This document is frozen while the conversion to Markdown is ongoing.

BIDS Derivatives

version RC1 (working copy)

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 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. 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.

RC1 marks freezing of addition of new data types and modalities. Subsequent RCs will only change how already included data and metadata types are represented or remove them altogether.

The work on this extension has been moved to GitHub. Please join the discusion at https://github.com/bids-standard/bids-specification/pull/109 and send PRs to https://github.com/chrisfilo/bids-specification/tree/enh/derivatives

1 Goals and scope of BIDS Derivatives

BIDS Derivatives is a representation of the outputs of common processing pipelines, capturing data and meta-data sufficient for a researcher to understand and (critically) reuse those outputs in subsequent processing. It is motivated by use cases where formalized machine-readable access to processed data enables higher level processing.

1.1 The relationship between BIDS-Derivatives datasets and BIDS-Raw datasets

The BIDS-Derivatives specification extends the BIDS-Raw specification. That is, **every BIDS-Derivative dataset must fully comply with the BIDS-Raw spec unless explicitly extended, negated or contradicted in this document or one of the domain-specific extensions**. In case of any apparent conflict between BIDS-Raw and BIDS-Derivatives, the latter always takes precedence. This inheritance principle (i.e., BIDS-Derivative inherits all BIDS-Raw rules) applies to *all* aspects of the spec. Some examples to illustrate:

- BIDS-Derivatives filenames must follow all BIDS-Raw file naming conventions. E.g., BOLD run 1 with task “rest” for subject “01” must begin with `”sub-01_task-rest_run-1”.`
- If a pipeline produces preprocessed versions of 2 different runs in each of 2 sessions for subject 01, the BIDS-Raw directory structure must be respected, producing paths like `“/deriv_directory/sub-01/ses-1/sub-01_ses-1_run-1…”`
- Every derivatives directory must include a `dataset_description.json` file at the root level.

2 General Principles

2.1 Filenaming conventions

- Filenames that are permissible under BIDS-Raw have a privileged status. Any *modification* of BIDS-Raw files must use a modified filename that does not conflict with the BIDS-Raw filename. Further, any files created as part of a BIDS-Derivative dataset must not match a permissible filename of a valid BIDS-Raw dataset. Stated equivalently, if any filename in a BIDS-Derivative dataset has a name permissible under BIDS-Raw, then that file must be an identical copy of that file in the associated BIDS-Raw dataset.

- Each Derivatives filename MUST be of the form: `<source_keywords>[_keyword-<value>]_<suffix>.<ext>`

    - When the derivatives chain involves outputs derived from a single BIDS-Raw input, `source_keywords` MUST be the entire filename from the BIDS-Raw input, including any optional components that were included in the BIDS-Raw filename, *except* that the `suffix` must always come at the end of the filename (and just before the extension). One exception to this rule is filename keywords that are no longer relevant (such as sub- in the case of group level derivatives). Depending on the nature of the derivative file, the `suffix` can either be the same as the original BIDS-Raw input if that suffix is still appropriate, or a new appropriate value selected from the controlled list.
    - When the derivatives output involves multiple BIDS-Raw inputs, `source_keywords` should preserve the elements in common between the input filenames in the process of creating a sensible `source_keywords`.

- The filename ends with a single `_suffix` tag and no other `_suffix` tags are allowed elsewhere in the filename. Only suffixes defined in this document (Table 1), the BIDS-Raw specification, or one of the other derivatives specifications are allowed. An arbitrary number of keyword-value pairs may be included, but all included keywords must be (a) either required or allowed by the BIDS-Raw specification (e.g., `sub`, `run`, `task`, etc.) or (b) explicitly allowed by this document or one of the derivatives extensions. In other words, users MUST NOT include arbitrary keyword-value pairs that are not explicitly allowed for the type of file in question.

- Unless specified otherwise, the `value` component of a `_keyword-value` pair is freeform and should be set appropriately by the user.
- Other than the privileged status of BIDS-Raw filenames, there is no prohibition against identical filenames in different BIDS-Derivatives directories, although users should be aware of the potential ambiguity this can create and use the sidecar JSON files to detail the specifics of individual files.
- When necessary to distinguish two files, the `_desc-<value>` keyword-value should be used. This includes the cases of needing to distinguish both differing inputs and differing outputs [e.g., `_desc-T1w` and `desc-T2w` to distinguish brain mask files derived from T1w and T2w images; or `_desc-sm4` and `_desc-sm8` to distinguish between outputs generated with two different levels of smoothing].

Data type	**Suffix**	**Description**	**Section**
anat func dwi	mask	Binary mask	7.2 Masks
xfm	xfm	Transform file	7.3 Mappings (transformation) between spaces
anat	dseg	Discrete segmentation file	7.4.1 Discrete Segmentations
anat	probseg	Probabilistic segmentation file	7.4.2 Probabilistic Segmentations
anat	wm	The gray matter / white matter border for the cortex surface	7.5 Reconstructed cortical surfaces
anat	smoothwm	The smoothed matter / white matter border surface	7.5 Reconstructed cortical surfaces
anat	pial	The gray matter / pial matter border surface	7.5 Reconstructed cortical surfaces
anat	midthickness	The midpoints between wm and pial surfaces	7.5 Reconstructed cortical surfaces
anat	inflated	An inflation of the midthickness surface (useful for visualization)	7.5 Reconstructed cortical surfaces
anat	sphere	The sphere (used for registration - see transforms for nomenclature)	7.5 Reconstructed cortical surfaces
anat	flat	The flattened surface (used for visualization)	7.5 Reconstructed cortical surfaces
anat	curv	Cortical surface curvature indices	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	thickness	Cortical thickness	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	area	Discretized surface area across regions	7.6 Surface-Mapped

			Anatomical Scalar Derivatives
anat	dist	Distance from a point	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	defects	Marked regions with surface defects	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	sulc	Sulcal depth	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	myelinmap	Myelin map calculated from T1w to T2 ratio	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	distortion	Distortion map calculated from a surface registration	7.6 Surface-Mapped Anatomical Scalar Derivatives
anat	morph	Structural statistics produced by segmentation routines	7.8 Morphometrics
func	mean	Mean across the temporal/4th dimension of the data	7.9 Functional derivatives maps
func	std	Standard deviation across the temporal/4th dimension of the data	7.9 Functional derivatives maps
func	tsnr	Temporal SNR (i.e. mean / std)	7.9 Functional derivatives maps
func	sfs	Signal fluctuation sensitivity	7.9 Functional derivatives maps
func	alff	Amplitude low frequency fluctuations	7.9 Functional derivatives maps
func	falff	Fractional amplitude of low frequency fluctuations	7.9 Functional derivatives maps
func	reho	Regional homogeneity (voxelwise only)	7.9 Functional derivatives maps
func	dcb, dcw	Voxelwise degree centrality binary and weighted	7.9 Functional derivatives maps
func	ecb, ecw	Voxelwise eigenvector centrality binary and weighted	7.9 Functional derivatives maps
func	lfcdb, lfcdw	Local functional connectivity density	7.9 Functional derivatives maps
func	vmhc	Voxel mirrored homotopic connectivity	7.9 Functional derivatives maps
func	regressors		7.11 General time series and regressors
func	timeseries		7.12 ROI-based time series extraction
func	motion		7.13 Motion-related regressors
func	physreg		7.14 Physiological regressors
func	outliers		7.15 Temporal outlier masks
func	mixing		7.17 Spatiotemporal decompositions

func	components		7.17 Spatiotemporal decompositions
func	decomposition		7.17 Spatiotemporal decompositions
dwi	diffmodel		7.19 Diffusion Models (and input parameters)
dwi	FA	Fractional Anisotropy	7.20.1 Scalar maps
	MD	Mean diffusivity (also called apparent diffusion coefficient, ADC)	7.20.1 Scalar maps
dwi	AD	Axial Diffusivity (also called parallel diffusivity)	7.20.1 Scalar maps
dwi	RD	Radial Diffusivity (also called perpendicular diffusivity)	7.20.1 Scalar maps
dwi	MODE	Mode of the tensor	7.20.1 Scalar maps
dwi	LINEARITY	Tensor linearity (Westin 1997)	7.20.1 Scalar maps
dwi	PLANARITY	Tensor planarity (Westin 1997)	7.20.1 Scalar maps
dwi	SPHERICITY	Tensor sphericity (Westin 1997)	7.20.1 Scalar maps
dwi	MK	Mean kurtosis	7.20.1 Scalar maps
dwi	RK	Radial kurtosis	7.20.1 Scalar maps
dwi	AK	Axial kurtosis	7.20.1 Scalar maps
dwi	ICVF	Intracellular volume fraction. This can be calculated from different models, so can have slightly different interpretation.	7.20.1 Scalar maps
dwi	DECFA	Directionally-encoded color (DEC) FA	7.20.1 Scalar maps
dwi	ODI	Orientation dispersion index	7.20.1 Scalar maps
dwi	ISOVF	Isotropic volume fraction	7.20.1 Scalar maps
dwi	TDI	Track Density Imaging	7.20.1 Scalar maps
dwi	GFA	Generalized Fractional Anisotropy	7.20.1 Scalar maps
dwi	AFD	Apparent Fiber Density	7.20.1 Scalar maps
dwi	INDEX	FSL BedpostX maps (FSUM, F_i, D and DSTD)	7.20.1 Scalar maps
dwi	RTPP	Return to the plane probability	7.20.1 Scalar maps
dwi	RTAP	Return to axis probability	7.20.1 Scalar maps
dwi	Fixel	Fixel (Fiber bundle element) images. Different voxels can have different numbers of fixels, and therefore a sparse file format is required. This is not specific to Fibre Density derived via CSD. Several diffusion models provide fixel-specific measures (e.g. Ball and stick, CHARMED, CuspMFM, Myelin content).	7.20.2 Vector-valued maps
dwi	ODF	Orientation distribution function	7.20.2 Vector-valued maps
dwi	PDF	Diffusion propagator	7.20.2 Vector-valued maps
dwi	FOD	Fiber orientation distribution (Sometimes known as fODF)	7.20.2 Vector-valued maps
dwi	EVECS	Eigenvectors of a model that has eigenvectors (such as DTI)	7.20.2 Vector-valued maps
dwi	PEAKS	Directions of ODF maxima on the sphere. It is a vector	7.20.2 Vector-valued maps
dwi	tractography		7.21 Describing tractography

Table 1.

