



# Accelting

Advancing movement & sleep research

[www.accelting.com](http://www.accelting.com)

# GGIR training: Session 3

Dr. Vincent van Hees

[v.vanhees@accelting.com](mailto:v.vanhees@accelting.com)

# BEFORE WE START

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



# Day 2: questions?



# Learning goals for this session

- Learn about sleep detection with GGIR:
  - Alternative guiders
  - Circadian rhythm
- Navigate the Part 4 Output.



# Alternatives to vanHees2015



[www.accelting.com](http://www.accelting.com)



# Count-based algorithms

- Sadeh1994
- ColeKripke1992
- Galland2012



For reflection on:

- Count-based algorithms see: [https://wadpac.github.io/GGIR/articles/chapter8\\_SleepFundamentalsSibs.html#sib-count-based-algorithms-experimental](https://wadpac.github.io/GGIR/articles/chapter8_SleepFundamentalsSibs.html#sib-count-based-algorithms-experimental)
- Replicating the original zero-crossing count see: [https://wadpac.github.io/GGIR/articles/chapter4\\_AccMetrics.html#notes-on-implementation-of-zero-crossing-counts](https://wadpac.github.io/GGIR/articles/chapter4_AccMetrics.html#notes-on-implementation-of-zero-crossing-counts)



# Rest/Wake detection

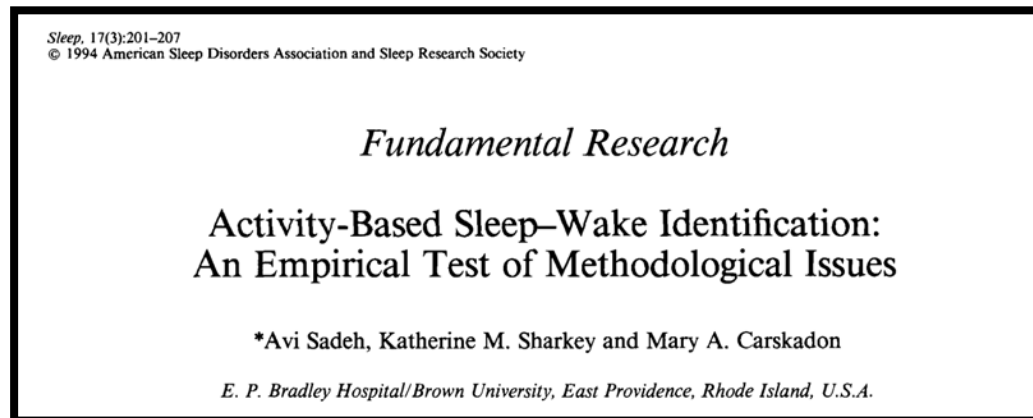
## Algorithms



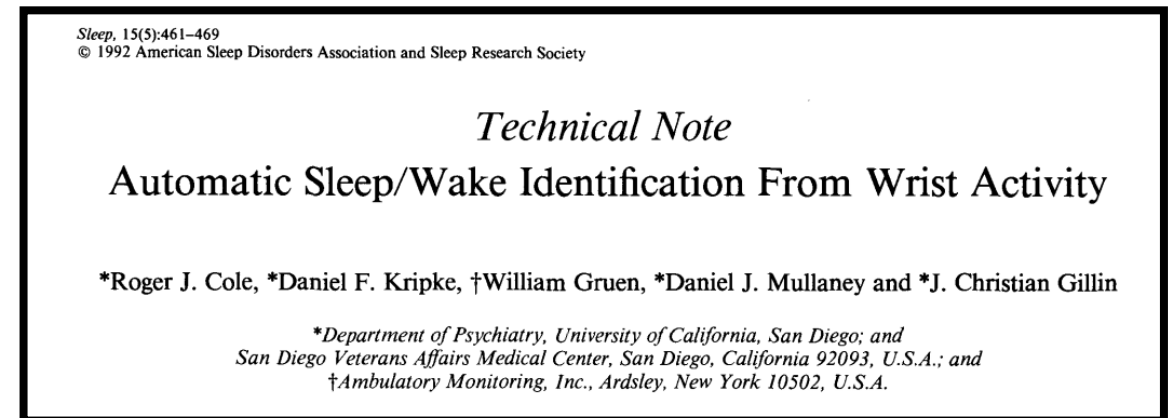
doi: 10.1371/journal.pone.0142533



doi: 10.1016/j.sleep.2012.01.018



doi: 10.1093/sleep/17.3.201



doi: 10.1093/sleep/15.5.461

# The GGIR()

*Sleep analysis*

GGIR(

[...]

*# Sleep analysis*

HASIB.algo = "vanHees2015",  
[...])

GGIR(

[...]

*# Sleep analysis*

HASIB.algo = "Sadeh1994",  
[...])

GGIR(

[...]

*# Sleep analysis*

HASIB.algo = "ColeKripke1992",  
[...])

GGIR(

[...]

