

[Articles Describing Code](#)

Automating data integration and publishing for neuroimaging via LSLAutoBIDS

Manpa Barman, BEng¹, Jan Range, MSc^{2,3}, Benedikt Valerian Ehinger, PhD^{1,3}^a¹ Institute for Visualization and Interactive Systems, University of Stuttgart, Stuttgart, Germany, ² Institute of Biochemistry, University of Stuttgart, Stuttgart, Germany, ³ Stuttgart Center for Simulation Science, University of Stuttgart, Stuttgart, Germany

Keywords: BIDS, EEG, lab streaming layer, data collection, datasets, open science, data standards, DataLad, Dataverse

<https://doi.org/10.52294/001c.159415>

Aperture Neuro

Vol. 6, Issue SI 1, 2026

Cognitive neuroscience routinely collects large datasets, yet data integration, curation, version control, and publishing are rarely automated. Here, we present such a workflow, offering open science by design, with an implementation in the Python based LSLAutoBIDS open-source package. We first describe our exemplary workflow based on LabStreamingLayer (to integrate), BIDS (to transform), DataLad (to version), and Dataverse (to publish), before discussing the place such tools can have in future data collection efforts.

INTRODUCTION

Modern neuroscience research often relies on collecting complex, multimodal datasets, for example, combining behavioral, eye-tracking, and neuroimaging (e.g., electroencephalography (EEG)) data. Such datasets are valuable for understanding cognitive processes, but are often challenging to collect due to the resource-intensive nature of experiments and the logistical demands. They are often collected by trainees or research assistants with limited prior experience or training. And even to more experienced data collectors, the following examples of what can go wrong will sound very familiar: 1) the same participant ID was used twice in a row, and data was irreversibly overwritten. 2) While running a study, the experiment was modified to fix a bug, but it was not documented after which subject it was fixed - later it remained unclear whether early subjects data were retrogradely fixed or not. 3) A master's student left the lab without documenting their last dataset, leaving a mess of various files on various computers. 4) To move files from the various recording computers to their analysis system, every lab user seems to have their own idiosyncratic workflow, from USB sticks to WiFi hotspots. In this paper, we show that these issues can be prevented while at the same time saving time and training effort.

To increase the robustness of data recording pipelines, as well as to encourage data sharing, we present an approach to automate parts of this workflow. The automation we propose can be understood under the umbrella of open science by design.¹ The central idea is to change infrastruc-

ture and workflows in a way that they are open from the point of creation, making open science a by-product and not an afterthought.¹ In our case, the new data is curated and inserted into an existing dataset, versioned, and (privately) archived in a remote repository immediately after the recording. Additionally, many open data check boxes are also immediately marked without additional effort by the experimenter.

Our proposed workflow automates four main aspects of such data recording pipelines: data integration, data curation, data versioning, and data publishing.

Data Integration: Collected data often comes from many sources. LabStreamingLayer (LSL)² has become a standard way to record multiple data streams and synchronize their time series. In addition, other data sources need to be collected from devices that cannot send them via LSL (e.g., experimental logging files, proprietary eye-tracking data).

Data Curation: To allow machines and humans to navigate such a diverse data collection, we want to structure them. The Brain Imaging Data Structure (BIDS)³ standard provides a well-described data curation environment, offering consistent directory structure, file naming conventions, and metadata descriptors for neuroimaging data. Once data is structured in BIDS format, the dataset can be seamlessly integrated into BIDS-compliant analysis pipelines.

Data Versioning: The next aspect is to include datasets into version control. While much progress has been made in version control for analysis code, datasets are only rarely versioned by default. DataLad⁴ combines git and git-annex,

^a Corresponding Author:
Benedikt Valerian Ehinger, PhD
benedikt.ehinger@vis.uni-stuttgart.de

allowing for efficient versioning of not only code, but also binary files. Crucially, adding new subjects, annotations, or corrections will leave a version trace, making the data collection robust to accidentally deleting or overwriting data and improving transparency of data provenance.⁵

Data Publishing: Our last step, which traditionally is often only taken after the data has been fully analyzed, is the long-term archiving and (access-restricted) publishing. Immediately publishing the raw data has the benefit of frontloading (and externalizing) the additional effort required to publish the data. Automating this is made possible by linking the versioned DataLad⁴ datasets directly with FAIR⁶ (findability, accessibility, interoperability, and reusability) enabling data repositories such as Dataverse,⁷ which will provide Digital Object Identifiers (DOI), customizable access controls, archival service, and further automatic versioning.