2.2 Metadata conventions

- Unless specified otherwise, individual sidecar JSON files and all metadata fields within are optional. However, the appropriate use of these files and pertinent fields is very valuable and thus encouraged. Moreover, for some types of files, there may be one or more required metadata fields, in which case at least one metadata file containing that field must be located somewhere within the file's hierarchy (per the Inheritance Principle defined in BIDS-Raw).

- When chaining derivative pipelines, any JSON fields that were specified as mandatory in the input files should be propagated forward in the output file's JSON provided they remain valid. Non-required JSON fields can be propagated, and are highly useful, but it is the pipeline's responsibility to ensure that the values are still relevant and appropriate to the type of output data.

- Lack of a given metadata field MUST NOT be interpreted as an indication that a given field is set to false.

3 Storage of derived datasets

Derivatives can be stored/distributed in two ways:

1. Under a `derivatives/` subfolder in the root of the BIDS-Raw dataset folder to make a clear distinction between raw data and results of data processing.

   Each pipeline has a dedicated directory under which it stores all of its outputs. There is no restriction on the directory name; however, it is RECOMMENDED to use the format <pipeline>-<variant> in cases where it is anticipated that the same pipeline will output more than one variant (e.g., "AFNI-blurring", "AFNI-noblurring", etc.).

   For example:

   `<dataset>/derivatives/fmripreprocess_01/sub-0001`

   `<dataset>/derivatives/spm/sub-0001`

   `<dataset>/derivatives/vbm/sub-0001`

2. As a standalone dataset independent of the source BIDS-Raw or BIDS-Derived dataset. This way of specifying derivatives is particularly useful when the original BIDS-Raw is provided with read-only access, and for publishing derivatives as independent bodies of work, or for describing derivatives that were created from more than one dataset.

The rest of the document assumes Case 1, but Case 2 applies after removing `/derivatives/<pipeline_name>` from template names.

4 Analysis levels

4.1 Compliance with BIDS-Raw

As noted above, the BIDS-Derivative standard extends the BIDS-Raw standard. Consequently, files should be organized to comply with BIDS-Raw to the full extent possible (i.e., unless explicitly contradicted by BIDS-Derivatives). Any subject-specific derivatives should be housed within each subject's directory; if session-specific derivatives are generated, they should be deposited under a session subdirectory within the corresponding subject directory; and so on.

In addition to the structure defined in the BIDS-Raw specification, the BIDS-Derivatives specification provides additional flexibility intended to account for derivatives that do not fit within the standard BIDS-Raw project layout. In particular, pipelines may generate derivatives that do not strictly belong to any single participant. To write group-level results, the `group/` folder under `derivatives/<pipeline_name>/` can be used:

```
<dataset>/derivatives/<pipeline_name>/group/
```

Derived dataset and pipeline description

In keeping with the BIDS-Raw standard, a `dataset_description.json` file MUST be found at the top level of the particular pipeline:

```
<dataset>/derivatives/<pipeline_name>/dataset_description.json
```

Keys. The dataset_description.json file in a BIDS-Derivatives dataset can contain arbitrary key-value pairs that encode metadata relevant to understanding the structure and contents of the directory. Although there is no strictly controlled vocabulary, users are strongly encouraged to use keys defined in the Raw-BIDS spec when available. In addition, the BIDS-Derivatives spec introduces a number of other required or recommended keys:

Key name	Description
PipelineDescription.Name	REQUIRED
PipelineDescription.Version	
PipelineDescription.CodeURL	
PipelineDescription.DockerHubContainerTag	
PipelineDescription.SingularityContainerURL	

`PipelineDescription.SingularityContainerVersion`	
`SourceDatasets`	A list of objects specifying the locations and relevant attributes of all source datasets. Valid fields in each object include `url`, `doi`, and `version`.

5 Modifications to Appendix VIII of the main specification

5.1 Definitions

5.1.1 Coordinate System

A coordinate system specifies the information necessary to interpret the coordinates associated with the data. A coordinate system can be described by its topology (cartesian, spherical, polar, etc.,.), dimensionality, and units of measurement along each dimension. Every device that makes spatial measurements or has sensors described in a space has a coordinate system (e.g., an MR scanner, an MEG scanner, an EEG cap, Polhemus digitizer). The data output by most of these devices describe the number of dimensions, the units along each dimension, and in some cases the details of origin (for example, time).

5.1.2 Space, atlas and map

A *space*|*atlas*|*map* is embedded in a coordinate system to ascribe meaning to the coordinates. For instance, *Scanner*, *Talairach88*, *MNI*, *FS* are different coordinate systems, while the *Talairach atlas*, *MNI152*, and *fsaverage* ascribe meaning to the coordinates in terms of brain anatomy. As an example to distinguish coordinate systems and spaces/atlas/maps, a 4D fMRI timeseries acquired without motion correction has individual time points all in the same coordinate system, but the atlas/map is different (due to motion) in different time points.

5.1.3 Landmarks

Landmarks are properties of the object being measured and can exist in any coordinate system in which the object is placed/measured. They can be relevant points of the imaged object's structure (i.e. *anatomical landmarks*) such as the anterior commissure in a brain. They can also be devices that generate a localizable signal that are placed in the intended position before scanning (i.e. *markers*). Positions of landmarks are identified via manual or automated algorithms that detect these landmarks in (e.g., MR) or from (e.g., MEG) the acquired data or via explicit encoding of position (e.g., EEG sensors, headshape/coil digitization).

These landmarks can be coordinates in a file or voxel/pixel intensity relations in image files.This information (coordinates or intensity relations) representing landmarks of the same or similar object (e.g., a template brain) in different coordinate systems can be used to transform the object between coordinate systems.

6 Metadata organization (file level)

Each derivative file SHOULD be described by a JSON file provided as a sidecar or higher up in the hierarchy of the derived dataset (according to Inheritance Principle - section 3.5 of the main specification) unless a particular derivative includes REQUIRED metadata fields in which case a JSON file is also REQUIRED. Each derivative type defines their own set of fields, but all of them share the following (non-required) ones:

Field name	Required	Description
`Description`	RECOMMENDED.	Freeform natural language description of the nature of the file.
`Sources`	OPTIONAL.	A list of paths relative to dataset root pointing to the file(s) that were directly used in the creation of this derivative. For example in a chain of A->B->C, "C" should only list "B" as Sources, and "B" should only list "A" as Sources. However in case X and Y jointly contribute to Z, then "Z" should list "X" and "Y" as Sources.
`RawSources`	OPTIONAL.	A list of paths relative to dataset root pointing to the BIDS-Raw file (s) that were used in the creation of this derivative.
`CoordinateSystem`	Mandatory if no implicit coordinate system.	Key indicates the coordinate system associated with the File. The coordinate system can be implicit to the File, for instance when data are images stored in NIfTI format. Can be a list. See Table below for list of allowed systems.
`ReferenceMap`	Required when coordinate system is Aligned.	Key indicates the reference atlas or map that the File is aligned to. See table below for list of common spaces.
`NonstandardReference`	Required when a non standard template or space is used. (e.g., a custom template in MNI305 space).	A path to a file that was used as, or can be used as, a reference image for determining the coordinate space of this file. If Space is a list, Space reference must also be a list.
`ReferenceIndex`	Required when an index into a	Used to index into a 4D spatial-reference

	4D (ReferenceMap|Nonstandard Reference) file is used.	file.
`TransformFile`	OPTIONAL.	A path to the file used to transform the source into this file.

6.1 CoordinateSystem reserved keys:

Name	Description
`Device`	The coordinate system of the device used to acquire the data.
`Aligned`	The coordinate system is specified by a target space (e.g., Talairach88, MNI305, etc.,.). See the Space keyword for details of targets.
`Custom`	A custom coordinate system that is not in alignment (dimensions, axis orientation, unit) with any device coordinate system.

6.2 ReferenceMap reserved keys:

`orig`	A (potentially unique) per-image space. Useful for describing the source of transforms from an input image to a target space.
`custom`	This can be used to point to the non-standard space of a file. This should be used only if the reference file is not in any of the other spaces.

