The Journal of Open Source Software

Bidsme: expandable BIDS-ifier of brain imagery datasets

Nikita Beliy[1], **Camille Guillemin**[1], **Emeline Pommier**[2], **Grégory Hammad**[1], **and Christophe Phillips**[1]

1 GIGA - Cyclotron Research Centre in vivo imaging, University of Liege, Liege, Belgium **2** Télécom Physique Strasbourg, Université de Strasbourg, France

DOI: 10.21105/joss.05575

Software

- Review
- Repository
- Archive

Editor: Elizabeth DuPre

Reviewers:

- @PeerHerholz
- @TheChymera

Submitted: 06 June 2023
Published: 30 November 2023

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

Summary

The purpose of Bidsme is to organize a given medical image dataset following the "Brain Imaging Data Structure" (BIDS; Gorgolewski et al., 2016). Bidsme is an all-in-one organizer tool, that not only renames and re-structures the original data files, but also extracts and formats the necessary metadata. During the data organization, Bidsme provides the user with the full control over these processes, allowing the use of non-standard metadata and file names, as well as the addition of modalities not yet described by the BIDS. Instead of strictly imposing this structure, Bidsme allows the user to fully configure how the source dataset will be organized and what metadata will be included. Bidsme can be used both as Python package and command-line tool, and includes a tutorial with a test dataset.

Statement of need

For a long time, the neuroimaging community suffered from a lack of standardized file structure, formats and metadata conventions. Different laboratories, and even different research groups within a laboratory, had their own, idiosyncratic ways to organize their data. Analyses were then performed with scripts tailored to specific data structures, which made analysis workflow and data sharing unnecessarily complicated. Consequently, it was difficult to ensure results repeatability and validation.

The Brain Imaging Data Structure (BIDS; Gorgolewski et al., 2016) was introduced to change this situation; it imposes a standard data structure and defines the required associated metadata. Once a dataset follows the imposed structure, any analysis tool, supporting the BIDS, should be able to automatically find the needed data to process it.

Prior to the introduction of the BIDS, the main challenge was to adapt the processing scripts to the different (typically inconsistent) dataset structures. With the advent of the BIDS, the main challenge is now to "BIDS-ify" a given dataset, i.e., adapt a dataset to the BIDS. Not only the image files must be renamed according to the standard, but also the associated metadata must contain all expected values, using expected conventions and measurement units.

The challenge increases for datasets acquired prior to the introduction of BIDS, where often mandatory information may not even be present in the original data, or be encoded in a non-common way. Often, organizer tools, for example, heudiconv (Halchenko et al., 2023) or dcm2bids (Boré et al., 2023), provide access only to a limited subset of source file metadata. As a consequence, any additional metadata, introduced by an experimental acquisition protocol, will be ignored, or at least will be difficult to retrieve. Developers will do their best to incorporate the most popular protocols, but the most exotic ones will be probably overlooked. Other tools, like, e.g., Bidscoin (Zwiers et al., 2022) may rely on conventions used in the laboratory of the developers, and may be difficult to use in laboratories following different conventions.

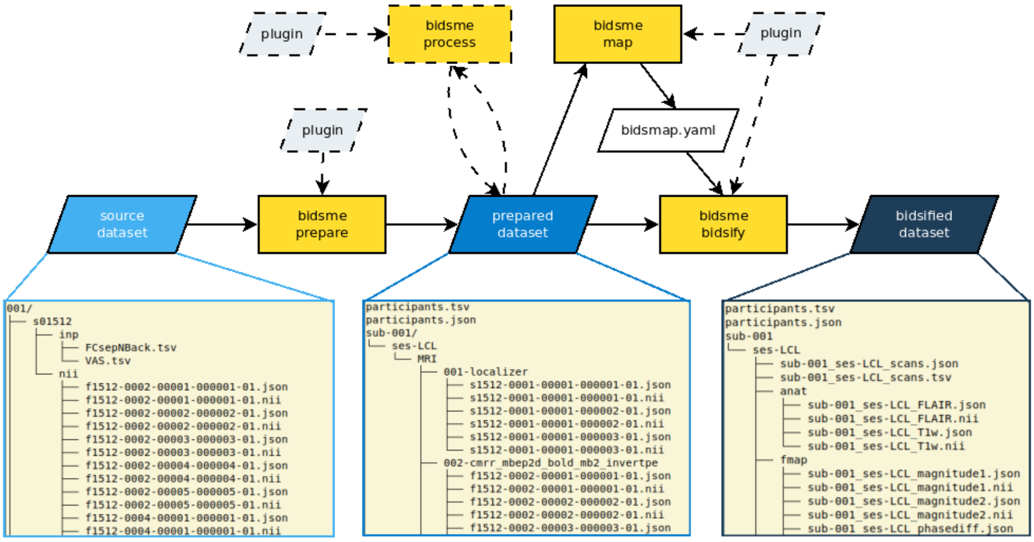
Multi-modal datasets provide an additional challenge: the organizer tools usually focuses only on one modality, for example pet2bids (Galassi et al., 2022) would BIDS-ify only PET data, forcing to use additional tools for EEG or MRI data.

