

SleepPy: A python package for sleep analysis from accelerometer data

Yiorgos Christakis¹, Nikhil Mahadevan¹, and Shyamal Patel¹

¹ Pfizer, Inc.

DOI: [10.21105/joss.01663](https://doi.org/10.21105/joss.01663)

Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Lindsey Heagy](#) ↗

Reviewers:

- [@eskofier](#)
- [@ikarosilva](#)
- [@nhejazi](#)

Submitted: 31 July 2019

Published: 18 December 2019

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC-BY](#)).

Introduction

Measures of sleep quality and quantity can provide valuable insights into the health and well-being of an individual. Traditionally, sleep assessments are performed in the clinic/hospital setting using polysomnography tests. Recent advances in wearable sensor technology have enabled objective assessment of sleep at home. Actigraphy has been widely used for this purpose and several algorithms have been published in the literature over the years. However, implementation of these algorithms is not widely available, which creates a barrier for wider adoption of wearable devices in clinical research.

SleepPy is an open source python package incorporating several published algorithms in a modular framework and providing a suite of measures for the assessment of sleep quantity and quality. The package can process multi-day streams of raw accelerometer data (X, Y & Z) from wrist-worn wearable devices to produce sleep reports and visualizations for each recording day (24-hour period). The reports are formatted to facilitate statistical analysis of sleep measures. Visualization acts as a quick debugging tool, provides insights into sleep patterns of individual subjects and can be used for presentation of data to diverse audiences.

Processing Pipeline

SleepPy has been developed with the goal of making it easy for researchers to derive measures of sleep from accelerometer data collected during daily life. While it currently provides out-of-the-box support for data files generated by the GeneActiv wrist-worn devices, it can be easily extended to support data files generated by other wrist-worn wearable devices. The package can work with both GeneActiv .bin files, as well as raw .csv outputs generated by GeneActiv's software. Each .bin file needs to be manually processed with the GeneActiv software to generate a .csv file. So while it takes longer for SleepPy to process .bin files, it is highly advisable to do so for large datasets.

SleepPy follows seven steps when processing data:

1. Split data by day: Load raw accelerometer data from input file and split it into 24-hour segments (noon to noon).
2. Derive activity index (Bai et al., 2016): Calculate activity index for each 1 minute epoch of the day.
3. Perform off-body detection (Hees et al., 2014),(Van Hees et al., 2015),(Hees et al., 2019): Run off-body detection algorithm and generate a label (on or off body) for each 15 minute epoch of the day.

4. Identify major rest period (Hees et al., 2018): Estimate the major rest period (i.e. sleep window) for each day.
5. Perform sleep/wake classification (Cole, Kripke, Gruen, Mullaney, & Gillin, 1992): Run sleep/wake classification algorithm to generate sleep/wake labels for each 1 minute epoch of the day.
6. Calculate sleep measures: Calculate sleep measures based on sleep/wake states only during the major rest period for each day.
7. Generate reports and visualizations: Create a set of tables and charts for analysis and presentation of processed outputs and sleep measures for each day.

Outputs

The user has access to all intermediate data, a table of sleep measures, and two kinds of visual reports.

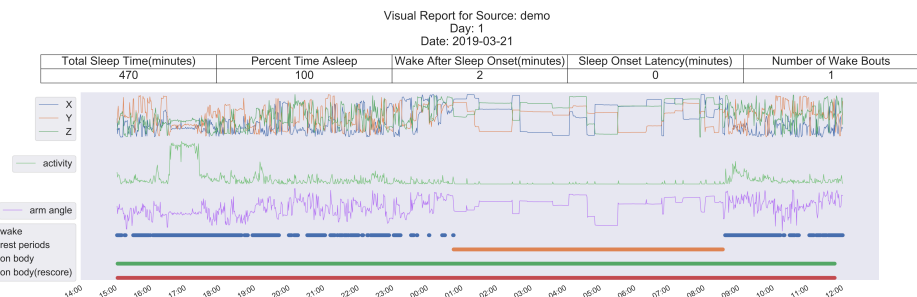


Figure 1: Visual report showing various raw and processed data streams

Data Streams Report

Raw and processed data streams generated at each stage of the processing pipeline are visualized in this report for each day (24 hour period, noon to noon). As shown in Figure 1, the report includes a source file name, number of the day in the sequence of days provided, start date of the data being shown, and a table of all calculated sleep measures for that day. Below the table are plots of raw and processed data streams.