6.3 Special keys for indicating a standard Atlas as ReferenceMap:

`FS305`	FreeSurfer variant of the MNI305 space
`MNI152Lin`	Also known as ICBM (version with linear coregistration) http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152Lin
`MNI152NLin6[Sym|Asym]`	Also known as ICBM 6th generation (non-linear coregistration). Used by SPM99 - SPM8 and FSL (MNI152NLin6Sym). http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin6
`MNI152NLin2009[a-c][Sym|Asym]`	Also known as ICBM (non-linear coregistration with 40 iterations, released in 2009). It comes in either three different flavours each in symmetric or asymmetric version. http://www.bic.mni.mcgill.ca/ServicesAtlases/ICBM152NLin2009
`MNIColin27`	Average of 27 T1 scans of a single subject http://www.bic.mni.mcgill.ca/ServicesAtlases/Colin27Highres. Also has variants
`MNI305`	Also known as avg305
`NIHPD`	Pediatric templates generated from the NIHPD sample. Available for different age groups (4.5–18.5 y.o., 4.5–8.5 y.o., 7–11 y.o., 7.5–13.5 y.o., 10–14 y.o., 13–18.5 y.o. This template also comes in either -symmetric or -asymmetric flavor.

	http://www.bic.mni.mcgill.ca/ServicesAtlases/NIHPD-obj1
Talairach88	http://www.talairach.org/ Mostly not used anymore. But the original Talairach transform was a piece affine transform of 12 boxes.
OASIS30Atropos	https://figshare.com/articles/ANTs_ANTsR_Brain_Templates/915436 http://www.mindboggle.info/data.html
ICBM452AirSpace	Reference space defined by the "average of 452 T1-weighted MRIs of normal young adult brains" with "linear transforms of the subjects into the atlas space using a 12-parameter affine transformation" http://www.loni.usc.edu/ICBM/Downloads/Downloads_452T1.shtml
ICBM452Warp5Space	Reference space defined by the "average of 452 T1-weighted MRIs of normal young adult brains" "based on a 5th order polynomial transformation into the atlas space" http://www.loni.usc.edu/ICBM/Downloads/Downloads_452T1.shtml
IXI549Space	Reference space defined by the average of the "549 [...] subjects from the IXI dataset" linearly transformed to ICBM MNI 452.Used by SPM12. http://www.brain-development.org/
fsnative	Images were sampled to the FreeSurfer surface reconstructed from the subject's T1w image
fsaverage[3|4|5|6|sym]	Images were sampled to the FreeSurfer surface reconstructed from the subject's T1w image, and registered to an fsaverage template
fsLR[164k|32k]	Images were sampled to the 164k or 32k fsaverage_LR surface reconstructed from the T1w image (high resolution space of the HCPPipelines)
UNCInfant[0|1|2]V[21|22|23]	Infant Brain Atlases from Neonates to 1- and 2-year-olds. https://www.nitrc.org/projects/pediatricatlas

6.4 Example sidecar files:

For a NIFTI file (Single coordinate system), one could have registered the File on to the standard MNI305 template. The way to write the metadata of such File is:

```
{
"Space": "MNI",
"TransformFile": "/path/to/xfm",
"ReferenceMap": "MNI305"
}
```

However, it could also be the case that a nonstandard derivative of MNI305 was used as standard space for the File. That can be written as follows:

```
{

"Space": "MNI",

"TransformFile": "/path/to/xfm",

"NonstandardReference": "uri or path to file"

}
```

Some derivatives such as CIFTI Files allow for multiple coordinate systems. Such possibility is enabled by using lists of spaces and references:

```
{

"Space": ["MNI", "fsLR"],

"TransformFile": "/path/to/xfm",

"ReferenceMap": ["MNI305", "fsLR32k"]

}
```

Differing references of the same spaces with respect to the above example can be expressed as follows:

```
{

"Space": ["MNI", "fsLR"],

"TransformFile": "/path/to/xfm",

"ReferenceMap": ["MNI152Lin", "fsLR164k"]

}
```

7 Datatypes

7.1 Processed, coregistered and/or resampled volumes

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              func|anat|dwi/
                     <source_keywords>[_space-<space>][_desc-<label>]_<suffix>.<ext>
```

Processing in this context means transformations of data that does not change the number of dimensions of the input and are not explicitly covered by other Datatypes in the specification. Examples:

- Motion-corrected, temporally denoised, and transformed to MNI space bold files.
- Inhomogeneity corrected and skull stripped T1w files.
- Motion-corrected DWI files.

The `space` keyword denotes the space of the file with a controlled vocabulary of values - see BEP014 for details. The `desc` keyword is a general purpose field with freeform values. To distinguish between multiple different versions of processing for the same input data the `desc` keyword should be used. **Note that even though `space` and `desc` are optional at least one of them needs to be defined to avoid name conflict with the BIDS-Raw file.**

Examples:

```
pipeline1/sub-001/
       func/
              sub-001_task-rest_run-1_space-MNI305_bold.nii.gz
              sub-001_task-rest_run-1_space-MNI305_bold.json

pipeline1/sub-001/
       func/
              sub-001_task-rest_run-1_desc-MC_bold.nii.gz
              sub-001_task-rest_run-1_desc-MC_bold.json
              OR
              sub-001_task-rest_run-1_desc-fmriprep_bold.nii.gz
              sub-001_task-rest_run-1_desc-fmriprep_bold.json
```

All REQUIRED metadata fields coming from a derivative file's source file(s) MUST be propagated to the JSON description of the derivative **unless the processing makes them invalid** (e.g., if a source 4D image is averaged to create a single static volume, a `SamplingFrequency` property would no longer be relevant). In addition, all processed files include the following metadata JSON fields:

Key	Description	Value for unprocessed data
`Space`	REQUIRED. Keyword denoting which space the processed file is in see BEP014 for details.	'orig'
`SkullStripped`	REQUIRED. Whether the volume was skull stripped (non-brain voxels set to zero) or not.	false

7.2 Masks

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              func|anat|dwi/
                     <source_keywords>[_space-<space>][_desc-<label>]_mask.nii.gz
```

A binary (1 - inside, 0 outside) mask in the space defined by <space> (see Table targets/spaces). By default (i.e., if no transformation has taken place) the value of `space` should be set to “orig”.

Key	Description	
`RawSources`	Elevated from OPTIONAL to REQUIRED	
`Space`	REQUIRED. Keyword denoting which space the processed file is in see [BEP014](BEP014) for details.	‘orig’
`Type`	RECOMMENDED. Short identifier of the mask. Restricted values: - “Brain” - brain mask - “Lesion” - lesion mask - “Face” - face mask - “ROI” - ROI mask	

Examples:

```
func_loc/
   sub-001/
       func/
          sub-001_task-rest_run-1_space-MNI305_desc-PFC_mask.nii.gz

manual_masks/
     sub-001/
         anat/
            sub-001_T1w_label-desc_mask.nii.gz
```

7.3 Mappings (transformation) between spaces

7.3.1 Definitions and introduction

A transformation file encapsulates a mathematical operation on the coordinates of those objects susceptible of being spatially transformed (i.e. images, surfaces, or points). For example, image analysis workflows that operate on a common standardized space for all the subjects will estimate one transform per subject to map the information (e.g. BOLD signal, the pial surface mesh, etc) from the individual to the common space. Although a transform is strictly just a coordinates mapping operation, it is generally understood by developers of registration algorithms that *applying a transform on an image* implicitly includes two operations (see below), which can be confused with *applying a transform on a surface of points* that just finds new coordinates through the transformation, for each vertex of the surface.

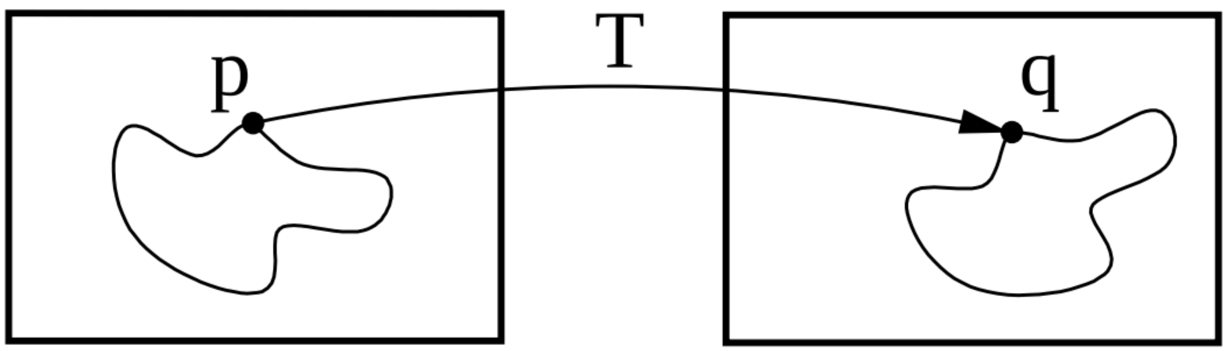

Figure 1. A transformation maps corresponding features or landmarks from one object to another. (Figure reproduced from the ITK Software Guide, Book 2, Figure 3.1).

Applying a transform on an image. Transformations allow to project information *from* one image (also called "moving" image) on *to* another image ("reference" image). As introduced before, this projection involves two operations: 1) the mapping of all points in the reference grid to the coordinates in the moving image; and 2) interpolating data on the mapped moving coordinates (as they are unlikely to fall on points of the moving image grid).

