



Accelting

Advancing movement & sleep research

www.accelting.com



GGIR training: Day 1

Dr. Vincent van Hees & Dr. Jairo H Migueles

v.vanhees@accelting.com, jairo@jhmigueles.com

Slide deck version: 01 August 2024

© Accelting

BEFORE WE START

- Focus of this course
- Questions
- Video recording
- Slides + Documentation + Example data:
<https://www.accelting.com/ggir-training-materials/>

Learning goals for this session

- Run GGIR with default settings
- How to look up GGIR documentation
- Understand what GGIR parts 1 & 2 do
- Configure part 1 & 2 to your own needs

Introduction to GGIR

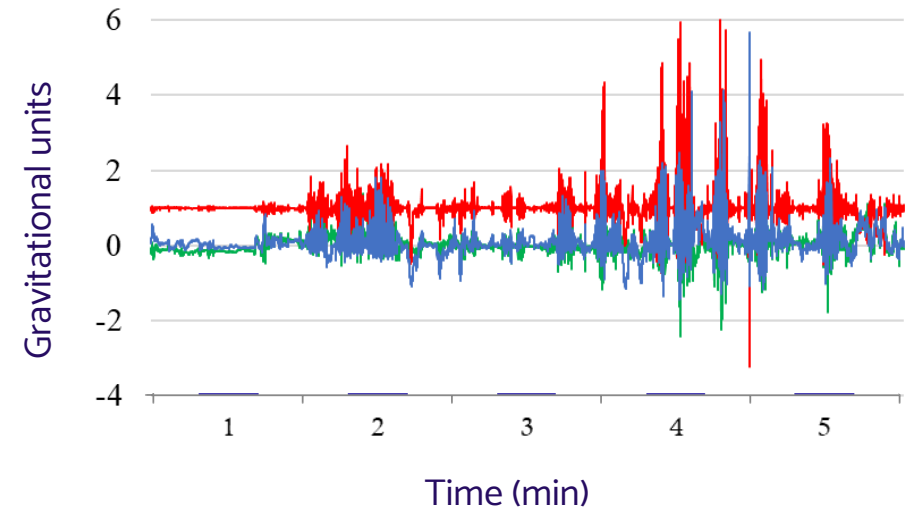
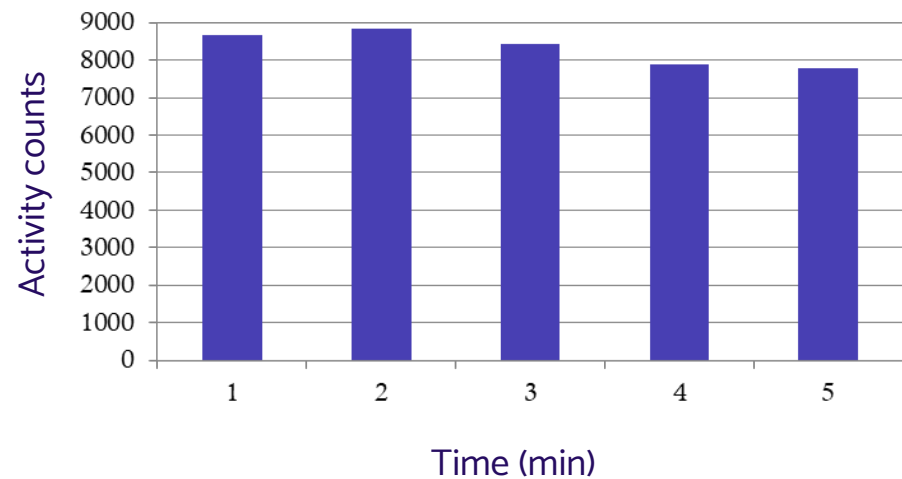
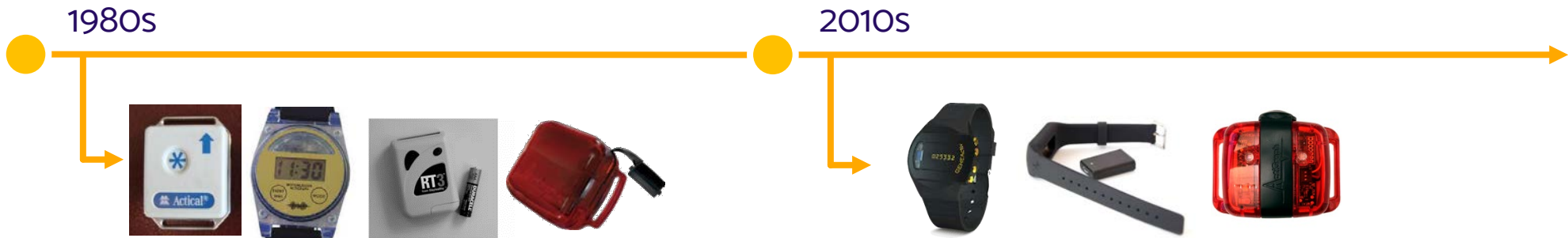


www.accelting.com

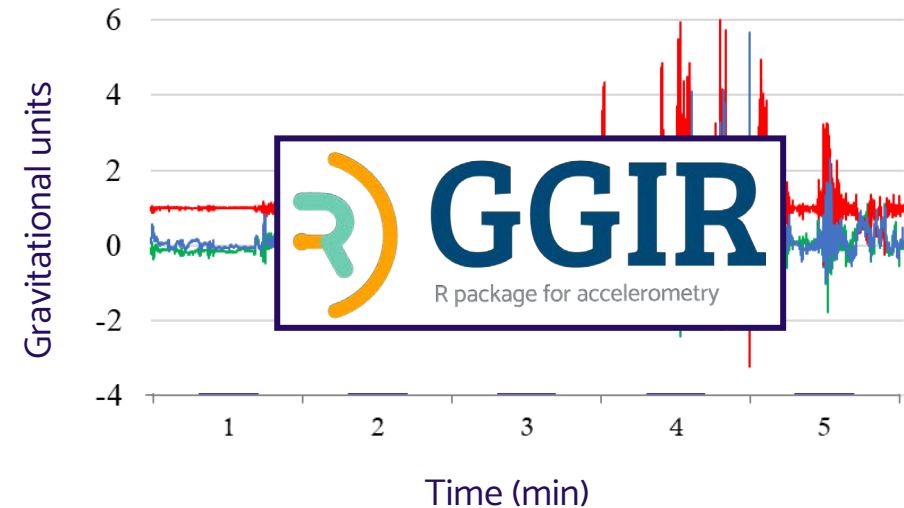
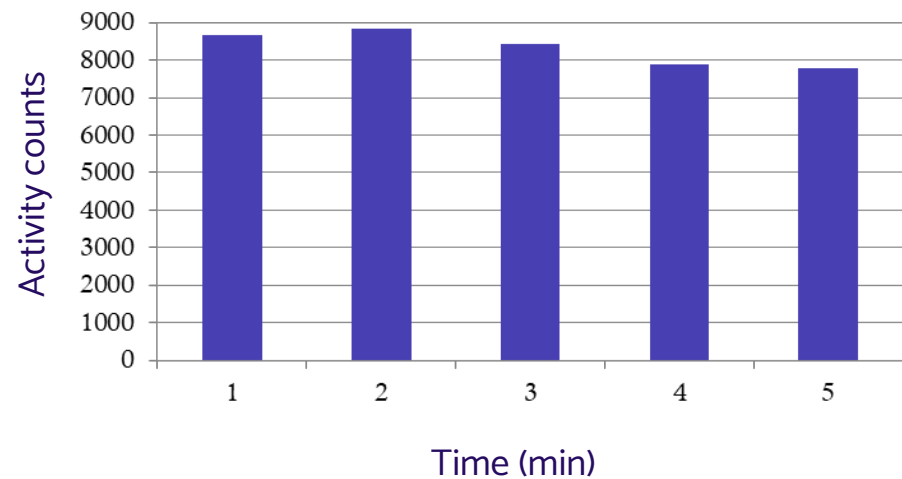
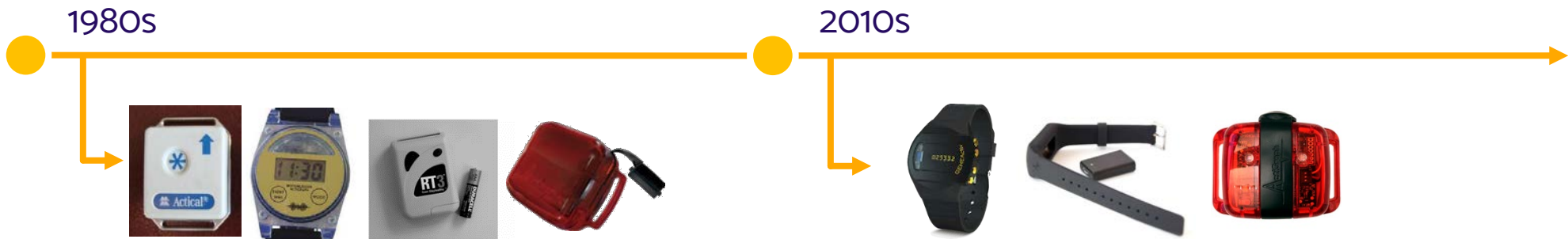


GGIR is an R-package to process
multi-day raw accelerometer data for
physical activity, sleep, and circadian rhythm
research.

Accelerometry



Accelerometry



In-built functionality to read

- Axivity AX3 and AX6 data (.cwa, and .csv)
- ActiGraph data (.gt3x and .csv)
- GENEActiv data (.bin)
- Movisens data (folder with inside .bin)
- Older devices epoch-level data:

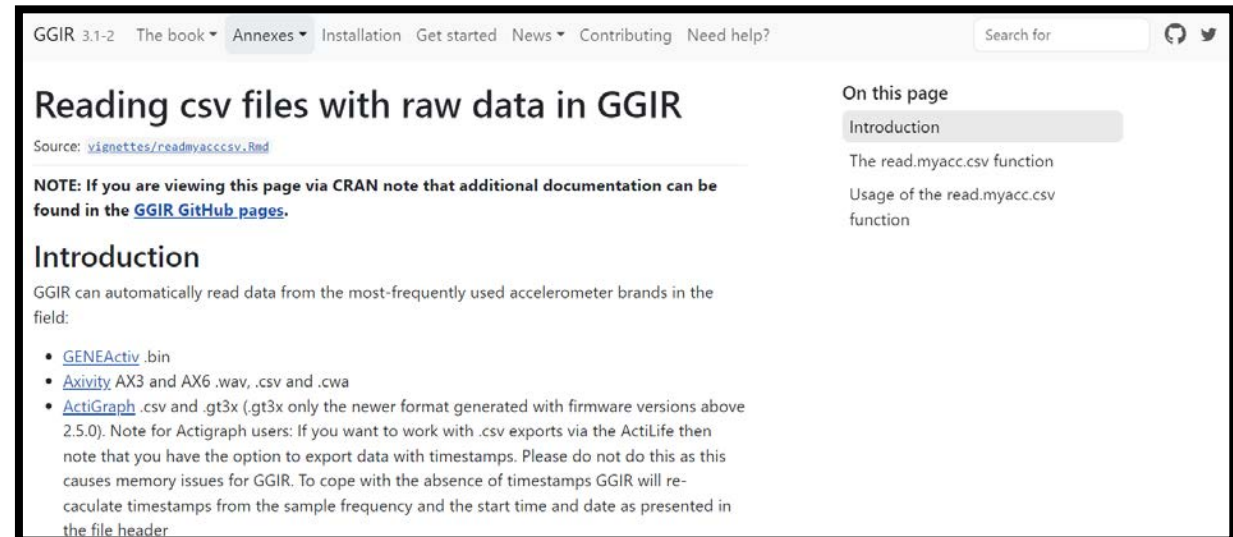
- UK biobank (.csv)
- Actiwatch (.csv, and .awd)
- Actigraph (.csv)
- Sensewear (.xls)

For details see:

https://wadpac.github.io/GGIR/articles/chapter2_Pipeline.html#externally-derived-epoch-level-data

And other csv files

- csv-files with raw acceleration data
 - Any brand
 - Flexible to variety of csv file structures
- For details see: <https://wadpac.github.io/GGIR/articles/readmyacc.csv.html>



Getting started with GGIR



www.accelting.com

The GGIR pipeline

Part 1

Part 2

Part 3

Part 4

Part 5

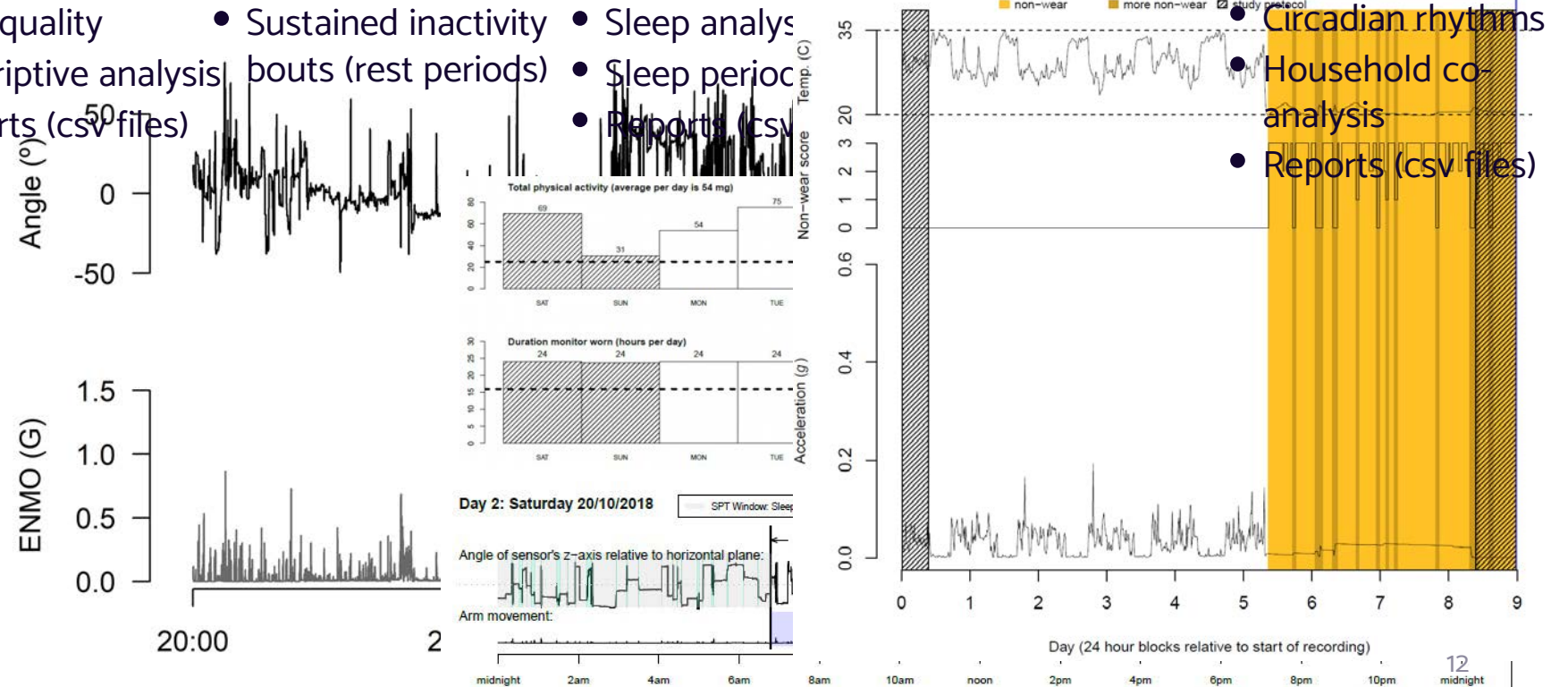
Part 6

- Reads file
- Extracts features
- Data quality

- Data quality
- Descriptive analysis
- Reports (csv files)

