

# neurone\_loader: A Python module for loading EEG data recorded with Bittium NeurOne

Felix A. Heilmeyer<sup>1</sup> and Tonio Ball<sup>1</sup>

<sup>1</sup> Translational Neurotechnology Lab, Department of Neurosurgery, University of Freiburg–Medical Center, Freiburg im Breisgau, Germany

DOI: [10.21105/joss.01626](https://doi.org/10.21105/joss.01626)

## Software

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

**Submitted:** 06 July 2019

**Published:** 01 October 2019

## License

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

## Summary

In the neuroimaging domain, most imaging equipment comes with its own binary format to store raw imaging data. It has always been a challenge for researchers to handle these formats and convert them into formats compatible with the processing tools used in their research. For example at this time the popular EEGLAB toolbox for MATLAB supports over 30 different data formats.

Bittium NeurOne is a research-grade EEG system that comes with a recording software which, like many others, has its own proprietary binary format for storing raw recordings. For a long time, MATLAB has been a popular processing environment in neuroscientific research, but lately especially with the rise of machine learning methods python has also become more present in neuroscientific workflows. Despite this, Bittium at this time only provides a MATLAB conversion script for their data format, so python-based workflows had to include a conversion step using the proprietary MATLAB software.

To enable pure python processing of data recorded with Bittium NeurOne we created the `neurone_loader` python package. It provides the ability to load data in the proprietary NeurOne binary format in pure python. The data can be used directly as `numpy` arrays or imported into `python-mne` (Gramfort et al., 2013) (Gramfort et al., 2014), a popular python package for processing neuroimaging data.

Since EEG data is often recorded at very high sampling rates and single recordings can reach sizes of hundreds of gigabytes `neurone_loader` was designed with memory restrictions in mind. To make it easier to work with data this large even in environments with (relatively) restricted memory resources, most data will be loaded lazily and redundant data will be removed from memory as soon as it has been copied.

## Acknowledgements

We acknowledge the work of Andreas Henelius at the Finnish Institute of Occupational Health figuring out how to read most the NeurOne binary format in pure python as part of his [export2hdf](#) project.

## References

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., et al. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7(7 DEC), 267. doi:[10.3389/fnins.2013.00267](https://doi.org/10.3389/fnins.2013.00267)

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Parkkonen, L., et al. (2014). MNE software for processing MEG and EEG data. *NeuroImage*, 86, 446–60. doi:[10.1016/j.neuroimage.2013.10.027](https://doi.org/10.1016/j.neuroimage.2013.10.027)