

# pyemgpipeline: A Python package for electromyography processing

Tsung-Lin Wu<sup>1</sup>, Amr A. Alhossary<sup>2</sup>, Todd C. Pataky<sup>3</sup>, Wei Tech Ang<sup>1, 2</sup>, and Cyril J. Donnelly<sup>\*2</sup>

<sup>1</sup> Nanyang Technological University, School of Mechanical & Aerospace Engineering <sup>2</sup> Nanyang Technological University, Rehabilitation Research Institute of Singapore <sup>3</sup> Kyoto University, Department of Human Health Sciences

DOI: [10.21105/joss.0XXXX](https://doi.org/10.21105/joss.0XXXX)

## Software

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

Editor: [Editor Name](#) ↗

Submitted: 01 January XXXX

Published: 01 January XXXX

## 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. As 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 normalized 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).

<sup>\*</sup>corresponding author

42 Within this package, class `DataProcessingManager` is designed for high-level, guided process-  
43 ing, and users are encouraged to use it to adhere to accepted EMG processing conventions.  
44 An example of using `pyemgpipeline` is shown in Figure 1, where the original signals and  
45 the processed ones are displayed in subfigure (a) and (b) respectively. Please refer to the full  
46 documentation and the source code for detailed information.

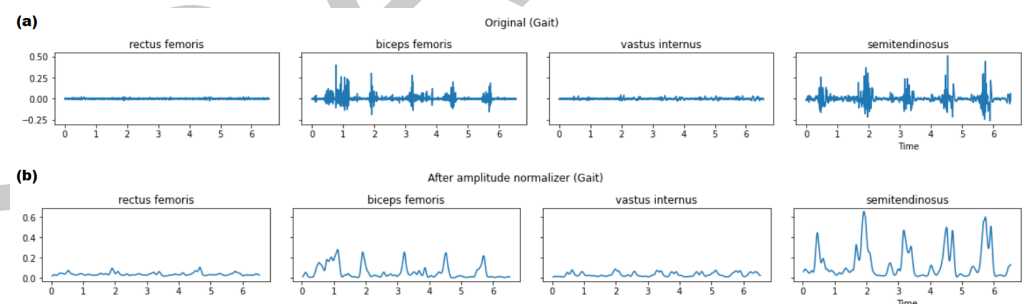
## 47 Statement of need

48 **Research purpose:** `pyemgpipeline` aims at creating a tool of EMG processing pipeline,  
49 ensuring users who are working on EMG processing and analysis to adhere to accepted EMG  
50 processing conventions.

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

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

## 59 Figures



**Figure 1:** An example of applying `pyemgpipeline` to EMG signals of a gait trial, which consists of four channels and lasts 6.6 seconds. (x-axis: seconds, y-axis: amplitude.) (a) The original signals. (b) The processed signals through steps until amplitude normalization. Note that the maximum amplitudes used in amplitude normalization are obtained from all trials (not shown here) including the gait trial.

## 60 References

- 61 Devaprakash, D., Weir, G. J., Dunne, J. J., Alderson, J. A., & Donnelly, C. J. (2016).  
62 The influence of digital filter type, amplitude normalisation method, and co-contraction  
63 algorithm on clinically relevant surface electromyography data during clinical movement  
64 assessments. *Journal of Electromyography and Kinesiology*, 31, 126–135. <https://doi.org/10.1016/j.jelekin.2016.10.001>  
65  
66 Drake, J. D. M., & Callaghan, J. P. (2006). Elimination of electrocardiogram contamination  
67 from electromyogram signals: An evaluation of currently used removal techniques. *Journal*

- 68     of *Electromyography and Kinesiology*, 16, 175–187. [https://doi.org/10.1016/j.jelekin.](https://doi.org/10.1016/j.jelekin.2005.07.003)  
69     2005.07.003
- 70     Editors. (2018). Standards for reporting EMG data. *Journal of Electromyography and Kine-*  
71     *siology*, 42, I–II. [https://doi.org/10.1016/S1050-6411\(18\)30348-1](https://doi.org/10.1016/S1050-6411(18)30348-1)
- 72     Merletti, R., & Torino, P. di. (1999). Standards for reporting EMG data. *Journal of Elec-*  
73     *tromyography and Kinesiology*, 9(1), III–IV.
- 74     Pataky, T. C., Naouma, H., & Donnelly, C. J. (2019). mwarpld: Manual one-dimensional  
75     data warping in python and PyQt. *Journal of Open Source Software*, 4(44), 1870. [https:](https://doi.org/10.21105/joss.01870)  
76     [//doi.org/10.21105/joss.01870](https://doi.org/10.21105/joss.01870)
- 77     Stegeman, D. F., & Hermens, H. J. (1999). *Standards for surface electromyography: The*  
78     *european project "surface EMG for non invasive assessment of muscles (SENIAM)"* (pp.  
79     108–112). European Community.

DRAFT