- Sustained inactivity bouts (rest periods)

- Sleep analysis
- Sleep period
- Reports (csv files)



- Circadian rhythms
- Household co-analysis
- Reports (csv files)

One R command

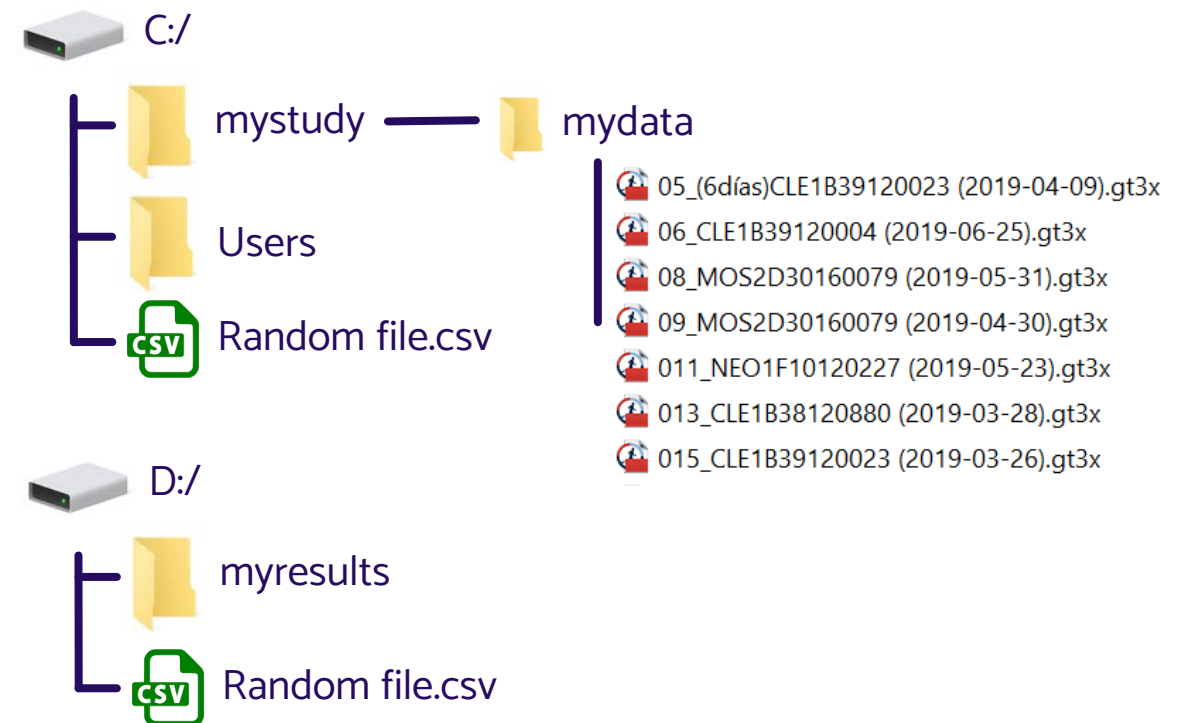
Using all defaults:

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

Notes:

- File paths are examples
- You can have data and output on different or on the same drive
- R uses single forward slashes '/' or double backward slashes '\\'
- Argument datadir must differ from argument outputdir

Example content of folder "mydata"

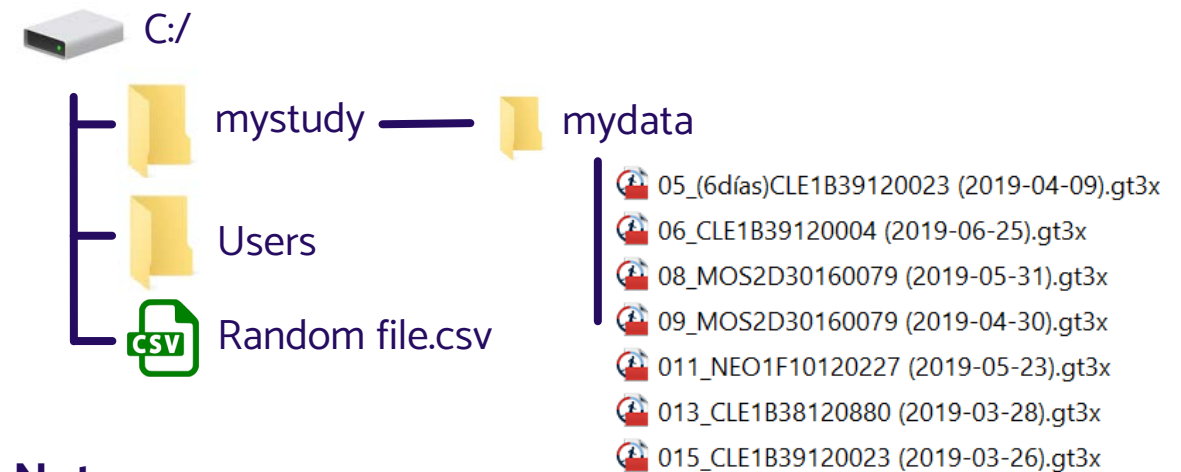


One R command

Toilored to a study:

```
library(GGIR)
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 = 6,
  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),
  includedaycrit.part5 = 2/3,
  #=====
  # Visual report
  #=====
  timewindow = c("WW"),
  visualreport=TRUE)
```

Example content of folder “mydata”



Notes:

- File paths are examples
- You can have data and output on different or on the same drive
- R uses single forward slashes '/' or double backward slashes '\\'
- Argument datadir must differ from argument outputdir

GGIR demo



General arguments



Running GGIR

General settings

GGIR(

general settings

`datadir` = "C:/mystudy/mydata",

`outputdir` = "D:/myoutput",

`mode` = c(1, 2, 3, 4, 5),

`dataFormat` = "raw",

`desiredtz` = "", *# <http://en.wikipedia.org/wiki/Zone.tab>*

`idloc` = 2,

`overwrite` = FALSE,

`do.parallel` = TRUE,

`[...]`

)

Running GGIR

General settings

GGIR(

```
# general settings  
datadir = "C:/mystudy/mydata",  
outputdir = "D:/myoutput",  
mode = c(1, 2, 3, 4, 5),  
dataFormat = "raw",  
desiredtz = "", # http://en.wikipedia.org/wiki/Zone.tab  
idloc = 2,  
overwrite = FALSE,  
do.parallel = TRUE,  
[...]  
)
```

mode

Numeric (default = 1:5). Specify which of the five parts need to be run, e.g., mode = 1 makes that g.part1 is run; or mode = 1:5 makes that the whole GGIR pipeline is run, from g.part1 to g.part5.

Running GGIR

General settings

GGIR(

general settings

`datadir` = "C:/mystudy/mydata",

`outputdir` = "D:/myoutput",

`mode` = c(1, 2, 3, 4, 5),

`dataFormat` = "raw",

`desiredtz` = "", # <http://en.wikipedia.org/wiki/Zone.tab>

`idloc` = 2,

`overwrite` = FALSE,

`do.parallel` = TRUE,

[...]

)

dataFormat

Character (default = "raw"). To indicate what the format is of the data in `datadir`. Alternatives: `ukbiobank_csv`, `actiwatch_csv`, `actiwatch_awd`, `actigraph_csv`, and `senswear_xls`, which correspond to epoch level data files.

Running GGIR

General settings

GGIR(

```
# general settings  
datadir = "C:/mystudy/mydata",  
outputdir = "D:/myoutput",  
mode = c(1, 2, 3, 4, 5),  
dataFormat = "raw",  
desiredtz = "", # http://en.wikipedia.org/wiki/Zone.tab  
idloc = 2,  
overwrite = FALSE,  
do.parallel = TRUE,  
[...]  
)
```

desiredtz

Character (default = "", i.e., system timezone). Timezone in which device was configured and experiments took place. If experiments took place in a different timezone, then use this argument for the timezone in which the experiments took place and argument configtz to specify where the device was configured. See also <http://en.wikipedia.org/wiki/Zone.tab>

Running GGIR

General settings

GGIR(

general settings

`datadir` = "C:/mystudy/mydata",

`outputdir` = "D:/myoutput",

`mode` = c(1, 2, 3, 4, 5),

`dataFormat` = "raw",

`desiredtz` = "", *# <http://en.wikipedia.org/wiki/Zone.tab>*

`idloc` = 2,

`overwrite` = FALSE,

`do.parallel` = TRUE,

Country code(s)	TZ identifier	Embedded comments	Type	UTC offset ±hh:mm		Time zone abbreviation		Source file	Notes
				STD	DST	STD	DST		
GB, GG, IM, JE	Europe/London		Canonical	+00:00	+01:00	GMT	BST	europe	
LU	Europe/Luxembourg		Link [†]	+01:00	+02:00	CET	CEST	backward	Link to Europe/Brussels
ES	Europe/Madrid	Spain (mainland)	Canonical	+01:00	+02:00	CET	CEST	europe	
MT	Europe/Malta		Canonical	+01:00	+02:00	CET	CEST	europe	
AX	Europe/Mariehamn		Link	+02:00	+03:00	EET	EEST	backward	Link to Europe/Helsinki

desiredtz

Character (default = "", i.e., system timezone). Timezone in which device was configured and experiments took place. If experiments took place in a different timezone, then use this argument for the timezone in which the experiments took place and argument `configtz` to specify where the device was configured. See also

<http://en.wikipedia.org/wiki/Zone.tab>

Running GGIR

General settings

GGIR(

general settings

`datadir` = "C:/mystudy/mydata",

`outputdir` = "D:/myoutput",

`mode` = c(1, 2, 3, 4, 5),

`dataFormat` = "raw",

`desiredtz` = "Europe/London", # <http://en.wikipedia.org/wiki/Zone.tab>

`idloc` = 2,

`overwrite` = FALSE,

`do.parallel` = TRUE,

[...]

)

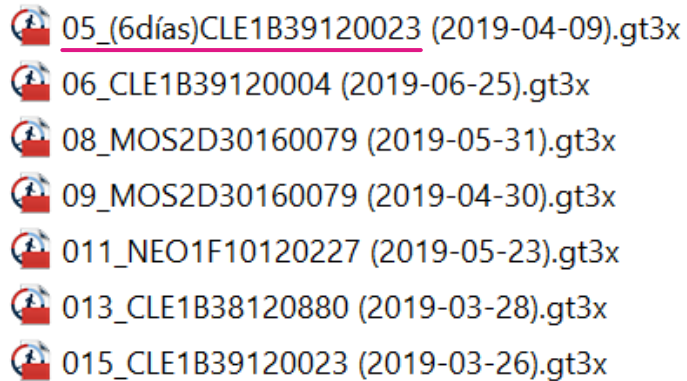
desiredtz

Character (default = "", i.e., system timezone). Timezone in which device was configured and experiments took place. If experiments took place in a different timezone, then use this argument for the timezone in which the experiments took place and argument `configtz` to specify where the device was configured. See also <http://en.wikipedia.org/wiki/Zone.tab>

Running GGIR

General settings

idloc = 8



05_(6días)CLE1B39120023 (2019-04-09).gt3x
06_CLE1B39120004 (2019-06-25).gt3x
08_MOS2D30160079 (2019-05-31).gt3x
09_MOS2D30160079 (2019-04-30).gt3x
011_NEO1F10120227 (2019-05-23).gt3x
013_CLE1B38120880 (2019-03-28).gt3x
015_CLE1B39120023 (2019-03-26).gt3x

GGIR(

general settings

`datadir = "C:/mystudy/mydata",`

`outputdir = "D:/myoutput",`

`mode = c(1, 2, 3, 4, 5),`

`dataFormat = "raw",`

`desiredtz = "Europe/London", # http://en.wikipedia.org/wiki/Zone.tab`

`idloc = 2,`

`overwrite = FALSE,`

`do.parallel = TRUE,`

`[...]`

`)`

idloc

Numeric (default: idloc = 1). If **idloc = 1** the code assumes that ID number is stored in the obvious header field. Note that for ActiGraph data the ID is never stored in the file header. For value set to 2, 5, 6, and 7, GGIR looks at the filename and extracts the character string preceding the first occurrence of a **'_' (idloc = 2)**, **' ' (space, idloc = 5)**, **'.' (dot, idloc = 6)**, and **'-' (idloc = 7)**, respectively. You may have noticed that idloc 3 and 4 are skipped, they were used for one study in 2012, and not actively maintained anymore, but because it is legacy code not omitted.

