MVPA_D, the existence of dur_INB_D10T30_min and dur_INB_D30T30_min indicates that dur_INB_D10T30_min corresponds to inactivity bouts lasting between 10 and 30 minutes and are defined by the threshold 30mg.

Motivation for default bout criteria for inactivity 0.9: Ssomewhat arbitrary decision, but the idea is that if you allow for bouts of 30 minutes it would not make sense to allow for breaks of 20 percent (6 minutes!) this is why I used a more stringent criteria for the highest category. Please note that you can change these criteria via arguments boutcriter.mvpa, boutcriter.in, and boutcriter.lig

4.3.2 Person level summary

Variables	Description
dur_nightsleep_min_pla	Sleep duration
XX_pla	plain average of variable XX across all days
XX_wei	weighted average of variable XX across all days where weekend days always weighted 2/5 relative to the contribution of week days.
L5VALUE_pla	Acceleration value for lowest 5hrs
M5VALUE_pla	Acceleration value for highest 5hrs
Acc_day_mg_pla	Average acceleration for daytime hours
ACC_night_mg_pla	Average acceleration for night hours
ACC_nightandday_mg_pla	Average acceleleration for night and day combined
Dur_day_SIB_min_pla	sustained inactivity bouts, are the periods of time during which the arm does not rotate at all for at least 5 or 10 minutes (depending on criteria used). This could be daytime sleep or the monitor not being worn for a very short period of time (these are not part of the inactivity bouts)
dur_day_OIN40_min_pla	Time spent below 40mg (OTHER INACTIVITY)
dur_day_LIG40_100_min_pla	Time spent in light (only unbouted)
dur_day_MOD100_400_min_pla	Time spent in moderate (only unbouted)
dur_day_VIG400_min_pla	Time spent in vigorous (only unbouted)

Variables	Description
dur_MVPA_D10T100_min_pla	Bouts of at least 10 minutes of MVPA (i.e above 100mg)
dur_MVPA_D1T100_min_pla	Bouts between 1-9.99mins above MVPA
dur_INB_D10T40_min_pla	Bouts of 10mins INACTIVE (other inactivity or SIB)
dur_LIGB_D10T40_100_min_pla	Bouts of 10mins light
dur_LIGB_D1T40_100_min_pla	Bouts between 1-99.9mins light
dur_TSIBday_min_pla	T=unbouted+bouted. THIS IS DIFFERENT TO SIB, because TSIB is regardless of whether they contributed or not to inactivity (INB) bouts or not, whereas SIB is just time not part of INB. So would expect TSIB to be higher
dur_TOINday_min_pla	Unbouted and bouted time spent in 'other inactivity'
dur_TLIGday_min_pla	Unbouted and bouted time spent in light
dur_TMODday_min_pla	Unbouted and bouted time spent in mod
dur_TVIGday_min_pla	Unbouted and bouted time spent in vig
dur_TINday_min_pla	Unbouted and bouted time spent in TOTAL INACTIVITY=TOIN+SIB
nonwear_perc_day_pla	non-wear time day
nonwear_perc_night_pla	non-wear time night
nonwear_perc_nightandday_pla	non-wear time night and day
sleep_efficiency_pla	Sleep efficiency
boutcriter.in_pla	0.9-90% of the bout has to be above threshold. The reason that the criteria is 90% rather than 80%, is "somewhat arbitrary decision, but the idea was that if you allow for bouts of 30 minutes it would not make sense if you allow for breaks of 20% of the time(6 minutes!) this is why we used a more stringent criteria for the highest category"
boutcriter.lig_pla	0.8-80% has to be above
boutcriter.mvpa_pla	0.8-80% has to be above threshold

5 Motivation and clarification

In this chapter we will try to collect motivations and clarification behind GGIR which may not have been clear from the existing publications

5.1 Reproducibilty of GGIR analyses

Some tips to increase reproducibility of your findings:

1. When you publish your findings please remember to add the GGIR package version number. All of GGIR are archived by CRAN and available from the archive section on the package website, so this will enable others to reproduce your analyses. GGIR has evolved over the years. To get a better understanding of how versions differ you should check the NEWS sections from the package website.
2. Keep a copy of your scripts with the functions calls
3. Keep a copy of your milestone data
4. Try to be explicit in referring to variable names