The ideal organizer tool must be able to be flexibly adapted to any original data structure and to any reasonable laboratory practices. It must try to retrieve as much necessary metadata as possible, but also allow the user to add additional metadata. It must suggest to follow the current standard but allow deviations from it, e.g., when a given modality is not defined in BIDS.

These fundamental principles have been adopted as guidelines for the development of Bidsme. Bidsme gives full control of the data organization workflow to the user, imposing only the core of BIDS – the directory structure, the file naming style and some minimal metadata (those included in `participants.tsv` and `*_scans.tsv` tabular files). The actual names and set of entities are suggested to the user but are not imposed. Likewise, the user is free to add, remove or modify any automatically retrieved metadata.

Bidsme overview and usage

The BIDS-ification workflow using Bidsme is presented in Figure 1. It is organized into two main steps: the "preparation" and the "BIDS-ification".



```
001/
├── s01512
    ├── inp
    │   ├── FCsepNBack.tsv
    │   └── VAS.tsv
    └── nii
        ├── f1512-0002-00001-000001-01.json
        ├── f1512-0002-00001-000001-01.nii
        ├── f1512-0002-00002-000002-01.json
        ├── f1512-0002-00002-000002-01.nii
        ├── f1512-0002-00003-000003-01.json
        ├── f1512-0002-00003-000003-01.nii
        ├── f1512-0002-00004-000004-01.json
        ├── f1512-0002-00004-000004-01.nii
        ├── f1512-0002-00005-000005-01.json
        ├── f1512-0002-00005-000005-01.nii
        ├── f1512-0004-00001-000001-01.json
        ├── f1512-0004-00001-000001-01.nii
```

```
participants.tsv
participants.json
sub-001/
└── ses-LCL
    └── MRI
        ├── 001-localizer
        │   ├── s1512-0001-00001-000001-01.json
        │   ├── s1512-0001-00001-000001-01.nii
        │   ├── s1512-0001-00001-000002-01.json
        │   ├── s1512-0001-00001-000002-01.nii
        │   ├── s1512-0001-00001-000003-01.json
        │   └── s1512-0001-00001-000003-01.nii
        ├── 002-cmrr_mbep2d_bold_mb2_invertpe
        │   ├── f1512-0002-00001-000001-01.json
        │   ├── f1512-0002-00001-000001-01.nii
        │   ├── f1512-0002-00002-000002-01.json
        │   ├── f1512-0002-00002-000002-01.nii
        │   ├── f1512-0002-00003-000003-01.json
```

```
participants.tsv
participants.json
sub-001
└── ses-LCL
    ├── sub-001_ses-LCL_scans.json
    ├── sub-001_ses-LCL_scans.tsv
    ├── anat
    │   ├── sub-001_ses-LCL_FLAIR.json
    │   ├── sub-001_ses-LCL_FLAIR.nii
    │   ├── sub-001_ses-LCL_T1w.json
    │   └── sub-001_ses-LCL_T1w.nii
    ├── fmap
    │   ├── sub-001_ses-LCL_magnitude1.json
    │   ├── sub-001_ses-LCL_magnitude1.nii
    │   ├── sub-001_ses-LCL_magnitude2.json
    │   ├── sub-001_ses-LCL_magnitude2.nii
    │   ├── sub-001_ses-LCL_phasediff.json
```

Figure 1: Workflow of a BIDS-ification using Bidsme. Dashed arrows and boxes represent optional steps.

The preparation step organizes the dataset into BIDS-like structure, with separate directories for each subject and session. The standardized structure of the prepared dataset not only facilitates the subsequent BIDS-ification, but also helps with visual inspection of data integrity and provides an opportunity for intervention on the dataset (e.g., with removal of corrupted or failed data samples), while keeping the original dataset untouched. Several original datasets can be prepared into the same dataset, as long as there is no overlapping data. This can be useful when BIDS-ifying datasets with several modalities (MRI, EEG, PET).

The proper BIDS-ification step is then performed on the prepared dataset. Bidsme scans for all data and with the help of a configuration file, i.e., `bidsmap.yaml`, it identifies each data file, generates the new BIDS-compliant name, and exports the desired metadata into a sidecar json file.