Running GGIR

General settings

05_(6días)CLE1B39120023 (2019-04-09).gt3x
06_CLE1B39120004 (2019-06-25).gt3x
08_MOS2D30160079 (2019-05-31).gt3x
09_MOS2D30160079 (2019-04-30).gt3x
011_NEO1F10120227 (2019-05-23).gt3x
013_CLE1B38120880 (2019-03-28).gt3x
015_CLE1B39120023 (2019-03-26).gt3x

GGIR(

general settings

`datadir = "C:/mystudy/mydata",`

`outputdir = "D:/myoutput",`

`mode = c(1, 2, 3, 4, 5),`

`dataFormat = "raw",`

`desiredtz = "Europe/London", # http://en.wikipedia.org/wiki/Zone.tab`

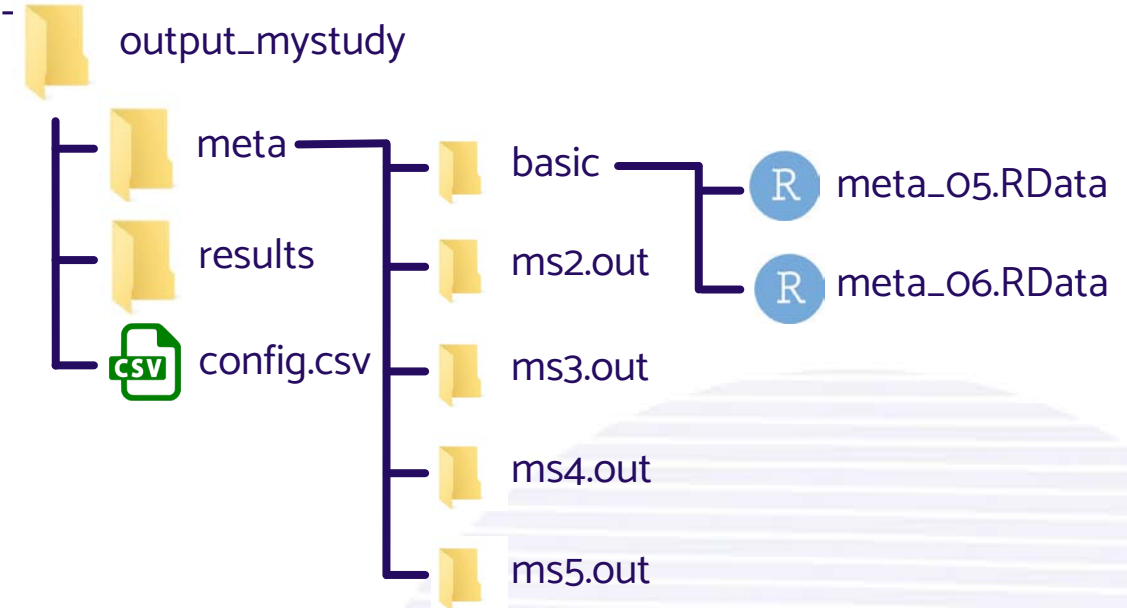
`idloc = 2,`

`overwrite = FALSE`

`do.parallel = FALSE`

`[...]`

`)`



overwrite

Boolean (default = FALSE). Do you want to overwrite analysis for which milestone data exists? If `overwrite=FALSE`, then milestone data from a previous analysis will be used if available and visual reports will not be created again.

Running GGIR

General settings

GGIR(

```
# general settings  
datadir = "C:/mystudy/mydata",  
outputdir = "D:/myoutput",  
mode = c(1, 2, 3, 4, 5),  
dataFormat = "raw",  
desiredtz = "Europe/London", # http://en.wikipedia.org/wiki/Zone.tab  
idloc = 2,  
overwrite = FALSE,  
do.parallel = TRUE,  
[...]  
)
```

do.parallel

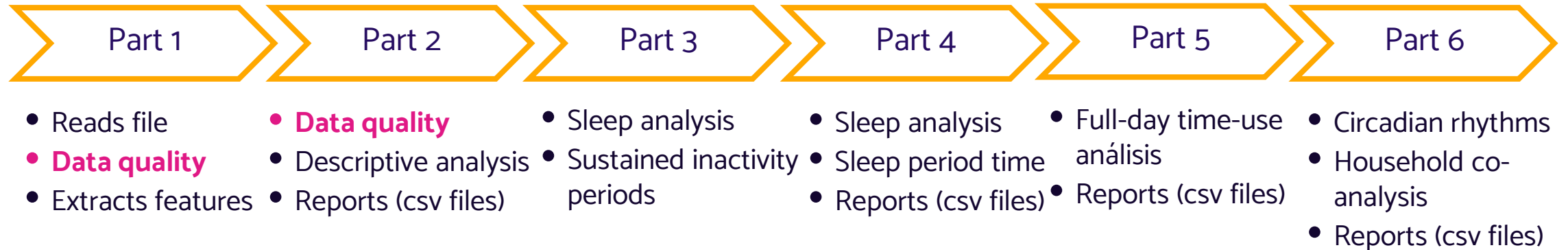
Boolean (default = TRUE). whether to use multi-core processing (only works if at least 4 CPU cores are available).

Data quality

Chapter 3



The GGIR pipeline



Data quality

- Time gap identification and imputation (gt3x and csv)
- Faulty block detection and imputation (.cwa)
- Auto-calibration of the accelerations (all)
- Non-wear detection (all)
- Clipping detection (all)

See https://wadpac.github.io/GGIR/articles/chapter3_QualityAssessment.html
for elaborate discussion

Non-wear time detection

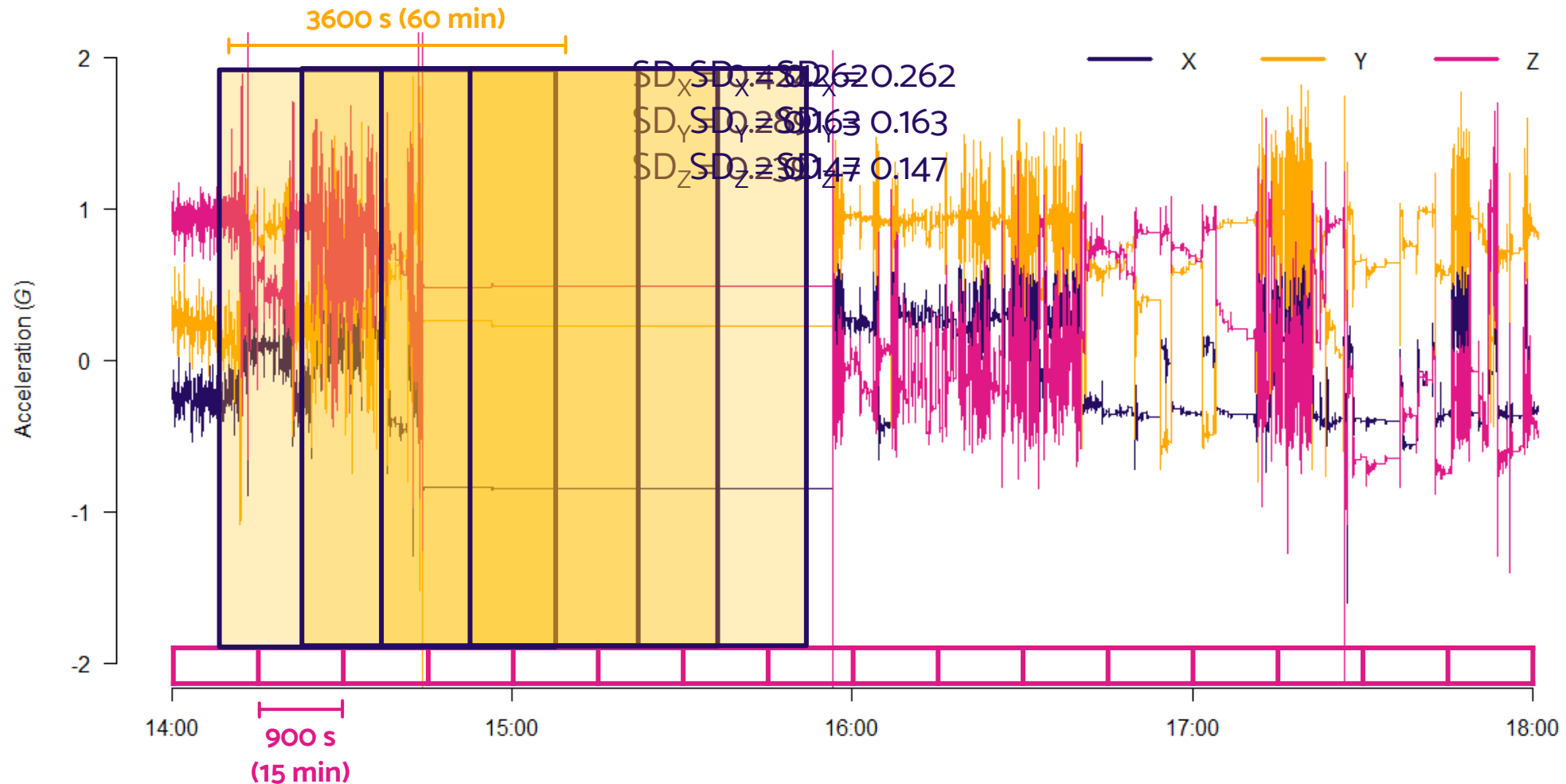
```
GGIR(# general settings  
[...]  
# data quality and metrics  
window sizes = c(5, 900, 3600),  
[...])
```

1. Standard deviation per axis per overlapping time window [1]
 - IF in at least 2 sensor axes:
 - standard deviation per hour [3600 seconds] \approx noise (noise threshold specific for each brand)
 - Difference between minimum and maximum value $<$ threshold
 - THEN:
 - Classification \Rightarrow not worn
2. Filter out 'wear' surrounded by a lot of non-wear [2]

Non-wear time detection

Device noise = 13 mg (0.013 g)

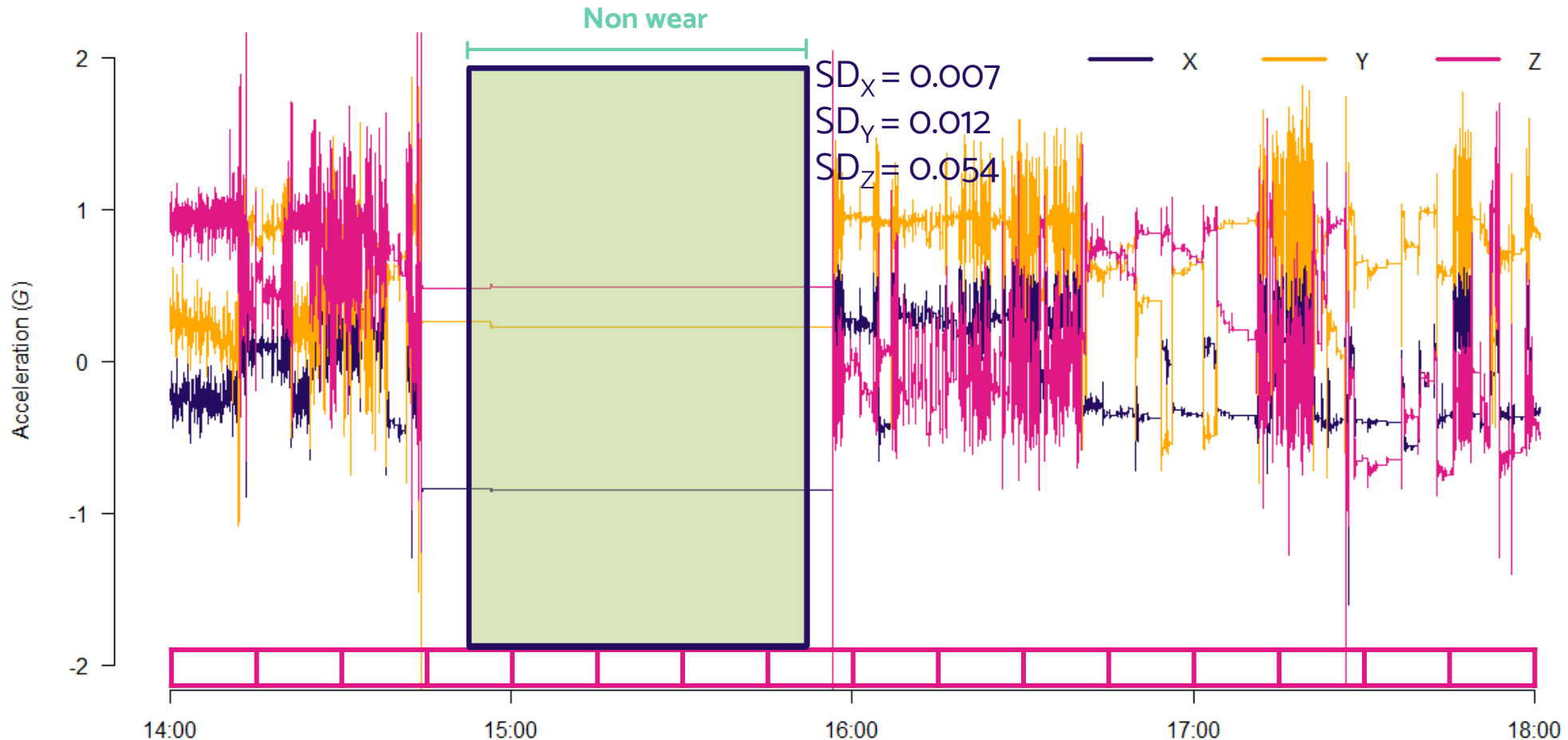
GGIR(# general settings
[...]
data quality and metrics
window sizes = c(5, 900, 3600),
[...])



Non-wear time detection

Device noise = 13 mg (0.013 g)

GGIR(# general settings
[...]
data quality and metrics
window sizes = c(5, 900, 3600),
[...])