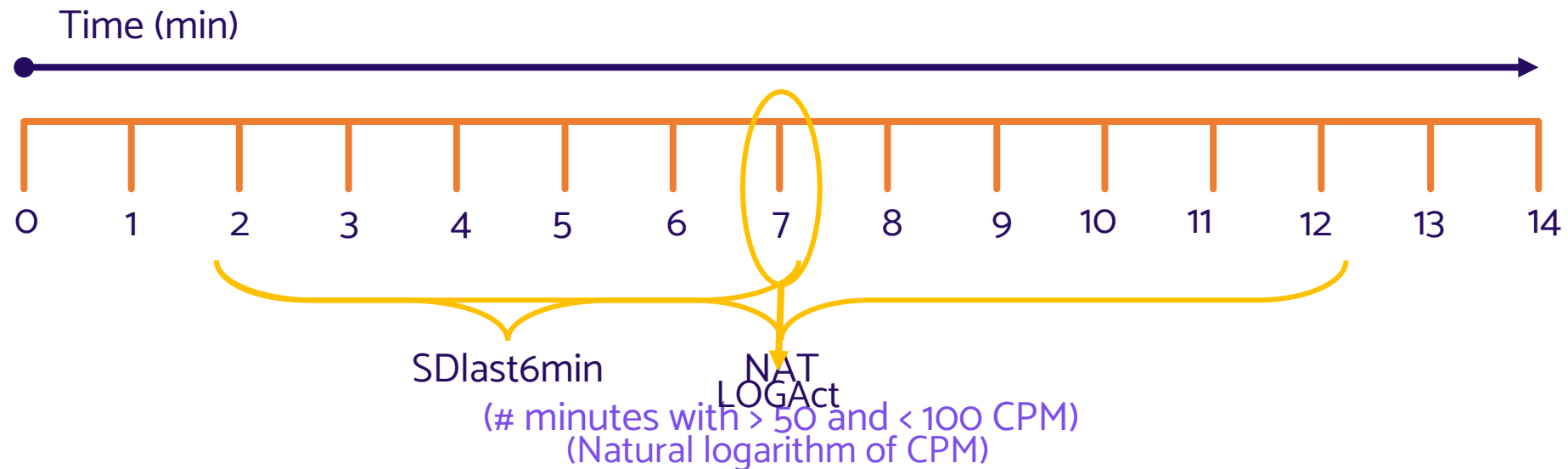
*# Sleep analysis*

HASIB.algo = "Galland2012",  
[...])

# Count-based algorithms

“Sadeh1994”

$$p(\text{Sleep}) = 7.601 - 0.065 \cdot \text{MeanCPM5 min} - 1.08 \cdot \text{NAT} - 0.056 \cdot \text{SDlast6min} - 0.703 \cdot \text{LOGAct} \\ \geq 0; \text{ sleep}$$



# The GGIR()

*Sleep analysis*

When using raw accelerometer data to imitate counts:

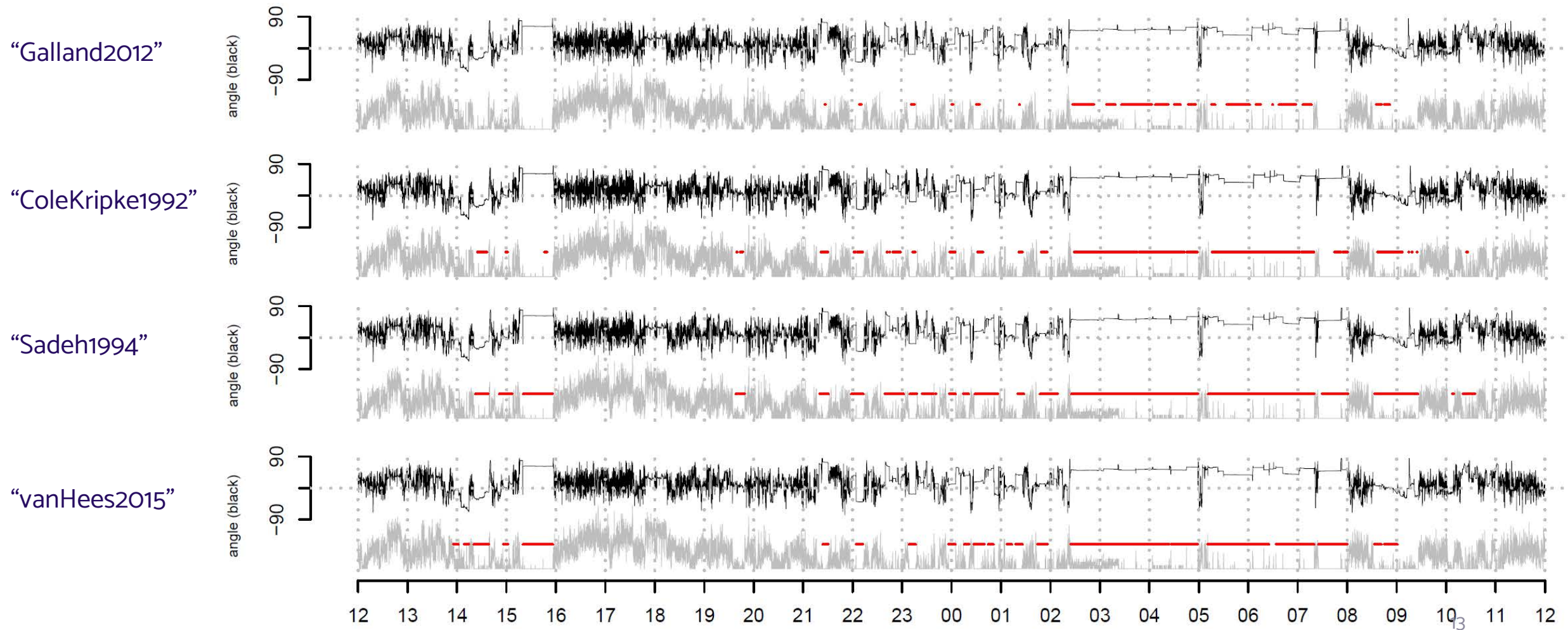
```
GGIR(  
  [...]  
  # Acceleration metrics  
  do.zcx = TRUE, do.zcy = TRUE, do.zcz = TRUE,  
OR  
  do.neishabouricounts= TRUE,  
  # Sleep analysis  
  HASIB.algo = "Sadeh1994",  
  Sadeh_axis = "Y",  
  [...])
```

When using device generated count data:

<https://wadpac.github.io/GGIR/articles/Cookbook.html>



# Rest/Wake detection



# Rest/Wake detection

## Summary of algorithms to detect SIBs in GGIR

Algorithm	Population	Device	Attachment site
vanHees2015	<b>Adults</b> n = 28 (11 female), 21-72 yr	GENEActiv	Wrist
Sadeh1994	<b>Adults</b> n = 20 (11 female), 21-25 yr <b>Children</b> n = 16 (11 female), 10-16 yr	AMI Motionlogger actigraph	Wrist
ColeKripke1992	<b>Adults</b> n = 41 (9 female), 50 ± 15 yr	AMI Motionlogger actigraph	Wrist
Galland2012	<b>Infants</b> n = 33 (9 female), 10-22 weeks	Actical	Shin

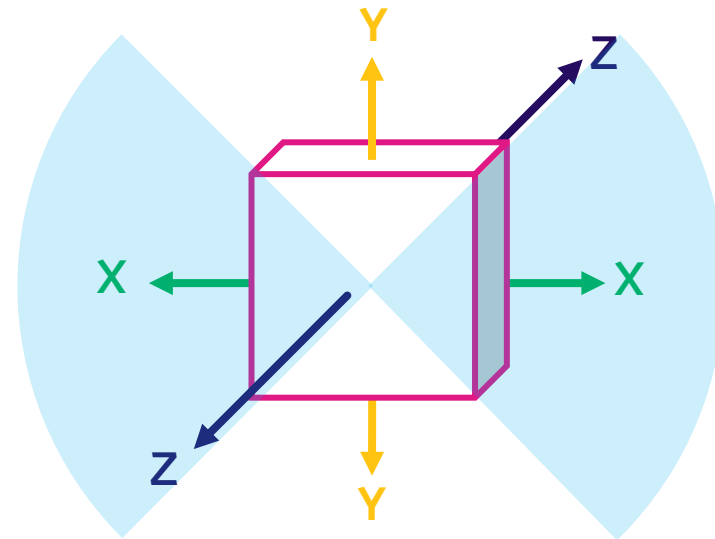
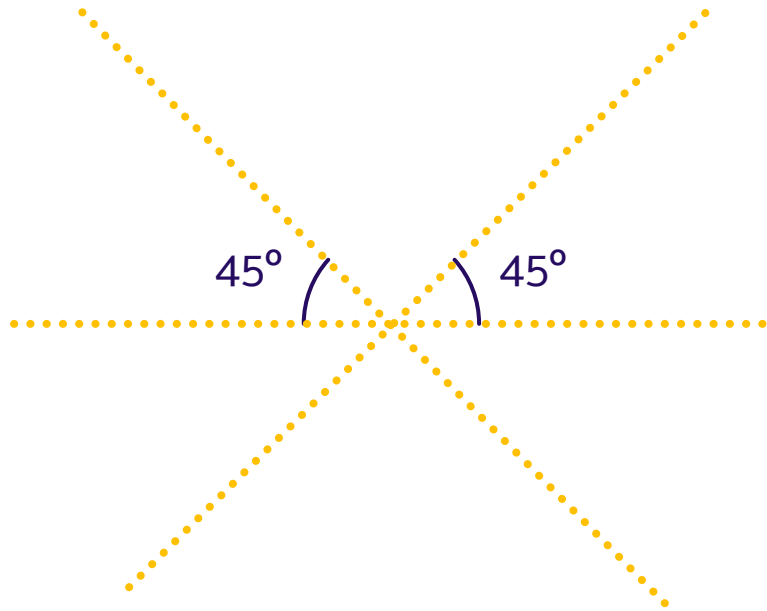


