biopeaks: a graphical user interface for feature extraction from heart- and breathing biosignals

Jan C. Brammer[1]

1 Behavioral Science Institute, Radboud University Nijmegen, Nijmegen, The Netherlands

Statement of need

Heart- and breathing biosignals increasingly gain popularity in academia and industry, sparked by the availability of easy-to-use, low-cost biosignal sensors along with a growing ecosystem of free, open-source software for the analysis of biosignals (Carreiras et al., n.d.; Makowski et al., 2020). However, an open-source, freely available graphical user interface (GUI) for biosignal analysis is currently lacking. `biopeaks` addresses this need. Compared to application programming interfaces (Carreiras et al., n.d.; Makowski et al., 2020), its GUI allows for more intuitive and immediate visual interaction with the biosignal, which is especially valuable during data preprocessing and exploration. At the time of writing, `biopeaks` is used in multiple projects at the Gemhlab.

Functionality

- processing of open biosignal formats EDF (Kemp, Värri, Rosa, Nielsen, & Gade, 1992), OpenSignals ("OpenSignals data visualization software bitalino," n.d.), as well as plain text files (.txt, .csv, .tsv)
- interactive biosignal visualization
- biosignal segmentation
- automatic extrema detection (R-peaks in electrocardiogram (ECG), systolic peaks in photoplethysmogram (PPG), as well as exhalation troughs and inhalation peaks in breathing biosignals) with biosignal-specific, sensible defaults
- automatic state-of-the-art artifact correction for ECG and PPG extrema
- point-and-click editing of extrema
- extraction of instantaneous features: rate and period for heart and breathing biosignals, as well as breathing amplitude
- automatic analysis of multiple files (batch processing)
- .csv export of extrema and instantaneous features for further analysis (e.g., heart rate variability)

An analyst who wants to extract information from heart or breathing biosignals performs multiple analysis steps. First, they verify if the biosignal's quality is sufficient for analysis, since biosignals can be corrupted for a number of reasons (e.g., movement artifacts, poor sensor placement). `biopeaks` allows the analyst to quickly visualize a biosignal and interact with it (panning, zooming) to evaluate its quality. If the analyst deems the biosignal's quality sufficient, they proceed to identify local extrema in the physiological time series. Local extrema include R-peaks in ECG and systolic peaks in PPG, representing the contraction of the ventricular heart muscle. In breathing biosignals, the relevant local extrema are inhalation peaks and exhalation troughs. `biopeaks` detects these extrema automatically using three

biosignal-specific algorithms. Breathing extrema are detected using a variant of the "zero-crossing algorithm with amplitude threshold" (Khodadad et al., 2018). Systolic peaks in PPG signals are identified using an implementation of "Method IV; Event-Related Moving Averages with Dynamic Threshold" introduced by Elgendi et al. (Elgendi, Norton, Brearley, Abbott, & Schuurmans, 2013). Lastly, the ECG R-peak detector is a custom algorithm that has been evaluated on the Glasgow University Database (GUDB) (Howell & Porr, 2018) which contains ECG signals along with R-peak annotations. The performance of the R-peak detector has been evaluated in terms of sensitivity (aka recall; i.e., how many of the correct extrema were detected?) and precision (i.e., how many of the detected extrema are correct extrema?). Peak detection has been evaluated on the records of all 25 participants included in the GUDB using the ECG channel corresponding to Einthoven lead II. The tolerance for true positive peak detection was set to one sample. The GUDB has not been used to optimize the R-peak detector prior to the performance evaluation. The performance at rest (sitting, 25 records) and in dynamic conditions (handbike, 24 records due to the missing R-peak annotations of participant 04) is as follows:

		sitting	handbike
precision	mean	.9995	.9855
	std	.0017	.0234
sensitivity	mean	.9974	.9853
	std	.0037	.0250

