BIDS Extension Proposal 21 (BEP021): Common Electrophysiological Derivatives

version 0.0.1 (working copy)

Extension moderators/leads: Stefan Appelhoff <stefan.appelhoff@mailbox.org>, Cyril R Pernet <cyril.pernet@ed.ac.uk>, Robert Oostenveld <r.oostenveld@donders.ru.nl>, Teon L Brooks <teon.brooks@gmail.com>, Mainak Jas <mainakjas@gmail.com>

This document contains a draft of the Brain Imaging Data Structure standard extension. It is a community effort to define standards in data / metadata. This is a working document in draft stage and any comments are welcome.

This specification is an extension of BIDS and BIDS Common Derivatives proposal, and general principles are shared. The specification should work for many different settings and facilitate the integration with other imaging methods.

To see the original BIDS specification, see this link, for Common Derivatives, see this link. This document inherits all components of the original specification (e.g. how to store imaging data, events, stimuli and behavioral data), and should be seen as an extension of it, not a replacement.

Further links:

- BIDS Specification Discussion and Development
- BIDS Extension Proposal 6 (BEP006): Electroencephalography (EEG)
- BIDS Extension Proposal 10 (BEP010): intracranial Electroencephalography (iEEG)
- Survey about common Electrophysiological data formats
- Chat with moderators

1. Table of Contents

1. Open Questions

1. Should we have a github branch for examples?
2. What are common results of processing steps on EEG data? (feel free to add)

    a. First-order derivatives
        ~~i. Epochs~~
        ii. Mixing matrices
            ~~1. ICA~~
                ~~a. Matrix~~
                ~~b. number of channel x~~
            ~~2. Signal Space Projectors (SSP)~~
            3. Inverse operator
        iii. Quality control measures that relate to the raw data
        ~~iv. Continuous variable~~
            ~~1. ICA quality~~
            ~~2. Head position~~
    b. Second-order derivatives
        i. Time-frequency representation -- Frequency representation of Epochs

~~ii. Evoked -- (weighted) average of Epochs~~
iii. Source Time Series (stc) -- Source space epochs
iv. Source reconstruction
~~v. Are we including all the manufacturing info here as we do with the raw files?~~
~~c. How about files existing derived files that may not have this rich information?~~
~~d. Do we need new file formats to be defined?~~

3. Will we be making recommendations of derivative file format? It is likely a short list given that these files typically come from the software package.
4. Are we interested in describing all downstream derivatives, which includes source space and statistical results, or are we interested in just the sensor space derivatives?
~~5. How will the shared derivatives be used by people that bother to download them?~~
~~a. Speed up or skip processing steps~~
~~b. ...~~

2. Preliminary classifications

This specification specifies common electrophysiological derivatives for the Brain Imaging Data Structure (BIDS) Specification. Core principles of the BIDS specification and BIDS Common Derivatives are inherited.

Please refer to general BIDS specification document for context and general guidelines (definitions, units, directory structure, missing values, stimulus and event information, etc.).

2.1 Definitions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Throughout this protocol we use a list of terms. To avoid misunderstanding we clarify them here.

1. **Epoch** - a brief segment of data that often time synchronized to an event, also sometimes called a trial.
2. **Average** - data average over epochs.
3. **Annotation** - label assigned to a certain time segment and channel in the continuous data.

Metadata for derived files should be saved in .json files. Inheritance principle applies.
Example at the bottom:

```
<dataset>/
    derivatives/
        <pipeline-name>/
            sub-<label>/
                [ses-<session-label>]/
                    <meg|eeg|ieeg>/
                        <source_keywords>_desc-<label>_<source_suffix>.mat
                        <source_keywords>_desc-<label>_<source_suffix>.json
```

3. Data property annotations

Annotations can relate to:

1) events that happened during recording,
2) behaviour during recording, or
3) features of the data that was recorded.