# Alternative guiders

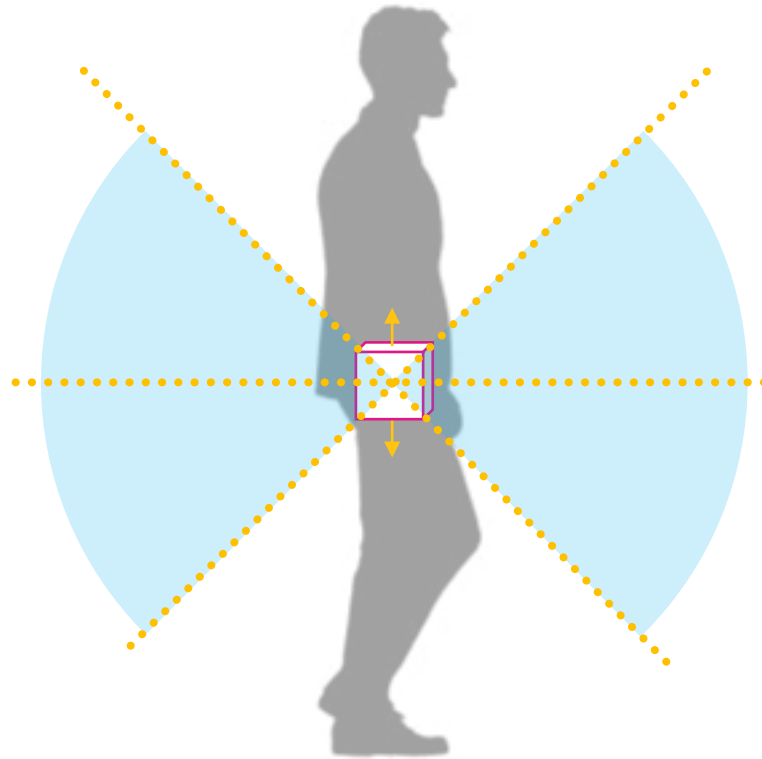




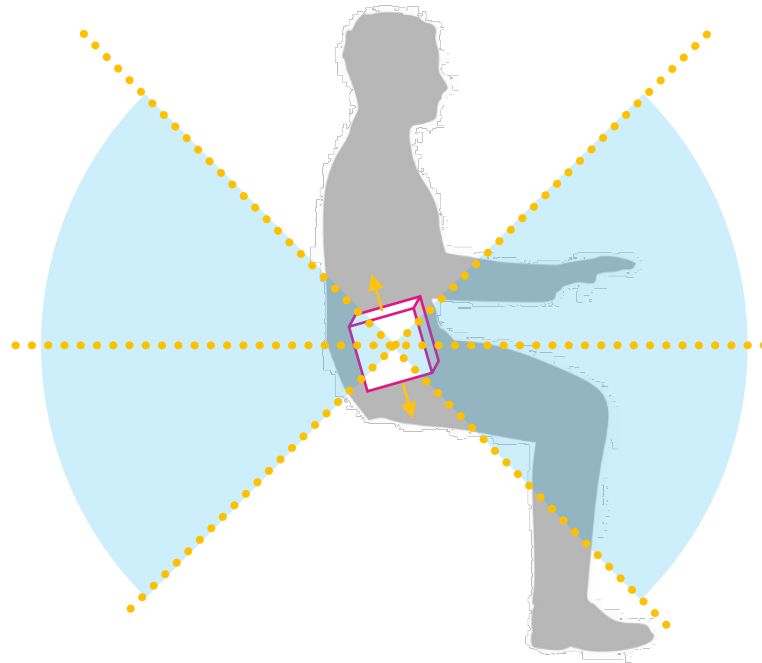
# HorAngle algorithm



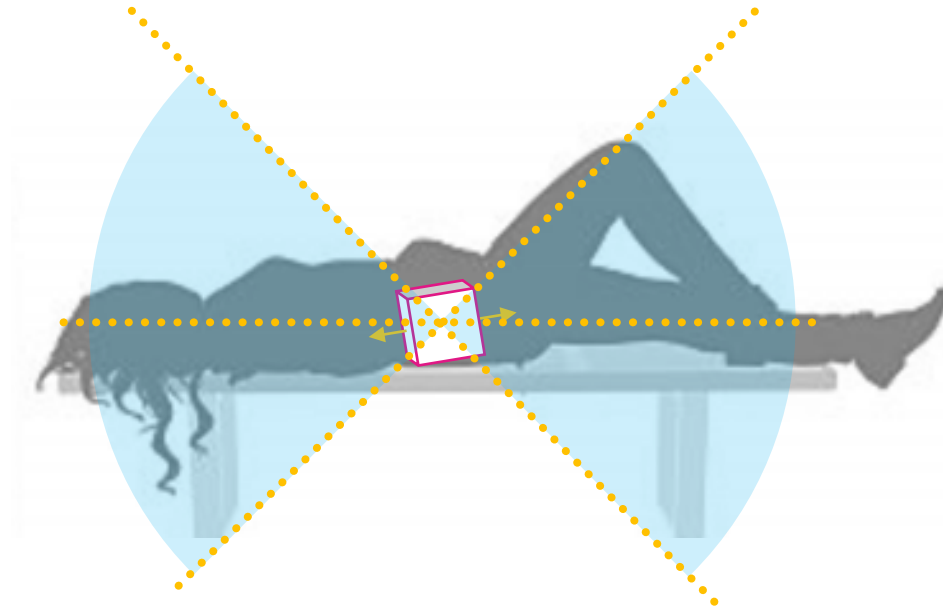
# HorAngle algorithm (experimental)



# HorAngle algorithm (experimental)



# HorAngle algorithm (experimental)

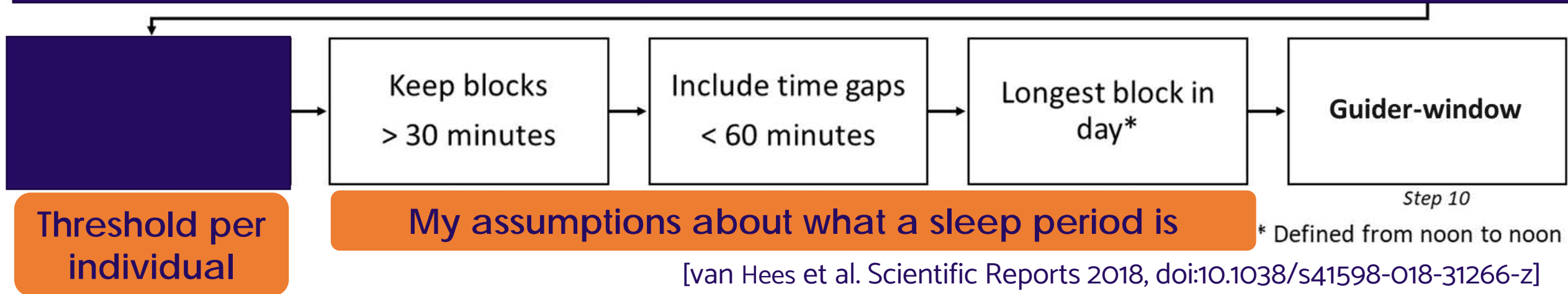




# HorAngle algorithm

Change in wrist angle over time invariant to sensor orientation

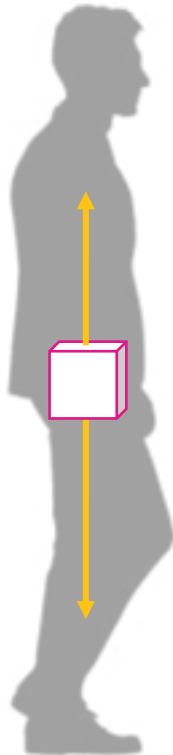
## DETECTION OF PERIODS OF THE DAY LYING DOWN