The code for performance evaluation is included in the `biopeaks` installation and can be run without downloading the GUDB (the database is streamed). Despite the robust performance of the extrema detectors, algorithmically identified extrema can be misplaced (false positives) or extrema might be missed (false negatives), if there are noisy segments in the biosignal. If left uncorrected, these errors can significantly distort subsequent analysis steps (Berntson & Stowell, 1998). To address this problem and ensure the correct placement of extrema, `biopeaks` offers intuitive point-and-click extrema editing (i.e., removing and adding extrema). Additionally, for cardiac biosignals, `biopeaks` offers state-of-the-art automatic extrema correction (Lipponen & Tarvainen, 2019). Finally, based on the local extrema, the analyst can extract features from the biosignal. The features are based on temporal or amplitude differences between the local extrema. Figure 1, Figure 2, and Figure 3 illustrate the extraction of instantaneous heart period, breathing period, and breathing (inhalation) amplitude respectively.
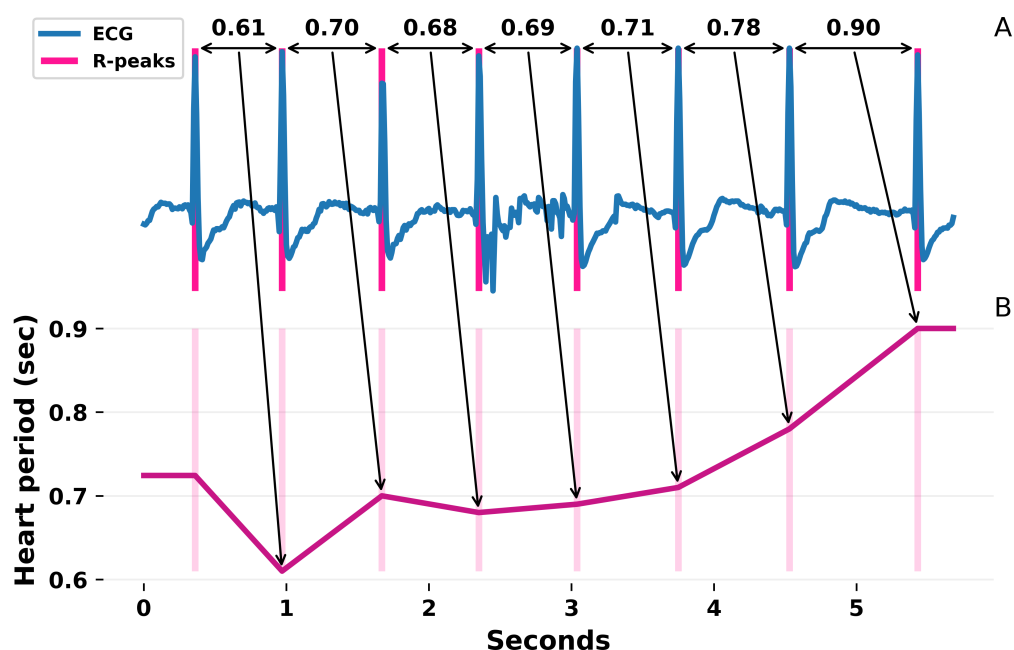
Figure 1: Extraction of heart period (panel B) based on R-peaks in an ECG (panel A). Note that this is conceptually identical to the extraction of heart period based on systolic peaks in PPG.
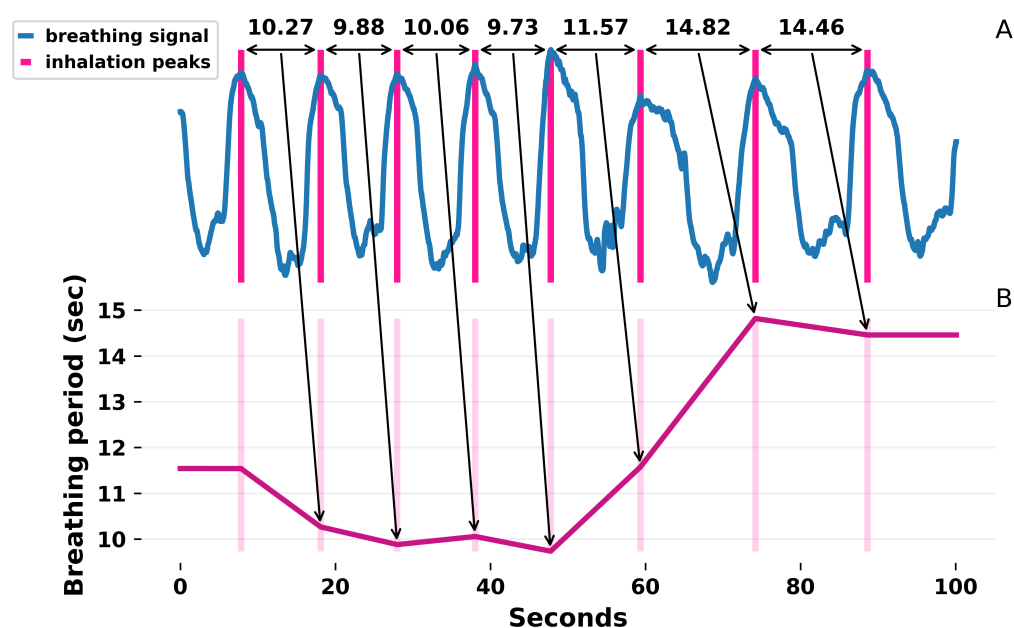

