Systole: A python package for cardiac signal synchrony and analysis

Nicolas Legrand[1] **and Micah Allen**[1, 2, 3]

1 Center of Functionally Integrative Neuroscience, Aarhus University Hospital, Denmark **2** Aarhus Institute of Advanced Studies, Aarhus University, Denmark **3** Cambridge Psychiatry, University of Cambridge, United Kingdom

DOI: 10.21105/joss.03832

Software

- Review
- Repository
- Archive

Editor: Øystein Sørensen

Reviewers:

- @axel-loewe
- @janfreyberg

Submitted: 04 October 2021
Published: 05 January 2022

Summary

Systole is a package for cardiac signal analysis in Python. It provides an interface for recording cardiac signals via electrocardiography (ECG) or photoplethysmography (PPG), as well as both online and offline data analysis methods extracting cardiac features, synchronizing experimental stimuli with different phases of the heart, removing artefacts at different levels and generating plots for data quality check and publication. Systole is built on the top of Numpy (Harris et al., 2020), Pandas [team (2020); mckinney-proc-scipy-2010] and Scipy (Virtanen et al., 2020), and can use both Matplotlib (Hunter, 2007) and Bokeh (Bokeh Development Team, 2018) to generate visualisations. It is designed to build modular pipelines that can interface easily with other signal processing or heart rate variability toolboxes, with a focus on data quality checks. Several parts of the toolbox utilize Numba (Lam et al., 2015) to offer more competitive performances with classic processing algorithms.

Statement of need

Analysis of cardiac data remains a major component of medical, physiological, neuroscientific, and psychological research. In psychology, for example, rising interest in interoception (i.e., sensing of cardiac signals) has led to a proliferation of studies synchronizing the onset or timing of experimental stimuli to different phases of the heart. Similarly, there is rising interest in understanding how heart-rate variability relates to mental illness, cognitive function, and physical wellbeing. This diverse interest calls for more open-source tools designed to work with cardiac data in the context of psychology research, but to date, only limited options exist. To improve the reproducibility and accessibility of advanced forms of cardiac physiological analysis, such as online peak detection and artefact control, we here introduce a fully documented Python package, Systole. Systole introduces core functionalities to interface with pulse oximeters devices, accelerated peaks detection and artefacts rejection algorithms as well as a complete suite of interactive and non-interactive plotting functions to improve quality checks and reports creation in the context of physiological signal analysis.

Overview

The package focuses on 5 core functional elements. The documentation of the package includes extensive tutorial notebooks and examples vignettes illustrating these points. It can be found at https://systole-docs.github.io/. The package has already been used in two publications that also describe some example uses (Legrand et al., 2021, 2020).

Core functionalities:

1. Signal extraction and interactive plotting. Systole uses adapted versions of ECG (Luis & Porr, 2019) and PPG (Gent et al., 2019) peaks detectors accelerated with Numba (Lam et al., 2015) for increased performances. It also implements plotting functionalities for RR time series and heart rate variability visualization both on Matplotlib (Hunter, 2007) and Bokeh (Bokeh Development Team, 2018). This API is designed and developed to facilitate the automated generation of interactive reports and dashboards from large datasets of physiological data.

2. Artefact detection and rejection. The current artefact detection relies on the Python adaptation of the method proposed by Lipponen & Tarvainen (2019). The correcting of the artefacts is modular and can be adapted to specific uses. Options are provided to control for signal length and events synchronization while correcting.

3. Instantaneous and evoked heart-rate analysis. Systole includes utilities and plotting functions for instantaneous and evoked heart rate analysis from raw signal or RR time series data.

4. Heart-rate variability analysis. The package integrates functions for heart rate variability analysis in the time, frequency and non-linear domain. This includes the most widely used metrics of heart rate variability in cognitive science. Other metric or feature extractions can be performed by interfacing with other packages that are more specialized in this domain (Gomes et al., 2019; Makowski et al., 2021).

5. Online systolic peak detection, cardiac-stimulus synchrony, and cardiac circular analysis. The package currently supports recording and synchronization with experiment software like Psychopy (Peirce et al., 2019) from Nonin 3012LP Xpod USB pulse oximeter together with the Nonin 8000SM 'soft-clip' fingertip sensors (USB) as well as Remote Data Access (RDA) via BrainVision Recorder together with Brain product ExG amplifier (Ethernet).

References

Bokeh Development Team. (2018). *Bokeh: Python library for interactive visualization*. https://bokeh.pydata.org/en/latest/

Gent, P. van, Farah, H., Nes, N. van, & Arem, B. van. (2019). HeartPy: A novel heart rate algorithm for the analysis of noisy signals. *Transportation Research Part F: Traffic Psychology and Behaviour*, *66*, 368–378. https://doi.org/10.1016/j.trf.2019.09.015

Gomes, P., Margaritoff, P., & Silva, H. (2019). pyHRV: Development and evaluation of an open-source python toolbox for heart rate variability (HRV). *Proc. Int'l Conf. On Electrical, Electronic and Computing Engineering (IcETRAN)*, 822–828.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-based python JIT compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, 1–6.

Legrand, N., Engen, S. S., Correa, C., Mathiasen, N. K., Nikolova, N., Fardo, F., & Allen, M. (2020). *Emotional Metacognition: Stimulus Valence Modulates Cardiac Arousal and Metamemory* [Preprint]. Neuroscience. https://doi.org/10.1101/2020.06.10.144428

Legrand, N., Nikolova, N., Correa, C., Brændholt, M., Stuckert, A., Kildahl, N., Vejlø, M., Fardo, F., & Allen, M. (2021). The heart rate discrimination task: A psychophysical method to estimate the accuracy and precision of interoceptive beliefs. *bioRxiv*, 2021.02.18.431871. https://doi.org/10.1101/2021.02.18.431871

Lipponen, J. A., & Tarvainen, M. P. (2019). A robust algorithm for heart rate variability time series artefact correction using novel beat classification. *Journal of Medical Engineering & Technology*, *43*(3), 173–181. https://doi.org/10.1080/03091902.2019.1640306

Luis, & Porr, B. (2019). *Popular ECG R peak detectors written in python*. Zenodo. https://doi.org/10.5281/ZENODO.3353396

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., & Chen, S. H. A. (2021). NeuroKit2: A Python toolbox for neurophysiological signal processing. *Behavior Research Methods*, *53*(4), 1689–1696. https://doi.org/10.3758/s13428-020-01516-y

Peirce, J., Gray, J. R., Simpson, S., MacAskill, M., Höchenberger, R., Sogo, H., Kastman, E., & Lindeløv, J. K. (2019). PsychoPy2: Experiments in behavior made easy. *Behavior Research Methods*, *51*(1), 195–203. https://doi.org/10.3758/s13428-018-01193-y

team, T. pandas development. (2020). *Pandas-dev/pandas: pandas* (latest) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2