The Journal of Open Source Software

wristpy: Fast, User-Friendly Python Processing of Wrist-worn Accelerometer Data

Adam Santorelli [1], **Freymon Perez**[1], **Reinder Vos de Wael**[1], **Florian Rupprecht**[1], **John Vito d'Antonio-Bertagnolli**[1], **Alexandre Franco**[1,2,3], and **Gregory Kiar** [1]

1 Child Mind Institute, New York, USA **2** Center for Biomedical Imaging and Neuromodulation, Nathan Kline Institute, Orangeburg, NY, USA **3** Department of Psychiatry, NYU Grossman School of Medicine, New York, NY, USA

DOI: 10.21105/joss.08637

Software

- Review
- Repository
- Archive

Editor: Stefan Appelhoff

Reviewers:

- @LukasAdamowicz
- @djmitche

Submitted: 05 May 2025
Published: 22 October 2025

Summary

`wristpy` is an open-source Python package designed to streamline the processing and analysis of wrist-worn accelerometer data. The package has been developed in a modular framework to process actigraphy data from both GENEActiv and Actigraph watches, while supporting the extension for other wearable standards in the future, and is supported across multiple platforms (Windows, macOS, and Ubuntu). This software can be accessed and used through either a command-line interface (CLI) or as an importable Python module. The main processing pipeline will generate outputs in either a `.csv` or `.parquet` format. The output contains timeseries data at a user-chosen temporal resolution, including: physical activity metrics, sleep status (including the sustained inactivity bouts (SIB) and the sleep period time (SPT) windows), non-wear period identification, and physical activity classification.

`wristpy` has been designed modularly, which allows for incremental improvements of bottlenecks in the processing pipeline — such as the independent creation and adoption of a Rust-based reader for actigraphy data that dramatically improves scalability over the vendor-provided reader. With `wristpy`, researchers can directly access a variety of functions that allow them to read specific sensor data stored on the watch, calculate a variety of physical activity metrics, compare various non-wear or sleep detection algorithms, or take advantage of them together in a 1-click end-to-end preprocessing pipeline.

Statement of Need

Wearable accelerometers are increasingly used to measure both physical activity and sleep patterns as they provide a simple, effective, non-invasive, and low-cost alternative to clinical observation, allowing for large scale data to be acquired in realistic scenarios. While several commercial solutions exist, such as consumer-grade products providing aggregate data (e.g., FitBit) or research-focused proprietary software (e.g., such as those from Actigraph and ActivInsights), a need exists for device-agnostic open-source tools that can process raw accelerometer data with transparency, reproducibility, and scalability, both as a base for evaluation and methodological advancement.

Several open-source packages have been developed, namely GGIR (Hees & Migueles, 2025), Scikit Digital Health (`skdh`) (Adamowicz et al., 2022), and pyActigraphy (Hammad et al., 2021). GGIR is one of the most commonly used software packages to process and analyze actigraphy data from various watches and has been cited in hundreds of publications over the past decade. `skdh` is a Python package that can process data from GENEActiv watches (with extensibility to support data from other data streams after conversion to `.csv`) to compute

physical activity metrics, sleep detection based on (Christakis et al., 2019), novel wear detection (Vert et al., 2022), gait detection, and numerous other functions for both processing and analyzing accelerometer data. Fruthermore, this package now supports combining multiple files into one data-stream for processing.