Figure 2: Extraction of breathing period (panel B) based on inhalation peaks in a breathing biosignal (panel A).
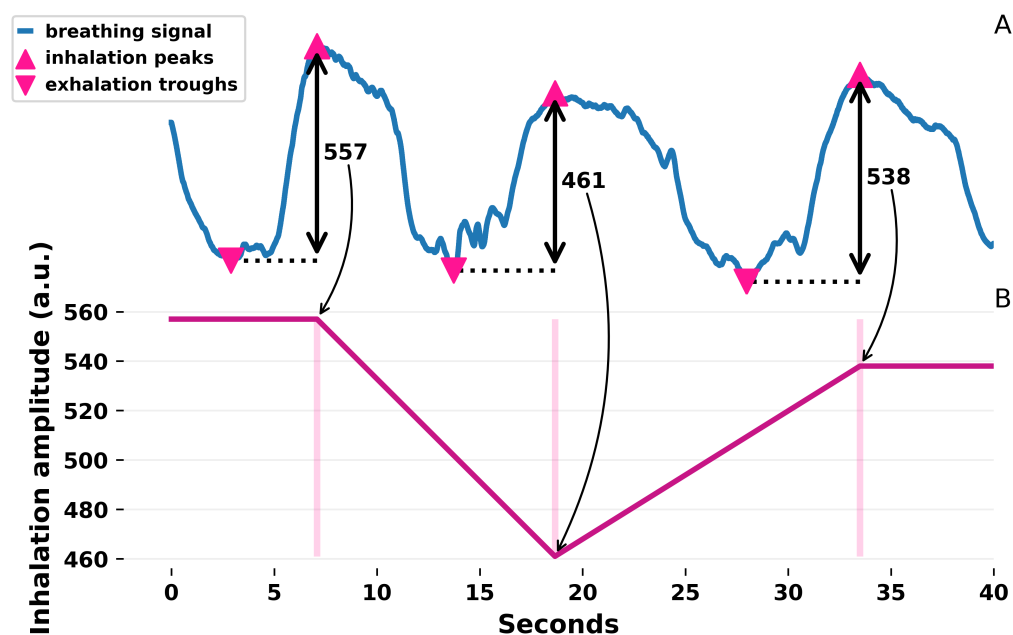
Figure 3: Extraction of inhalation amplitude (panel B) based on breathing extrema in a breathing biosignal (panel A).

