NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing

Dominique Makowski [1,*], Tam Pham [1], Zen J. Lau [1], Jan C. Brammer [2], Hung Pham [3], Francois Lesspinasse [4], Christopher Sch"olzel [5], & S.H. Annabel Chen [1, 6, 7]

[1] School of Social Sciences, Nanyang Technological University, Singapore

[2] ???

[3] ???

[4] ???

[5] ???

[6] Centre for Research and Development in Learning, Nanyang Technological University, Singapore

[7] Lee Kong Chian School of Medicine, Nanyang Technological University, Singapore

Author Note

* Correspondence concerning this article should be addressed to Dominique Makowski (HSS 04-18, 48 Nanyang Avenue, Singapore; dmakowski@ntu.edu.sg).

Abstract

The NeuroKit2 toolbox is an open-source Python package aimed at providing users with comprehensive and flexible functionality in neurophysiological signal processing. It developed from a collaborative project aimed at offering programming ease for both novice and advanced users to perform elaborate analyses of electrocardiogram (ECG), respiratory (RSP), electrodermal activity (EDA), and electromyography (EMG) data. It comprises of a consistent set of user-friendly, high-level functions that implements an all-in-one cleaning, preprocessing, and processing pipeline with sensible defaults. At the same time, greater flexibility and parametric control can be achieved by using Neurokit2's mid-level functions to build a custom analysis pipeline. (talk about novelty?)

Keywords: neurophysiology, ECG, EDA, EMG, RSP

Word count:

NeuroKit2: A Python Toolbox for Neurophysiological Signal Processing

The field of cognitive neuroscience and psychology is increasingly relying on neurophysiological methods. One of the reason is that such approaches often offer low monetary cost (especially compared with other imaging techniques, such as MRI) and high user convenience (e.g., portability). At the same time, the fields of signal processing and computational data science are strongly growing, tackling issues and limitations, and pushing the horizon of possibilities and opportunities. However, as these methods are often not easily accessible and user-friendly, neurophysiological data processing remains a challenge for many researchers without a formal programming training.

NeuroKit2 aims at addressing this gap by offering a free and user-friendly solution for neurophysiological data processing. It is an open-source Python package, developed in a collaborative environment that continues to welcome contributors from different countries and fields. Historically, *NeuroKit2* is the re-forged successor *NeuroKit* (*https://github.com/neuropsychology/NeuroKit.py*), a PhD side project that ended attracting a lot of users and success (236 GitHub stars as of 13-03-2020). The new version takes on its best features and design choices, and re-implements them in a professional and well-thaught way. It aims at being 1) accessible, 2) well-documented, 2) reliable, 4) cutting-edge and 5) powerful.

The package is available for Python (Van Rossum & Drake, 2009) and thus benefits of its important base of users, existing tutorials and large online community. It's also relatively lightweight, enabling its use as a dependency in other software. The package source code is available under a permissive license on GitHub (*https://github.com/neuropsychology/NeuroKit*), along with its documentation, automatically built and hosted at *https://neurokit2.readthedocs.io/*. Aside for installation and contribution instructions, and a decription of the package's functions, is also includes several "hands-on" examples providing a walk-through on how to address specific issues (for instance, how to exctract and visualize individual heartbeats). The accessibiltiy for newcommers is reinforced by the issue tracker of GitHub,

allowing users to create public issues to inquiry for help.

The packages is made to be reliable, and functions are tested against existing implementations of established reference software such as *BioSPPy* (Carreiras et al., 2015), *hrv under review*, *PySiology* (Gabrieli, Azhari, & Esposito, 2019), *HeartPy* (Gent, Farah, Nes, & Arem, 2019), *systole* (Legrand & Allen, 2020) or *nolds* (Schölzel, 2020). The code itself includes a comprehensive test suite to ensure stability and prevent error. Moreover, the issue tracker allows users to easily report any bugs and track their fixation. Thanks to its collaborative and open developpment, as well as its modular organization, *NeuroKit2* is being developped with a longterm perspective in mind, aiming at remaining cutting-edge through its ability to evolve, adapt, and integrate new methods as they are being developped.

Finally, we believe that the design philosophy contributes to a powerful (allowing to achieve a lot with very few functions) yet flexible (enabling fine control and precision over what is done) user interface. It is described below.

Design Philosophy

Organization

3 levels of functions.

Processing and analysis

This design withholds the promise of being able to do a full processing and features extraction of your data with only 2 functions.

Example

Despite not having a Graphical User Interface (GUI), NeuroKit2 is accessible to people with very little knowledge of python or programming in general.

Event-related Paradigm

```
import neurokit2 as nk
```

ADD HERE THE EXAMPLE ABOUT EVENT RELATED FEATURES using the example dataset

Resting-state Features

ADD here an example on resting feature dataset (to select from our data, downsample, transform to csv and add to the datasets)

Conflict of Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgements

All the contributors (https://neurokit2.readthedocs.io/credits.html) that reported bugs, and the users.

References

Carreiras, C., Alves, A. P., Lourenço, A., Canento, F., Silva, H., Fred, A., & others. (2015). BioSPPy: Biosignal processing in Python. Retrieved from https://github.com/PIA-Group/BioSPPy/

Gabrieli, G., Azhari, A., & Esposito, G. (2019). PySiology: A python package for physiological feature extraction. In *Neural approaches to dynamics of signal exchanges* (pp. 395–402). Springer Singapore. https://doi.org/10.1007/978-981-13-8950-4_35

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

Legrand, N., & Allen, M. (2020). Systole: A python toolbox for preprocessing, analyzing, and synchronizing cardiac data. Retrieved from https://github.com/embodied-computation-group/systole

Schölzel, C. (2020). NOnLinear measures for dynamical systems (nolds). Retrieved from https://github.com/CSchoel/nolds

Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.