5.2 Auto-calibration

An acceleration sensor works on the principle that acceleration is captured mechanically and converted into an electrical signal. The relationship between the electrical signal and the acceleration is usually assumed to be linear, involving an offset and a gain factor. We shall refer to the establishment of the offset and gain factor as the sensor calibration procedure. Accelerometers are usually calibrated as part of the manufacturing process under non-movement conditions using the local gravitational acceleration as a reference. The manufacturer calibration can later be evaluated by holding each sensor axis parallel (up and down) or perpendicular to the direction of gravity; readings for each axis should be ±1 and 0 g, respectively. However, this procedure can be cumbersome in studies with a high throughput. Furthermore, such a calibration check will not be possible for data that have been collected in the past and for which the corresponding accelerometer device does not exist anymore. Techniques have been proposed that can check and correct for calibration error based on the collected triaxial accelerometer data in the participant's daily life without additional experiments, referred to as autocalibration. The general principle of these techniques is that a recording of acceleration is screened for nonmovement periods. Next, the moving average over the nonmovement periods is taken from each of the three orthogonal sensor axes and used to generate a three-dimensional ellipsoid representation that should ideally be a sphere with radius 1 g. Here, deviations between the radius of the three-dimensional ellipsoid and 1 g (ideal calibration) can then be used to derive correction factors for sensor axis-specific calibration error. This auto-calibration performed by GGIR uses this technique and a more detailed description and demonstration can be found in the published paper.

Reference:

- van Hees VT, Fang Z, Langford J, Assah F, Mohammad A, da Silva IC, Trenell MI, White T, Wareham NJ, Brage S. Autocalibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. J Appl Physiol (1985). 2014 Oct 1;117(7):738-44. doi: 10.1152/japplphysiol.00421.2014. Epub 2014 Aug 7. PMID: 25103964 link

Key decisions to be made:

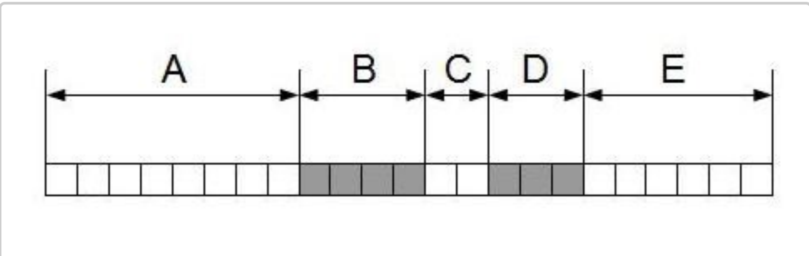
1. Whether to apply auto-calibration or not (default and recommended setting is YES). You can turn this off by changing do.call in g.shell.GGIR to do.call=FALSE.
2. Other variables are probably best left in their default setting

Key output variables:

1. Variable value cal.error.end as stored in data_quality_report.csv or variable value calib_err in summary.csv. These should be less than 0.01 g (10mg).

5.3 Non-wear detection

Accelerometer non-wear time was estimated on the basis of the standard deviation and the value range of the raw data from each accelerometer axis. Classification was done per 15 minute block and based on the characteristics of the 60 minute window centred at these 15 minutes. A block was classified as non-wear time if the standard deviation of the 60 minute window was less than 13.0 mg (1 mg = 0.00981 m·s−2) for at least two out of the three axes or if the value range, for at least two out of three axes, was less than 50 mg. The procedure for non-wear detection was modified in comparison to the procedure as applied in the 2011 PLoSONE publication link. Instead of 30-minute time windows 60-minute time windows were used to decrease the chance of accidently detecting short sedentary periods as non-wear time. The windows were overlapping (15 minute steps, window overlap of 45 minutes), which was done to improve the accuracy of detecting the boundaries of non-wear time as opposed to non-overlapping time windows. Inspection of unpublished data on non-wear classification by the algorithm as described in our published work indicated that the algorithm does not cope well with periods of monitor transportation per post. Here, long periods of non-wear are briefly interrupted by periods of movement, which are normally interpreted as monitor wear. Therefore, the algorithm was expanded with an additional stage in which the plausibility of “wear-periods” in-between non-wear periods is tested. Short periods of detected wear-time in-between longer periods of detected non-wear were classified as non-wear time based on the duration and the proportion of the duration relative to the bordering periods of detected non-wear-periods. The following criteria were derived from visual observation of various datasets using knowledge about study protocols. All detected wear-periods of less than six hours and less than 30% of the combined duration of their bordering non-wear periods were classified as non-wear. Additionally, all wear-periods of less than three hours and which formed less than 80% of their bordering non-wear periods were classified as non-wear. The motivation for selecting a relatively high criteria (< 30%) in combination with a long period (6hrs) and a low criteria (< 80%) in combination with a short period (3 hrs) was that long period are more likely to be actually related to monitor wear time. A visual model was created, see Figure 1. Here, units of time are presented in squares and marked grey if detected as non-wear time. Period C is detected as wear-time and borders to non-wear periods B and D, see Figure 1. If the length of C is less than six hours and C divided by the sum of B and D is less than 0.3 then the first criteria is met and block C is turned into a non-wear period.