The aforementioned `bidsmap.yaml` configuration file is the central piece of the BIDS-ification workflow. For each supported data format and data type (Table 1), it defines a set of criteria to identify a given modality and a set of rules to BIDS-ifiy the identified files. Identification criteria will match a given data file metadata with user-defined values, and in case of success, the BIDS-ification rules will be applied. The file naming rules are defined as a list of entities and corresponding values, which can be either provided by the user or retrieved dynamically from the metadata. The metadata rules in the sidecar json file are defined in the same way, allowing the user to automatically export specific values from the metadata, or provide a value manually in case of missing metadata.

In addition, Bidsme implements a flexible system of plugins that can be used at any stage. A plugin is a Python file with a set of user-implemented functions, which are executed at specific processing steps and gives access to the relevant data. Plugins allow users, for example, to rename subjects and sessions, to provide subject-related metadata, to incorporate auxiliary data (e.g., physiological) into the dataset, to add user calculated values into the metadata or to control data integrity. To help with the implementation of the plugins, Bidsme provides a template that describes the signature of the plugin functions, and a set of helper functions that implement common tasks like, for example, assembling a set 3D MRI images into one 4D MRI image, or extracting b-values from diffusion MRI image.

Supported data types and formats

Bidsme was developed to work with multiple data types and data formats. At the time of writing, Bidsme supports MRI, PET and EEG data types and a variety of data formats, summarized in Table 1.

Table 1: List of supported data formats together with Python3 modules

Modality	Data format	Module required
MRI	NIfTI	nibabel
	NIfTI+JSON	
	dicom	pydicom
PET	NIfTI	nibabel
	NIfTI+JSON	
	dicom	pydicom
	ECAT	nibabel
EEG	BrainVision	mne
	EDF/EDF+	mne

Bidsme was implemented using an object-oriented approach, where the interactions with the actual data files are implemented in base class in the Modules package. Every data type inherits from the base class and implements the BIDS requirements for that data type. Interactions with the data files are defined in a class which inherits from data type class and implements the metadata extraction, file validation, copy etc. Hence, it is relatively easy to expand Bidsme to support new data modalities and formats, simply by creating a new class and defining a handful of low-level functions. This allows users to quickly include additional data modalities, even if they are not currently supported by BIDS (for example, MEG or actigraphy data).

Acknowledgements

This work and Nikita Beliy were supported by the Fonds National de la Recherche Scientifique (F.R.S.-FNRS, Belgium) through Grant No. EOS 30446199 and the University of Liège. Camille Guillemin was supported by University of Liege. Christophe Phillips is supported by the Fonds National de la Recherche Scientifique (F.R.S.-FNRS, Belgium).

As Bidsme was developped based on Bidscoin(Zwiers et al., 2022) package, we would like to thanks its developpers, and in particular its lead developper Marcel Zweirs.

References

Boré, A., Guay, S., Bedetti, C., Meisler, S., & GuenTher, N. (2023). *Dcm2Bids* (Version 3.1.0). https://doi.org/10.5281/zenodo.8342572

Galassi, A., Norgaard, M., Thomas, A. G., Gonzalez-Escamilla, G., Svarer, C., Rorden, C., Matheson, G., Knudsen, G., Innis, R., Ganz-Benjaminsen, M., & Pernet, C. (2022). *PET2BIDS* (Version 1.0.0). https://github.com/openneuropet/PET2BIDS

Gorgolewski, K. J., Auer, T., Calhoun, V. D., Craddock, R. C., Das, S., Duff, E. P., Flandin, G., Ghosh, S. S., Glatard, T., Halchenko, Y. O., Handwerker, D. A., Hanke, M., Keator, D., Li, X., Michael, Z., Maumet, C., Nichols, B. N., Nichols, T. E., Pellman, J., … Poldrack, R. A. (2016). The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Scientific Data*, *3*(160044). https://doi.org/10.1038/sdata.2016.44

Halchenko, Y., Goncalves, M., Velasco, P., Oleggio Castello, M. V. di, Ghosh, S., Salo, T., Wodder II, J. T., Hanke, M., Sadil, P., Christian, H., Dae, M., Tilley, S., Kent, J., T, I., Brett, M., Amlien, I., Gorgolewski, C., Markiewicz, C., Lukas, D. C., … Michael. (2023). *nipy/heudiconv* (Version 1.0.0). https://doi.org/10.5281/zenodo.8364586

Zwiers, M. P., Moia, S., & Oostenveld, R. (2022). BIDScoin: A user-friendly application to convert source data to brain imaging data structure. *Frontiers in Neuroinformatics*, *15*. https://doi.org/10.3389/fninf.2021.770608