These annotations could be manual or the output of an automated software. Annotations are associated with an onset, a duration and a label. Annotations may overlap, and they can be channel-specific and are stored in an `_annotation.tsv` file. Annotations only exist in relation to the dataset as it was originally created for, links to the dataset and how the annotations were created is stored in an accompanying <>_annotation.json file

Template:

```
<dataset>/
    derivatives/
        <pipeline-name>/
            sub-<label>/
                [ses-<session-label>]/
                    <meg|eeg|ieeg>/
                            <source_keywords>_annotations.tsv
                            <source_keywords>_annotations.json
```

3.1 Sidecar TSV document (*_annotations.tsv)

`onset`	REQUIRED. Onset in seconds.
`duration`	REQUIRED. Duration in seconds. 'n/a' if it affects entire recording.
`label`	REQUIRED. Label for the annotation.
`channels`	OPTIONAL. Label specific to a channel. If 'n/a', applies to all the channels. Else, it must be a list of channel names or labels for the time series (e.g., name of ICA component).
`absolute_time`	OPTIONAL. Onset in absolute datetime which should be expressed in the following format `2009-06-15T13:45:30` (year, month, day, hour (24h), minute, second; this is equivalent to the RFC3339 "`date-time`" format, time zone is always assumed as local time). If the corresponding unprocessed source file has been anonymized, using BIDS convention: i.e., time stamps have been shifted by a randomly chosen number of days, then the `absolute_time` must also be shifted by the same amount.

	The advantage of providing `absolute_time` is that it is unaffected by signal cropping or resampling.

Examples:

```
derivatives/
    hcp-preprocessed/
        sub-control01/
            ses-01/meg/sub-control01_ses-01_task-rest_run-01_annotations.tsv:
```

Example 1: annotation for events only

onset	duration	label
1.2	1.000	bad_eye_blink
8	10.000	bad_motion
10	3	radio_on

Example 2: annotation for events and entire channels

onset	duration	label	channels
0	n/a	bad_line_noise	['EEG002']
1.2	1.000	bad_eye_blink	n/a
8	10.000	bad_motion	['EEG001', 'EEG004']
10	3	radio_on	n/a

Example 3: annotation for ICA components

onset	duration	label	channels
0	n/a	bad_blink	['ICA001']
0	n/a	bad_heart_beat	['ICA015', 'ICA018']

For annotations common to the entire channel: for instance, if it is globally bad, use onset=0, and duration=n/a. In this case, the channel column MUST not be n/a.

Recommended labels

Here is a list of RECOMMENDED labels to use for certain types of annotations:

- **artifact_eye_blink:** artifactual data due to eye blink artifact
- **artifact_motion:** motion related artifact

- **artifact_flux_jump:** artifactual data due to flux jump
- **artifact_line_noise:** artifactual data due to line noise (e.g., 50Hz)
- **epilepsy_preictal:** onset of preictal state prior to onset of epilepy
- **epilepsy_seizure:** onset of epilepsy
- **epilepsy_postictal:** postictal seizure period
- **sleep_N1:** sleep stage N1
- **sleep_N2:** sleep stage N2
- **sleep_N3:** sleep stage N3
- **sleep_REM:** REM sleep
- **sleep_wake:** sleep stage awake

Users are free to add other annotations that are more task-specific and describe them in the `_annotations.json` file.

Sometimes, it is also interesting to have continuous annotations. An example is the likelihood of an ICA model or a metric to quantify head movement. For such annotations, one can augment a synthetic data channel in the raw file and add an entry in the `channels.tsv` by setting the channel type to ANNOTATION.

3.2 Sidecar JSON document (*_annotations.json)

`Description`	REQUIRED. Description of the annotations.
`IntendedFor`	REQUIRED. Link to the data file for which the annotations are intended. The link should be relative the BIDS source directory: e.g., "/sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.<ext>" For multi-file formats, the link should point to the header file, which would also be indicated when reading the data into a software package (e.g., for BrainVision files (.eeg, .vmrk, and .vhdr), the .vhdr should be indicated; or for EEGLAB files (.set and .fdt), the .set should be indicated.).
`Sources`	RECOMMENDED. Relative link or description of the file(s) used to make the annotations . e.g.: "/sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.<ext>" "Video not shared with data"
`Author`	RECOMMENDED. Which algorithm or person did the annotation.
`LabelDescription`	OPTIONAL. More extensive description of the labels in the _annotations.tsv file.