By visual inspection of >100 traces from a large observational study it turned out that applying this stage in three iterative stages allowed for improved classification of periods characterised by intermittent periods of non-wear and apparent wear. Further, an additional rule was introduced for the final 24 hours of each measurement. The final 24 hours are often considered the period in which the accelerometer is potentially taken off but moved because of transportation, e.g. by the mail service. All wear-periods in the final 24 hrs of each measurement shorter than three hours and preceded by at least one hour of non-wear time were classified as non-wear. Finally, if the measurement starts or ends with a period of less than three hours of wear followed by non-wear (any length) then this period of wear is classified as non-wear. These additional criteria for screening the beginning and end of the accelerometer file reflect the likelihood of movements that are involved when starting the accelerometer or downloading the data from the accelerometer.

Reference:

- van Hees VT, Gorzelniak L, Dean León EC, Eder M, Pias M, Taherian S, Ekelund U, Renström F, Franks PW, Horsch A, Brage S. Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. PLoS One. 2013 Apr 23;8(4):e61691. doi: 10.1371/journal.pone.0061691. Print 2013 link.

Key decisions to be made:

1. Size of windows
2. Whether to utilize the non-wear detection

Key output variables:

1. Raw classification
2. Non-wear duration
3. Non-wear duration taking into account the protocol

5.4 Clipping score

The acceleration signal was screened for 'clipping'. If more than 50% of the data points in a 15 minute time window are higher than 7.5g (close to the maximal dynamic range of this sensor) the corresponding time period is considered as potentially corrupt data, which may be explained by the sensor getting stuck at its extreme value.

Reference:

- van Hees VT, Gorzelniak L, Dean León EC, Eder M, Pias M, Taherian S, Ekelund U, Renström F, Franks PW, Horsch A, Brage S. Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. PLoS One. 2013 Apr 23;8(4):e61691. doi: 10.1371/journal.pone.0061691. Print 2013 link

5.5 Why collapse information to epoch level?

Although many data points are collected we decide to only work with aggregated values (e.g. 1 or 5 second epochs) for the following reasons:

1. Accelerometers are often used to describe patterns in metabolic energy expenditure. Metabolic energy expenditure is typically defined per breath or per minute (indirect calorimetry), per day (room calorimeter), or per multiple days (doubly labelled water method). In order to validate our methods against these reference standards we need to work with a similar time resolution.
2. Collapsing the data to epoch summary measures helps to standardise for differences in sample frequency between studies.
3. There is little evidence that the raw data is an accurate representation of body acceleration. All scientific evidence on the validaty of accelerometer data has so far are based on epoch averages.
4. Collapsing the data to epoch summary measures may help to average out different noise levels and make sensor brands more comparable.

5.6 Sleep and full day time-use analysis in GGIR

5.6.1 Sleep analysis

In GGIR sleep analysis has been developed in part 4. Here the objective is to detect nocturnal Sleep Period Time window (often in bed at night). To do so, **sustained inactivity bouts (abbreviated as SIB)** are detected with the heuristic algorithm proposed in 2015 link which looks for periods of time where the z-angle does not change by more than 5 degrees for at least 5 minutes. SIBs can occur anytime in the day. In order to differentiate SIBs that correspond to daytime rest/naps from SIBs that correspond to the Sleep Period Time window (abbreviated as SPT), a guiding method referred as **guider** is used. This guiding method relies either on sleep log (sleep diaries) if available or on the Heuristic algorithm looking at Distribution of Change in Z-Angle (HDCZA), an algorithm developed to detect sleep window in the absence of sleep diary. These approaches are quickly described below.