Currently, none of these packages allow for the computation of multiple sleep windows within a day and, aside from `skdh`, these libraries do not extensibly support data from other sensors, including those on the devices they currently support (such as skin conductance and light). Furthermore, the implementation and utilization of these toolkits and pipelines may require some training on the end user side. Crucially, we have developed `wristpy` with the end user's experience in mind, creating a true `point and click solution`. It is a simple, inuitive, and light-weight toolbox. Additionally by expanding to a docker image, it is easy to run the same one-click implementation on servers. `wristpy` achieves these goals by providing a development environment and utilities that allow it to: - Process raw accelerometer data from various watch manufacturers, including both GENEActiv and Actigraph watches (with plans to support more research grade devices in future development). - Access to all sensors and metadata from these watches. - Support modular processing algorithms that can be easily extended while enforcing strict code quality guidelines, exhaustive test coverage, and documentation. Additionally, this documentation extends to a thorough and robust `logger` allowing users to easily track down and identify the source of any `WARNING` or `ERROR` messages. - Run in a 1-click fashion with an easy to use interface that requires no domain specific knowledge. - Allows batch processing for entire directories of data (e.g. a directory containing all data from a specific protocol can be processed in a 1-click fashion). - Provide a suite of functionality for more advanced users to request specific outputs (choice of physical activity metrics, user-defined temporal resolution, activity thresholds, different non-wear algorithms). - Execute a streamlined implementation that can lead to computational savings on the order of magnitudes.

Critically, `wristpy` leverages the history and experience of each of these tools, and supports the algorithms they have pioneered. While the need for `wristpy` was apparent due to the noted limitations, it cannot be overstated how influential these prior toolboxes have been in its construction.

Processing Pipeline

`wristpy` provides the following key functionalities within the main processing pipeline:

- Data loading using `actfast` (Rupprecht, 2025) a Rust-based reader that provides up to nanosecond temporal resolution with direct access to all sensor information on the watch and key metadata information.
- Post-manufacturer calibration to remove any bias in the device. The two primary methods are a direct minimization method and the default method in GGIR (Vincent T. Van Hees et al., 2014).
- Calculation of essential physical activity metrics such as ENMO (Euclidean norm minus one), angle-Z (orientation of the watch relative to the x-y axis), MAD (mean amplitude deviation) (Aittasalo et al., 2015), MIMS (monitor independent motor summary unit) (John et al., 2019), and Actigraph activity counts (agcounts) (Neishabouri et al., 2022). Users can request any number of metrics as part of the output.
- Implementation of validated algorithms for on-body wear detection namely from GGIR (Hees & Migueles, 2025), the combined temperature and acceleration algorithm from (Zhou et al., 2015), and the DETACH algorithm from (Vert et al., 2022). Most recently, we have added the option for users to test out the ensemble classification of multiple non-wear detection algorithms (any number from the above list can be chosen).
- Sleep period detection: following the default parameters from GGIR, we use the HDCZ (Vincent Theodoor Van Hees et al., 2018) and HSPT algorithms (Vincent T. Van Hees et al., 2015) to find sleep period candidates. Unlike the implementation within GGIR, we output all viable sleep window candidates across the entire available data. Furthermore,

any overlap between sleep periods and non-wear times are removed, eliminating the need for the researcher to verify or remove those overlaps manually.
- Physical activity level classification: Categorize periods of physical activity as `inactive`, `light`, `moderate`, or `vigorous` activity, based on established thresholds for ENMO (Hildebrand et al., 2014), MAD (Aittasalo et al., 2015), agcounts (Neishabouri et al., 2022) or custom user-defined thresholds. If multiple physical activity metrics have been requested, categorization for each metric will be provided.

Acknowledgements

Financial support has been provided by Dr Michael P. Milham. Financial support has been provided by the California Department of Health Care Services (DHCS) as part of the Children and Youth Behavioral Health Initiative (CYBHI). We would also like to thank Dr. Michelle Freund, Dr. Vadim Zipunnikov, Dr. Andrew Leroux, and Dr. Kathleen R. Merikangas and her team at the NIMH, for their technical and administrative support.

References

Adamowicz, L., Christakis, Y., Czech, M. D., & Adamusiak, T. (2022). SciKit Digital Health: Python Package for Streamlined Wearable Inertial Sensor Data Processing. *JMIR mHealth and uHealth*, *10*(4), e36762. https://doi.org/10.2196/36762

Aittasalo, M., Vähä-Ypyä, H., Vasankari, T., Husu, P., Jussila, A.-M., & Sievänen, H. (2015). Mean amplitude deviation calculated from raw acceleration data: A novel method for classifying the intensity of adolescents' physical activity irrespective of accelerometer brand. *BMC Sports Science, Medicine and Rehabilitation*, *7*(1), 18. https://doi.org/10.1186/s13102-015-0010-0