Example 1:

{

```
        "Description":      "Annotations as marked by visual inspection of the
                            data.",
        "IntendedFor":      "/sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.vhdr",
        "Sources":          "/sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.vhdr",
        "Author":           "Hans Berger",
        "LabelDescription": {
                            “tele_sync”: “synchronization of the data with our tele
                            equipment”,
                            “train_noise”: “typical noise pattern when the train is
                            running next to our lab.”,
                            “sleep_N1”: “Sleep stage N1”
                            }
}
```

Example 2:

```
{
        "Description":      "Annotations of electrodes as marked by visual
                            inspection of an operative photo.",
        "IntendedFor":      "/sub-01/ses-01/ieeg/sub-01_ses-01_task-01_ieeg.vhdr",
        "Sources":          "operative photo, not in data",
        "Author":           "Wilder Penfield",
        "LabelDescription": {
                            “on_blood_vessel”: “electrode is located on top of a
                            large blood vessel”
                            }
}
```

Example 3:

```
{
        "Description":      "Annotations from a video.",
        "IntendedFor":      "/sub-01/ses-01/ieeg/sub-01_ses-01_task-01_ieeg.vhdr",
        "Sources":          "synchronized video, not shared with data",
        "Author":           "Lab Technician",
        "LabelDescription": {
                            “eyes_closed”: “subject has the eyes closed”,
                            “eyes_open”: “subject has the eyes open”,
                            “watching_tv”: “subject is watching tv”
                            }
}
```

4. Common preprocessing steps

The source data can be preprocessed using temporal filtering, spatial projection or downsampling. The resulting data is still a continuous time series per channel type. To allow for the numerous ways in which these preprocessing steps could be applied, the `desc` keyword is provided as in the BEP003 common derivatives specification. The `desc`key-value pair is a unique identifier that MUST be accompanied by a sidecar json file containing the details of the parameter combinations used to generate the derived file.

Thus, the same file could be stored at different stages of preprocessing by simply using a different `desc`label. At the same time, the same data preprocessed using different parameters can als be given a different label.

Template:

```
<dataset>/
    derivatives/
       <pipeline-name>/
           sub-<label>/
                [ses-<session-label>]/
                     <meg|eeg|ieeg>/
                         <source_keywords>_desc-<label>_<meg|eeg|ieeg>.<ext>
                         <source_keywords>_desc-<label>_<meg|eeg|ieeg>.json
```

Sidecar JSON file (`*_desc-<label>_<eeg|meg|ieeg>.json`)

We provide a list of key-value pairs for some common operations that can be added *in addition* to the required and recommended fields in the <_meg|_eeg|_ieeg>.json file in the BIDS core specification.

Sidecar json files should all have the following fields as per the common BIDS derivatives.

`Description`	REQUIRED. Description of the annotations.
`Sources`	REQUIRED. Link to the file for which the annotations were originally created. The link should be relative the BIDS source directory: e.g. /sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.<ext>

4.1 Detrending

`Detrending`	RECOMMENDED. Indicate how detrending was done, e.g. 'constant', 'linear', 'none'.

4.2 Filtering

Template for filter names and properties stored in the json file in the `SoftwareFilters` field:

```
"SoftwareFilters": {
                    "<Filter1Name>": {"<Filter1Property1>": "<Filter1Setting1>"},
                    "<Filter2Name>": {"<Filter2Property1>": "<Filter2Setting1>",
                                      "<Filter2Property2>": "<Filter2Setting2>",},
                    },...
```

Filter names can be defined by the user, the following filter properties are recommended/optional.

FilterProperties

`FilterType`	RECOMMENDED. Could be 'low-pass butterworth', 'high-pass fir', 'band-pass iir', etc.
`HighCutoff`	RECOMMENDED. Cutoff at which higher frequencies are attenuated.

LowCutoff	RECOMMENDED. Cutoff at which lower frequencies are attenuated.
HighCutoffDefinition	RECOMMENDED. Can be “half-amplitude (-6dB)” or “half-power (-3dB)”
LowCutoffDefinition	RECOMMENDED. Can be “half-amplitude (-6dB)” or “half-power (-3dB)”
FilterOrder	RECOMMENDED. The order of the filter.
FilterLength	OPTIONAL. The length of the filter
RollOff	OPTIONAL. Could be “12 dB/Octave”, etc.
TransitionBandwidth	OPTIONAL. Could be “10 Hz”, etc.
PassbandRipple	OPTIONAL.
StopbandAttenuation	OPTIONAL.
FilterDelay	OPTIONAL. Could be “linear-phase” or “non-linear-phase”
FilterCausality	OPTIONAL. Could be “causal” or “non-causal”
Direction	RECOMMENDED. Could be ‘forward’, ‘backward’ or ‘bidirectional’
DirectionDescription	REQUIRED if “Direction” is “bidirectional”. Else OPTIONAL. Could be “cutoff frequencies and filter order apply to a single filter pass” or “cutoff frequencies and filter order apply to the final two-pass filter”
Purpose	OPTIONAL. ‘antialiasing’

