

pyemgpipeline: A Python package for electromyography processing

Tsung-Lin Wu¹, Amr A. Alhossary², Todd C. Pataky³, Wei Tech Ang^{1,2}, and Cyril J. Donnelly^{2¶}

¹ Nanyang Technological University, School of Mechanical & Aerospace Engineering ² Nanyang Technological University, Rehabilitation Research Institute of Singapore ³ Kyoto University, Department of Human Health Sciences ¶ Corresponding author

DOI: [10.21105/joss.04156](https://doi.org/10.21105/joss.04156)

Software

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

Editor: Øystein Sørensen ↗

Reviewers:

- [@tuliofalmeida](#)

Submitted: 24 January 2022

Published: 08 April 2022

License

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

Summary

We have developed an electromyography (EMG) signal processing pipeline package called pyemgpipeline, which is suitable for both surface EMG and intramuscular EMG processing. pyemgpipeline implements internationally accepted EMG processing conventions and provides a high-level interface for ensuring user adherence to those conventions, in terms of (1) processing parameter values, (2) processing steps, and (3) processing step order. The international standards are from surface EMG for non-invasive assessment of muscles (SENIAM) (Stegeman & Hermens, 1999). The seven processing steps included in the package are DC offset removal, bandpass filtering, full wave rectification, linear envelope, end frame cutting, amplitude normalization, and segmentation. In sport tasks particularly, it has been observed that amplitudes greater than 100% maximum voluntary contraction (MVC) can be observed (Devaprakash et al., 2016). Therefore, we will be using amplitude normalization recommendations from Devaprakash et al. (2016) for amplitude normalization – this method guarantees a muscle will not exceed 100% MVC as all EMG trials from the experiment are used to identify maximal muscle activation. What is not included in the package is time event detection and time normalization as different laboratories are interested in different phases of gait and prefer to use either linear or nonlinear time normalization techniques.

As stated by Stegeman & Hermens (1999) although EMG is easy to use it is also easy to abuse. Researchers have thus tried to standardize the use of EMG regarding implementation, analysis, and reporting (Editors, 2018; Merletti & Torino, 1999; Stegeman & Hermens, 1999). These fine works however have arguably been unsuccessfully adopted by many researchers to date. Based on the published standards and decades of practical experience, we summarize EMG signal processing conventions into the seven steps mentioned above and implement them within an easy-to-use package so that researchers and clinicians with all levels of experience process their EMG signals using correct conventions. In addition to the processing steps and their order, the choice of processing/filtering parameters can be changed to match the needs of the research or clinician. For example, within the bandpass filtering step, proper values of the low and high cutoff frequencies of the Butterworth bandpass filter depend on different use cases. We carefully set default values in the package, and we also include suggested values under different scenarios in [related API documentation](#) (Drake & Callaghan, 2006; Editors, 2018; Merletti & Torino, 1999; Stegeman & Hermens, 1999).

After the aforementioned seven processing steps, sometimes it is necessary to perform the time normalization step before further analysis. Time normalization can be executed by an external package `mwarp1d` (Pataky et al., 2019).

Within this package, class `DataProcessingManager` is designed for high-level, guided processing, and users are encouraged to use it to adhere to accepted EMG processing conventions.

The depiction of pyemgpipeline data processing flow is shown in [Figure 1](#), which includes the original signal and the processed signals of all processing steps. Please refer to the [full documentation](#) and the [source code](#) for detailed information.

Statement of need

Research purpose: pyemgpipeline aims to provide software tools for electromyography (EMG) data processing, while ensuring adherence to internationally accepted EMG processing conventions.

Problem solved: pyemgpipeline implements internationally accepted EMG processing conventions and provides a high-level interface for ensuring user adherence to those conventions. It facilitates the convenience and correctness of processing EMG data. To our best knowledge, no other package provides tools of EMG processing pipeline to ensure users adhere to accepted conventions in terms of (1) processing parameter values, (2) processing steps, (3) processing step order, and (4) amplitude normalization.

Target audience: The target audience is anyone working with surface or intramuscular EMG data such as gait biomechanics, sports science, rehabilitation, and robotics.

