BIDS Extension Proposal 39 (BEP039): Dimensionality reduction-based networks

BIDS extension proposal for dimensionality reductions
that produce spatiotemporal components for fMRI/PET/EEG/MEG/iEEG data

Extension moderators/leads: **Arianna Sala <arianna.sala@uliege.be>, Anibal Heinsfeld <anibalsolon@utexas.edu>, Alejandro de la Vega <delavega@utexas.edu>, Ross Blair <>, Cyrus Eierud <ceierud@gsu.edu>, Dora Hermes <hermes.dora@mayo.edu>, Angela Laird <alaird@fiu.edu>, Franco Pestilli <pestilli@utexas.edu>**

Contributors: Russ Poldrack, Robert Smith, Eugene Duff, Oscar Esteban, Cyril Pernet, Tom Nichols, Peer Herholz, Vince Calhoun

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. Discussion can take place here, or on the Brainhack Mattermost channel.

This specification is an extension of BIDS Common Derivatives proposal. 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.

This specification is part of BIDS-derivatives, for a complete list of the BIDS-derivatives sub-committees, see BIDS Derivatives sub-committees.

Related BEPs: BEP017 generic connectivity schema, BEP12 - Functional Derivatives

1. Table of Contents

2. Scope

This schema is intended to cover a range of use cases organizing and describing data deriving from dimensionality reduction pipelines, including - but not limited to - independent component analysis, principal component analysis and diffusion embedding. Dimensionality reduction is a key outcome measure in a number of imaging modalities, including fMRI, sMRI, PET and electrophysiology (MEG/EEG/iEEG) and across a broad range of analytical approaches. The BIDS dimensionality reduction-based networks extension aims to capture and describe the majority of the respective outcome measures.

Most generally, the schema aims to harmonize directory structures, file names and meta-data pertaining to the results obtained from:

1) different methods for dimensionality reduction, such as but not exclusively (i) independent component analysis: used to separate a mixture of signals into different sources by maximizing their statistical independence; (ii) principal component analysis: an approach for decomposing data into ranked projections, each capturing maximal variance; (iii) spatial gradients: spatial transitions based on n-dimensional spaces derived by dimensionality reduction methods (e.g. ICA, PCA, diffusion-map embedding, etc.).

2) different axes of dimensionality reduction, such as but not exclusively (i) spatial dimensionality reduction, to identify spatial components of functional interest in data from neuroimaging modalities. This can be achieved either based on spatiotemporal data at single-subject and group level (as is typically the case for fMRI data) or based on spatial data at the group level (as is typically the case for sMRI and PET data); (ii) temporal dimensionality reduction, which can be considered the transpose of the spatial case, used for temporal ICA of fMRI data or in data from electrophysiological modalities where there are many more timepoints than spatial points.

While not directly addressed in this extension proposal, the schema could be easily extended and applied to variations of the methods mentioned above for dimensionality reduction, like variations of principal component analysis, e.g. partial least squares or multimodal fusion approaches such as joint or parallel ICA.

2.1 Goals

While a wide range of methods can be used to achieve dimensionality reduction across neurophysiological modalities, there are fundamental similarities in the organization of the outputs. The goal of this BEP, therefore, is to define a schema for BIDS derivatives that can be used to give structure to the outputs of a variety of dimensionality reduction methods, applied to any type of neurophysiological modalities.

This extension will allow to capture and describe the outputs of dimensionality reduction approaches such as but not limited to independent component (vector) analysis (ICA/IVA), principal component analysis (PCA), and spatial gradients, performing dimensionality reduction along the spatial or temporal axes, at single-subject or group-level, and based on any neurophysiological modality. Dimensionality reduction outputs will be captured and described via a set of factors that allow abstracting from a given modality-analysis combination into rather agnostic and FAIR representations. This entails two main types of files:

1) files including the spatial (components), temporal (mixing matrices) or spatiotemporal components resulting from the dimensionality reduction;
2) files that describe the parameters used in the execution of the dimensionality reduction method, entailing a set of common information shared/required across all modalities and analyses, as well as information specific to/required by a given modality and analysis.