The GGIR()

Data quality

```
GGIR(  
  [...]  
  window sizes = c(5, 900, 3600),  
  [...])
```

Acceleration metrics

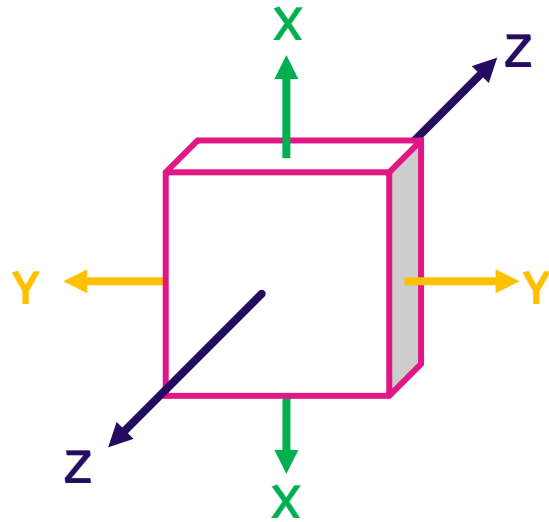
Chapter 4



The GGIR pipeline



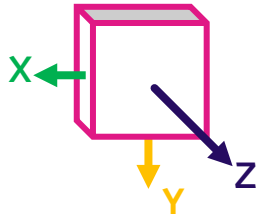
Raw data



X	Y	Z
-0.510	0.144	-0.850
-0.510	0.144	-0.845
-0.510	0.144	-0.850
-0.507	0.144	-0.845
-0.510	0.144	-0.850
-0.510	0.144	-0.845
-0.510	0.144	-0.845
-0.513	0.144	-0.842
-0.510	0.144	-0.845
-0.510	0.144	-0.845
-0.507	0.141	-0.845
-0.510	0.147	-0.845
-0.510	0.147	-0.845
-0.510	0.147	-0.842
-0.510	0.144	-0.845
-0.510	0.144	-0.842
-0.510	0.150	-0.842
-0.510	0.147	-0.845
-0.501	0.150	-0.845
-0.510	0.144	-0.845
-0.510	0.147	-0.850
-0.510	0.147	-0.845
-0.513	0.147	-0.842
-0.510	0.144	-0.842
-0.510	0.144	-0.845
-0.510	0.144	-0.845
-0.510	0.147	-0.842
-0.510	0.150	-0.845
-0.510	0.144	-0.842
-0.507	0.144	-0.845
-0.507	0.144	-0.845

Raw data

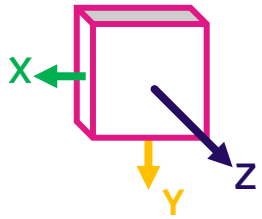
Gravity



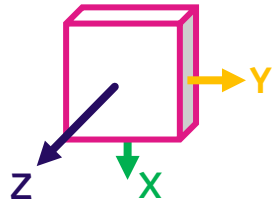
X-Axis	Y-Axis	Z-Axis

Raw data

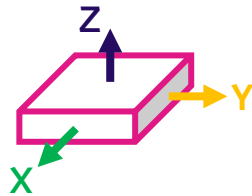
Gravity



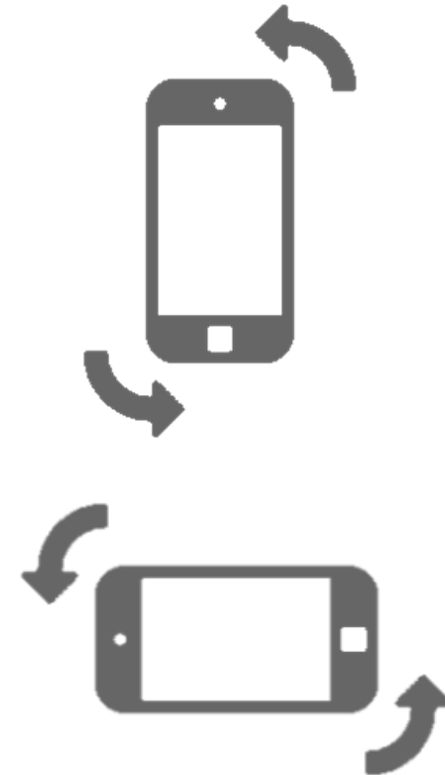
X-Axis	Y-Axis	Z-Axis
0g	-1g	0g



X-Axis	Y-Axis	Z-Axis
-1g	0g	0g



X-Axis	Y-Axis	Z-Axis
0g	0g	+1g



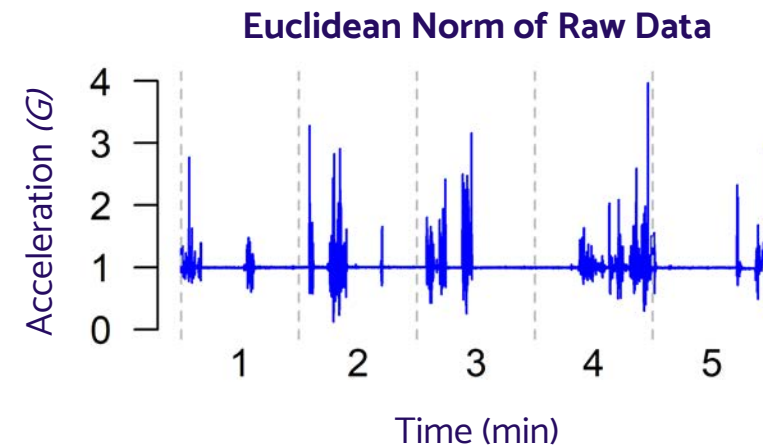
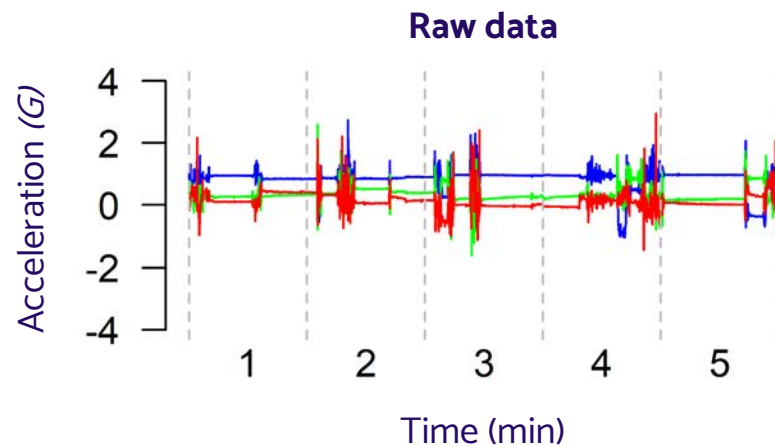
Euclidean Norm

X	Y	Z
-0.510	0.144	-0.850
-0.510	0.144	-0.845
-0.510	0.144	-0.850
-0.507	0.144	-0.845
-0.510	0.144	-0.850
-0.510	0.144	-0.845
-0.510	0.144	-0.845
-0.513	0.144	-0.842
-0.510	0.144	-0.845
-0.510	0.144	-0.845

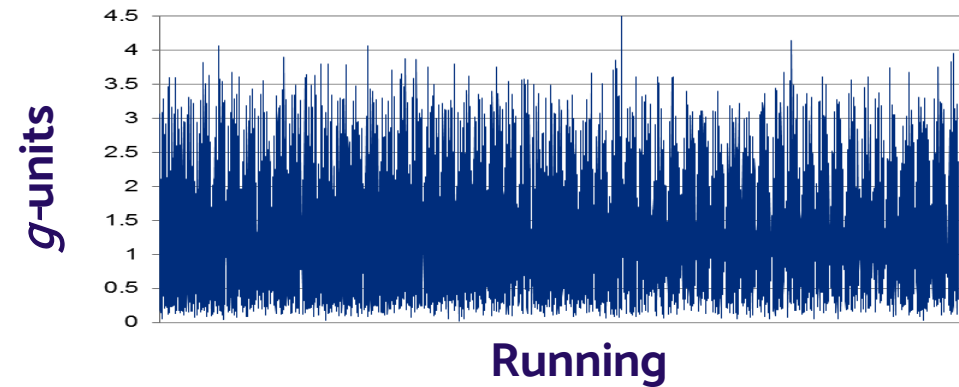
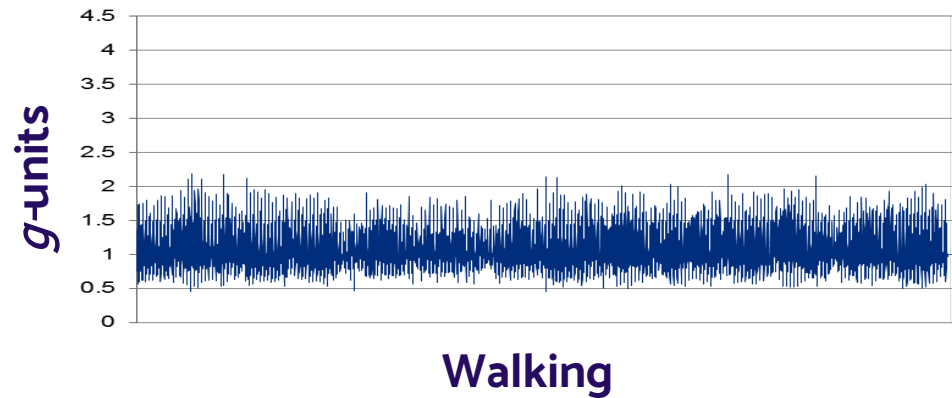
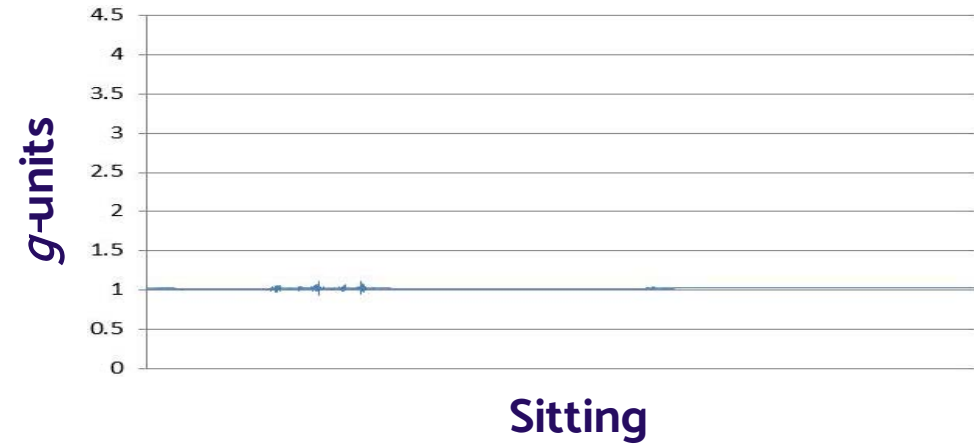
EN
1.002
0.997
1.002
0.996
1.002
0.997
0.997
0.996
0.997
0.997

Euclidean Norm (Vector Magnitude)

$$EN = \sqrt{x^2 + y^2 + z^2}$$



Euclidean Norm



Default ENMO metric in GGIR

Formula:

$$\max \left\{ \sqrt{acc_x^2 + acc_y^2 + acc_z^2} - 1, 0 \right\}$$

Broken down:

1. Euclidean norm:

$$EN = \sqrt{x^2 + y^2 + z^2}$$

2. Subtract 1g: $EN - 1$

3. **when** $(EN - 1) < 0$ **replace by 0**

4. $g \rightarrow mg$

5. Average per epoch

	A	B	C	D	E	F	G	H
1	Accelerometer.X	Accelerometer.Y	Accelerometer.Z	Euclidean Norm	Euclidean Norm Minus One	ENMO	ENMO_mg	ENMO_1s_mg
2	-0.791	-0.253	0.548	0.995	-0.005	0.000	0.000	4.147
3	-0.794	-0.251	0.545	0.996	-0.004	0.000	0.000	
4	-0.805	-0.253	0.551	1.008	0.008	0.008	8.156	
5	-0.797	-0.251	0.545	0.998	-0.002	0.000	0.000	
6	-0.794	-0.253	0.542	0.994	-0.006	0.000	0.000	
7	-0.797	-0.245	0.545	0.997	-0.003	0.000	0.000	
8	-0.797	-0.256	0.542	0.998	-0.002	0.000	0.000	
9	-0.660	-0.493	0.551	0.991	-0.009	0.000	0.000	
10	-0.557	-0.548	0.655	1.019	0.019	0.019	19.450	
11	-0.531	-0.539	0.675	1.014	0.014	0.014	13.864	
12	-0.516	-0.600	0.614	1.001	0.001	0.001	1.418	4.579
13	-0.575	-0.571	0.587	1.000	0.000	0.000	0.381	
14	-0.545	-0.589	0.576	0.988	-0.012	0.000	0.000	
15	-0.575	-0.586	0.573	1.001	0.001	0.001	0.908	
16	-0.528	-0.618	0.563	0.989	-0.011	0.000	0.000	
17	-0.548	-0.612	0.584	1.008	0.008	0.008	7.644	
18	-0.619	-0.481	0.655	1.021	0.021	0.021	20.926	
19	-0.572	-0.236	0.784	0.998	-0.002	0.000	0.000	
20	-0.805	-0.423	0.449	1.015	0.015	0.015	14.514	
21	-0.773	-0.452	0.419	0.989	-0.011	0.000	0.000	
22	-0.788	-0.408	0.407	0.977	-0.023	0.000	0.000	5.406
23	-0.776	-0.446	0.493	1.022	0.022	0.022	22.204	
24	-0.779	-0.455	0.451	1.009	0.009	0.009	8.912	
25	-0.773	-0.464	0.425	0.997	-0.003	0.000	0.000	
26	-0.817	-0.311	0.487	1.001	0.001	0.001	1.191	
27	-0.867	-0.251	0.419	0.995	-0.005	0.000	0.000	
28	-0.773	-0.493	-0.227	0.944	-0.056	0.000	0.000	
29	-0.976	-0.302	0.009	1.022	0.022	0.022	21.751	

Acceleration metrics in GGIR

Magnitude-based removal of gravity

- ENMO
- ENMOa
- LFENMO

No attempt to remove gravity

- EN
- LF_x , LF_y , LF_z
- LFEN

Frequency-content based removal of gravity

- BFEN, BF_x , BF_y , BF_z
- HFEN, HF_x , HF_y , HF_z
- $HFEN_+$
- MAD
- (Brond counts)
- Neishabouri counts

Zero-crossing

- Zero-crossing counts
 - ZC_x , ZC_y , ZC_z

Embedding your own metrics

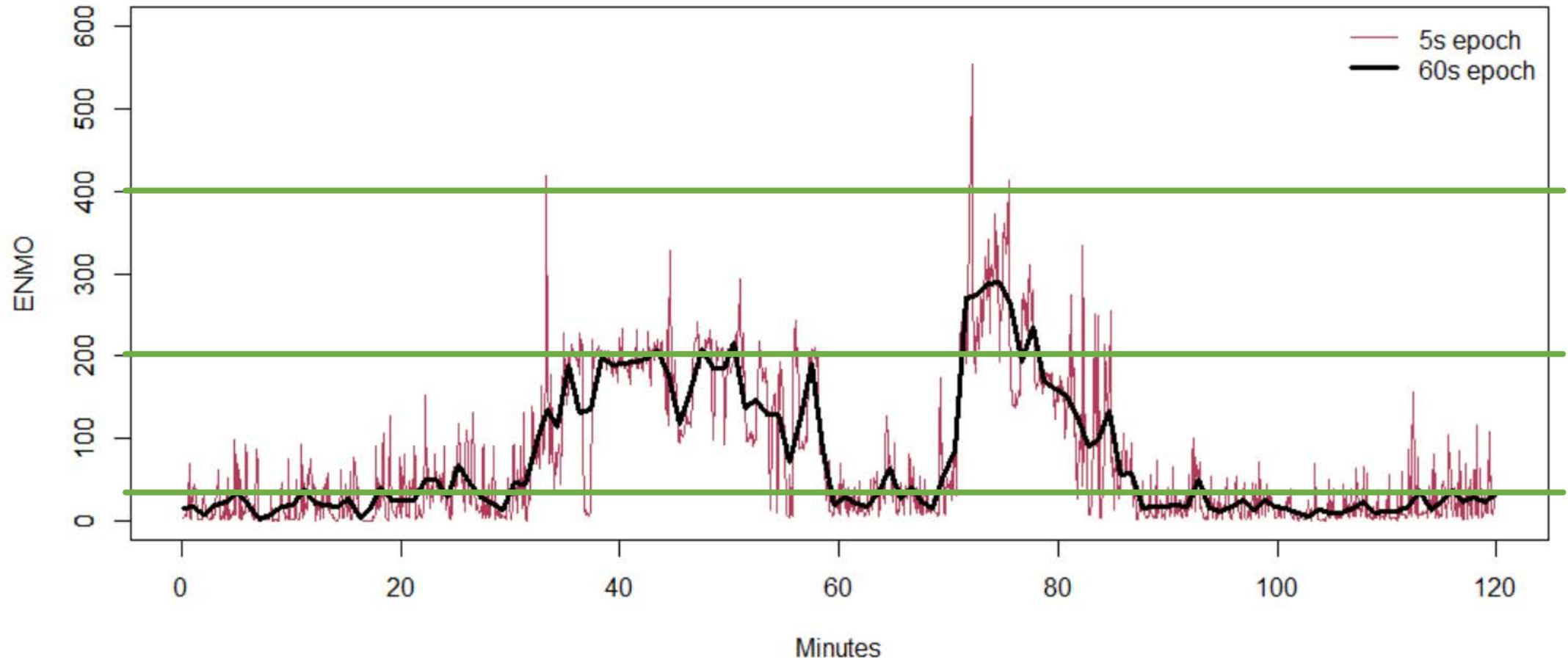
1. Download *verisense_count_steps.R* and *myscript.R* to a local location on your computer
2. Update the source path to the *verisense_count_steps.R* function in *myscript.R*. It should be local on your machine
3. Open R-studio and Source all GGIR functions as normal
4. Source *myscript.R*
5. Add 'myfun = myfun' as a line to your entry function to GGIR
6. Run GGIR and step count per day will be added to *part2_daysummary.csv*

Why do we aggregate per epoch?