In our work, we addressed all four aspects in a single automated workflow. Our contributions are: 1) we discuss workflow to integrate community standards for data curation, automated publishing, and version control, with the potential to generalize across modalities and experimental paradigms, 2) we introduce LSLAutoBIDS, a Python package as an example implementation for such a workflow, using Lab Streaming Layer,² converting to BIDS, versioning via DataLad,⁴ and publishing via Dataverse.

RELATED WORKS

The pioneering work of Dobson et al.⁸ should be highlighted: They present ReproIn, a software package quite similar to ours, which converts DICOM (Digital Imaging and Communications in Medicine) data from a magnetic resonance (MR) scanner to BIDS, and finally saves it in a DataLad repository. Compared to our implementation, ReproIn is optimized to work with different MR scanners, whereas we use LSL to integrate time-series data more directly. Instead of `heudiconv`, we use `mne-lab`, `pyls1`, and `mne-bids` for the data curation stage. While ReproIn does not automatically link the DataLad repository to a Dataverse, this additional step could be readily implemented for ReproIn as well.

Another pioneering approach is taken at the Donders Institute for Brain and Cognition.⁹ There, all functional magnetic resonance imaging (fMRI) and magnetoencephalography (MEG) data are automatically archived in a data acquisition collection (DAC). Other data needs to be added manually to the DAC. While the DAC is centralized, it is typically only used internally, and later, a data sharing collection (DSC) is created, which can be publicly shared.

A different example for a data curation workflow, incorporating standardization and publishing, i.e., from raw data acquisition over standardization to analysis, is presented in Stawiski et al.¹⁰ In this project, they collected heterogeneous raw data of more than 100 participants in two clinics, transformed them to BIDS, and used SQLite to manage the resulting dataset. Even though they do not discuss versioning and publishing, it is easily conceivable how these steps could be added to their workflow.

The previously introduced Brain Imaging Data Structure,³ originally proposed for magnetic resonance imaging (MRI), is a community standard for data curation and sharing of brain data within research communities. This standard is designed using the FAIR principles,⁶ contributing to efficient scientific data curation and stewardship. Following the release of the BIDS standard, numerous extensions were developed, concerning integration of BIDS with different neuroimaging techniques like MEG,¹¹ intracranial electroencephalography (iEEG),¹² EEG,¹³ positron emission tomography (PET),¹⁴ and others. Data sharing and reuse of these BIDS-complaint datasets is even further simplified using open-source sharing platforms, for instance, OpenNeuro,¹⁵ which currently hosts more than 600 datasets in compliance with BIDS.

To convert datasets to be BIDS compliant, packages like `mne-bids`^{16,17} or `BIDSCoin`¹⁸ (using the `sova2coin` plugin) are essential to move and convert source data to BIDS-compliant datasets, and subsequently, the BIDS validator³ can be used to validate the data collection. To make the data conversion even more user-friendly for researchers, `BIDSCoin`¹⁸ offers a graphical user interface, making it accessible even to those without programming experience. As far as we know, there is no dedicated LSL-to-BIDS converter available.

Unlike software and code versioning, data versioning is more rarely practiced, given issues in practicality to version large binary files. Nevertheless, version control is an important aspect in data collection to ensure data traceability and an error-free workflow. DataLad,⁴ based on `git-annex`, nicely extends the capabilities of classical `git` versioning tools to large, binary datasets. DataLad is actively used in hundreds of studies and underlies all datasets on OpenNeuro.

LSLAUTOBIDS

LSLAutoBIDS is an open-source Python package developed and actively used by the Computational Cognitive Science Lab at the University of Stuttgart. It offers a modular and reproducible workflow tailored for studies using LSL based data acquisition, specifically targeting the integration of EEG and eye-tracking modalities.