- **Guider = sleep log**: As presented in before mentioned 2015 article). All SIBs that overlapped with the window defined by self-reported sleep window are considered as sleep within the SPT window. The start of the first SIB identified as nocturnal sleep period and the end of the last SIB identified as nocturnal sleep period define the beginning and the end of the SPT window. In this way the classification is relies on the accelerometer for detecting the timing of sleep onset, but relies on the guider to tell it in what part of the day it should look, as SPT window will be defined only if SIB is detected during the guider specified window.
- **Guider = HDCZA**: As presented in our 2018 article. The HDCZA algorithm guides us in identifying which SIB periods are to be labelled as part of SPT and those SIB periods then determine the start and end of the SPT window. See details of the method in the above reference. The time segment over which the HDCZA is derived are by default from noon to noon. However, if the HDCZA ends between 11am and noon then it will be applied again but to a 6pm-6pm window.
- **Guider = L5+/-12**: As presented in our 2018 article. Twelve hour window centered around the least active 5 hours of the day.
- **Guider = setwindow**: Window times are specificed by user, constant at specific clock

times with argument def.noc.sleep.

GGIR uses by default the sleep log if available, if the sleep log is not available it falls back on the HDCZA algorithm and if that for whatever reason is not available it falls back on the L5+/-12 definition. So, when we refer to guider then we refer to one of these three methods.

If the guider indicates that the person woke up after noon the sleep analysis in part 4 are performed on the window 6m-6pm. In this way our method is sensitive to people who have their main sleep period starting before noon and ending after noon, referred as daysleeper=1 in daysummary.csv file, which you can interpret as night workers. Note that the L5+/-12 algorithm is not configured to identify daysleepers, it will only consider the noon-noon time window.

To monitor possible problems with the sleep assessment, the variable **cleaningcode** is recorded for each night. Cleaningcode per night (noon-noon or 6pm-6pm as described above) can have one of the following values:

- 0: no problem, sleep log available and SPT is identified;
- 1: sleep log not available, thus HDCZA is used and SPT is identified,
- 2: not enough valid accelerometer data based on the non-wear and clipping detection from part summarized over the present night where the argument includenightcrit indicates the minimum number of hours of valid data needed within those 24 hours.
- 3: no accelerometer data available,
- 4: there were no nights to be analyzed for this person,
- 5: SPT estimated based on guider only, because either no SIB was found during the entire guider window, which complicates defining the start and end of the SPT, or the user specified the ID number of the recording and the night number in the *data_cleaning_file* to tell GGIR to rely on the guider and not rely on the accelerometer data for this particular night
- 6: no sleep log available and HDCZA also failed for this specific night then use average of HDCZA estimates from other nights in the recording as guider for this night. If HDCZA estimates are not available during the entire recording then use L5+/-12 estimate for this night. The last scenario seems highly unlikely in a recording where the accelerometer was worn for at least one day.

All the information for each night is stored in the results/QC folder allowing tracing of the data analysis and night selection. The cleaned results stored in the results folder. In part 4 a night is excluded from the 'cleaned' results based on the following criteria:

- If the study proposed a sleep log to the individuals, then nights are excluded for which the sleep log was not used as a guider (i.o.w. nights with cleaningcode not equal to 0 or variable sleep log used equals FALSE).
- If the study did not propose a sleep log to the individuals, then all nights are removed with cleaningcode higher than 1.

The package allows some adjustments to be made after data quality check. The *data_cleaning_file* argument allows you to specify individuals and nights for whom part4 should entirely rely on the guider (for example if we decide to use sleep log only information.

5.6.2 Waking-waking or 24 hour time-use analaysis

In part 5 the sleep estimates from part 4 are used to describe 24-hour time use. Part 5 allows you to do this in two ways: Literally 24 hours which start and end at of a calendar day (default midnight, but modifiable with argument dayborder) or from waking up to waking up. In GGIR we refer to the former as **MM** windows and to the latter as **WW** windows. The onset and waking times are guided by the estimates from part 4, but if they are missing part 5 will attempt to retrieve the estimate from the guider method, because even if the accelerometer was not worn during the night, or a sleep log missing in a study where sleep log was proposed to the participants, estimates from a sleep log or HDCZA can still be considered a reasonable estimate of the SPT window in the context of 24-hour time use analysis.

If WW is used in combination with ignoring the first and last midnight, argument excludefirstlast, then the first wake-up time (on the second recording day) needs to be extracted for the first WW day. This is done with the guider method. This also means that the last WW window ends on the before last morning of the recording.

If you want to inspect the time series corresponding to these windows then see argument save_ms5rawlevels, which allows you to export the time series including behavioral classes and non-wear information to csv files. A distinction is made between the full results stored in the results/QC folder and the cleaned results stored in the results folder.