- Reduces dependency on sampling frequency, which varies between studies
- Evidence on the value of raw accelerometer data primarily based on epoch aggregates
- Computational speed of subsequent analyses

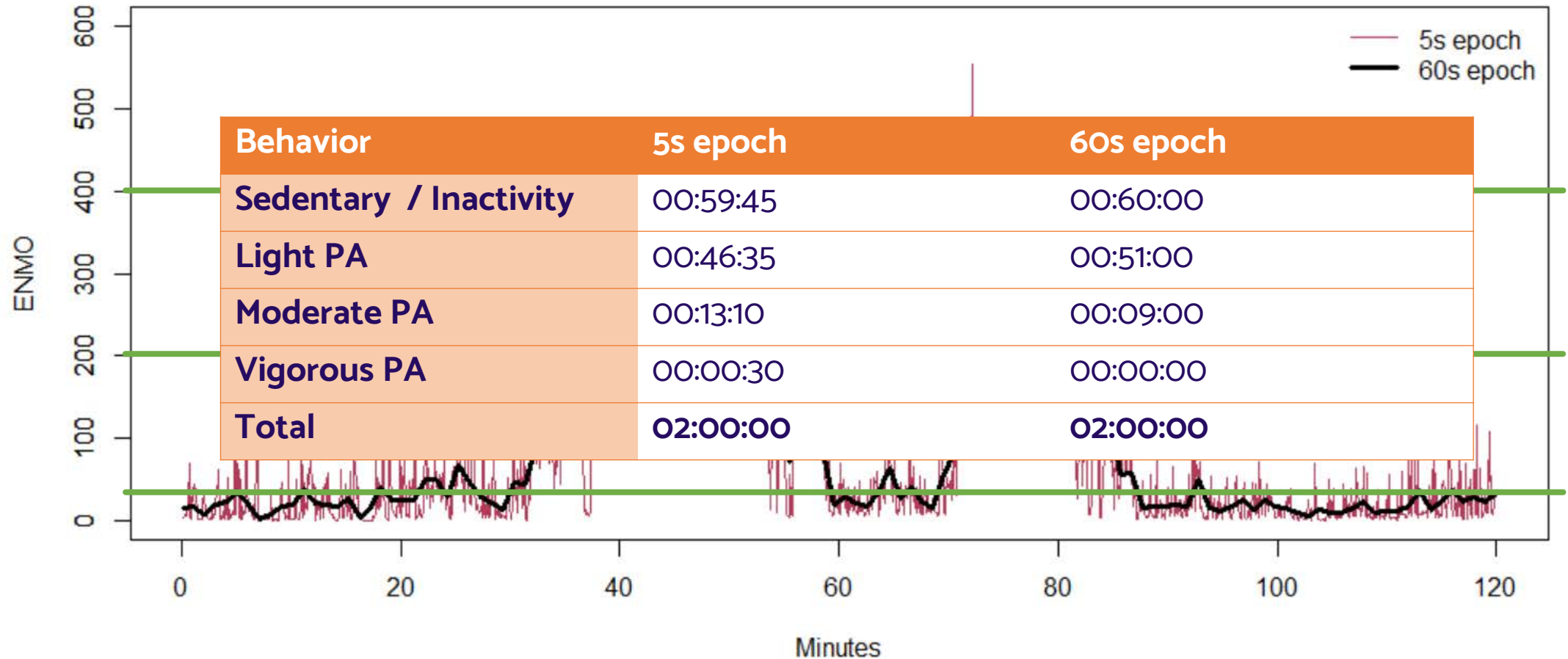
Epoch length

GGIR(# general settings
[...]
data quality and metrics
windowsizes = c(**5**, 900, 3600),
[...])



Epoch length

GGIR(# general settings
[...]
data quality and metrics
windowsizes = c(**5**, 900, 3600),
[...])



The GGIR()

Acceleration metrics

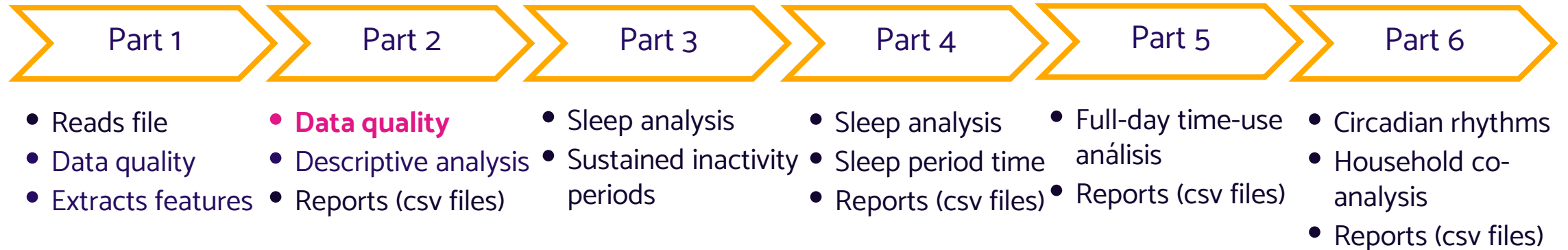
```
GGIR(  
  [...]  
  windowsizes = c(5, 900, 3600),  
  do.enmo = TRUE,  
  do.enmoa = FALSE,  
  do.anglex = FALSE,  
  do.angley = FALSE,  
  do.anglez = TRUE,  
  # [see all “do...” metrics]  
  [...])
```

Study protocol

Chapter 5



The GGIR pipeline



Need to select/mask data

- Non-wear detection may not be perfect
 - Accelerometer may be in the mail
 - Accelerometer may be left in a bag
 - Recording is expected to run longer than wear instruction
- Some days may be expected to include non-representative data
 - Participant is invited to come to the clinic

Available options in GGIR to select/mask data

- Exclude X hours from start
- Exclude X hours from end
- Exclude all data before first and after last midnight
- Exclude all data before first midnight
- Include X days with the highest activity levels
- Include only first X 24 hour blocks in data
- Include only first X calendar days
- Date-oriented masking of the data

Set maximum number of days or calendar days

```
GGIR(  
  [...]  
  # Study protocol  
  maxdur = 0,  
  max_calendar_days = 0,  
  [...])
```

The GGIR()

Study protocol

Purpose: analyze all the data available (default)

GGIR(

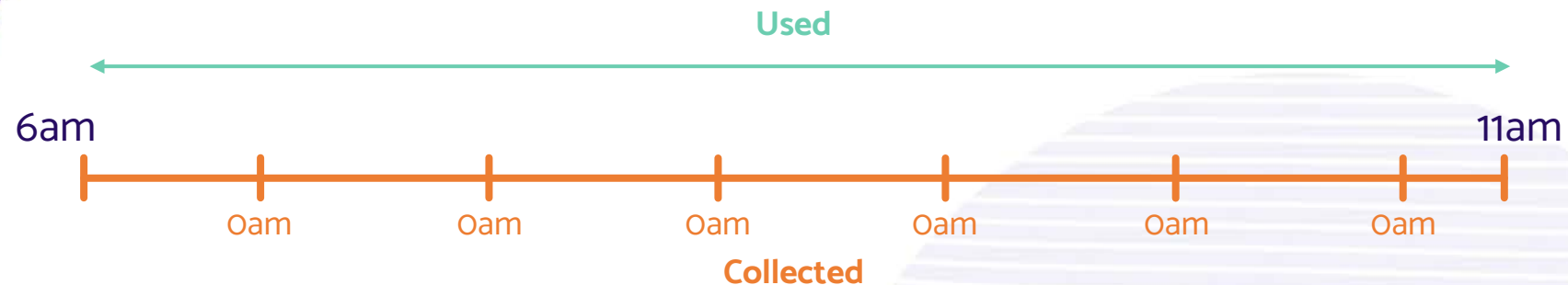
[...]

Study protocol

data_masking_strategy = 1,

hrs.del.start = 0, hrs.del.end = 0,

[...])



data_masking_strategy

Numeric (default = 1). How to deal with knowledge about study protocol.

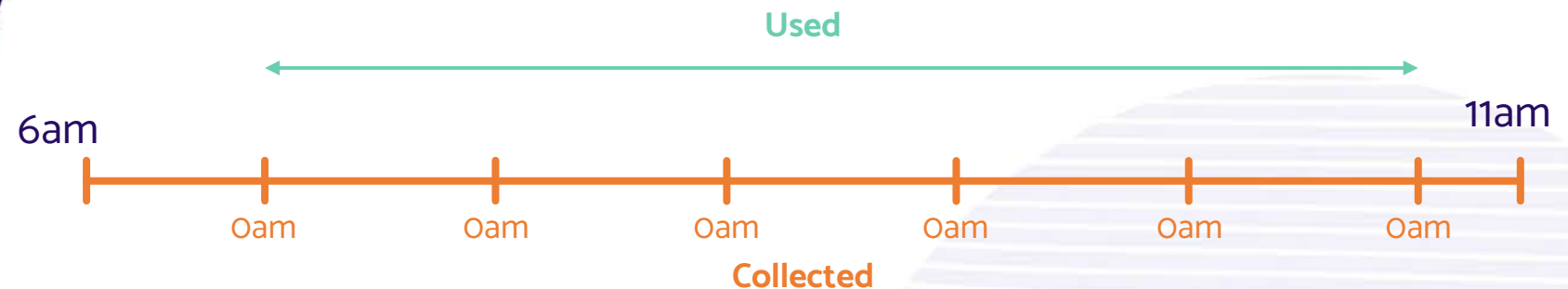
data_masking_strategy = 1 means select data based on **hrs.del.start** and **hrs.del.end**.

The GGIR()

Study protocol

Purpose: Skip first and last day

```
GGIR(  
  [...]  
  # Study protocol  
  data_masking_strategy = 2,  
  [...])
```



data_masking_strategy

Numeric (default = 1). How to deal with knowledge about study protocol.

data_masking_strategy = 2 makes that only the data between the first midnight and the last midnight is used.

The GGIR()

Study protocol

Purpose: Handle study protocol when recording lasts longer than instructed wear period

GGIR(

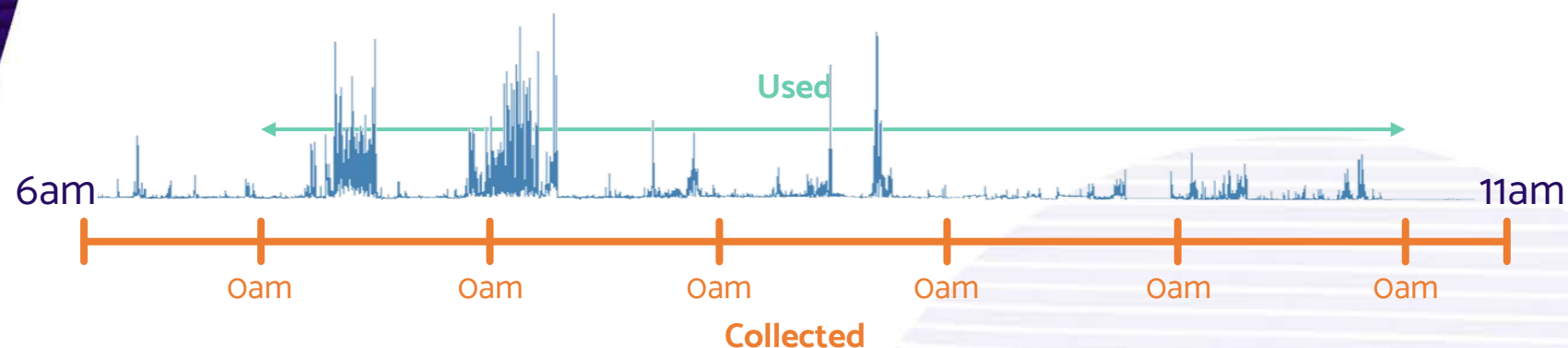
[...]

Study protocol

data_masking_strategy = 3,

ndayswindow = 3,

[...])



data_masking_strategy

Numeric (default = 1). How to deal with knowledge about study protocol.

data_masking_strategy = 3 only selects the most active X days in the file where X is specified by argument **ndayswindow**

The GGIR()

Study protocol

Purpose: Handle study protocol when recording lasts longer than instructed wear period

GGIR(

[...]

