The Journal of Open Source Software

MNE-LSL: Real-time framework integrated with MNE-Python for online neuroscience research through LSL-compatible devices.

Mathieu Scheltienne [1], **Eric Larson** [2], **Arnaud Desvachez** [1], and **Kyuhwa Lee** [3]

1 Fondation Campus Biotech Geneva, Geneva, Switzerland **2** Institute for Learning & Brain Sciences, University of Washington, Seattle, WA, United States of America **3** Wyss Center for Bio and Neuroengineering, Geneva, Switzerland

DOI: 10.21105/joss.08088

Software

- Review
- Repository
- Archive

Editor: Marcel Stimberg

Reviewers:

- @matthiasdold
- @teonbrooks

Submitted: 03 September 2024
Published: 23 July 2025

Summary

Neural, physiological, and behavioral data plays a key role in understanding the human brain and body. Neurophysiology, as an umbrella term, encompasses various modalities for measuring brain and nervous system activity. These include electrophysiology (EEG, EMG), magnetophysiology (MEG), electrocorticography (ECoG), functional near-infrared spectroscopy (fNIRS), and other techniques that capture different aspects of neural function. Through these complementary approaches, researchers can measure electrical brain activity, heart rate, muscle contractions, eye movements, and other physiological signals. These diverse measurements provide crucial insights into cognitive processes, emotional states, physical health, and behavioral responses.

With the growing popularity of brain-computer interfaces (BCIs) and neurofeedback applications, the demand for real-time processing of neurophysiological data has surged. Real-time applications present unique challenges, requiring the continuous collection, synchronization, and processing of data from multiple sources with minimal latency. This requires the implementation of advanced methods for handling data in real-time, including the reduction of noise and management of artifacts.

A critical challenge in real-time applications is the need for direct access to data streams from measurement devices, which often rely on device-specific APIs. Adapting a real-time application from one device to another can be labor-intensive, as it requires modifications to the entire communication protocol. To address this issue, the Lab Streaming Layer (LSL; LabStreamingLayer contributors, n.d.) offers a standardized protocol for streaming time-series data from multiple devices in real-time. LSL has gained significant popularity, particularly among EEG manufacturers, many of whom now provide LSL streaming capabilities out-of-the-box with their devices. By abstracting the complexities of device-specific APIs, LSL allows researchers to concentrate on data analysis rather than the intricacies of device communication.

Beyond data acquisition, the real-time analysis of signals also presents significant challenges. MNE-Python (Gramfort et al., 2013) is a comprehensive toolset for processing and analyzing neurophysiological data in Python, a widely-used programming language in the scientific community. MNE-LSL enhances the integration of LSL with MNE-Python, providing robust objects for handling and processing both continuous and epoch-based data from real-time sources, thereby streamlining the development of real-time neurophysiological applications.

Statement of need

The rise of brain-computer interfaces (BCIs) and neurofeedback applications has led to an increased demand for tools capable of real-time acquisition and processing of neural, physiological, and behavioral data. As these applications become more sophisticated and widespread, researchers and developers require flexible, robust solutions to handle the complexities of real-time data streams, which often involve multiple devices and modalities.

Existing platforms offer different approaches to neurophysiological data acquisition and processing. BCI2000 (Schalk et al., 2004) represents one of the most established solutions, providing a comprehensive framework for BCI research with its own modules for data acquisition, signal processing, and application development. However, its monolithic architecture and C++-based implementation can present challenges for researchers seeking to integrate custom Python-based analyses or leverage existing Python scientific libraries for neurophysiological data processing. OpenBCI (OpenBCI contributors, n.d.) represents a specific hardware ecosystem with its own software tools, limiting researchers to devices within that ecosystem. BrainFlow (BrainFlow contributors, n.d.), while providing a unified API to interact with biosensor data from various manufacturers, still has limitations in supporting the full range of devices used in neurophysiological research. Similarly, GUI-based platforms like Neuromore (Neuromore, n.d.) prioritize user-friendliness, often at the expense of customization and advanced functionality, which can be a limitation for experienced users requiring more control over data processing pipelines.

In contrast, Lab Streaming Layer (LSL) has emerged as a widely adopted, device-agnostic protocol within the neurophysiological research community, especially for EEG data. LSL's ability to unify data streaming from various devices under a common framework has made it an invaluable tool for researchers who need to integrate multiple data sources seamlessly. However, while LSL provides a solid foundation for real-time data acquisition, its integration with Python, particularly for processing and analysis, has been less intuitive and less accessible.

To bridge this gap, MNE-LSL was developed as a solution that not only facilitates the acquisition of real-time LSL streams but also integrates seamlessly with MNE-Python, a leading toolset for neurophysiological data analysis. Unlike its predecessor, MNE-realtime (MNE-realtime contributors, n.d.), which had limitations in LSL support and user interface design, MNE-LSL offers an intuitive MNE-Python-like API allowing for real-time processing of continuous and epoch-based data streams in a familiar environment.

MNE-LSL further enhances the real-time data acquisition workflow by re-implementing the low-level pylsl (pyLSL contributors, n.d.) library with improved structure and documentation. The implementation defaults to efficient NumPy (Harris et al., 2020) operations and ensures cross-platform compatibility by packaging the liblsl library directly within Python wheels. These improvements provide researchers with a powerful, user-friendly tool for real-time data analysis within the Python ecosystem, meeting the growing needs of the neurophysiological research community.

Acknowledgements

We would like to acknowledge the LSL developers, with special thanks to Tristan Stenner and Chadwick Boulay.

MNE-LSL development is supported by the Fondation Campus Biotech Geneva, Geneva, Switzerland.

References

BrainFlow contributors. (n.d.). *BrainFlow*. https://brainflow.org.

Gramfort, A., Luessi, M., Larson, E., Engemann, D. A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., & Hämäläinen, M. S. (2013). MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, *7*(267), 1–13. https://doi.org/10.3389/fnins.2013.00267

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

LabStreamingLayer contributors. (n.d.). LabStreamingLayer. In *GitHub repository*. GitHub. https://github.com/sccn/labstreaminglayer

MNE-realtime contributors. (n.d.). *MNE-realtime*. https://mne.tools/mne-realtime/.

Neuromore. (n.d.). *Neuromore studio*. https://www.neuromore.com.

OpenBCI contributors. (n.d.). *OpenBCI*. https://openbci.com.

pyLSL contributors. (n.d.). pyLSL. In *GitHub repository*. GitHub. https://github.com/labstreaminglayer/pylsl

Schalk, G., Mcfarland, D., Hinterberger, T., Birbaumer, N., & Wolpaw, J. (2004). BCI2000: A general-purpose brain-computer interface (BCI) system. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1034–1043. https://doi.org/10.1109/TBME.2004.827072