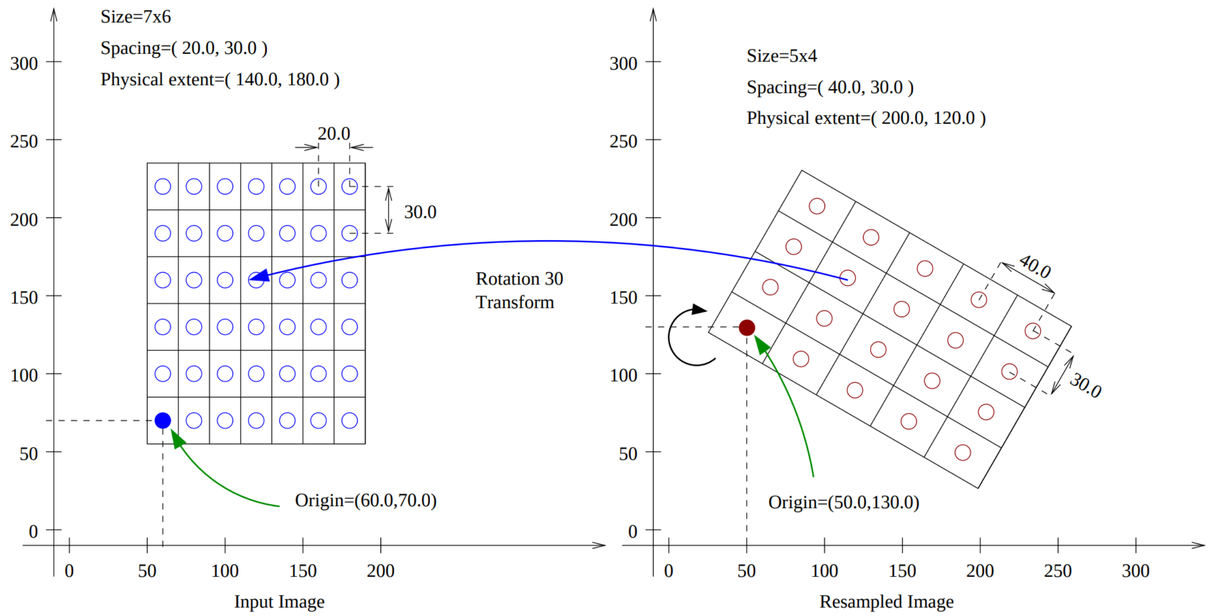

Figure 2. Example of applying a transform on an image, in particular a rotation of 30 degrees clockwise. The image on the right-hand side is the *resampled image* in the reference image's space. To generate such result, all the coordinates of all pixels in the reference (red circles) are iterated and corresponding coordinates in the moving (*input*) image at the left side are computed through the transform (blue arrow). Since the mapped coordinates will not likely fall at the center of a pixels in the input image (blue circles), then a value for the image is interpolated from the neighboring pixels before it can be assigned to the corresponding resampled pixel (red circle at the source of the blue arrow). This figure has been reproduced from the ITK Software Guide, Book 2, Figure 2.51.

7.3.2 Naming transform files - BIDS entities

Transform-files are labeled with the `from-<space>` and `to-<space>` BIDS entities. An additional entity, `mode-<mode_label>`, indicates what kinds of objects the transform operates to, in order to allow accommodating the widely extended use case of *applying a transform on an image* that implicitly involves two operations.

BIDS entities specific of the transform file template:

`from-<space_label>`	(MANDATORY) Source space in the information flow when applying a transform.
`to-<space_label>`	(MANDATORY) Destination space in the information flow when applying a transform.
`mode-<mode_label>`	(MANDATORY) Acceptable `mode_label` values are `image`, `points`, `surface` and `sphere`. Indicates how the `from` and `to` labels should be interpreted: as pure coordinates mapping (`points`, `surface`) or as a coordinates mapping internal to a resampling process (`image`, `sphere`).

7.3.3 Transform file mode

Due to the different ways in which *apply a transform* is understood (described above), the mode-<mode_label> defines such convention. Sometimes, the transform needed for one use case can be calculated from another transform, however, sometimes these type of calculations are not appropriate. For example, a mode-points transform can sometimes be calculated as the inverse of a mode-image transform, but only if the mode-image transform is “Invertible”. To clarify the acceptable use of a particular transform file, the mode key value pair has been introduced.

<mode_label>	**description**
points|surface	The File maps individual coordinates (i.e. location of implanted electrodes, sets of coordinates for a surface mesh) given in the from-<space_label> space into the to-<space_label> space.
image|sphere	Indicates that this File is intended to apply a transform on discretely sampled data defined on the from-<space_label> space (e.g. 2D image, 3D image, sphere, etc.) and resample such those data on a discretely sampled to-<space_label> space.

Please note that, given these definitions, a the transform file from-A_to-B_mode-points is completely equivalent to from-B_to-A_mode-image.

7.3.4 Examples of usage of various neuroimaging tools that calculate transforms:

	from	**to**	**mode**	Example command line to moving the same image from “MNI” to “bold” space
FS	mov	target		
FSL (flirt)	in	ref	image	flirt -in MNItemplate.nii.gz -ref <source_keywords>_bold.nii.gz -omat <source_keywords>_from-MNI_to-BOLD_mode-image_xfm.mat -dof 6
FSL (fnirt)	in	ref	image	fnirt -in=MNItemplate.nii.gz -ref=<source_keywords>_bold.nii.gz --cout=<source_keywords>_from-MNI_to-BOLD_mode-image_xfm.mat
ANTs	moving	fixed		antsRegistration --dimensionality 3 \ --output <source_keywords>_from-MNI_to-BOLD_mode-image_xfm] \ --metric MI[<source_keywords>_bold.nii.gz,MNItemplate.nii.gz,...] \ ...

7.3.5 Transform files template

Since there is no standard file-format for transform objects, any transform format will be accepted until a common format for spatial transforms is designed and accepted. In the following, we will use the .h5 extension to refer to the File storing the actual transform, and such extension can be replaced by that generated with any registration software (e.g. .nii[.gz],

`.mat`, `.lta`, `.xfm`, `.h5`). Transforms are a new file-type `xfm`, and thus they will use such key as file suffix and will be stored under a `xfm/` folder.

A folder under the participant level will contain all transforms from/to those subject spaces:

```
<pipeline_name>/
  sub-<participant_label>/
    xfm/
      <source_keywords>_from-MNI_to-BOLD_mode-image_xfm.h5
      <source_keywords>_from-MNI_to-BOLD_mode-image_xfm.json
      <source_keywords>_from-BOLD_to-MNI_mode-points_xfm.h5
      <source_keywords>_from-BOLD_to-MNI_mode-points_xfm.json
      <source_keywords>_from-MNI_to-BOLD_mode-points_xfm.h5
      <source_keywords>_from-MNI_to-BOLD_mode-points_xfm.json
```

where `<source_keywords>_from-BOLD_to-MNI_mode-points_xfm.h5` is a (soft)link to `<source_keywords>_from-MNI_to-BOLD_mode-image_xfm.h5`, to encode the said equivalence between both.

For example, for a participant with label 001 with both BOLD and T1w images:

```
sub-001/
       xfm/
           sub-001_task-rest_run-01_from-orig_to-index10_mode-points_xfm.h5
           sub-001_from-T1w_to-meanBOLD_mode-points_xfm.h5
           sub-001_from-T1w_to-meanBOLD_mode-points_xfm.json
           sub-001_from-MNI_to-T1w_mode-image_xfm.h5
           sub-001_from-MNI_to-T1w_mode-image_xfm.json
```

Example JSON sidecar for transform files:

```
{
  "Type": ["rigid", "displacementfield"],
  "Multiplexed": [true, false],
  "Software": "antsRegistration",
  "SoftwareVersion": "you wish!",
  "Invertible": true,
  "FromFile": ...,
  "ToFile": ...,
  "FromFileSHA": ...,
  "ToFileSHA": ...,
  "CommandLine" "antsRegistration --collapse-output-transforms 0 --dimensionality 3
--initial-moving-transform [ trans.mat, 0 ] --initialize-transforms-per-stage 0
--interpolation Linear --output [ output_, output_warped_image.nii.gz ] --transform
Affine[ 2.0 ] --metric Mattes[ fixed1.nii, moving1.nii, 1, 32, Random, 0.05 ]
--convergence [ 1500x200, 1e-08, 20 ] --smoothing-sigmas 1.0x0.0vox --shrink-factors
2x1 --use-estimate-learning-rate-once 1 --use-histogram-matching 1 --transform SyN[
0.25, 3.0, 0.0 ] --metric Mattes[ fixed1.nii, moving1.nii, 1, 32 ] --convergence [
100x50x30, 1e-09, 20 ] --smoothing-sigmas 2.0x1.0x0.0vox --shrink-factors 3x2x1
--use-estimate-learning-rate-once 1 --use-histogram-matching 1
```

```
--winsorize-image-intensities [ 0.0, 1.0 ]  --write-composite-transform 1"
}
```

`Type`	List of elements that are `translation|rotation|rigid|similarity|affine|displacementfield|parametric|composite`
`Multiplexed`	Whether several transforms are to be applied to the last dimension of the dataset (e.g. a list of affine matrices to correct for head motion each 3D volume)
`Invertible`	Whether the transform is diffeomorphic or not.
`FromFile`	The fixed (reference) file used by `Software` to compute the transform
`FromFileSHA`	
`ToFile`	The moving (target) file used by `Software` to compute the transform
`ToFileSHA`	
`CommandLine`	The command line used to create the transform

7.4 Segmentation Metadata

The `desc-manual` key-value pair should be used to specify whether a segmentation (or parcellation) has been created manually. For automatic segmentations, the `atlas` key should indicate which atlas, if any, was used to create the labels.

7.4.1 Discrete Segmentations

Discrete segmentations of brain tissue represent each tissue class with a unique integer label in a 3D volume.