The hereby created data will enable a broad range of results obtained from diverse dimensionality reduction methods to be more easily shared and utilized, for example, in the context of multi-modal data integration and/or for comparison of results obtained from different software or pipelines.

2.2. Relationship to BIDS and other BEPs

Most core principles of the original BIDS-Raw and BIDS-Derivative specification are inherited by the BIDS-dimensionality reduction-base networks specification, though some special considerations and additional fields are noted below.

In addition, this BEP refers to terminology introduced in other in-progress PRs:

https://github.com/bids-standard/bids-extensions/pull/26

Please refer to the general BIDS specification document for context and general guidelines (definitions, units, directory structure, missing values, stimulus and event information, etc.).: https://bids-specification.rtfd.io

The keywords "REQUIRED", "SHOULD", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

The terminology inherited from BIDS-Derivative includes the following:

- desc
- mask
- source

The terminology inherited from the in-progress PRs includes the following:

- map

The terminology introduced in this BEP includes the following:

- item
- param-mixing
- param-components
- model

3. BIDS - dimensionality reduction-based networks

BIDS contains “required”, “recommended” and “optional” fields. These are indicated throughout the document:

- REQUIRED: essential to be BIDS compliant (i.e. MUST as per RFC2199)
- RECOMMENDED: gives a warning if not present (i.e. SHOULD as per RFC2199)
- OPTIONAL: no warning if missing (i.e. MAY as per RFC2199)

As in BIDS-Raw, the following apply:

1) All specifications of paths need to use forward slashes.
2) The inheritance principle applies: any metadata file (.json, .tsv, etc.) may be defined at any directory level. The values from the top level are inherited by all lower levels unless they are overridden by a file at the lower level. For details see BIDS-Raw (The Inheritance Principle).

3.1. File formats for the raw data

This proposal aims to describe dimensionality reduction-based networks via the following files. The former entail: a REQUIRED file containing information about the applied model(s) (in .tsv), REQUIRED map files, depending on the data and model, either a file containing spatial maps or “components” (e.g. in .nii, .nii.gz, .gii, .eeg) or decomposition time course or “mixing matrix” in (.tsv), a REQUIRED file containing information about the subjects included in the model (in .tsv) and a RECOMMENDED file entailing information on the masking applied within the model (e.g. in .nii, .nii.gz, .gii, .eeg or .tsv). Each file is accompanied by a .json entailing respective meta-data.

As outlined above, the map files can include the spatial maps or “components” (e.g. in .nii, .nii.gz, .gii, .eeg) or decomposition time course or “mixing matrix” in (.tsv) files or both. This is dependent on the data and applied model at hand. While some data/models result in one or the other (e.g. assessing spatial gradients in fMRI), others result in both (e.g. ICA in M/EEG). In the latter case, both files, the “components” and “mixing matrix” file, will be accompanied by a respective .json sidecar file following BIDS principles.