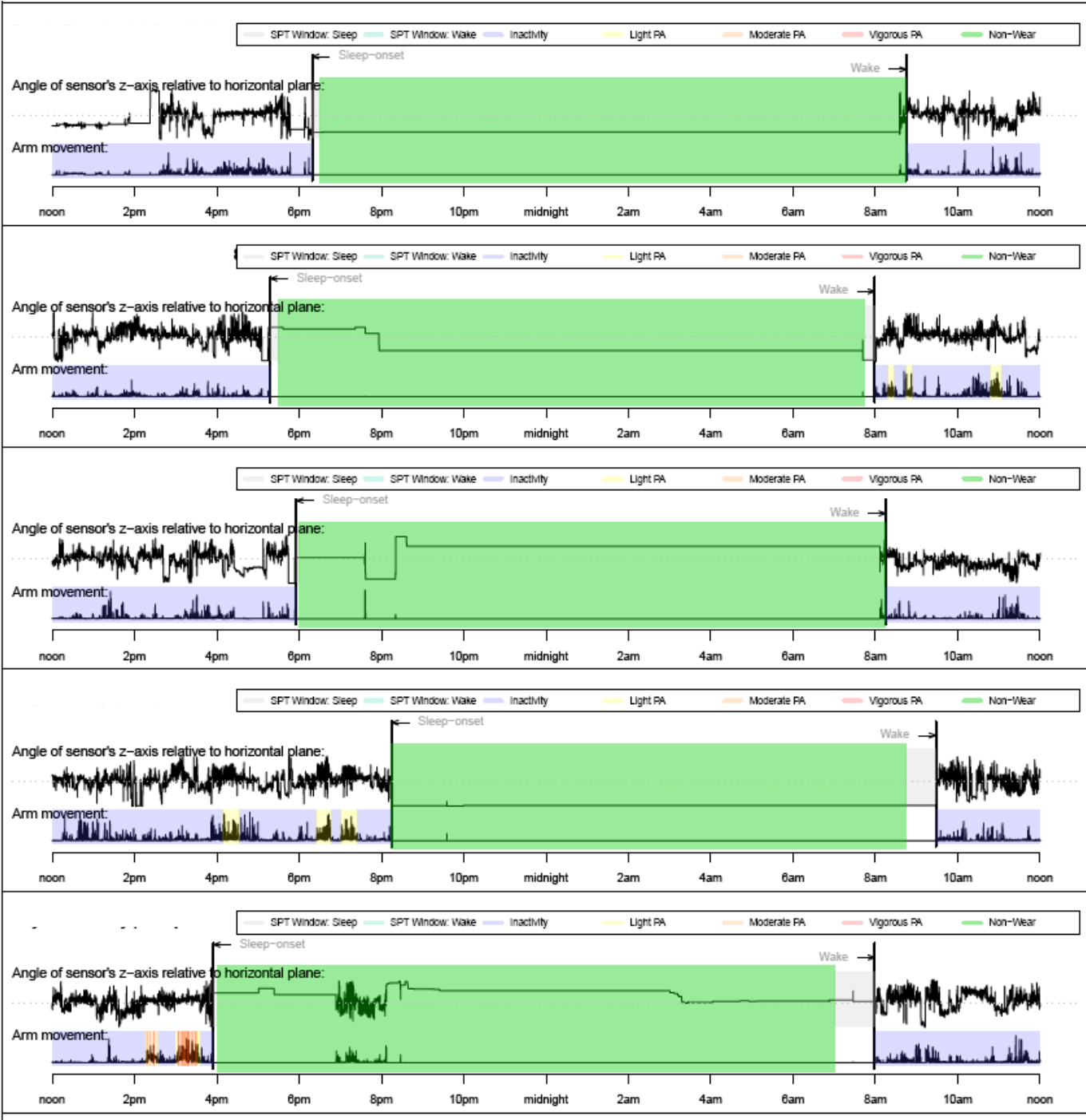
# The GGIR()

*Sleep analysis*

```
GGIR(  
  [...]  
  # Sleep analysis  
  HASPT.algo = "HorAngle",  
  HASPT.ignore.invalid = NA,  
  sensor.location = "hip",  
  longitudinal_axis = 2, # if not provided it will be estimated  
  [...])
```



# NotWorn



# The GGIR()

*Sleep analysis*

```
GGIR(  
  [...]  
  # Sleep analysis  
  HASPT.algo = "NotWorn",  
  do.imp = FALSE,  
  HASPT.ignore.invalid = NA,  
  ignorenonwear = FALSE,  
  includenightcrit = 8,  
  includedaycrit = 8,  
  [...])
```

```
GGIR(  
  [...]  
  HASPT.algo = c("NotWorn", "HDCZA"),  
  [...])
```

When worn for at  
least 75% of the  
time (in that night)

```
GGIR(  
  [...]  
  HASPT.algo = c("NotWorn", "HorAngle"),  
  [...])
```

# The GGIR()

*Sleep analysis guided by heuristic algorithm*

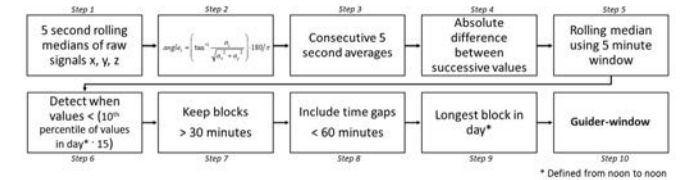
GGIR(

[...]

# Sleep analysis

HASPT.algo = "HDCZA",

[...])



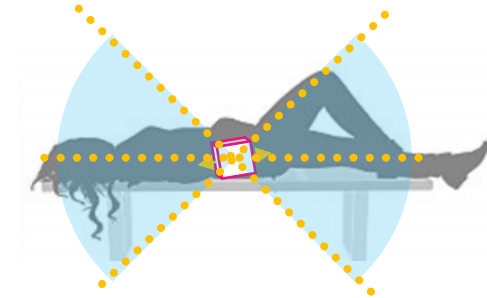
GGIR(

[...]