Christakis, Y., Mahadevan, N., & Patel, S. (2019). SleepPy: A python package for sleep analysis from accelerometer data. *Journal of Open Source Software*, *4*(44), 1663. https://doi.org/10.21105/joss.01663

Hammad, G., Reyt, M., Beliy, N., Baillet, M., Deantoni, M., Lesoinne, A., Muto, V., & Schmidt, C. (2021). pyActigraphy: Open-source python package for actigraphy data visualization and analysis. *PLOS Computational Biology*, *17*(10), e1009514. https://doi.org/10.1371/journal.pcbi.1009514

Hees, V. van, & Migueles, J. H. (2025). *GGIR*. Zenodo. https://doi.org/10.5281/ZENODO.1051064

Hildebrand, M., Van Hees, V. T., Hansen, B. H., & Ekelund, U. (2014). Age Group Comparability of Raw Accelerometer Output from Wrist- and Hip-Worn Monitors. *Medicine & Science in Sports & Exercise*, *46*(9), 1816–1824. https://doi.org/10.1249/MSS.0000000000000289

John, D., Tang, Q., Albinali, F., & Intille, S. (2019). An Open-Source Monitor-Independent Movement Summary for Accelerometer Data Processing. *Journal for the Measurement of Physical Behaviour*, *2*(4), 268–281. https://doi.org/10.1123/jmpb.2018-0068

Neishabouri, A., Nguyen, J., Samuelsson, J., Guthrie, T., Biggs, M., Wyatt, J., Cross, D., Karas, M., Migueles, J. H., Khan, S., & Guo, C. C. (2022). Quantification of acceleration as activity counts in ActiGraph wearable. *Scientific Reports*, *12*(1), 11958. https://doi.org/10.1038/s41598-022-16003-x

Rupprecht, F. (2025). *Childmindresearch/actfast: v1.1.2*. Zenodo. https://doi.org/10.5281/ZENODO.14721947

Van Hees, Vincent T., Fang, Z., Langford, J., Assah, F., Mohammad, A., Da Silva, I. C. M., Trenell, M. I., White, T., Wareham, N. J., & Brage, S. (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. https://doi.org/10.1152/japplphysiol.00421.2014

Van Hees, Vincent T., Sabia, S., Anderson, K. N., Denton, S. J., Oliver, J., Catt, M., Abell, J. G., Kivimäki, M., Trenell, M. I., & Singh-Manoux, A. (2015). A Novel, Open Access Method to Assess Sleep Duration Using a Wrist-Worn Accelerometer. *PLOS ONE*, *10*(11), e0142533. https://doi.org/10.1371/journal.pone.0142533

Van Hees, Vincent Theodoor, Sabia, S., Jones, S. E., Wood, A. R., Anderson, K. N., Kivimäki, M., Frayling, T. M., Pack, A. I., Bucan, M., Trenell, M. I., Mazzotti, D. R., Gehrman, P. R., Singh-Manoux, B. A., & Weedon, M. N. (2018). Estimating sleep parameters using an accelerometer without sleep diary. *Scientific Reports*, *8*(1), 12975. https://doi.org/10.1038/s41598-018-31266-z

Vert, A., Weber, K. S., Thai, V., Turner, E., Beyer, K. B., Cornish, B. F., Godkin, F. E., Wong, C., McIlroy, W. E., & Van Ooteghem, K. (2022). Detecting accelerometer non-wear periods using change in acceleration combined with rate-of-change in temperature. *BMC Medical Research Methodology*, *22*(1), 147. https://doi.org/10.1186/s12874-022-01633-6

Zhou, S.-M., Hill, R. A., Morgan, K., Stratton, G., Gravenor, M. B., Bijlsma, G., & Brophy, S. (2015). Classification of accelerometer wear and non-wear events in seconds for monitoring free-living physical activity. *BMJ Open*, *5*(5), e007447. https://doi.org/10.1136/bmjopen-2014-007447