In this setup, participant-level EEG data streams, and LSL streams (e.g., motion tracking, mobile eye-tracking), are recorded using the LSL protocol,¹⁹ which allows for sub-millisecond time-synchronization of heterogeneously sampled streams and results in an XDF container file, most commonly via the `LSLRecorder`²⁰ package. Project metadata such as authors, license, experimental description, but also Dataverse details are specified in one central configuration-toml file, which is then used to retrieve these project-specific metadata during the conversion process. The recorded raw data streams are then converted into the BIDS³ standard using the `mne-lab`,²¹ `pyls1`,¹⁹ `mne`,²² and `mne-bids`¹⁶ packages. The converted EEG data are then validated with the `BIDSValidator`, while non-BIDS files like project metadata, configuration files, and additional modalities such as eye-tracking data are listed in a `.bidsignore` file to exclude them

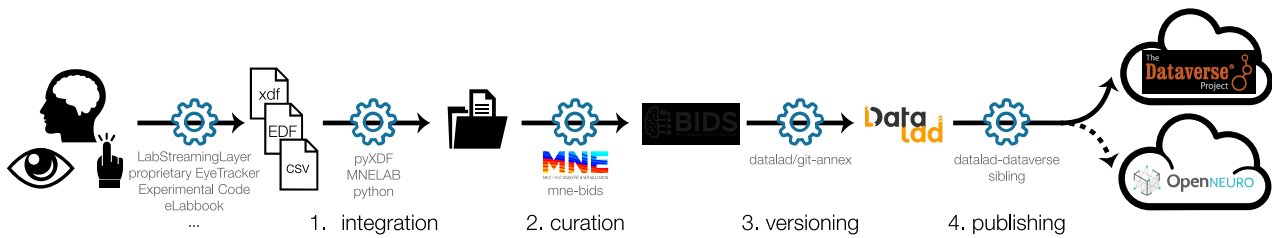


Figure 1. Flowchart of LSLAutoBIDS. We start with different data streams recorded using LabStreamingLayer and as other proprietary file formats. (1) We use LSLAutoBIDS to integrate data that is collected across different computers and synchronize the data streams temporally. (2) Next, we organize these data into the BIDS structure. (3) Using DataLad, we version them, and (4) finally upload them to an open repository like Dataverse or OpenNeuro.

from validation. Thus, we are compliant with the BIDSval-idator for our output. Once validated, the BIDS-compliant dataset is automatically deposited into a Dataverse repository, along with the experiment stimulus files and the raw source data.

In addition to LSL-based EEG data acquisition, LSLAutoBIDS supports the incorporation of non-LSL data. For example, in our use case, eye-tracking data are collected using the EyeLink 1000 Plus eye tracker simultaneously with EEG data collection, which produces proprietary EyeLink data format (EDF) files that cannot easily be streamed via LSL. Other examples are an electronic lab notebook record, log files of the experiment, and a compressed archive of the experimental code used in that individual session. LSLAutoBIDS accommodates these files as a secondary data modality currently implemented via user-defined folders, filename specifications, and regular expression matching, and is able to organize them within the appropriate BIDS subdirectories to publish them alongside EEG data.

Version control is integrated into the pipeline using DataLad,⁴ enabling precise tracking of all data and metadata changes across the research lifecycle, including modification or re-transformation of the individual files. DataLad is very powerful, but we use only the basic functionalities to simplify adoption for new users. We are not using the `datalad run` or `datalad rerun` capabilities, but are only concerned with the versioning of large files (`datalad save` and `datalad push`). The two minimal commands a user needs to know are `datalad clone` to checkout a repository and `datalad get` to download large files. This split between providing only symbolic links to all binary files first and only then explicitly downloading, e.g., a subset of the dataset, allows for easy exploration of terabyte-sized datasets.

After each recording session, the dataset is then uploaded to an initially private Dataverse repository, where it is persistently archived with the specified metadata under a DOI. After data collection is finished, the dataset can be publicly released and shared, potentially under a data sharing agreement.