# Sleep analysis

HASPT.algo = "HorAngle",

[...])



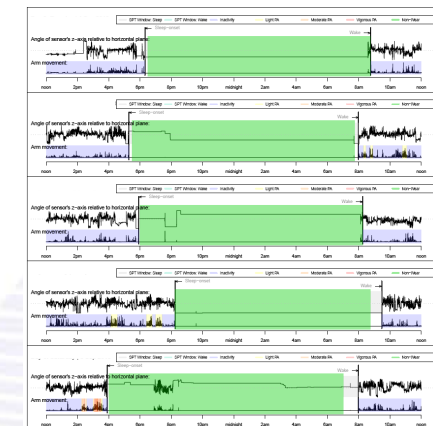
GGIR(

[...]

# Sleep analysis

HASPT.algo = "NotWorn",

[...])



# Guiders

Guider	Definition	Relevant arguments to use it
HDCZA		
HorAngle		
sleep log		
NotWorn		



# Guiders

## Summary of guiders

Guider	Definition	Relevant arguments to use it
HDCZA	HDCZA algorithm (van Hees 2018)	HASPT.algo = "HDCZA"
HorAngle	HorAngle algorithm intended to detect lying posture	HASPT.algo = "HorAngle" longitudinal_axis = 2 sensor.location = "hip"
sleep log	Reported sleep diaries (basic or advanced)	loglocation = "C:/mystudy/sleeplog.csv" sleepwindowType = "TimeInBed" or sleepwindowType = "SPT" colid = 1 If basic sleeplog: coln1 = 2
NotWorn	NotWorn algorithm aimed to use the longest non-wear period	HASPT.algo = "NotWorn" do.imp = FALSE, HASPT.ignore.invalid = NA, ignorenonwear = FALSE, includenightcrit = 8, includedaycrit = 8

# “Unusual” sleepers

- More than one sleep period time per day?
- Daysleeper?
  - If sleeplog wake-up > 12pm → re-do sleep analysis from 6pm-to-6pm
  - If other guider wake-up > 11am → re-do sleep analysis from 6pm-to-6pm
  - Classified as daysleeper in reports
  - Intended to adapt the algorithm to night workers



# Part 3 & 4 Output



# The GGIR pipeline



# Output from Part 4

## Night-level features (full report)

ID	Onset_n1	Wakeup_n1	Onset_n2	Wakeup_n2	Onset_n3	...
01	23:00:00	07:00:00			23:15:00	...

## part4\_nightsummary\_sleep\_full.csv

ID	night	cleaningcode	sleeplog_used	acc_available	guider
1	1	0	1	1	sleeplog
1	2	1	0	1	HDCZA
1	3	0	1	1	sleeplog
1	4	0	1	1	sleeplog
1	5	0	1	1	sleeplog
1	6	0	1	1	sleeplog
1	7	2	1	0	sleeplog

Cleaning code	Meaning
0	No problem
1	Sleep log not available (HDCZA used)
2	Not enough valid data
3	No accelerometer data available
4	No nights to be analyzed
5	Guider-defined SPT
6	SPT could not be defined either by sleeplog or algorithms



# Output from Part 4

## Night-level features (full report)

do.report = 4

part4\_nightsummary\_sleep\_full.csv

ID	night	sleeponset	wakeup	SptDuration	SleepDurationInSpt	WASO
1	1	27.201	32	4.799	3.696	1.103
1	2	26.11	31.936	5.826	4.888	0.939
1	3	26.5	32	5.5	4.997	0.503
1	4	25	32.475	7.475	7.115	0.36
1	5	26.413	32	5.588	5.071	0.517
1	6	27.131	32	4.869	3.526	1.343
1	7	25.074	31.761	6.688	6	0.688

# Output from Part 4

## Night-level features (clean report)

`do.report = 4`

GGIR(  
[...]  
# Data cleaning  
includenightcrit = 16,  
excludefirst.part4 = FALSE,  
excludelast.part4 = FALSE,  
[...])

### part4\_nightsummary\_sleep\_cleaned.csv

ID	night	sleeponset	wakeup	SptDuration	SleepDurationInSpt	WASO
1	1	27.201	32	4.799	3.696	1.103
1	2	26.11	31.936	5.826	4.888	0.939
1	3	26.5	32	5.5	4.997	0.503
1	4	25	32.475	7.475	7.115	0.36
1	5	26.413	32	5.588	5.071	0.517
1	6	27.131	32	4.869	3.526	1.343
1	7	25.074	31.761	6.688	6	0.688

# Output from Part 4

## Night-level features (clean report)

`do.report = 4`

GGIR(  
[...]  
# Data cleaning  
includenightcrit = 16,  
excludefirst.part4 = **TRUE**,  
excludelast.part4 = FALSE,  
[...])

### part4\_nightsummary\_sleep\_cleaned.csv

ID	night	sleeponset	wakeup	SptDuration	SleepDurationInSpt	WASO
1	1	27.201	32	4.799	3.696	1.103
1	3	26.5	32	5.5	4.997	0.503
1	4	25	32.475	7.475	7.115	0.36
1	5	26.413	32	5.588	5.071	0.517
1	6	27.131	32	4.869	3.526	1.343

# Output from Part 4

## Night-level features (clean report)

do.report = 4

GGIR(

[...]

# Data cleaning

includenightcrit = 16,

excludefirst.part4 = **TRUE**,

excludelast.part4 = FALSE,

[...])

