

pantompkins: A Python Implementation of the Pan-Tompkins Real-Time QRS Detection Algorithm

Hatem Zehir

Laboratory of Study and Research in Instrumentation
and Communication of Annaba, Faculty of Technology,
Badji Mokhtar-Annaba University, Annaba, Algeria
hatem.zehir@univ-annaba.dz

15 July 2026

Summary

`pantompkins` is an open-source Python package implementing the Pan-Tompkins algorithm (Pan and Tompkins 1985) for real-time QRS complex detection in electrocardiogram (ECG) signals. The QRS complex represents ventricular depolarisation and is the most prominent feature of the ECG waveform. Accurate and reliable detection of QRS complexes is fundamental to a wide range of clinical and research applications, including heart rate computation, heart rate variability (HRV) analysis, arrhythmia detection, and the synchronisation of physiological measurements to the cardiac cycle.

The package exposes a single function, `pan_tompkins(ecg, fs, gr)`, which accepts a one-dimensional NumPy array and a sampling frequency, and returns the amplitudes and sample indices of detected R-peaks along with the accumulated filter delay. It is designed to integrate directly with the scientific Python ecosystem (NumPy, SciPy, and Matplotlib) and supports ECG signals recorded at any sampling rate without manual parameter adjustment.

Statement of Need

The Pan-Tompkins algorithm [pan1985] remains one of the most cited and widely taught QRS detection methods in biomedical engineering, achieving sensitivity and specificity above 99% on the MIT-BIH Arrhythmia Database (Moody and Mark 2001). Despite its importance, the most complete and frequently referenced implementation has historically been written in MATLAB (Sedghamiz

2014), a proprietary environment that requires an expensive commercial license. This constitutes a significant barrier for researchers, students, and clinicians who conduct their analyses in Python, which has become the dominant language for biomedical data science, machine learning, and wearable health monitoring.

`pantompkins` addresses this gap by providing a free, open-source, and pip-installable Python implementation that requires no proprietary software. It is intended for researchers working on ECG-based cardiac analysis, educators teaching biomedical signal processing, and engineers building real-time or offline cardiac monitoring pipelines.

State of the Field

Several Python packages offer QRS detection functionality. `NeuroKit2` (Makowski et al. 2021) is a comprehensive neurophysiological signal processing library that includes multiple QRS detection algorithms alongside a wide range of other biosignal processing tools. `BioSPPy` (Carreiras et al. 2015) provides a broader biosignal processing framework with ECG-specific modules. `HeartPy` (van Gent et al. 2019) focuses primarily on photoplethysmography (PPG) but includes basic ECG peak detection. The `py-ecg-detectors` package (Porr et al. 2024) offers a collection of classical QRS detectors implemented in Python, including Pan-Tompkins among others.

`pantompkins` differs from these alternatives in scope and intent. Rather than embedding the algorithm within a larger framework with numerous dependencies, `pantompkins` provides a self-contained, minimal-dependency implementation of a single well-defined algorithm. This makes it particularly well suited for educational use, for integration into existing pipelines without introducing framework-level dependencies, and as a reproducible reference implementation. The package also faithfully replicates the intermediate visualisation pipeline of the original MATLAB implementation (Sedghamiz 2014), which is valuable for teaching and debugging.

Software Design

The implementation follows the five-stage processing pipeline described in the original publication (Pan and Tompkins 1985):

1. **Bandpass filtering (5–15 Hz):** removes baseline wander and high-frequency noise. For signals sampled at 200 Hz, two sequential third-order Butterworth filters are applied (low-pass at 12 Hz, then high-pass at 5 Hz), following the original design. For all other sampling rates, a single band-pass Butterworth filter is used.
2. **Derivative filtering:** a five-point derivative approximation emphasises the steep slopes characteristic of QRS complexes. The filter kernel is

interpolated to match the input sampling rate, enabling correct operation at any frequency without manual reconfiguration.