Study protocol

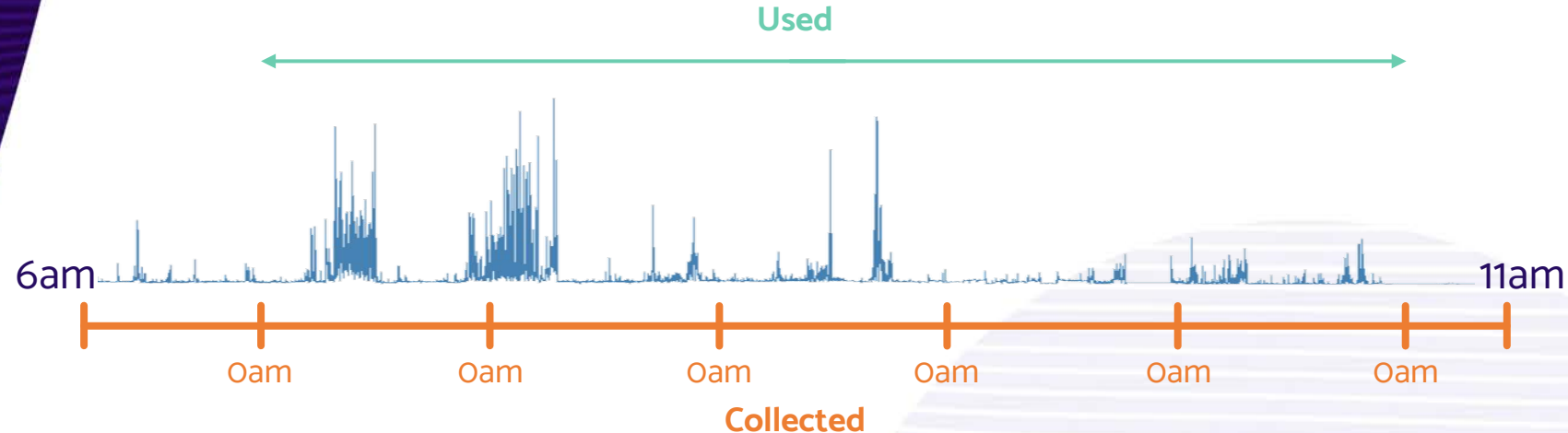
data_masking_strategy = 3,

ndayswindow = 3,

[...]

Assumption!

the days of data collection record higher activity than the other days



data_masking_strategy

Numeric (default = 1). How to deal with knowledge about study protocol.

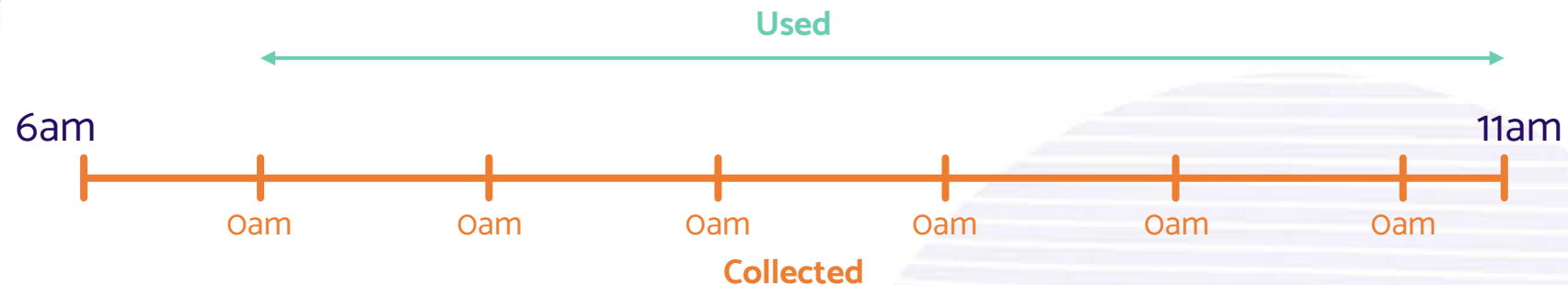
data_masking_strategy = 3 only selects the most active X days in the file where X is specified by argument **ndayswindow**

The GGIR()

Study protocol

Purpose: Skip only the first day

```
GGIR(  
  [...]  
  # Study protocol  
  data_masking_strategy = 4,  
  [...])
```



data_masking_strategy

Numeric (default = 1). How to deal with knowledge about study protocol.

data_masking_strategy = 4 to only use the data after the first midnight.

The GGIR()

Study protocol

Purpose: Not consistent but known start and end dates

GGIR(

[...]

Study protocol

study_dates_file = "C:/mystudy/diaries/study_dates_file.csv",

study_dates_dateformat = "%d/%m/%Y",

[...])



<https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/strptime>

	A	B	C
1	ID	Clinic.Date	Watch.Removal.Day
2	id01	25/10/2019	02/11/2019
3	id02	26/02/2020	07/03/2020
4	id03	01/06/2021	09/06/2021
5	id04	29/09/2021	07/10/2021
6	id05	13/07/2023	21/07/2023
7			

study_dates_file

Character (default = c()). Full path to csv file containing the first and last date of the expected wear period for every study participant (dates are provided per individual).

[...] Note that these dates are used on top of the **data_masking_strategy** selected

The GGIR()

Study protocol

GGIR(

[...]

Study protocol

dayborder = 0,

data_masking_strategy = 1,

hrs.del.start = 0, hrs.del.end = 0,

ndayswindow = 7,

study_dates_file = "C:/mystudy/diaries/study_dates_file.csv",

study_dates_dateformat = "%d/%m/%Y",

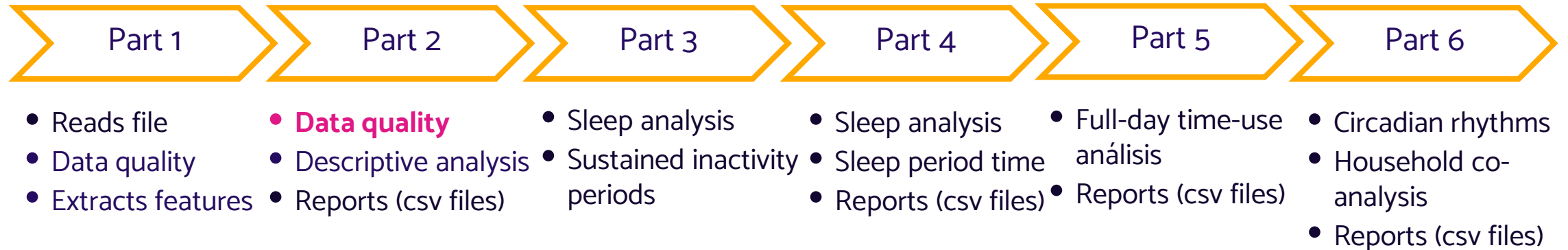
[...])

Dealing with Invalid Data

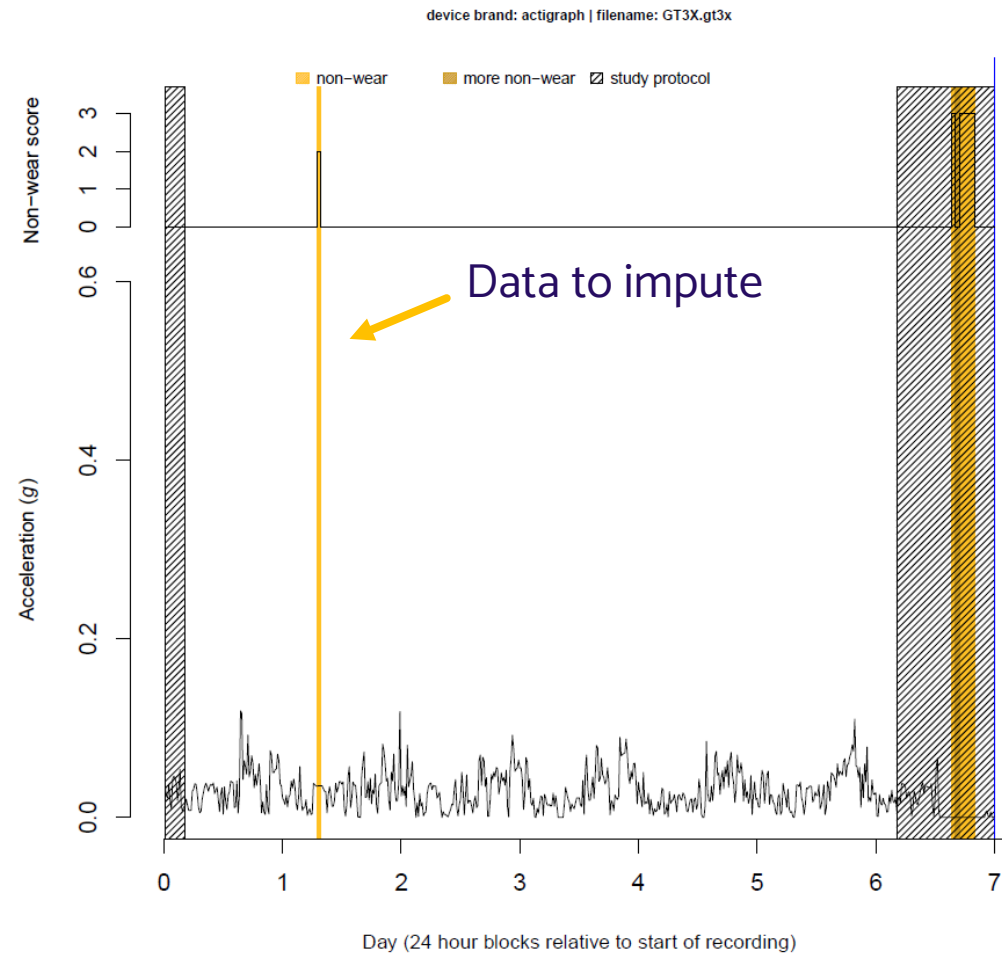
Chapter 6



The GGIR pipeline



Imputation of invalid data points



Imputation of invalid data points

Scenario 1: Monitor was not worn on Thursday from 9:00 to 9:30 AM

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Avg
9:00:00 - 9:00:05	3	4	3	2.2	2	0	1	2.2
9:00:05 - 9:00:10	3	5	2	2	1	0	1	2
9:00:10 - 9:00:15	2	4	2	1.8	1	0	2	1.8
9:00:15 - 9:00:20	3	4	3	2.3	2	1	1	2.3
...
9:29:55 - 9:30:00	5	2	4	2.8	2	1		2.8

Imputation of invalid data points

Scenario 2: Monitor was not worn any day from 9:00 to 9:30 AM

	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Avg
9:00:00 - 9:00:05	○	○	○	○	○	○	○	○
9:00:05 - 9:00:10	○	○	○	○	○	○	○	○
9:00:10 - 9:00:15	○	○	○	○	○	○	○	○
9:00:15 - 9:00:20	○	○	○	○	○	○	○	○
...
9:29:55 - 9:30:00	○	○	○	○	○	○	○	○

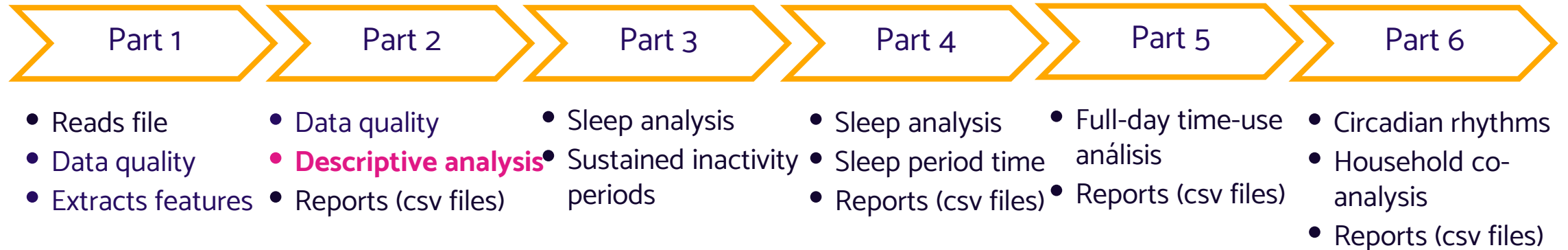
Do you want to turn off the data imputation?  **GGIR**(*[...]*
data quality and metrics
do.imp = **FALSE**,
[...])

Describing the data

Chapter 7



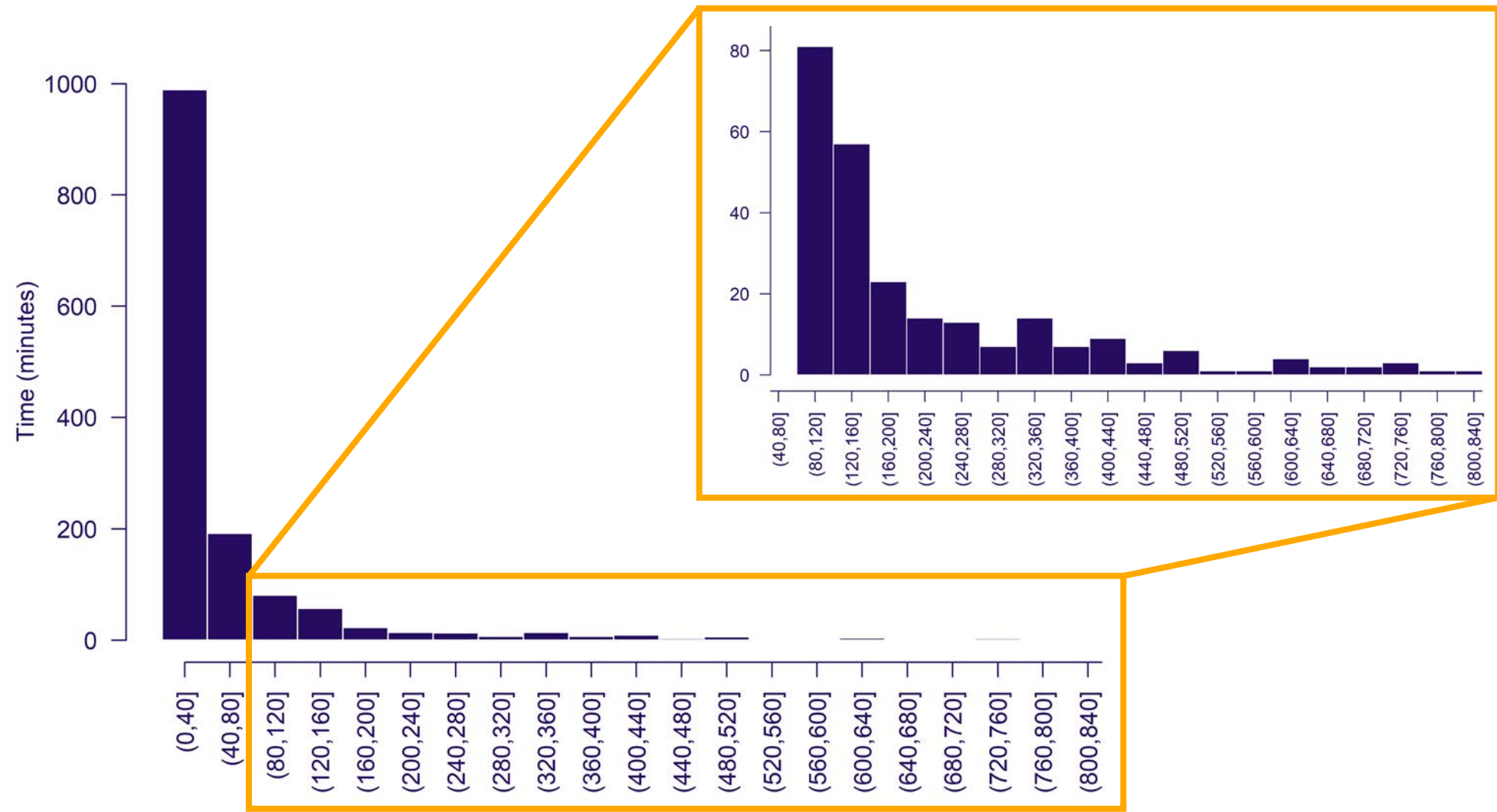
The GGIR pipeline




Describing acceleration data

- Average
 - Time-weighted of all valid data points across recording
 - Per day with invalid data points imputed
- Distribution
- Circadian rhythm analysis (session 3)
- Time spent in MVPA (session 3)

Acceleration distribution



MX metrics (Rowlands et al.)