The architecture of LSLAutoBIDS is deliberately designed to be extensible. While we developed this tool with our setup in mind (EyeLink 1000 Plus, ANT Neuro EEG, VR-setups), we will continuously add new devices and allow

for further customizability. Already now, additional files, such as behavioral data, audio recordings, or physiological signals, can be integrated by extending configuration templates and adding corresponding processing steps. This generalizability makes LSLAutoBIDS not just a tool for EEG and eye-tracking studies, but a proof of concept for broader multimodal data workflows in cognitive neuroscience.

How do workflows like LSLAutoBIDS address the examples raised in the introduction? 1) If data is immediately versioned and archived, then overwritten data can always be recovered. 2) As experimental files are included in the archive for each subject, it is easy to find any changes in the experimental code during data recording. 3) Even if a student leaves without further documenting the dataset, it is already in a cleaned up state, including metadata, greatly simplifying further use of the data. 4) Training of new users is greatly simplified, manual copying of data via USB sticks and other idiosyncratic workflows are no longer necessary.

CONCLUSION

Practicing open science by design and frontloading the conversion to a citable data publication efficiently addresses many hurdles researchers face in data collection and publishing. Not only is any recorded dataset immediately converted to BIDS, it also is archived, findable, shareable, backed up, and versioned.

Whether the final dataset is publicly available or only used internally (e.g., due to privacy issues in non-defaced MRIs), we think such a workflow will be helpful to many laboratories.

CODE AND DATA AVAILABILITY

The LSLAutoBIDS package is continuously developed and freely available: <https://github.com/s-ccs/LSLAutoBIDS> or via Zenodo <https://zenodo.org/records/15525822>.

CONFLICT OF INTEREST

The authors declare no conflicts of interest that may bias or could be perceived to bias this work.

FUNDING SOURCES

Funded by Deutsche Forschungsgemeinschaft (DFG, German Research Foundations) in the Emmy Noether Programme - Project-ID 538578433 - and Germanys Excellence Strategy EXC 2075 - 390740016.

Submitted: May 30, 2025 CDT. Accepted: January 28, 2026 CDT. Published: April 06, 2026 CDT.



This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CCBY-4.0). View this license's legal deed at <http://creativecommons.org/licenses/by/4.0> and legal code at <http://creativecommons.org/licenses/by/4.0/legalcode> for more information.