## part4\_nightsummary\_sleep\_cleaned.csv

ID	night	sleeponset	wakeup	SptDuration	SleepDurationInSpt	WASO
1	1	27.201	32	4.799	3.696	1.103
1	3	26.5	32	5.5	4.997	0.503
1	4	25	32.475	7.475	7.115	0.36
1	5	26.413	32	5.588	5.071	0.517
1	6	27.131	32	4.869	3.526	1.343

# Output from Part 4

## Person-level features

do.report = 4

part4\_summary\_sleep\_cleaned.csv

ID	sleeplog_used	n_nights_acc	n_nights_sleeplog	n_WE_nights_complete	n_WD_nights_complete	n_WEnights_daysleeper	n_WDnights_daysleeper
11	1	6	6	2	4	0	0

All days

ID	SptDuration_ <b>AD</b> _T5A5_mn	SptDuration_ <b>AD</b> _T5A5_sd	SleepDurationInSpt_ <b>AD</b> _T5A5_mn	SleepDurationInSpt_ <b>AD</b> _T5A5_sd	WASO_ <b>AD</b> _T5A5_mn	WASO_ <b>AD</b> _T5A5_sd
11	5.372	1.142	4.685	1.375	0.687	0.383

Weekdays

ID	SptDuration_ <b>WD</b> _T5A5_mn	SptDuration_ <b>WD</b> _T5A5_sd	SleepDurationInSpt_ <b>WD</b> _T5A5_mn	SleepDurationInSpt_ <b>WD</b> _T5A5_sd	WASO_ <b>WD</b> _T5A5_mn	WASO_ <b>WD</b> _T5A5_sd
11	5.64	1.277	5.045	1.404	0.594	0.34

Weekend days

ID	SptDuration_ <b>WE</b> _T5A5_mn	SptDuration_ <b>WE</b> _T5A5_sd	SleepDurationInSpt_ <b>WE</b> _T5A5_mn	SleepDurationInSpt_ <b>WE</b> _T5A5_sd	WASO_ <b>WE</b> _T5A5_mn	WASO_ <b>WE</b> _T5A5_sd
11	4.838	0.877	3.966	1.409	0.872	0.532

# Output from Part 4

## Complete list of variables

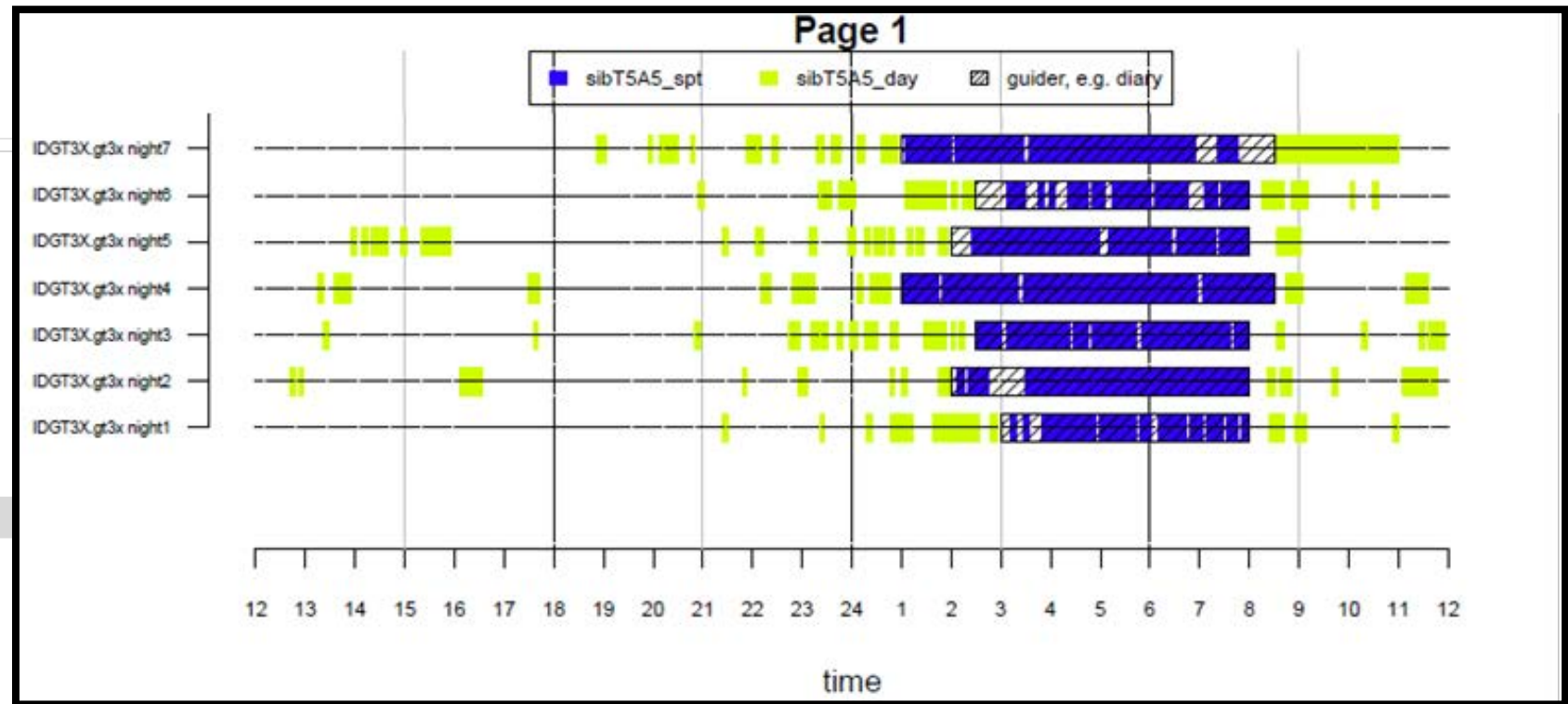
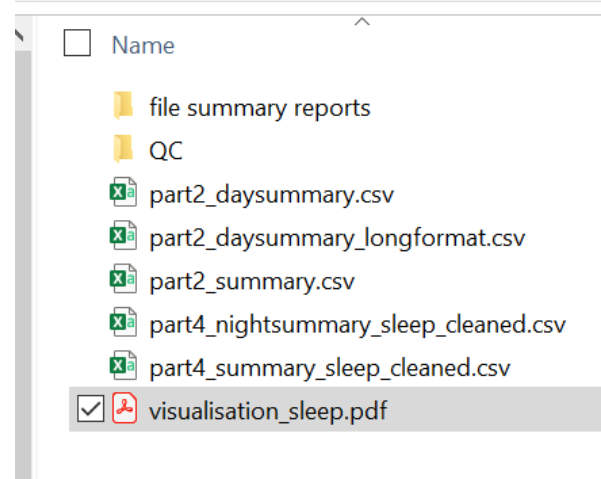
Sleeponset, sleepset_ts	duration_sib_wakinghours	calendar_date
Wakeup, wakeup_ts	number_sib_sleepperiod	filename
SptDuration	number_of_awakenings	cleaningcode
Sleepparam	number_sib_wakinghours	sleeplog_used
guider_inbedStart, guider_inbedStart_ts	duration_sib_wakinghours_atleast15min	acc_available
guider_inbedEnd, guider_inbedEnd_ts	sleeplatency	guider
guider_inbedDuration	sleepefficiency	SleepRegularityIndex
fraction_night_invalid	page	SriFractionValid
SleepDurationInSpt	daysleeper	longitudinal_axis
WASO	weekday	nonwear_perc_spt