- **Quantiles**  Percentiles of acceleration over the day expressed as a probability (e.g., value 0.5 refers to 12 hours [i.e., 0.5 over 24 hours])
- Intensity levels
- Intensity gradient

```
GGIR(  
  [...]  
  # Physical activity and acceleration distribution  
  qlevels = c(0.5, 0.7, 0.9, 0.95),  
  [...])
```

MX metrics (Rowlands et al.)

- **Quantiles** → Percentiles of acceleration over the day expressed as a probability (e.g., value 0.5 refers to 12 hours [i.e., 0.5 over 24 hours])
- Intensity levels
- Intensity gradient

MX metrics

$$M120 = (24 - 2) / 24 \sim 0.917$$

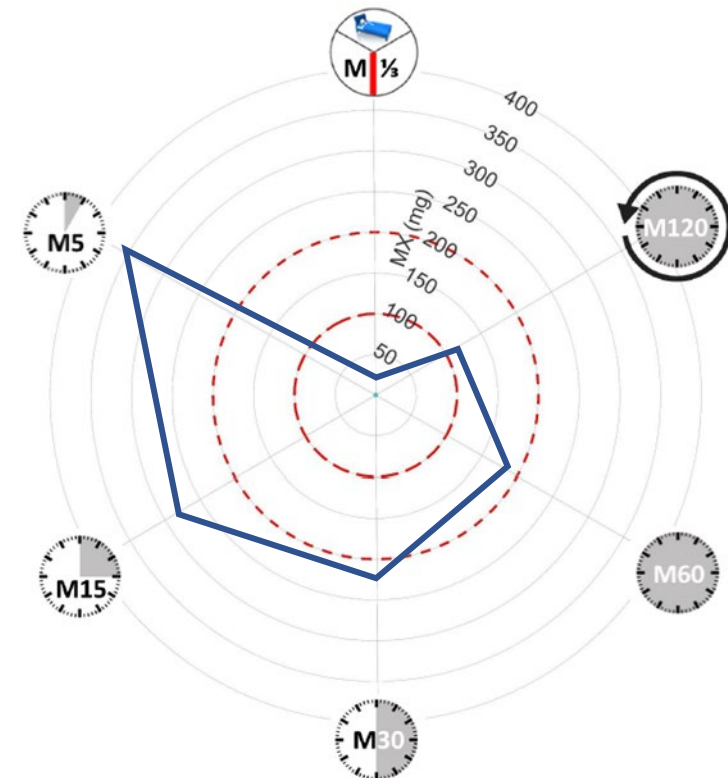
Rowlands et al. *Sports Medicine - Open* (2019) 5:47
<https://doi.org/10.1186/s40798-019-0225-9> Sports Medicine - Open

CURRENT OPINION Open Access


Enhancing the value of accelerometer-assessed physical activity: meaningful visual comparisons of data-driven translational accelerometer metrics

Alex V. Rowlands^{1,2,3*}, Nathan P. Dawkins^{1,2}, Ben Maylor^{1,2}, Charlotte L. Edwardson^{1,2}, Stuart J. Fairclough⁴, Melanie J. Davies^{1,2}, Deirdre M. Harrington^{1,2}, Kamlesh Khunti^{1,5} and Tom Yates^{1,2}

*Check for updates



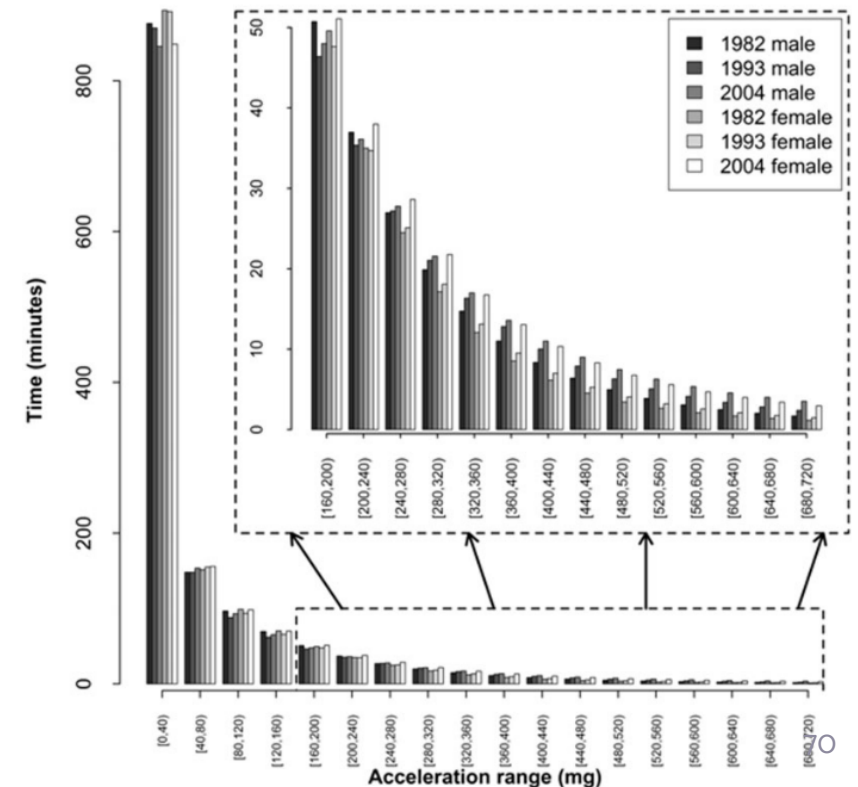
MX metrics (Rowlands et al.)

- **Quantiles**  Percentiles of acceleration over the day expressed as a probability (e.g., value 0.5 refers to 12 hours [i.e., 0.5 over 24 hours])
- Intensity levels
- Intensity gradient

```
GGIR(  
  [...]  
  # Physical activity and acceleration distribution  
  qlevels = c((24 - 2) / 24, # M120  
              (24 - 1) / 24), # M60  
              (1440 - 30) / 1440, # M30  
  [...])
```

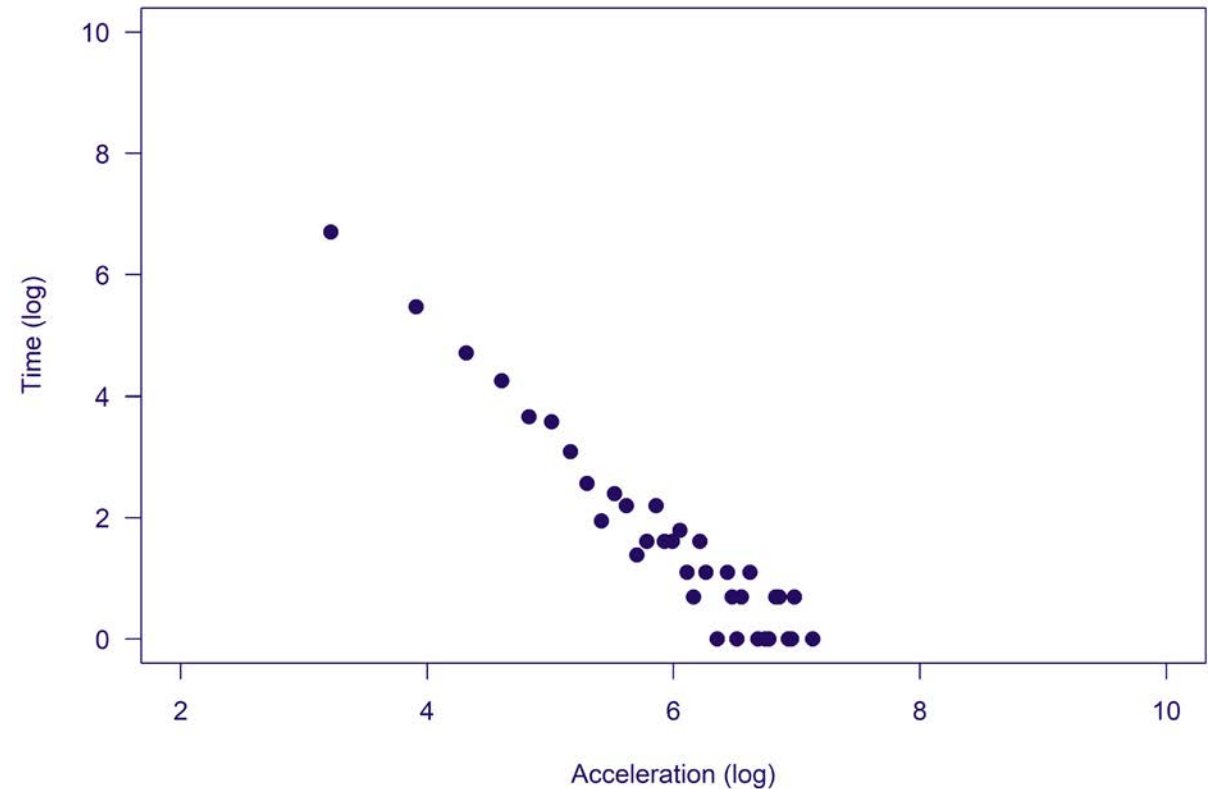
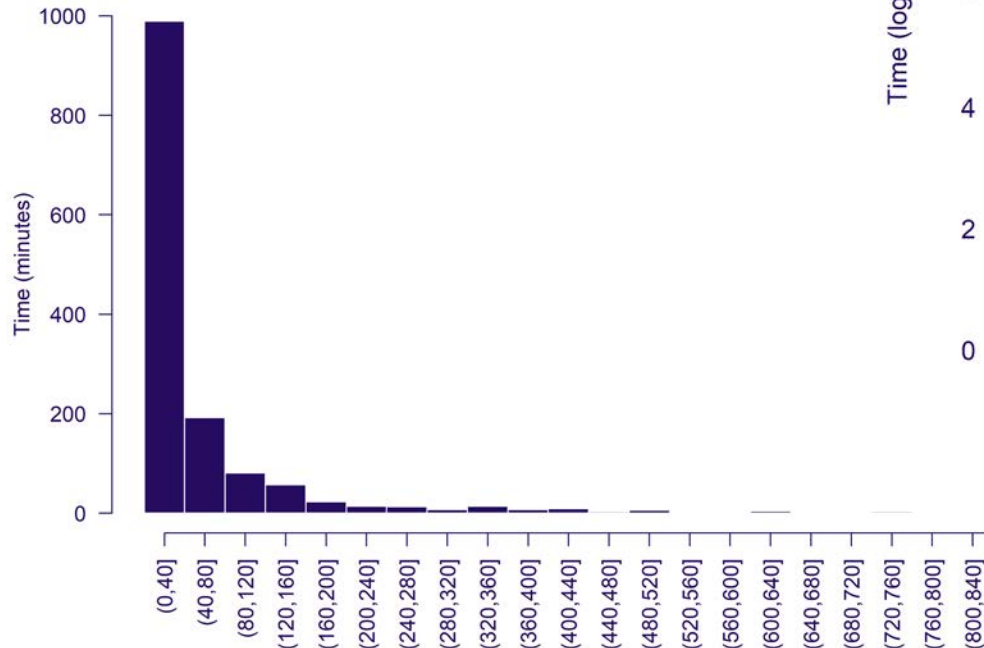
Acceleration distribution

- Quantiles
- **Intensity levels** → Time spent in intensity levels, e.g., ilevels = c(0, 50, 100, 200, 8000)
- Intensity gradient



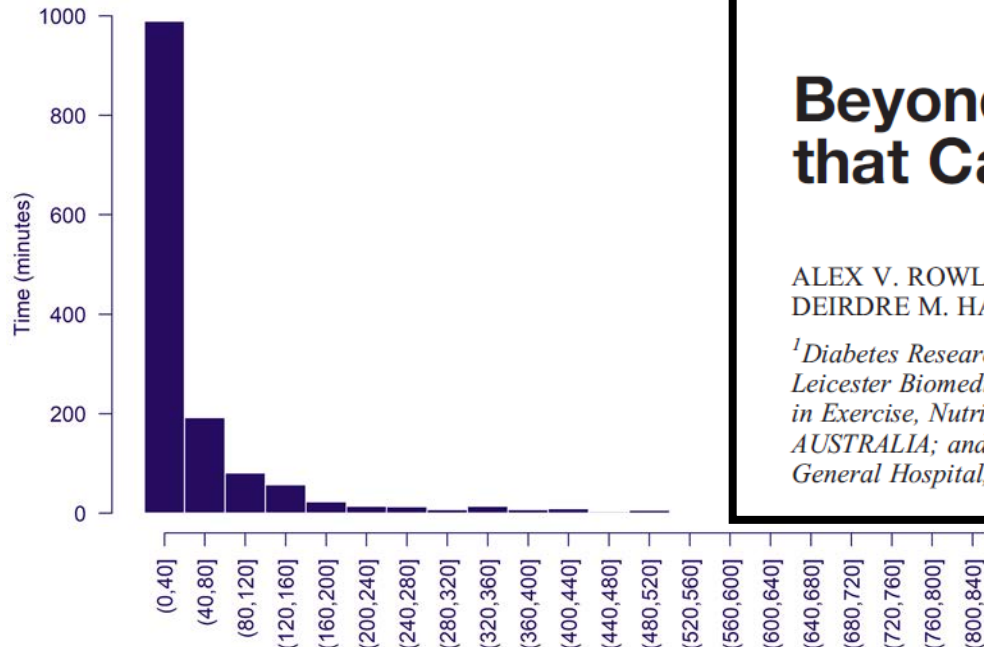
Intensity gradient

- Quantiles
- Intensity levels
- **Intensity gradient**



Intensity gradient

- Quantiles
- Intensity levels
- **Intensity gradient**



10

SPECIAL COMMUNICATIONS
Methodological Advances

Beyond Cut Points: Accelerometer Metrics that Capture the Physical Activity Profile