The plots are laid out as follows:

1. XYZ: The first chart is a plot of the raw tri-axial accelerometer signals (X, Y, and Z).
2. Activity index: The second chart is a plot of minute-by-minute activity index values, which reflect the intensity of activity for each minute.
3. Arm-angle: The third chart is a plot of the arm-angle over 24 hours, this data stream is used to determine the major rest period.
4. Wake: The fourth chart represents sleep/wake classification for the entire day. However, sleep measures are computed during the major rest period only.
5. Rest periods: The fifth chart is a plot of all rest periods detected by the algorithm. Only the longest rest period (i.e. major rest period) is used for calculating sleep measures.

6. On-body: The sixth chart is a plot of all periods identified by the on-body detection algorithm (without filtering or re-scoring).
7. On-body (re-scored): The seventh chart is a plot of the on-body periods after re-scoring has been applied.

Sleep Measures Report

Sleep measures derived from the sleep/wake classifications for each day are provided in a table as well as a visual report (Figure 2).



Figure 2: Visual report showing a plot of sleep measures across all days.

Sleep measures table (.csv file) contains the following sleep measures for each night:

1. Total sleep time (minutes): Total time spent asleep during the major rest period.
2. Percent time asleep (%): Percentage of the major rest period spent in the sleep state.
3. Wake after sleep onset (WASO, minutes): The amount of time spent awake after the first sleep state.

4. Sleep onset latency (SOL, minutes): The time it took for the subject to fall asleep.
5. Number of wake bouts: The number of times the subject transitioned from sleep to wake, after the first sleep state.

Availability

The software is available as a pip installable package, as well as on GitHub at: <https://github.com/elyiorgos/sleepy>

Acknowledgements

The Digital Medicine & Translational Imaging group at Pfizer, Inc supported the development of this package.

License

This project is licensed under the MIT License - see the [LICENSE](#) file for details

References

- Bai, J., Di, C., Xiao, L., Evenson, K. R., LaCroix, A. Z., Crainiceanu, C. M., & Buchner, D. M. (2016). An activity index for raw accelerometry data and its comparison with other activity metrics. *PLoS ONE*, *11*(8). doi:[10.1371/journal.pone.0160644](https://doi.org/10.1371/journal.pone.0160644)
- Cole, R. J., Kripke, D. F., Gruen, T., Mullaney, D. J., & Gillin, J. C. (1992). *Automatic Sleep/Wake Identification From Wrist Activity* (No. 5) (Vol. 15, pp. 461–469). doi:[10.1093/sleep/15.5.461](https://doi.org/10.1093/sleep/15.5.461)
- Hees, V. T. V., Fang, Z., Zhao, J., Heywood, J., Mirkes, E., Sabia, S., & Migueles, J. (2019). GGIR: Raw Accelerometer Data Analysis. doi:[10.5281/zenodo.1051064](https://doi.org/10.5281/zenodo.1051064)
- Hees, V. T. van, Fang, Z., Langford, J., Assah, F., Mohammad, A., Silva, I. C. M. da, Trenell, M. I., et al. (2014). Autocalibration of accelerometer data for free-living physical activity assessment using local gravity and temperature: an evaluation on four continents. *Journal of Applied Physiology*, *117*(7), 738–744. doi:[10.1152/jappphysiol.00421.2014](https://doi.org/10.1152/jappphysiol.00421.2014)
- Hees, V. T. van, Sabia, S., Jones, S. E., Wood, A. R., Anderson, K. N., Kivimäki, M., Frayling, T. M., et al. (2018). Estimating sleep parameters using an accelerometer without sleep diary. *Scientific Reports*, *8*(1). doi:[10.1038/s41598-018-31266-z](https://doi.org/10.1038/s41598-018-31266-z)
- Van Hees, V. T., Sabia, S., Anderson, K. N., Denton, S. J., Oliver, J., Catt, M., Abell, J. G., et al. (2015). A novel, open access method to assess sleep duration using a wrist-worn accelerometer. *PLoS ONE*, *10*(11). doi:[10.1371/journal.pone.0142533](https://doi.org/10.1371/journal.pone.0142533)