REFERENCES

1. National Academies of Sciences, Engineering, and Medicine, Policy and Global Affairs, Board on Research Data and Information, Committee on Toward an Open Science Enterprise. *Open Science by Design*. National Academies Press; 2018.
2. Kothe C, Shirazi SY, Stenner T, et al. The lab streaming layer for synchronized multimodal recording. Published online February 14, 2024. doi:[10.1101/2024.02.13.580071](https://doi.org/10.1101/2024.02.13.580071)
3. Gorgolewski KJ, Auer T, Calhoun VD, et al. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci Data*. 2016;3(1):160044. doi:[10.1038/sdata.2016.44](https://doi.org/10.1038/sdata.2016.44)
4. Halchenko YO, Meyer K, Poldrack B, et al. DataLad: Distributed system for joint management of code, data, and their relationship. *Journal of Open Source Software*. 2021;6(63):3262. doi:[10.21105/joss.03262](https://doi.org/10.21105/joss.03262)
5. Hanke M, Pestilli F, Wagner AS, Markiewicz CJ, Poline JB, Halchenko YO. In defense of decentralized research data management. *Neuroforum*. 2021;27(1):17-25. doi:[10.1515/nf-2020-0037](https://doi.org/10.1515/nf-2020-0037)
6. Wilkinson MD, Dumontier M, Aalbersberg IJJ, et al. The FAIR guiding principles for scientific data management and stewardship. *Sci Data*. 2016;3(1):160018. doi:[10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18)
7. Crosas M. The dataverse network: An open-source application for sharing, discovering and preserving data. *D-Lib Magazine*. 2011;17. doi:[10.1045/january2011-crosas](https://doi.org/10.1045/january2011-crosas)
8. Dobson J, Sackett T, Kodiweera C, et al. Presentations 2000 : ReproIn : Automatic generation of shareable , version-controlled BIDS datasets from MR scanners. In: ; 2018. Accessed May 8, 2025. <https://www.semanticscholar.org/paper/Presentations-2000-%3A-ReproIn-%3A-automatic-generation-Dobson-Sackett/f8f4a4fbadb6033b545b57bd51bf0356ac105566>
9. The RDM handbook — DCCN RDM handbook documentation. Accessed May 28, 2025. <https://rdm.dccn.nl/index.html>
10. Stawiski M, Bucciarelli V, Vogel D, Hemm S. Optimizing neuroscience data management by combining REDCap, BIDS and SQLite: A case study in deep brain stimulation. *Front Neuroinform*. 2024;18:1435971. doi:[10.3389/fninf.2024.1435971](https://doi.org/10.3389/fninf.2024.1435971)
11. Niso G, Gorgolewski KJ, Bock E, et al. MEG-BIDS, the brain imaging data structure extended to magnetoencephalography. *Sci Data*. 2018;5(1):180110. doi:[10.1038/sdata.2018.110](https://doi.org/10.1038/sdata.2018.110)
12. Holdgraf C, Appelhoff S, Bickel S, et al. iEEG-BIDS, extending the brain imaging data structure specification to human intracranial electrophysiology. *Sci Data*. 2019;6(1):102. doi:[10.1038/s41597-019-0105-7](https://doi.org/10.1038/s41597-019-0105-7)
13. Pernet CR, Appelhoff S, Gorgolewski KJ, et al. EEG-BIDS, an extension to the brain imaging data structure for electroencephalography. *Sci Data*. 2019;6(1):103. doi:[10.1038/s41597-019-0104-8](https://doi.org/10.1038/s41597-019-0104-8)
14. Norgaard M, Matheson GJ, Hansen HD, et al. PET-BIDS, an extension to the brain imaging data structure for positron emission tomography. *Sci Data*. 2022;9(1):65. doi:[10.1038/s41597-022-01164-1](https://doi.org/10.1038/s41597-022-01164-1)
15. Markiewicz CJ, Gorgolewski KJ, Feingold F, et al. OpenNeuro: An open resource for sharing of neuroimaging data. Published online June 29, 2021. doi:[10.1101/2021.06.28.450168](https://doi.org/10.1101/2021.06.28.450168)
16. Appelhoff S, Sanderson M, Brooks TL, et al. MNE-BIDS: Organizing electrophysiological data into the BIDS format and facilitating their analysis. *Journal of Open Source Software*. 2019;4(44):1896. doi:[10.21105/joss.01896](https://doi.org/10.21105/joss.01896)
17. Yorguin-Jose MR, Brayan-Andres HM, Steffen B, et al. SOVABIDS: EEG-to-BIDS conversion software focused on automation, reproducibility and interoperability. Published online March 20, 2025. Accessed September 12, 2025. <https://zenodo.org/records/15056451>
18. Zwiers MP, Moia S, Oostenveld R. BIDScoin: A user-friendly application to convert source data to brain imaging data structure. *Front Neuroinform*. 2022;15. doi:[10.3389/fninf.2021.770608](https://doi.org/10.3389/fninf.2021.770608)
19. Kothe C. Chkothepy/plysl. Published online May 18, 2025. Accessed May 30, 2025. <https://github.com/chkothepy/plysl>
20. Labstreaminglayer/app-LabRecorder. Published online November 20, 2025. Accessed November 21, 2025. <https://github.com/labstreaminglayer/App-LabRecorder>
21. Brunner C. MNELAB: A graphical user interface for MNE-python. *Journal of Open Source Software*. 2022;7(78):4650. doi:[10.21105/joss.04650](https://doi.org/10.21105/joss.04650)

22. Gramfort A, Luessi M, Larson E, et al. MEG and EEG data analysis with MNE-python. *Frontiers in Neuroscience*. 2013;7(267):1-13. doi:[10.3389/fnins.2013.00267](https://doi.org/10.3389/fnins.2013.00267)