4.3 Downsampling

SamplingFrequency	REQUIRED. Sampling frequency after downsampling. Anti aliasing filters used in this procedure should be specified as in filtering.
IsDownsampled	RECOMMENDED. Boolean ‘true’ if data are downsampled.

Example 1: sub-01_ses-01_task-rest_desc-filtered_meg.json

```
{
        "InstitutionName": "Stanford University",
        "InstitutionAddress: "450 Serra Mall, Stanford, CA 94305-2004, USA"
        "Manufacturer": "CTF",
        "ManufacturersModelName": "CTF-275",
        "DeviceSerialNumber": "11035"
        "SoftwareVersions": "Acq 5.4.2-linux-20070507",
        "PowerLineFrequency": 60,
        "SamplingFrequency": 2400,
```

```
        "MEGChannelCount": 270,
        "MEGREFChannelCount": 26,
        "EEGChannelCount": 0,
        "EOGChannelCount": 2,
        "ECGChannelCount": 1,
        "EMGChannelCount": 0,
        "DewarPosition": "upright",
        "SoftwareFilters": {
                        "SpatialCompensation":{"GradientOrder": "3rd"},
                        "LowPass":
                               {"LowCutoff": 100, "FilterType": "butterworth",
                                "FilterOrder": 4,"Direction":"bidirectional"},
                        },
        "RecordingDuration": 600,
        "RecordingType": "continuous",
        "EpochLength": 0,
        "TaskName": "rest",
        "ContinuousHeadLocalization": true,
        "HeadCoilFrequency": [1470,1530,1590],
        "DigitizedLandmarks": true,
        "DigitizedHeadPoints": true
}
```

4.4 Re-referencing and projection

Linear data projections such as re-referencing (EEG, iEEG) and 3rd order gradient projections (MEG) can be saved. A projection matrix should be saved in a .tsv file that specifies how the hardware electrode contacts map onto the channels in the data.

When data are re-referenced, the projection matrix should be saved as a .tsv file:

```
<dataset>/
    derivatives/
       <pipeline-name>/
           sub-<label>/
                [ses-<session-label>]/
                    <meg|eeg|ieeg>/
                        <source_keywords>_desc-<label>_proj.tsv
```

The first row and the first column should be matching and correspond to the names of the channels.

Example of the .tsv file:

Example 1: Average reference

output_name	O1	O2	P1
O1	0.66	-0.33	-0.33
O2	-0.33	0.66	-0.33
P1	-0.33	-0.33	0.66

Example 2: Bipolar reference or fancy stuff

output_name	O1	O2	P1
O1-O2	1	-1	0
O2-P1	0	1	-1
fancy	1	-0.5	-0.5

Each row in the matrix shows how the data in the channel should be weighted by different channels.

5. Segmented data

File format: mat? Or same as raw file format.

Template:

```
<dataset>/
    derivatives/
        <pipeline-name>/
            sub-<label>/
                [ses-<session-label>]/
                    <meg|eeg|ieeg>/

                    <source_keywords>_src-<source_suffix>_desc-<label><_epochs|ave
                    rage>.mat
                        <source_keywords>_desc-<label>_<epochs|average>.tsv
                        <source_keywords>_desc-<label>_<epochs|average>.json
                        <source_keywords>_desc-<label>_channels.tsv
```

Sidecar _<epochs|average>.json

Sidecar json files should all have the following fields as per the common BIDS derivatives.

`Description`	REQUIRED. Description of the annotations.
`Sources`	REQUIRED. Link to the file for which the annotations were originally created. The link should be relative the BIDS source directory: e.g. /sub-01/ses-01/eeg/sub-01_ses-01_task-01_eeg.<ext>

These are the fields *in addition* to the fields in source files.