3.2. Directory Structure

The files introduced in the prior section can be situated at two different levels within a dedicated derivatives directory: at the subject level and at the top level. For example, if a given model was applied to the data of 10 subjects, each subject would have its specific files in its directory and general model files would be stored at the top level. The former would consist of either spatial maps or decomposition time courses or both and their meta-data files, whereas the latter would entail the files describing the applied models.
If no subject-specific but group-level files were obtained through the model, these files will be stored at the top level as well.

```
OPTION 1: model generates subject-level files only
<dataset>/derivatives/<pipeline_name>/
        models.tsv
        model-<model-id>_<modalitysuffix>.json
        model-<model-id>_participants.tsv
        sub-<label>/<modality>/
                <source_entities>_model-<label>_param-<mixing|components>_map.<ext>
                <source_entities>_model-<label>_param-<mixing|components>_map.json
                <source_entities>_model-<label>_mask.<ext>
                <source_entities>_model-<label>_mask.json
```

```
OPTION 2: model generates group-level and subject-level files
<dataset>/derivatives/<pipeline_name>/
        models.tsv
        model-<model-id>_description.json
        model-<model-id>_participants.tsv
        group-<label>/<modality>/
          <source_entities>_model-<label>_param-<mixing|components>_map.<ext>
          <source_entities>_model-<label>_param-<mixing|components>_map.json
          <source_entities>_model-<label>_mask.<ext>
          <source_entities>_model-<label>_mask.json
        sub-<label>/<modality>/
          <source_entities>_model-<label>_param-<mixing|components>_map.<ext>
          <source_entities>_model-<label>_param-<mixing|components>_map.json
          <source_entities>_model-<label>_mask.<ext>
          <source_entities>_model-<label>_mask.json
```

3.3 models.tsv

This file indicates the models included in a single derivative pipeline output. This includes the name of the model type employed by the model, which also acts as the model identifier, ie model_id for the corresponding JSON meta-data file(s), the datatype it was run in, as well as an OPTIONAL long-form description of the model. *models.tsv* is be accompanied by a model-<model_id>.json file describing the included model(s) further. Importantly, all models applied within a single derivative pipeline have to be included in the models.tsv file.

The following REQUIRED and OPTIONAL fields are defined as standards for dimensionality reduction models.

Column name	Description
model_id	REQUIRED. model type.
datatype	REQUIRED. Datatype(s) included in model
description	OPTIONAL. Human interpretable long description of model

Example

model_id	datatype	description
GroupSpatialICA	func	Group independent component analysis for multi-subject fMRI data as proposed by Guo & Pagnoni 2010.
TemporalICA	func	Temporal independent component analysis for fMRI data as proposed by Glasser et al. 2019.

3.4 model-<model_id>_<modalitysuffix>.json

model-<model_id>_<modalitysuffix>.json describes the parameters used in the execution of the respective model and thus dimensionality reduction method. Each model listed in models.tsv is required to have one corresponding *model-<model_id>_<modalitysuffix>.json* with the correspondence being indicated by the model_id. Several common parameters have been derived, with some being REQUIRED and others OPTIONAL. Importantly, these parameters are intended to be independent of the software used to conduct the dimensionality reduction. All software-specific parameters are placed in the SoftwareParameters key.

Key name	**Description**
modelID	REQUIRED. model identifier should be equal to the corresponding identifier used in models.tsv.
modelType	REQUIRED. Overarching/general model class/type (e.g. PCA, ICA, Gradients, etc.).
Algorithm	REQUIRED. Name of the specific algorithm use in the model (e.g. SVD, FastICA, InfoMax, DiffusionEmbedding, etc.). The valid values of this field are not controlled.
AlgorithmParams	REQUIRED. Initiation parameters chosen before running the algorithm
Mask	OPTIONAL. Spatial mask applied prior to dimensionality reduction
Filter	OPTIONAL. Temporal filter applied prior to dimensionality reduction
NDimensions	OPTIONAL. Number of dimensions (same as model order) requested by the dimensional reduction algorithm. This is RECOMMENDED for algorithms that require manually specifying the number of dimensions for the reduction.
Software	REQUIRED. Reference to software and library used to produce results
SoftwareParameters	OPTIONAL (dictionary). Uncontrolled dictionary of (software-specific) model parameters