# Sleep visualizations

do.visual = TRUE

est files > output > output\_GT3X > results





# Circadian rhythm



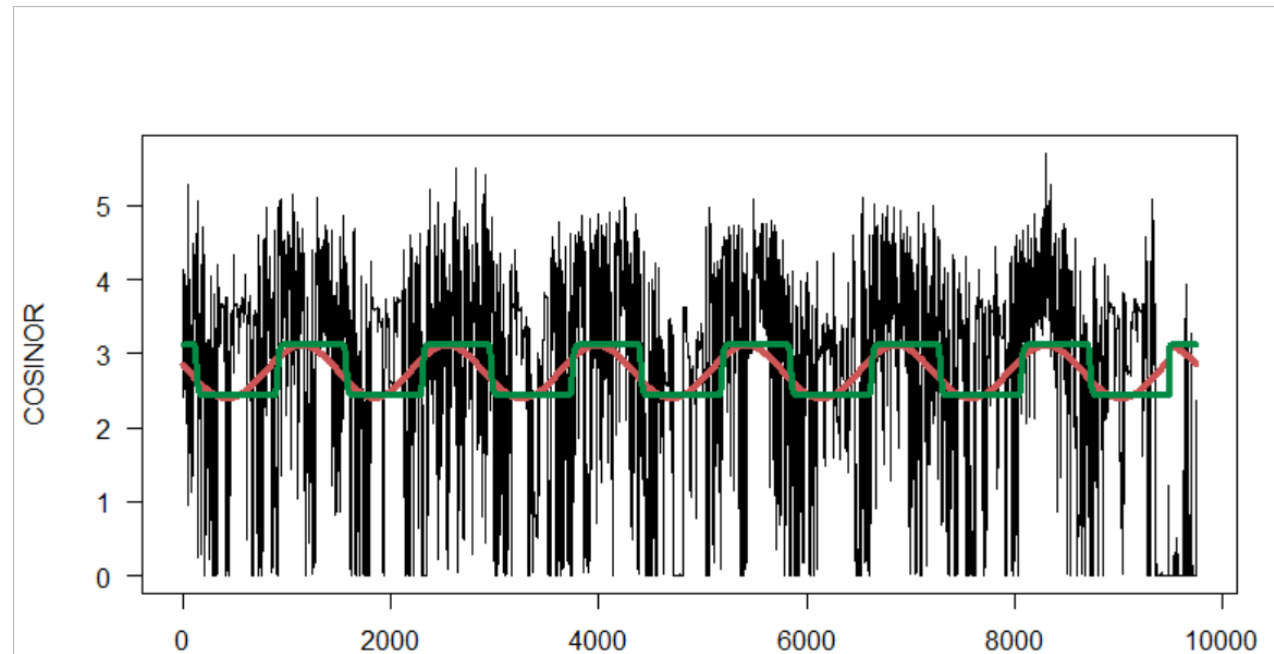
# Other variables

## Circadian rhythms

Part 2 & 6

- **MXLX** – (consecutive) Most and Least active X hours
  - Timing (start time)
  - Average acceleration
  - Intensity gradient
- Cosinor & Extended Cosinor (Marler et al. 2006)

```
GGIR(  
  [...]  
  winhr = c(5, 10),  
  cosinor = TRUE,  
  part6CR = TRUE, mode = 6,  
  part6Window = c("start", "end")  
  save_ms5rawlevels = TRUE,  
  save_ms5raw_format = "RData",  
  [...])
```



# Other variables

## Circadian rhythms

Part 2 & 6

- IV & IS – Intradaily Variability & Interdaily Stability (after van Witting 1990 and Someren 1996, both in Bio Psychiatry)

```
GGIR(  
  [...]  
  part6CR = TRUE, mode = 6,  
  part6Window = c("start", "end")  
  [...])
```

# Group assignment

## Task:

- Prepare a mini-presentation where you explain how you would process parts 1 to 4 for a real or imaginary study
- Groups of 4

## What do we expect:

- Research question of interest (ideally related to sleep)
  - Including population and variables of interest that you would require
- Details about the study protocol
  - Accelerometers configuration, body attachment site, recording duration, start and end times of the recording,...
- How you would approach the data processing to get your variables of interest from GGIR parts 3 and 4
  - GGIR parameters and how you would configure them



# Day Evaluation

**Thank you!**