3. **Pointwise squaring:** ensures all values are positive and nonlinearly amplifies larger slopes, suppressing lower-amplitude P and T waves relative to QRS complexes.
4. **Moving-window integration (~150 ms):** produces a smooth envelope capturing the total energy in each QRS region.
5. **Adaptive dual-threshold decision logic:** two independent sets of adaptive thresholds (one applied to the integrated signal, one to the bandpass-filtered signal) classify candidate peaks as QRS complexes or noise. A T-wave rejection rule discards peaks occurring within 360 ms of a confirmed QRS, and a search-back procedure recovers missed beats when the RR interval exceeds 1.66 times the running mean.

A key design decision was to use `scipy.signal.filtfilt` throughout rather than a causal filter. This applies each filter forward and backward to achieve zero-phase distortion, which is appropriate for offline analysis and eliminates the need to track and compensate for per-filter phase delays. The accumulated delay reported by the function reflects only the moving-average window contribution.

The package follows the `src/` layout convention, uses `hatchling` as its build backend, and declares dependencies conservatively (`numpy>=1.21`, `scipy>=1.7`, `matplotlib>=3.4`). A test suite of 16 tests across five test classes covers basic detection accuracy at multiple sampling rates (200 Hz, 360 Hz, 500 Hz), index validity, RR interval plausibility, delay properties, and edge cases including two-dimensional input rejection. Continuous integration is provided via GitHub Actions, running the full test suite on Python 3.9 through 3.12 across Ubuntu, macOS, and Windows.

Research Impact Statement

`pantompkins` package enables researchers to perform reproducible QRS detection within fully open-source Python pipelines. It is directly compatible with the `wfdb` library (Goldberger et al. 2000) for reading PhysioNet datasets, allowing immediate application to benchmark databases such as the MIT-BIH Arrhythmia Database without additional data conversion steps. The package is intended to serve as a reproducible reference implementation for educational use and as a component in larger cardiac analysis workflows.

AI Usage Disclosure

Generative AI tools were used during the preparation of this software and paper. Specifically, AI assistance was used to support code structuring, documentation drafting, and consistency checking across project files. All AI-generated content

was reviewed, verified for correctness, and revised by the author. The core algorithmic logic was ported directly from the original MATLAB implementation (Sedghamiz 2014) with manual verification against the reference paper (Pan and Tompkins 1985).

Acknowledgements

The author thanks Hooman Sedghamiz for making the original MATLAB implementation publicly available, and for the clarity of documentation that made this port straightforward to verify. No external funding was received for this work.

References

- Carreiras, Carlos, Ana Priscila Alves, André Antas Figueiredo, and Hugo Plácido Da Silva. 2015. *BioSPPy: Biosignal Processing in Python*. <https://github.com/PIA-Group/BioSPPy>.
- Goldberger, Ary L, Luis AN Amaral, Leon Glass, et al. 2000. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals.” *Circulation* 101 (23): e215–20.
- Makowski, Dominique, Tam Pham, Zen J Lau, et al. 2021. “NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing.” *Behavior Research Methods* 53 (4): 1689–96.
- Moody, George B, and Roger G Mark. 2001. “The Impact of the MIT-BIH Arrhythmia Database.” *IEEE Engineering in Medicine and Biology Magazine* 20 (3): 45–50.
- Pan, Jiapu, and Willis J Tompkins. 1985. “A Real-Time QRS Detection Algorithm.” *IEEE Transactions on Biomedical Engineering*, no. 3: 230–36.
- Porr, Bernd, Luis Howell, Ioannis Stournaras, and Yoav Nir. 2024. “ECG Detectors.” *Zenodo*.
- Sedghamiz, Hooman. 2014. “Matlab Implementation of Pan Tompkins ECG QRS Detector.” *Code Available at the File Exchange Site of MathWorks*, 1–3.
- van Gent, Paul, Haneen Farah, Nicole van Nes, and Bart van Arem. 2019. “HeartPy: A Novel Heart Rate Algorithm for the Analysis of Noisy Signals.” *Transportation Research Part F: Traffic Psychology and Behaviour* 66: 368–78. <https://doi.org/https://doi.org/10.1016/j.trf.2019.09.015>.