In summary, `biopeaks` is designed to make heart- and breathing biosignal inspection, extrema detection and editing, as well as feature extraction fast and intuitive. It does not aim to offer a suite of low-level signal processing tools (e.g., digital filter design), and abstracts these details away with sensible biosignal-specific defaults. `biopeaks` is implemented in Python using the cross-platform PySide2 framework (official Python bindings for Qt) ("Qt for Python — Qt for Python," n.d.), and leverages Matplotlib (Hunter, 2007), NumPy (Harris et al., 2020), SciPy (Virtanen et al., 2020) and pandas (McKinney, 2010) for visualization and signal processing. There are freely available alternatives to `biopeaks` that are implemented in MATLAB (Kaufmann, Sütterlin, Schulz, & Vögele, 2011; Sjak-Shie, 2019) or C# (Plesinger, Jurco, Halamek, & Jurak, 2016). However, the source code of these tools is not available (Kaufmann et al., 2011; Plesinger et al., 2016) or they are not released under an open source license (Kaufmann et al., 2011; Plesinger et al., 2016; Sjak-Shie, 2019).

Acknowledgements

References

Berntson, G. G., & Stowell, J. R. (1998). ECG artifacts and heart period variability: Don't miss a beat! *Psychophysiology*, *35*(1), 127–132. doi:10.1111/1469-8986.3510127

Carreiras, C., Alves, A. P., Lourenço, A., Canento, F., Silva, H., Fred, A., & others. (n.d.). BioSPPy: Biosignal processing in Python. Retrieved from https://github.com/PIA-Group/BioSPPy/

Elgendi, M., Norton, I., Brearley, M., Abbott, D., & Schuurmans, D. (2013). Systolic Peak Detection in Acceleration Photoplethysmograms Measured from Emergency Responders in Tropical Conditions. *PLoS ONE*, *8*(10), e76585. doi:10.1371/journal.pone.0076585

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., et al. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. doi:10.1038/s41586-020-2649-2

Howell, L., & Porr, B. (2018). High precision ECG Database with annotated R peaks, recorded and filmed under realistic conditions. Data Collection. Retrieved from http://researchdata.gla.ac.uk/716/

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

Kaufmann, T., Sütterlin, S., Schulz, S. M., & Vögele, C. (2011). ARTiiFACT: A tool for heart rate artifact processing and heart rate variability analysis. *Behavior Research Methods*, *43*(4), 1161–1170. doi:10.3758/s13428-011-0107-7

Kemp, B., Värri, A., Rosa, A. C., Nielsen, K. D., & Gade, J. (1992). A simple format for exchange of digitized polygraphic recordings. *Electroencephalography and Clinical Neurophysiology*, *82*(5), 391–393. doi:10.1016/0013-4694(92)90009-7

Khodadad, D., Nordebo, S., Müller, B., Waldmann, A., Yerworth, R., Becher, T., Frerichs, I., et al. (2018). Optimized breath detection algorithm in electrical impedance tomography. *Physiological Measurement*, *39*(9), 094001. doi:10.1088/1361-6579/aad7e6

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. doi:10.1080/03091902.2019.1640306

Makowski, D., Pham, T., Lau, Z. J., Brammer, J. C., Lespinasse, F., Pham, H., Schölzel, C., et al. (2020). NeuroKit2: A python toolbox for neurophysiological signal processing. Zenodo. doi:10.5281/ZENODO.3597887

McKinney, W. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th python in science conference* (pp. 56–61). doi:10.25080/Majora-92bf1922-00a

OpenSignals data visualization software bitalino. (n.d.). Retrieved March 12, 2020, from https://bitalino.com/en/software

Plesinger, F., Jurco, J., Halamek, J., & Jurak, P. (2016). SignalPlant: An open signal processing software platform. *Physiological Measurement*, *37*(7), N38–N48. doi:10.1088/0967-3334/37/7/N38

Qt for Python — Qt for Python. (n.d.). Retrieved from https://doc.qt.io/qtforpython/

Sjak-Shie, E. E. (Elio). (2019). *PhysioData toolbox.* Retrieved from https://PhysioDataToolbox.leidenuniv.nl

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, *17*, 261–272. doi:10.1038/s41592-019-0686-2