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              func|anat|dwi/
                     <source_keywords>[_space-<space>]_dseg.nii.gz
```

Example:

```
pipeline/sub-001/
       anat/
              sub-001_space-ACPC_dseg.nii.gz
              sub-001_space-ACPC_dseg.json
```

A segmentation could be a binary mask that functions as a discrete label for a single structure. In this case, the `label` key must be used to specify the corresponding structure. For example:

```
pipeline/sub-001/
	anat/
		sub-001_space-ACPC_label-GM_dseg.nii.gz
```

7.4.2 Probabilistic Segmentations

Probabilistic segmentations of brain tissue represent a single tissue class with values ranging from 0 to 1 in individual 3D volumes or across multiple frames. Similarly to a discrete, binary segmentation, the `label` key can be used to specify the corresponding structure.

Template:

```
<pipeline_name>/
	sub-<participant_label>/
		func|anat|dwi/
			<source_keywords>[_space-<space>][_label-<label>]_probseg.nii.gz
```

Example:

```
pipeline/sub-001/
	anat/
		sub-001_space-ACPC_label-BG_probseg.nii.gz
		sub-001_space-ACPC_label-WM_probseg.nii.gz
```

A 4D probabilistic segmentation, in which each frame corresponds to a different tissue class, must provide a label mapping in its JSON sidecar. For example:

```
pipeline/sub-001/
	anat/
		sub-001_space-ACPC_probseg.nii.gz
		sub-001_space-ACPC_probseg.json
```

The JSON sidecar must include the `label-map` key that specifies a tissue label for each frame:

```
{
	"label-map": [
		"BG",
		"WM",
		"GM"
	]
}
```

7.4.3 Anatomical Labels

BIDS supplies a standard, generic label-index dictionary, defined in the table below, that contains common tissue classes and can be used to map segmentations (and parcellations) between lookup tables.

Index	Name	Abbreviation
0	Background	BG
1	Grey Matter	GM
2	White Matter	WM
3	Cerebrospinal Fluid	CSF
4	Grey and White Matter	GWM
5	Bone	B
6	Soft Tissue	ST
7	Non-brain	NB
8	Lesion	L
9	Cortical Grey Matter	CGM
10	Subcortical Grey Matter	SCGM
11	Brainstem	BS
12	Cerebellum	CBM

These definitions can be overridden (or added to) by providing custom labels in a sidecar `<matches>.tsv` file, in which `<matches>` corresponds to segmentation filename.

Example:

```
pipeline/sub-001/
        anat/
                sub-001_space-ACPC_dseg.nii.gz
                sub-001_space-ACPC_dseg.tsv
```

Definitions can also be specified with a top-level `dseg.tsv`, which propagates to segmentations in relative subdirectories.

Example:

```
pipeline/
        dseg.tsv
        sub-001/anat/
                sub-001_space-ACPC_dseg.json
```

These tsv lookup tables should contain the following columns:

Column name	Description
`index`	The label integer index
`name`	The unique label name
`abbr`	OPTIONAL The unique label abbreviation
`mapping`	OPTIONAL Corresponding integer label in the standard BIDS label lookup
`color`	OPTIONAL Label color for visualization

An example, custom `dseg.tsv` that defines three labels:

```
index  name                abbr   color         mapping
100    "Grey Matter"       GM     #ff53bb       1
101    "White Matter"      WM     #2f8bbe       2
102    "Brainstem"         BS     #36de72       11
```

When specifying a particular label within a filename or JSON, it's acceptable to supply either the label index, abbreviation, or name (with whitespace removed).

7.5 Reconstructed cortical surfaces

Reconstructed cortical surfaces should be stored as GIFTI files, and each hemisphere should be stored separately.

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              anat/
                     <source_keywords>_hemi-{L|R}[_space-<space>]_[surftype].surf.gii
```

Example:

```
pipeline/sub-001/
       anat/
              sub-001_hemi-L_pial.surf.gii
              sub-001_hemi-R_pial.surf.gii
```

The supported surface types are:

Surface type	Description
`wm`	The gray matter / white matter border for the cortex
`smoothwm`	The smoothed matter / white matter border
`pial`	The gray matter / pial matter border
`midthickness`	The midpoints between wm and pial surfaces
`inflated`	An inflation of the midthickness surface (useful for visualization)
`sphere`	The sphere (used for registration - see transforms for nomenclature)
`flat`	The flattened surface (used for visualization)

Note: reconstructed cortical surfaces are unique in that they contain both 2-dimensional and 3-dimensional elements of "space". More details on this concept are given in BEP14 Spaces and Transforms.

7.6 Surface-Mapped Anatomical Scalar Derivatives

Surface-mapped scalar overlays should be stored as either GIFTI or CIFTI files (which allow for the combination of left and right hemispheres).

Template:

```
<pipeline_name>/
        sub-<participant_label>/
                anat/
                        <source_keywords>[_hemi-{L|R}][_space-<space>]_<suffix>.<ext>
```

The preferred extension for scalar GiFTI files is `.shape.gii`. The `hemi` tag is required for GiFTI files. For example:

```
pipeline/sub-001/
        anat/
                sub-001_hemi-L_curv.shape.gii
                sub-001_hemi-R_curv.shape.gii
```

The preferred extension for scalar CIFTI files is `.dscalar.nii`. For example:

```
pipeline/sub-001/
        anat/
                sub-001_curv.dscalar.nii
```

The file suffix should concisely describe the parameter that is represented in the overlay, and while the suffix can be individually customized, the following values should be reserved for their common use-cases:

Suffix	Description
`curv`	Cortical surface curvature indices
`thickness`	Cortical thickness
`area`	Discretized surface area across regions
`dist`	Distance from a point
`defects`	Marked regions with surface defects
`sulc`	Sulcal depth
`myelinmap`	Myelin map calculated from T1w to T2 ratio
`distortion`	Distortion map calculated from a surface registration

7.7 Surface Parcellations

Discrete parcellations (surface segmentations) of cortical structures should also be stored as GIFTI or CIFTI files, and like for volumetric segmentations, parcellations should inherit the appropriate label-names from top-level or sidecar `dseg.json` files.

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              anat/
                     <source_keywords>[_hemi-{L|R}][_space-<space>]_dseg.<ext>
```

The RECOMMENDED extension for GIFTI parcellations is `.label.gii`. The `hemi` tag is required for GIFTI files. For example:

```
pipeline/sub-001/
       anat/
              sub-001_hemi-L_dparc.label.gii
              sub-001_hemi-R_dparc.label.gii
```

The RECOMMENDED extension for CIFTI parcellations is `.dlabel.nii`. For example:

```
pipeline/sub-001/
       anat/
              sub-001_dparc.dlabel.nii
```

```
            sub-001_dparc.dlabel.nii
```

7.8 Morphometrics

Structural statistics produced by segmentation routines should be stored within tsv files, which could contain common parameters specified in the table below.

Template:

```
<pipeline_name>/
       sub-<participant_label>/
              func|anat|dwi/
                     <source_keywords>[_desc-<label>]_morph.tsv
```

Example:

```
pipeline/sub-001/
       anat/
              sub-001_desc-volumetric_morph.tsv
```

Column name	**Description**
`index`	RECOMMENDED Label integer index
`name`	RECOMMENDED Structure name
`centroid`	Center coordinate of structure
`volume`	Volume of structure
`intensity`	Intensity of voxels within structure
`thickness`	Thickness of cortical structure
`area`	Surface area of cortical structure
`curv`	Curvature index of cortical structure

Some parameters might require unit specification or have multiple associated statistics (such as `avg`, `std`, `min`, `max`, `range`). The suggested syntax for this is `<parameter>[-<stat>][-<units>]`. An example volumetric stats file might look something like this:

```
index  name               volume-mm3   intensity-avg       intensity-std
11     Brainstem          23415.9      80.11               3.40
32     Left-Hippocampus   5349.7       75.23               2.27
32     Right-Hippocampus  4112.1       76.98               4.01
```

7.9 Functional derivatives maps

Template:

```
<pipeline_name>/
    sub-<participant_label>/
        func/
            <source_keywords>[_space-<space>][_desc-<label>]_<suffix>.nii[.gz]
            <source_keywords>[_space-<space>][_desc-<label>]_<suffix>.json
```

for example:

```
pipeline1/
    sub-001/
        func/
            sub-001_task-rest_run-1_space-MNI305_mean.nii.gz
            sub-001_task-rest_run-1_space-MNI305_mean.json
```

The following table lists known suffixes and their corresponding measures:

Suffix	Measure
`mean`	Mean across the temporal/4th dimension of the data
`std`	Standard deviation across the temporal/4th dimension of the data
`tsnr`	Temporal SNR (i.e. mean / std)
`sfs`	Signal fluctuation sensitivity
`alff`	Amplitude low frequency fluctuations
`falff`	Fractional amplitude of low frequency fluctuations
`reho`	Regional homogeneity (voxelwise only)
`dcb, dcw`	Voxelwise degree centrality binary and weighted
`ecb, ecw`	Voxelwise eigenvector centrality binary and weighted
`lfcdb, lfcdw`	Local functional connectivity density
`vmhc`	Voxel mirrored homotopic connectivity

All REQUIRED metadata fields coming from a derivative file's source file(s) MUST be propagated to the JSON description of the derivative, **unless the processing makes them invalid** (e.g., if a source 4D image is averaged to create a single static volume, a

SamplingFrequency property would no longer be relevant). In addition the following metadata JSON fields are valid for derivative maps:

Key	Description	Required for suffix
BandpassFilter	String describing all relevant parameters of the applied bandpass filter.	alff, falff
Neighborhood	String describing neighborhood for regional measures.	reho
Threshold	String describing threshold used for determining graph edges.	dcb, dcw, ecb, ecw
Method	String describing method used to calculate measure.	dcb, dcw, ecb, ecw

7.10 Time series and regressors

For the purposes of this section, a time series or regressor is a numeric series of values corresponding to volumes in a BOLD series. **Time series** is specifically used to indicate an aggregate time series within an ROI, while **regressor** is used as a more general term indicating series that may be derived from other sources, such as head motion estimates or physiological recordings.

Regressors will generally be stored as tables, with a row of column headers indicating the name of the regressor. In the case where every voxel has a regressor, then the data should be stored in a 4D NIfTI file.

Tabular regressor files MUST be accompanied by a data dictionary in JSON format, consistent with Raw-BIDS section 4.2.

Volumetric regressor files MAY be accompanied by a data dictionary in JSON format, with OPTIONAL `LongName`, `Description`, `Levels`, `Units`, and `TermURL` fields.

7.11 General time series and regressors

Template:

```
<pipeline_name>/
    sub-<participant_label>/
        [ses-<session_label>/]
            func/
                <source_keywords>_[desc-<label>_]regressors.tsv
                <source_keywords>_[desc-<label>_]regressors.json
```

for example:

```
sub-001/
```

```
        func/
              sub-001_task-rest_run-1_desc-confounds_regressors.tsv
              sub-001_task-rest_run-1_desc-confounds_regressors.json
```

Any time series or regressor with one value per BOLD volume may be stored in a regressors file. If a column is specified in another sub-section, then any additional required metadata MUST be stored in the JSON description of the column.

Column names

Column names are unique alphanumeric values that are defined in a relevant JSON sidecar file. For a series of related columns, it is RECOMMENDED to use zero-based indexing as a suffix. For example, the first six aCompCor components may be named `a_comp_cor_00..a_comp_cor_05`. Custom column names must not conflict with reserved names specified in this document.

Reserved names are specified in the remaining subsections.

Time series transformations

Any column may have the following suffixes appended to indicate transformations applied to the original time series data:

Transformation	Description
`_shift_back`	The time series has been lagged by one TR `x_shift_back[i] = x[i-1]`
`_dt`	The discrete first derivative of the time series `x_dt[i] = x[i+1] - x[i]`
`_sq`	The square of the time series
`_var_norm`	The time series has been variance normalized
`_centered`	The time series has had its mean subtracted

For example, `rot_z_shift_back_sq` means the square of the lagged version of the Z rotation (see Motion-related regressors).

7.12 ROI-based time series extraction

Template:

```
<pipeline_name>/
    sub-<participant_label>/
        [ses-<session_label>/]
```

```
        func/
            <source_keywords>_[atlas-<atlas_label>_][desc-<label>_]timeseries.<tsv|ni
i.gz>
            <source_keywords>_[atlas-<atlas_label>_][desc-<label>_]timeseries.json
```

for example:

```
sub-001/
    func/
        sub-001_task-rest_run-1_atlas-AAL_timeseries.tsv
        sub-001_task-rest_run-1_atlas-AAL_timeseries.json
        sub-001_task-rest_run-1_desc-anaticor_timeseries.nii.gz
        sub-001_task-rest_run-1_desc-anaticor_timeseries.json
```

Time series will generally be stored as tables, with a row of column headers indicating the name of the time series. In the case where every voxel has a time series (i.e., voxel-wise regressors, as in ANATICOR), then the time series should be saved as a NIfTI file.

Metadata fields

Atlas (optional) - A label indicating an atlas that defines a region or set of regions in the volume. An atlas may be three- or four-dimensional, which affects the interpretation of the `roi` index.

ROI (optional) - For 3D atlases, the ROI label should be numeric, corresponding to the value of the voxels in the ROI. For 4D atlases, the ROI label should be numeric, corresponding to the volume index containing the ROI mask. The following special labels correspond to common ROIs that may be defined by many atlases or segmentation algorithms:

Column name	
`white_matter`	Signal derived from white matter ROI.
`CSF`	Signal derived from cerebro-spinal fluid ROI.
`Background`	Signal derived from background (out of brain) ROI.
`GreyMatter`	Signal derived from grey matter ROI.
`Ventricles`	Signal derived from ventricles ROI.
`CircleOfWillis`	Signal derived from circle of Willis ROI.
`GlobalSignal`	Vector of mean values within the brain mask. Can be used for Global Signal Regression.

Time series tabular files MUST be accompanied by a data dictionary in JSON format, consistent with Raw-BIDS section 4.2, which describes metadata for each column name. In addition to LongName, Description, Levels, TermURL column fields, the following new fields are added:

`Method`	RECOMMENDED. Summarization method (defined below).
`Transformations`	RECOMMENDED.
`ROI`	

Column names

Column names are unique alphanumeric values that are defined in a relevant JSON sidecar file. A naming convention might be to concatenate the atlas, ROI index, summarization method (defined below) and transformations (see Time series transformations, above) using snake case. For example, `harvard_oxford_cortical_4_PC` could indicate ROI 4 of the Harvard-Oxford cortical atlas, summarized by taking the first principal component.

Summarization methods

To indicate the summarization method applied to construct a single time series for an ROI, the following column suffixes are defined:

Column suffix	Description of the transformation
`_mean`	The mean of voxel time series
`_median`	The median of voxel time series
`_pc[_<x>]`	The ith eigenvariate from principal component analysis, where i is 0 indexed. If `x` is not specified, the first component is implied (i.e., `pc_0`).
`_spat_reg`	Spatial regression

Examples (tsv + json):

1. Just reserved keywords
2. Timeseries from functional localizer
3. Timeseries from a commonly used parcellation (AAL)

7.13 Motion-related regressors

Template:

```
<pipeline_name>/
    sub-<participant_label>/
        [ses-<session_label>/]
            func/
                <source_keywords>_[desc-<label>_]motion.tsv
                <source_keywords>_[desc-<label>_]motion.json
```

Column names

The six basic motion parameters derived from motion correction have the following names and units:

Column name	Units	Description
`trans_x, trans_y, trans_z`	mm	Translation parameters
`rot_x, rot_y, rot_z`	radians	Rotation parameters

Transformations (see Transformations) of motion parameters may be included in the same file. For example, `rot_z_shift_back_sq` means square of the lagged version of Z rotation.

The following columns indicate summarized motion, as defined in the corresponding references:

Column name	Units	Description
`framewise_displacement`	mm	Framewise displacement (Power, et al., 2012)
`rmsd`	mm	Root mean square deviation (Jenkinson, 1999)
`rms`	mm	Root mean square of translation parameters (Van Dijk, et al., 2012)

7.14 Physiological regressors

Template:

```
<pipeline_name>/
   sub-<participant_label>/
      [ses-<session_label>/]
         func/
            <source_keywords>_[desc-<label>_]physreg.tsv
            <source_keywords>_[desc-<label>_]physreg.json
```

Column names

Column Name	Description
`retroicor_pulse`	Pulse regressor calculated using RETROICOR method (Glober, et al., 2000)
`retroicor_resp`	Respiration regressor calculated using RETROICOR method (Glober, et al., 2000)
`respiration_volume_per_time`	Smoothed respiration belt time series (Birn, et al., 2006)

7.15 Temporal outlier masks

Template:

```
<pipeline_name>/
   sub-<participant_label>/
      [ses-<session_label>/]
         func/
            <source_keywords>_[desc-<label>_]outliers.tsv
            <source_keywords>_[desc-<label>_]outliers.json
```

Outlier masks are columns of zeros (0), with ones (1) indicating volumes that have been identified as outliers by some method.

Column names

Column names for outliers correspond to the type of outlier or method for identifying the outlier. Column names should take the form `method[XX]`, where `XX` is an optional index.

For example, if a BOLD series included two initial dummy scans, `non_steady_state` may be a column with a `1` for the first two volumes, or `non_steady_state_00` and `non_steady_state_01` may be columns with a `1` in the first and second position, respectively.

The following methods are defined as reserved words:

Column name	Description

`non_steady_state_<x>`	Initial non-steady-state volumes

7.16 Other time series and regressors

Time series and regressors that are not otherwise specified should be placed in a `regressors.tsv` file (see General time series and regressors).

Column names

The following regressors are defined as reserved words:

Column name	Description
`dvars`	Change in variance (Brett, 2006, Power, et al., 2012)
`std_dvars`	Standardized DVARS (Nichols, 2013)
`cosine_<X>`	Discrete cosine basis vectors
`legendre_<X>`	Legendre polynomial basis vectors

7.17 Spatiotemporal decompositions

Template:

```
<pipeline_name>/
    sub-<participant_label>/
        [ses-<session_label>/]
            func/
                <source_keywords>_[desc-<label>_]<mixing|components>.tsv
                <source_keywords>_[desc-<label>_]<mixing|components>.nii.gz
                <source_keywords>_[desc-<label>_]decomposition.json
```

for example:

```
sub-001/
        func/
                sub-001_task-rest_run-1_desc-MELODIC_components.nii.gz
                sub-001_task-rest_run-1_desc-MELODIC_mixing.tsv
                sub-001_task-rest_run-1_desc-MELODIC_decomposition.json

sub-001/
        func/
                sub-001_task-rest_run-1_desc-tICA_mixing.nii.gz
                sub-001_task-rest_run-1_desc-tICA_components.tsv
                sub-001_task-rest_run-1_desc-tICA_decomposition.json
```

Spatiotemporal decompositions produce either spatial or temporal components, along with conjugate mixing matrices that can be represented as time series and spatial maps, respectively. The combination of suffix and extension indicates which class of algorithm produced the outputs.

Mixing matrices may be omitted for algorithms that are temporal decompositions with no spatial component, such as CompCor variants.

Spatiotemporal decomposition files MUST be accompanied by a data dictionary in JSON format, consistent with Raw-BIDS section 4.2, which describes metadata for each column name. In addition to column names, the following MANDATORY field is added:

`Method`	Algorithm name and (if applicable) version.

Column names

Column names in spatiotemporal decompositions take the form of `<decomposition>_<index>`, where `<decomposition>` is the name of the decomposition algorithm, and `<index>` is a numeric identifier. Indices SHOULD start at 0. If there is a natural ordering, indices should reflect that ordering. For example, the aCompCor component which explains the most variance should be named `aCompCor00`.

The following reserved words indicate common algorithms:

Label	**Description**
`[a|t|w|c]_comp_cor_<x>`	CompCor (Behzadi, et al., 2007) calculated with voxels chosen based on: ● a: anatomically derived ROIs (white matter and CSF) ● t: temporal variance ● w: white matter voxels only ● c: CSF voxels only
`melodic_<x>`	Columns from the mixing matrix in FSL MELODIC

7.18 Preprocessed diffusion weighted images

Multiple different versions of preprocessing can be stored for the same source data. To distinguish them from each other, the “desc” keyword can be used. Details of preprocessing performed for each variation of the processing should be included in the pipeline documentation.