The default exclusion criteria for days in part 5 that leads to the cleaned results are:

- For both MM and WW defined days: is that tThe invalid time percentage during waking hours needs to be below a certain percentage, defined as: 100 x (1 – (the value of argument includedaycrit divided by 24)), with default 100 x (1-(16/24)) = 33%.
- For MM defined days only: The day needs to last at least 23 hours.

Note, no criteria is set for the amount of valid data during the SPT window, because all we are interested in part 5 is knowing the borders of the night and we trust that this was sufficiently estimated by part 4. If you disagree then please note that these all days are included in the full report available in results/QC folder.

The *data_cleaning_file* argument allows you to specify which person(s) and day(s) should be omitted in part 5.

5.7 Why use data metric ENMO?

In most studies we have been involved we used metric ENMO (Euclidean Norm Minus One with negative values rounded to zero). In 2013 Vincent wrote a paper in which he investigated different ways of summarising the raw acceleration data. In short, different metrics exist and there is very little literature to support the supperiority of any metric. As long as different studies use different metrics their findings will not be comparable. Therefore, the choice for metric ENMO is partially pragmatic. Vincent is using ENMO because:

1. Has demonstrated value in describing variance in energy expenditure, correlated with questionnaire data, able to describe patterns in physical activity
2. Easy to describe mathematically and by that higher chances of reproducibility

3. Attempt to quantify the actual biomechanical acceleration in universal units. I prefer to avoid using abstract index for movement quantification
4. The 2013 paper showed that when ENMO is used in combination with auto-calibration it has similar validity to filter-based metrics like HFEN (which is essentially the same as the MAD metric recently used in literature). The few studies I am aware of who contested this, did not use auto-calibration by which ENMO was evalauted under unrealistic conditions. Metric ENMO depends by design on well calibrated acceleration signals, while metrics like HFEN and MAD are not affected by calibration offset. The metric HFENplus proposed in the 2013 paper may actually be a better candidate, but is more computationally heavy and more difficult to describe and replicate.

I recently wrote a blog post on this topic that you may find interesting: https://medium.com/@vincentvanhees/ten-misunderstandings-surrounding-information-extraction-from-wearable-accelerometer-data-a4f767a865b6

5.8 What does GGIR stand for?

I (Vincent) wanted a short name and not to spend too much time finding it. At the time when I created GGIR I was working with GENEActiv and GENEA data In R, and that's how the name GGIR was born: Short, easy to remember, and as acronnym sufficiently vague to not be tight up with a specific functionality.

6 Other Resources

- The GGIR package manual provides documentation on individual functions.
- For general questions about how to use GGIR join our google group/mailing list.
- For bug reports please post them here.

7 Citing GGIR

A correct citation of research software is important to make your research reproducible and to acknowledge the effort that goes into the development of the software.

To do so, please report the GGIR version you used in the text. Additionally, please also cite:

1. Migueles JH, Rowlands AV, et al. GGIR: A Research Community–Driven Open Source R Package for Generating Physical Activity and Sleep Outcomes From Multi-Day Raw Accelerometer Data. Journal for the Measurement of Physical Behaviour. 2(3) 2019. doi: 10.1123/jmpb.2018-0063.

If your work depends on the quantification of **physical activity** then also cite:

2. van Hees VT, Gorzelniak L, et al. Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE 8(4) 2013. link
3. Sabia S, van Hees VT, Shipley MJ, Trenell MI, Hagger-Johnson G, Elbaz A, Kivimaki M, Singh-Manoux A. Association between questionnaire- and accelerometer-assessed

physical activity: the role of sociodemographic factors. Am J Epidemiol. 2014 Mar 15;179(6):781-90. doi: 10.1093/aje/kwt330. Epub 2014 Feb 4. PMID: 24500862 link

If you used the **auto-calibration functionality** then also cite:

4. van Hees VT, Fang Z, et al. Auto-calibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. J Appl Physiol 2014. link

If you used the **sleep detection** then also cite:

5. van Hees VT, Sabia S, et al. A novel, open access method to assess sleep duration using a wrist-worn accelerometer, PLoS ONE, 2015 link

If you used the **sleep detection without relying on sleep diary** then also cite:

6. van Hees VT, Sabia S, et al. Estimating sleep parameters using an accelerometer without sleep diary. Scientific Reports 2018. doi: 10.1038/s41598-018-31266-z. link