Figures

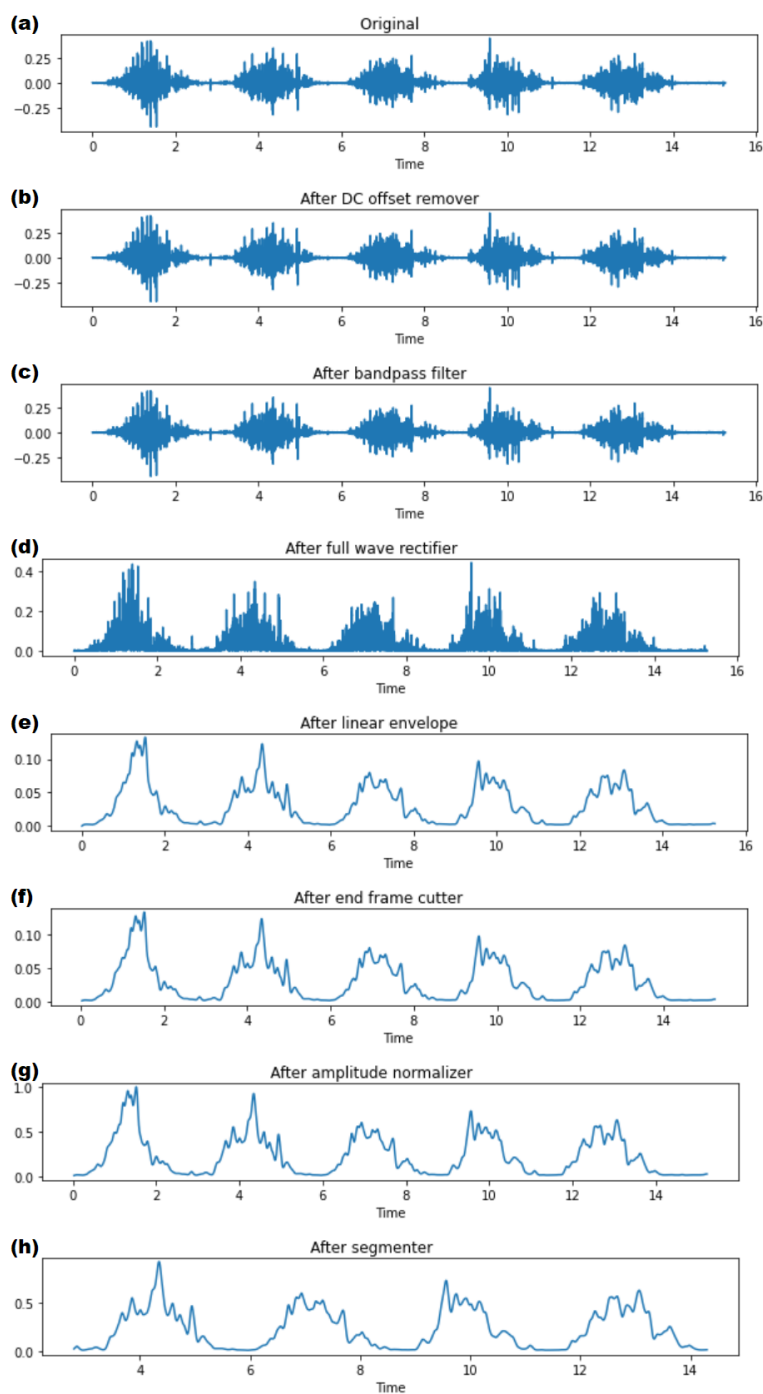


Figure 1: Depiction of pyemgpipeline data processing flow. (x-axis: seconds, y-axis: amplitude.)

References

- Devaprakash, D., Weir, G. J., Dunne, J. J., Alderson, J. A., & Donnelly, C. J. (2016). The influence of digital filter type, amplitude normalisation method, and co-contraction

- algorithm on clinically relevant surface electromyography data during clinical movement assessments. *Journal of Electromyography and Kinesiology*, 31, 126–135. <https://doi.org/10.1016/j.jelekin.2016.10.001>
- Drake, J. D. M., & Callaghan, J. P. (2006). Elimination of electrocardiogram contamination from electromyogram signals: An evaluation of currently used removal techniques. *Journal of Electromyography and Kinesiology*, 16, 175–187. <https://doi.org/10.1016/j.jelekin.2005.07.003>
- Editors. (2018). Standards for reporting EMG data. *Journal of Electromyography and Kinesiology*, 42, I–II. [https://doi.org/10.1016/S1050-6411\(18\)30348-1](https://doi.org/10.1016/S1050-6411(18)30348-1)
- Merletti, R., & Torino, P. di. (1999). Standards for reporting EMG data. *Journal of Electromyography and Kinesiology*, 9(1), III–IV. <https://doi.org/10.1201/9781420036985.ax2>
- Pataky, T. C., Naouma, H., & Donnelly, C. J. (2019). mwarpld: Manual one-dimensional data warping in python and PyQt. *Journal of Open Source Software*, 4(44), 1870. <https://doi.org/10.21105/joss.01870>
- Stegeman, D. F., & Hermens, H. J. (1999). *Standards for surface electromyography: The european project "surface EMG for non-invasive assessment of muscles (SENIAM)"* (pp. 108–112). European Community.