```
<pipeline_name>/
        sub-<participant_label>/
                dwi/
                    <source_keywords>[_space-<space>][_desc-<label>]_dwi.nii[.gz]
```

```
<source_keywords>[_space-<space>][_desc-<label>]_dwi.bvals
<source_keywords>[_space-<space>][_desc-<label>]_dwi.bvecs
```

The space keyword is described in the table of target spaces of the main BIDS Derivatives proposal.

The `.json` sidecar file is optional, and if present can be used to store information (in addition to fields listed in 7.1) about what preprocessing options were used (for example whether motion compensation was performed, non-linear corrections were applied, whether Eddy current correction was performed, denoising, or intensity normalization were applied, etc).

Preprocessing flags		Type
MotionCompensation	Motion compensation	boolean
EddyCurrentCorrection	Eddy currents corrections	boolean
Denoising	Denoising method	string
IntensityNormalization	Intensity normalization	boolean
NonLinearCorrections	non-linear corrections	boolean
GradientNonLinearities	Correction for non-linear gradients	string [geometry/gradients/geometry&gradients]

7.19 Diffusion Models (and input parameters)

Diffusion MRI can be modeled using various paradigms to extract a more easily understandable representation of the diffusion process and the underlying structure. To do so, parameters might be needed to control how the signal is fit to the model. Those parameters are called **input parameters** in the following. Once the model is fit, the resulting representation can be saved using a number of values per voxel. Those values will be called **output** or **estimated parameters** in the following.

Estimated parameters are saved as NIFTI files (see section Models for the expected content of each model), and **input parameters** are saved in the sidecar JSON file.

The following is a general example of naming convention:

```
<pipeline_name>/sub-<participant_label>/dwi/
    <source_keywords>[_space-<space>][_desc-<label>]_model-<label>_diffmodel.nii[.gz]
    <source_keywords>[_space-<space>][_desc-<label>]_model-<label>_diffmodel.json
```

The following models are codified and should be used in the `model-<label>` field. If a new model is used, common sense should be used to derive a name following the BIDS standard, and should ideally be integrated in a follow-up version of the specification.

Model `<label>` **field accepted values**		What/how?
DTI	Diffusion tensor imaging (Basser et al. 1994)	4D image with Dxx, Dxy, Dxz, Dyy, Dyz, Dzz; the 6 unique parameters of the diffusion tensor.
DKI	Diffusion kurtosis imaging (Jensen et al., 2005)	4D image with Dxx, Dxy, Dxz, Dyy, Dyz, Dzz, Wxxxx, Wyyy, Wzzzz, Wxxxy, Wxxxz, Wxyyy, Wyyyz, Wxzzz, Wyzzz, Wxxyy, Wxxzz, Wyyzz, Wxxyz, Wxyyz, Wxyzz; Where D is the diffusion tensor and W is the kurtosis tensor.
WMTI	White matter tract integrity (Fieremans et al., 2011)	4D image with Dxx, Dxy, Dxz, Dyy, Dyz, Dzz, Wxxxx, Wyyy, Wzzzz, Wxxxy, Wxxxz, Wxyyy, Wyyyz, Wxzzz, Wyzzz, Wxxyy, Wxxzz, Wyyzz, Wxxyz, Wxyyz, Wxyzz, Dhxx, Dhxy, Dhxz, Dhyy, Dhyz, Dhzz, Drxx, Drxy, Drxz, Dryy, Dryz, Drzz, AWF; where D is the diffusion tensor, W is the kurtosis tensor, AWF is the additional axonal water fraction parameter
CSD	Constrained Spherical Deconvolution (Tournier et al. 2007; Descoteaux et al. 2009)	4D image with SHM coefficients (size and ordering, depends on the model order and basis set, specified in the sidecar)
NODDI	Neurite Orientation Dispersion and Density Imaging (Zhang et al. 2012, Daducci et al., 2015)	4D image with ICVF, OD, ISOVF; where ICVF is the “intracellular volume fraction” (also known as NDI), OD is the “orientation dispersion” (the variance of the Bingham; also known as ODI) and ISOVF is the isotropic component volume fraction (also known as IVF).
DSI	Diffusion Spectrum Imaging (Wedeen et al. 2008; Paquette et al 2017)	DSI is generally used mostly to compute the diffusion ODF (Orientation Distribution Function). No parameters are generally returned, but an ODF can be saved as a map.
CSA	Constant solid angle (Aganj et al. 2010)	4D image with SHM coefficients (size and ordering, depends on the model order and basis set, specified in the side-car)
SHORE	Simple Harmonic Oscillator based Reconstruction and Estimation. (Ozarslan et al. 2008)	4D image with basis coefficients depending on choice of basis.
MAPMRI	Mean Apparent Propagator MRI. (Ozarslan, 2013)	4D image with basis coefficients depending on choice of basis.
FORECAST	Fiber ORientation Estimated using	4D image with SHM coefficients.

	Continuous Axially Symmetric Tensors. (Zuchelli et al. 2017)	
fwDTI	Free water DTI (Hoy et al. 2014)	4D image with D_c_xx, D_c_xy, D_c_xz, D_c_yy, D_c_yz, D_c_zz, FWF; where D_c is the fw-corrected diffusion tensor in this voxel.
BedpostX	Ball-and-Stick model (Behrens et al. 2007; Jabdbi et al, MRM 2012)	5D image (xyz, m, n) for m parameters and n MCMC samples (n=1 implies best-fit parameters). Parameters are f_i, th_i, ph_i (volume fraction, polar angle, azimuthal angle) for up to 3 fibres, d, d_std (for "multiexponential" model)

The JSON sidecar contains the following generic key/value pairs:

- `Shells` (optional): shells that were utilized to fit the model, as a list of b-values. If the key is not present, all shells were used.
- `Gradients` (optional): subset of gradients utilized to fit the model, as a list of three-elements lists. If not present, all gradients were used.

For each model type, some additional fields are mandatory, and some are optional.

```
● Model:
    ○ {"DTI"}:
            fitMethod : {"WLS","OLS","NLLS","RESTORE"}
                          -  RESTORE : sigma
    ● {"DKI","WMTI"} :
            fitMethod : {"WLS","OLS","NLLS"}
    ● {"CSD"} :
            sphericalHarmonicOrder   : value
            responseFunctionOrder    : value
            responseFunction         :
      [1:response_function_order_value,b0mean]
            basis : {"MRtrix 0.2","MRtrix3","DESCOTEAUX"}
    ● {"CSDM"} :
            sphericalHarmonicOrder   : [values (1 x number of tissues)]
            responseFunctionOrder    : [values (1 x number of tissues)]
            responseFunction         :  [values (1 x number of
      tissues)][1:response_function_order_value x number of
      tissues,b0mean x number of tissues]
            basis : {"MRtrix 0.2","MRtrix3","DESCOTEAUX"}
    ● {"NODDI:AMICO"} :
            dPar        : value
            dIso        : value
            lambda1     : value
            lambda2     : value
    ● {"DSI"} :
            gridSize    : value
```

```
        rStart      : value
        rStep       : value
        rEnd        : value
        filterWidth : value
```

- {"CSA"} :

```
        sphericalHarmonicOrder        : value
        smoothing                     : value
        basis                         : value
```

- {"SHORE"} :

```
        radialOrder                   : value
        zeta                          : value
        lambdaN                       : value
        lambdaL                       : value
        tau                           : value
        constrainE0                   : value
        positiveConstraint      : value
        posGrid                       : value
        posRadius                     : value
```

- {"MAPMRI"} :

```
        radialOrder                   : value
        laplacianRegularization       : bool
        LaplacianWeighting            : value
        positivityConstraint          : bool
        tau                           : value
        constrainE0                   : value
        positiveConstraint      : value
        posGrid                       : value
        posRadius                     : value
        anisotropicScaling      : bool
        eigenvalueThreshold           : value
        posGrid                       : value
        bvalThreshold                 : value
        dtiScaleEstimation      : bool
        staticDiffusivity             : value
```

- {"FORECAST"} :

```
        sphere                  : value
        decAlg                  : value
        lambdaLb                      : value
        sphericalHarmonicsOrder : value
```

- {"fwDTI"} :

```
        fitMethod : {"WLS","NLLS"}
```

- {"BEDPOSTX"} :

```
        nFibers                       : value
        fudge                         : value
        burnIn                        : value
        nJumps                        : value
        sampleEvery                   : value
        model       : {"monoexponential",
"multiexponentialStick",”multiexponentialZeppelin”}
```

- `ModelDescription <optional>`: extended information to describe the model.

- `ModelURL` `<optional>`: URL to the implementation of the specific model utilized.

Examples:

```
my_diffusion_pipeline/sub-01/dwi

sub01-dwi_space-T1w_desc-WLS_model-DTI_diffmodel.nii.gz:
```

A 4D volume with floating point numbers:

```
sub01-dwi_space-T1w_desc-WLS_model-DTI_diffmodel.json:
       {“Model”: “DTI”,

       “DTI: {“fitMethod”: “WLS”}}
```

7.20 Model-derived maps (output parameters)

Models output maps of estimated parameters. Commonly one parameter is estimated per voxel, we call these maps of parameters. Two types of maps are described in this specification: scalar and vector maps. Scalar maps are saved as 3D `.nii/nii.gz` files. Vector maps are saved as 4D `.nii/nii.gz` files. Below the specification for the naming of the files and a the list of currently accepted fields for the maps.