ALEX V. ROWLANDS^{1,2,3}, CHARLOTTE L. EDWARDSON^{1,2}, MELANIE J. DAVIES^{1,2}, KAMLESH KHUNTI^{1,2,4}, DEIRDRE M. HARRINGTON^{1,2}, and TOM YATES^{1,2}

¹Diabetes Research Centre, University of Leicester, Leicester General Hospital, Leicester, UNITED KINGDOM; ²NIHR Leicester Biomedical Research Centre, Leicester, UNITED KINGDOM; ³Division of Health Sciences, Alliance for Research in Exercise, Nutrition and Activity (ARENA), Sansom Institute for Health Research, University of South Australia, Adelaide, AUSTRALIA; and ⁴NIHR Collaboration for Leadership in Applied Health Research and Care East Midlands, Leicester General Hospital, Leicester, UNITED KINGDOM

Acceleration distribution

- Quantiles
- Intensity levels
- **Intensity gradient**

```
GGIR(  
  [...]  
  # Physical activity and acceleration distribution  
  qlevels = c(0.5, 0.7, 0.9, 0.95),  
  ilevels = c(0, 50, 100, 200, 8000),  
  iglevels = TRUE,  
  [...])
```

The GGIR pipeline

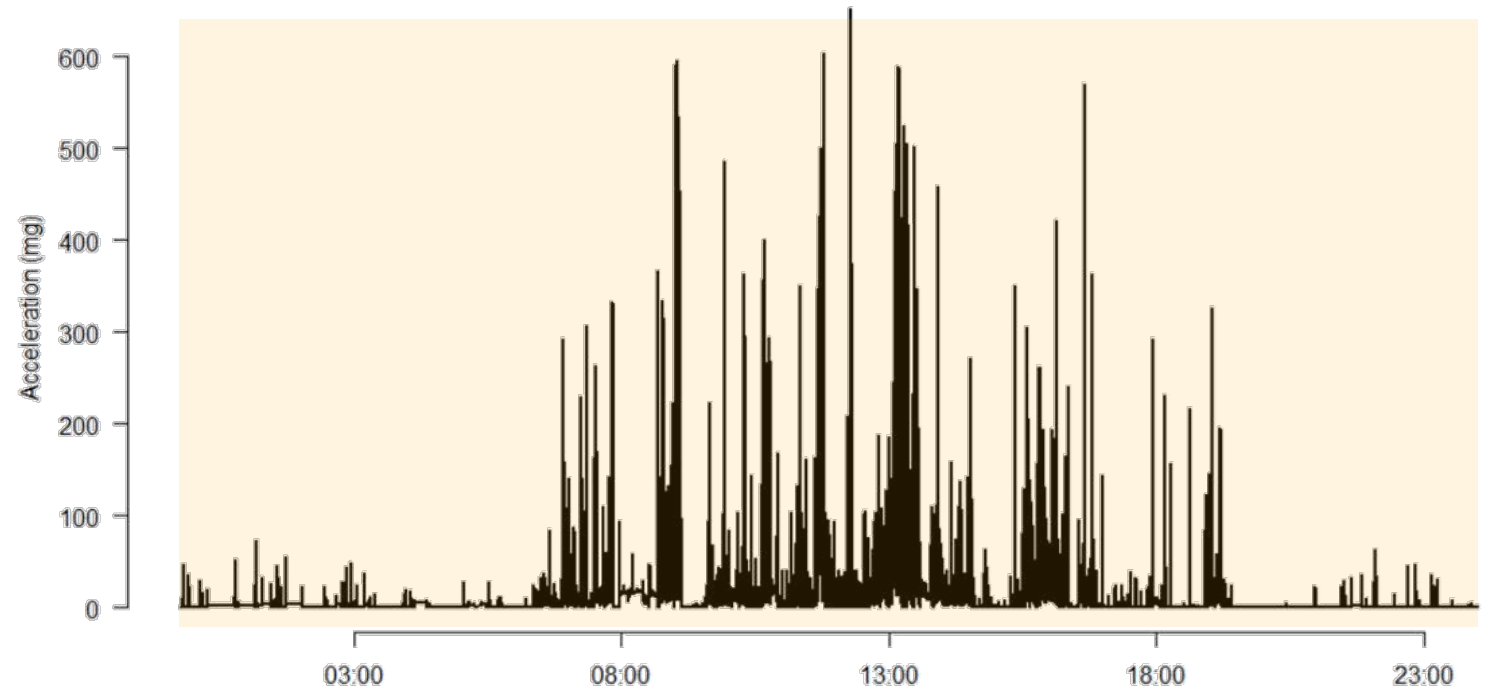


Windows definition

The argument **qwindow**

Numeric or character (default = c(0, 24)).

```
GGIR(  
  [...]  
  qwindow = c(0, 24),  
  [...])
```

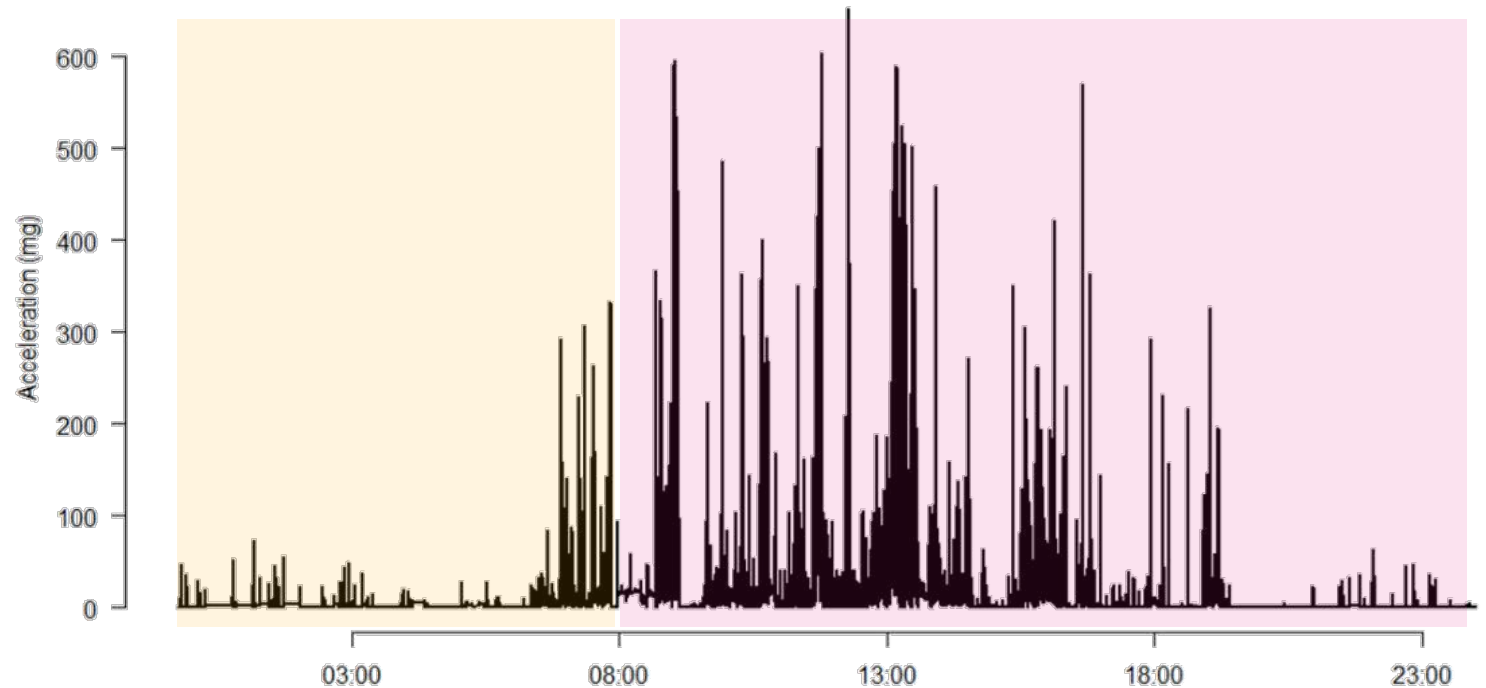


Windows definition

The argument **qwindow**

Numeric or character (default = c(0, 24)).

```
GGIR(  
  [...]  
  qwindow = c(0, 8, 24),  
  [...])
```



Windows definition

The argument **qwindow**

Numeric or **character**.

If you want to use a day specific segmentation, then you can set qwindow to be the **full path to activity diary file (csv file)**.

```
GGIR(  
  [...]  
  qwindow = "C:/mystudy/activitylog.csv",  
  [...])
```

ID	Date	PE_1	PE_2	Date	PE_1	PE_2
ID01	20-01-2022	09:00:00	10:00:00	21-01-2022		
ID02	22-01-2022	11:30:00	12:30:00	23-01-2022	09:00:00	10:00:00
ID03	02-02-2022			03-02-2022	10:00:00	11:00:00
ID04	15-01-2022	09:00:00	10:00:00	16-01-2022		
ID05	04-02-2022			05-02-2022	11:30:00	12:30:00

Will only get the 24h indicators

The GGIR()

Physical activity & distribution

GGIR(

[...]

Physical activity and acceleration distribution

qlevels = c(0.5, 0.7, 0.9, 0.95),

ilevels = c(0, 50, 100, 200, 8000),

iglevels = 1,

mvpthreshold = 100,

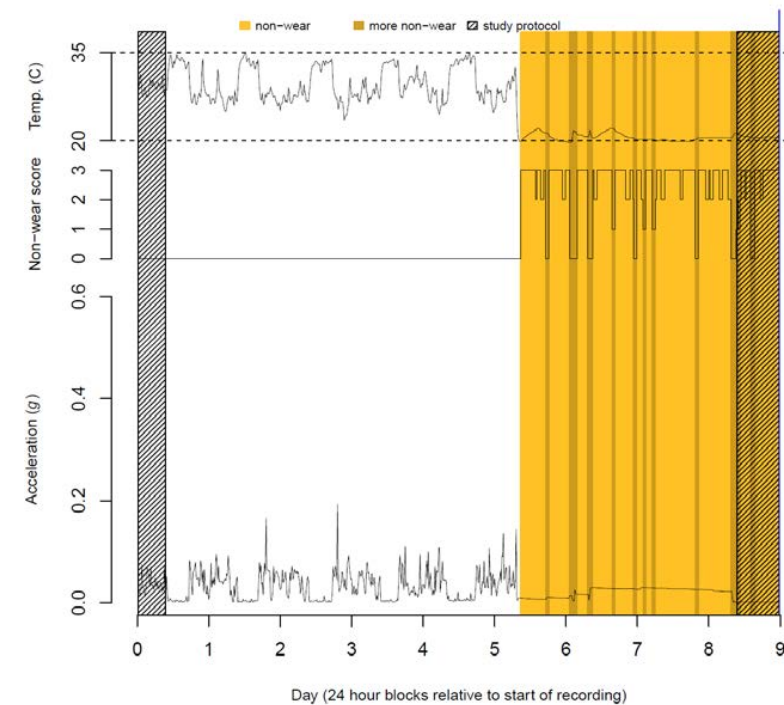
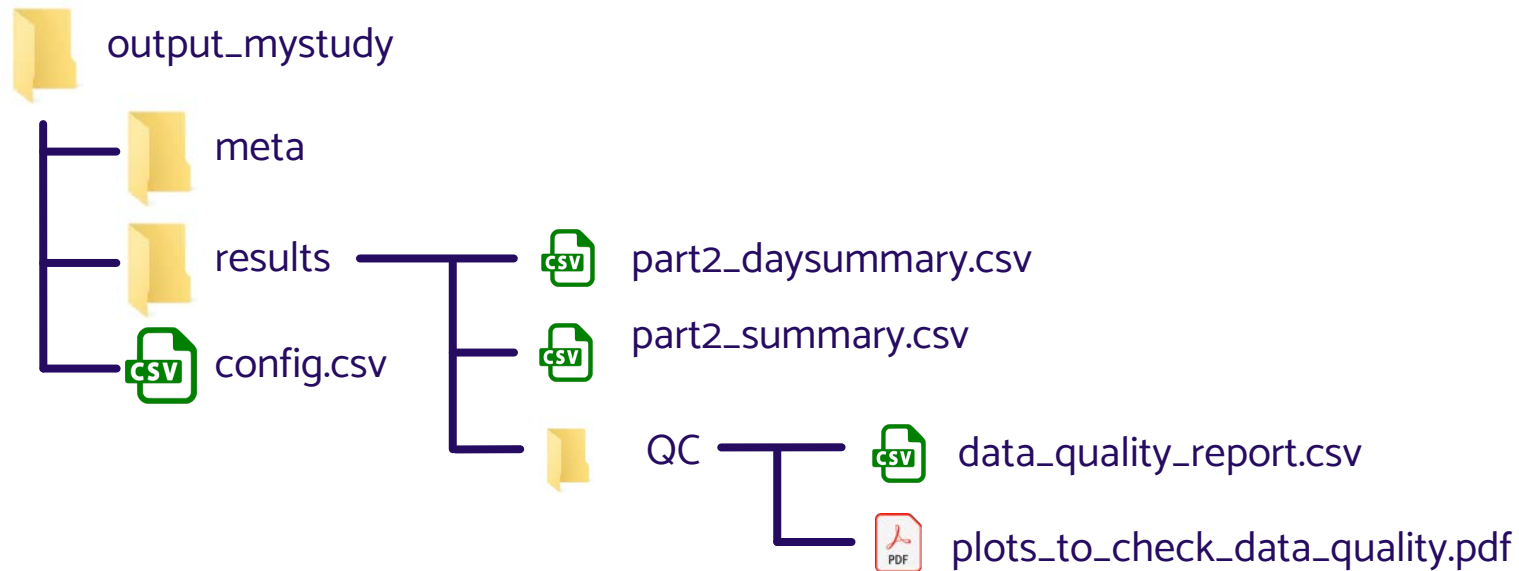
mvpadur = c(1, 5, 10),

boutcriter = 0.8,

qwindow = c(0,24),

[...])

GGIR output part 2



Assignment 1

Preparation:

- Get the example data from <https://www.accelting.com/ggir-training-materials/>
- Open a new R script in RStudio
- Copy this assignment to the R script and turn it into a comment by adding a # at the start of each line **Hint: A quicker way is to select the lines and press Ctrl+Shift+C**

Task:

- Process the data files with GGIR part 1 and 2:
 - Specify idloc = 2 to ensure the ID is correctly extracted
 - Treat the first day as not trustworthy
 - Extract time spent in intensity levels.
- Compare part2_summary.csv and part2_daysummary.csv, can you see the relationship and differences?
- Look at the part2_summary.csv column names, can you figure out what they mean?

Hint: The following page provides some guidance

<https://wadpac.github.io/GGIR/articles/GGIRoutput.html#ggir-part-2>

Day Evaluation

Thank you!