`BaselineCorrection`	OPTIONAL. Boolean. If true, baseline correction has been performed.
`BaselineCorrectionMethod`	OPTIONAL. Freeform description for how baseline correction was performed: e.g. “mean subtraction”.

`BaselinePeriod`	OPTIONAL. List with two numbers indicating the start and end of the baseline period with respect to zero_time, e.g. `[-0.100, 0]`.

Sidecar _<epochs|average>.tsv

Data are epoched with respect to a certain event. This event happens at a certain time in the continuous data. This information is stored in the _events.tsv file. Here we define the terminology.

`duration`	REQUIRED. The total length of the epoch (in seconds).
`zero_time`	REQUIRED. Latency with respect to the start of the epoch at which time is zero (in seconds). This could be the onset of an event.
`trial_type`	OPTIONAL. Primary categorisation of each trial to identify them as instances of the experimental conditions. For example: for a response inhibition task, it could take on values "go" and "no-go" to refer to response initiation and response inhibition experimental conditions.
`latency`	OPTIONAL. The `event_time` with respect to the start of the recording of the continuous data. Not applicable to average data.
`number_averages`	OPTIONAL. If an average is stored, this column indicates the number of epochs that are averaged
`HED`	OPTIONAL. Hierarchical Event Descriptor (HED) tag. See Appendix III of main specification for details.

Example:

Latency	Condition	Duration	Baseline
5.2	Auditory/Right	1.0	0.2
8.2	Auditory/Left	1.0	0.2

The shape of the array in the mat file is number of epochs x number of data channels x max(duration)

`_channels.tsv` should also be included using the convention in the main BIDS spec.

FUTURE EXTENSIONS:

7. ICA decompositions

Following the convention in BEP012, ICA mixing matrices (typically topographic maps in the context of denoising) should be stored in tsv files and the source components in the same file format as the unprocessed time series. The mixing matrices SHOULD be stored in the same format as projection matrices, i.e., along with their channel names.
The source components are optional as they can be recovered by a dot product between the source time series and the unmixing matrix (inverse of the mixing matrix). Annotations describing the ICA components can be stored in a sidecar file.

```
<dataset>/
    derivatives/
       <pipeline-name>/
           sub-<label>/
                [ses-<session-label>]/
                     <meg|eeg|ieeg>/
                          <source_keywords>_desc-<label>_mixing.tsv
                          <source_keywords>_desc-<label>_annotations.tsv
                          [<source_keywords>_desc-<label>_components.<ext>]
```

It's common to do a PCA before the ICA ... how to integrate?

- *Include PCA rotation matrix as .tsv as well?*

8. Inverse modeling

A typical inverse modeling pipeline follows the steps shown here. While it does involve storing many intermediary files, the specification for some of these steps are sometimes part of another BIDS extension. The following steps are typically required in the pipeline

8.1 Coregistration

For information on storing coregistration transformation matrices, for example the head to device transformation: see BEP014

8.2 Freesurfer surfaces for BEM

For creating Boundary Element Model (BEM), typically surfaces segmented using Freesurfer are required. These could be the surface corresponding to the "outer skin", the "outer skull", and the "inner skull". For details on how these should be stored: see BEP011

8.4 Source estimate

9. Connectivity

See BEP017

Appendix I: Contributors

Feel free to add yourself after making a contribution.

Legend (source: https://github.com/kentcdodds/all-contributors)

Emoji	Represents
💬	Answering Questions (on the mailing list, NeuroStars, GitHub, or in person)
🐛	Bug reports
📝	Blogposts
💻	Code
📖	Documentation and specification
🎨	Design
💡	Examples
📋	Event Organizers
💵	Financial Support
🔍	Funding/Grant Finders
🤔	Ideas & Planning
🔌	Infrastructure (Hosting, Build-Tools, etc)
🔧	Plugin/utility libraries
👀	Reviewed Pull Requests
🔧	Tools
🌍	Translation
⚠	Tests
✔	Tutorials
📢	Talks
📹	Videos

- Stefan Appelhoff 📖🤔
- Teon Brooks 📖🤔
- Mainak Jas 📖🤔
- Robert Oostenveld 📖🤔

- Dora Hermes 📖🤔