```
<pipeline_name>/
       sub-<participant_label>/
              dwi/

       <source_keywords>[_space-<space>][_model-<label>][_desc-<label>]_<map_label>.ni
       i[.gz]
```

7.20.1 Scalar maps

These maps are saved 3D NIfTI files (x,y,z, 1).

`<map_label>` **field accepted values**		**Possible MODEL sources**	**Units or scale**
FA	Fractional Anisotropy	DTI, DKI, WMTI (D, Dh, Dr)	Unitless [0-1]
MD	Mean diffusivity (also called apparent diffusion coefficient, ADC)	DTI, DKI, WMTI (D, Dh, Dr)	microns^2/ms (for example, for free water in body temperature, this should be 3)
AD	Axial Diffusivity (also called parallel diffusivity)	DTI, DKI, WMTI (D, Dh, Dr)	microns^2m/s (for example, for free water in body temperature, this should be 3)

RD	Radial Diffusivity (also called perpendicular diffusivity)	DTI, DKI, WMTI (D, Dh, Dr)	microns^2/ms (for example, for free water in body temperature, this should be 3)
MODE	Mode of the tensor	DTI, DKI, WMTI (D, Dh, Dr)	
LINEARITY	Tensor linearity (Westin 1997)	DTI, DKI, WMTI (D, Dh, Dr)	
PLANARITY	Tensor planarity (Westin 1997)	DTI, DKI, WMTI (D, Dh, Dr)	
SPHERICITY	Tensor sphericity (Westin 1997)	DTI, DKI, WMTI (D, Dh, Dr)	
MK	Mean kurtosis	DKI, WMTI	Unitless
RK	Radial kurtosis	DKI, WMTI	Unitless
AK	Axial kurtosis	DKI, WMTI	Unitless
ICVF	Intracellular volume fraction. This can be calculated from different models, so can have slightly different interpretation.	WMTI (AWF), NODDI	Proportion [0-1]
DECFA	Directionally-encoded color (DEC) FA	DTI, DKI, WMTI (D, Dh, Dr)	3 8-bit floating point numbers that encode x,y,z of the principal diffusion direction in the voxel. The norm of the vector is the FA in the voxel. Need to set the NiFTI dtype to RGB
ODI	Orientation dispersion index	NODDI	Normalised to [0,1]
ISOVF	Isotropic volume fraction	NODDI	microns^2/ms (for example, for free water in body temperature, this should be 3)
TDI	Track Density Imaging		
GFA	Generalized Fractional Anisotropy	CSA, CSD, SHORE, MAPMRI, Forecast (or any model that can be represented as an ODF)	Proportion [0 - 1]
AFD	Apparent Fiber Density		positive, unitless

INDEX	Scalar maps from bedpostx	FSUM (sum of partial volume fractions of the sticks) F_i (volume fraction of stick i) D (diffusivity) DSTD (std of diffusivity)	FSUM [0 - 1] F_i [0 - 1] D and DSTD microns^2/ms (for example, for free water in body temperature, this should be 3)
RTPP	Return to the plane probability	MAPMRI	Probability [0 - 1]
RTAP	Return to axis probability	MAPMRI	Probability [0 - 1]

7.20.2 Vector-valued maps

These maps are saved 4D NIfTI files (x,y,z, n*(xyz)).

`<map_label>` **field accepted values**		**Possible sources**
fixel	Fixel (Fiber bundle element) images. Different voxels can have different numbers of fixels, and therefore a sparse file format is required. This is not specific to Fibre Density derived via CSD. Several diffusion models provide fixel-specific measures (e.g. Ball and stick, CHARMED, CuspMFM, Myelin content).	
ODF	Orientation distribution function	DSI, SHORE, MAPMRI
PDF	Diffusion propagator	DSI, SHORE, MAPMRI
FOD	Fiber orientation distribution (Sometimes known as fODF)	CSD,
EVECS	Eigenvectors of a model that has eigenvectors (such as DTI)	DTI, DKI, WMTI, ...
PEAKS	Directions of ODF maxima on the sphere.	Pretty much any model that can be projected on the sphere

7.21 Describing tractography

Tractography normally generates one of two primary file types: tractograms or NIfTi files containing maps. Tractograms are files containing a collection of streamlines, which are objects describing the path of idealized brain fiber fascicles (ensembles of neural fibers that travel together). Tractography maps contain visitation counts (i.e. 3D spatial histograms) of a collection of streamlines but do not contain the actual streamline objects. The main type of map represent a "probability of connection", which is the result of FSL's FDT tractography process. Two tractogram file formats are supported, tractography maps should be saved as NIfTIs.

```
<pipeline_name>/
        sub-<participant_label>/
                dwi/

<source_keywords>[_space-<space>][_desc-<label>][_subset-<label>]_tractography.trk
```

OR

```
<source_keywords>[_space-<space>][_desc-<label>][_subset-<label>]_tractography.tck
```

OR

```
        <source_keywords>[_space-<space>][_desc-<label>][_subset-<label>]_tractography.
nii[.gz]
```

Desc (optional) – A way to refer to a specific instance of the tractography process. The combination of tractography methods and parameters that were used to create this tractography result should be described in the associated .json file.

Subset (optional) – A label descriptor for the the subset of streamlines included in this file; if not specified, a whole brain tractography is assumed (a tractogram). Example of subsets can be "short" if only short streamlines were kept, or "subsampled50" if only 50% of the streamlines were kept.

File formats	
.trk - Diffusion Toolkit (+TrackVis)	http://www.trackvis.org/docs/?subsect=fileformat ● Embeds all of the information about the space ● Does not manage properly the orientation (everything has to be in LPS) ● In new trackvis some of the problems are corrected. Also .trk file can be corrected using anatomical information in advance.
.tck - MRTrix	http://mrtrix.readthedocs.io/en/latest/getting_started/image_data.html#tracks-file-format-tck ● Assumes 1mm space ● Allows storage of key-value pairs within the file header ● Streamlines are stored in the same reference frame as the input DWI files, i.e. are in scanner coordinates.

We have two main classes of tractography algorithms: Global or Local tractography, each with various supported algorithms.

Class	Algorithms	Reference	
LOCAL	PROB	Sherbondy, A.J., *et al.* (2008). Tournier J.D.*et al.* (2012). Behrens et al. (2007)	Probabilistic tracking
	DET	Conturo, T. *et al.* (1999) Mori, S. *et al.* (1999)	Deterministic tracking
	EUDX	Garyfallidis *et al.* (2010) Mori, S. *et al.* (1999)	Euler delta crossings
	FACT	Jiang, H. *et al.* (2005)	Fiber Assignment by Continuous Tracking
	STT	Lazar, M., *et al.* (2003)	Tensor deflection
GLOBAL	UKF	Malcolm J-G., *et al.* (2009)	Unscented Karman Filter
	SpinGlass	Fillard P., *et al.* (2009)	Spin Glass tractography
	ENS	Takemura *et al.* (2016)	Combines multiple algorithms.
	Other	Mangin, J.F. et al. (2013). Aganj, I. et al. (2011). Neher, P.F. et al. (2012). Jbabdi, S., et al. (2007). Sherbondy, A.J., et al. (2009). Reisert, M. et al. (2011)	Various algorithms

The JSON sidecar contains the following key/value pairs:

Mandatory:

- `sform`: 4x4 transformation matrix of the NIfTI file used to generate the tracts. This means that the tracts coordinates are expressed in the tracts files in this space.
- `qform`: 4x4 transformation matrix of the NIfTI file used to generate the tracts. This means that the tracts coordinates are expressed in the tracts files in this space.
- `TractographyClass : ["local", "global"]`
- `TractographyMethod :`
    - `"local" : ["probabilistic","deterministic","eudx","fact","stt","other"]`
    - `"Global": ["UKF","SpinGlass","other"]`

Recommended:

- `When TractographyMethod is ["probabilistic","deterministic","eudx","fact","stt","other"]`
    - `StepSizeUnits : ["mm","norm"]`
    - `StepSize : value`
    - `AngleCurvature : value`

    - SeedingMethod : ["seedsPerVoxel", "globalSeedNumber", "totalStreamlinesNumber"]
    - SeedingNumberMethod : [integer for the number of seeds]
    - TerminationCriterion : ["leaveMask", "reachingTissueType”]
    - TerminationCriterionTest : ["binary","ACT","CMC"]
    - StreamlineIndex : <optional>

- When TractographyClass is ["global"]
    - TractographyMethod : ["UKF","SpinGlass","other"]
        - Param1 : ["mm","norm"]
        - Param2 : value
        - Param3 : <optional>
        - Param4 : <optional>

Acronyms for tractography parameters:

- ACT: anatomically constrained tractography. (Smith et al. 2012)
- CMC: continuous map criterion. (Girard et al. 2014)

Example:

```
sub01-dwi_desc-DET1_tractography.trk
sub01-dwi_desc-DET1_tractography.json:
      {"sform": 4x4 matrix,
       "qform": quaternion explanation,
       "TractographyClass": "local",
             "TractographyMethod": "deterministic",
             "StepSize": 0.5}
sub01-dwi_subset-AFLeft_tractography.trk
sub01-dwi_subset-short_fibers_tractography.json:
      {"sform": 4x4 matrix,
       "qform": quaternion explanation,
       "TractographyClass": "global",
             "TractographyMethod": "UKF",
             "Description": "UKF tracking where only short fibers were kept."
```