Example

```
{
    "modelID": "InfoMax-attempt2",
    “modelType: “ICA”
    "Algorithm": "InfoMax",
    "AlgorithmParams": {
            "block less than 50485": 129,
            "stopCriteria": 1e-6,
            "weight": 0,
            "irate": 0.005,
            "maxSteps": 512,
            "annealDegrees": 60,
            "momentum": 0,
            "select posact": 0,
            "select sphering": true,
            "select bias": true
        }
    "Mask": "default&icv",
    "NDimensions": 20,
    "Software": "GIFT@v4.0.3.0",
    "SoftwareParameters": {
        "MDLEstimated": false,
        "stabilityType": "regular",
        "backReconAlgo": "GICA",
        "normalization": "z-scores",
        "GroupICAtype": "spatial",
        "PCAtype": "standard",
        "GroupPCAtype": "subject",
        "PCAsteps": 2,
        "numPCstep1": 30,
        "numPCstep2": 20,
        "autofill": false,
        "dataPreprocessingType": "demean",
        "extended": 0,
    }

}
```

3.5 model-<model-id>_participants.tsv

The list of participants indicates which participants were included in the model via their subject IDs. The model-<model-id>_participants.tsv file MAY contain additional columns sampled from the participants.tsv file of the original BIDS dataset. Using this subject list may help future researchers understand what analysis the original researcher performed.

Column name	Description
participant_id	Subject number (e.g., 05)
…	...

Example

participant_id
01
02
03
04
05
06
…

3.6 model-<label>_param-components[_item-<item>]_<modality>map.<ext>

Spatial components obtained through the dimensionality reduction are represented via dedicated files storing the corresponding spatial maps. Independent of the data modalities, the spatial maps SHOULD be provided in 4D, ie all obtained components in one file stacked along the 4th dimension. If there is a natural ordering, indices in the 4th dimension should reflect that ordering. For example, the spatial component which explains the most variance should be stored at the first index. However, depending on the software used, components might also be output as a list of respective 3D files. In such cases, the `item` key has to be used to denote the item of the given component, again starting with the most variance explaining component at the first index. The respective file extension is set based on the data modality. For example, spatial components yielded from fMRI data SHOULD be provided in .nii/.gz and vice versa in .bdf/.edf for EEG data. The naming convention for these files is:

```
<source_entities>_model-<label>_param-<mixing|components>[_item-<item>]_map.<ext>
```

Example

In case the utilized software outputs a 4D file with components stacked along the 4th dimension, the respective file would be named as follows (please note the absence of the `item` key):

```
sub-01_task-rest_model-corticalGradientsDE_param-components_boldmap.nii.gz
```

In case the utilized software outputs a list of 3D files with components, the respective files would be named as follows (please note the `item` key):

```
sub-01_task-rest_model-corticalGradientsDE_param-components_item-1_boldmap.nii.gz

sub-01_task-rest_model-corticalGradientsDE_param-components_item-2_boldmap.nii.gz

sub-01_task-rest_model-corticalGradientsDE_param-components_item-3_boldmap.nii.gz

…
```

3.7 model-<label>_param-components[_item-<item>]_<modality>map.json

This file describes the components in the corresponding model-<label>_param-components[_item-<item>]_map.<ext> file and provides additional metadata. It entails, for example information on the components' ordering.

Key name	Description
Description	REQUIRED.
modelDescriptionPath	REQUIRED. Path to the model description metadata json describing the model used to derive this file.
ComponentsOrdered	REQUIRED. Boolean field to specify whether the components contained within the file are ordered (ie. 1 or 0 only) or not.
ComponentsOrderedDescription	RECOMMENDED. If the components are ordered, a free-form description of the ordering should be provided, e.g. "based on variance explained, descending order".
ComponentsType	REQUIRED. String field. Indicates whether weight values refer to time points (typically the case for fMRI/EEG data types) or participants (typically the case for PET data type)
Sources	REQUIRED. A list of files with the paths specified using BIDS URIs; these files were directly used in the creation of this derivative data file. For example, if a derivative A is used in the creation of another derivative B, which is in turn used to generate C in a chain of A->B->C, C should only list B in "Sources", and B should only list A in "Sources". However, in case both X and Y are directly used in the creation of Z, then Z should list X and Y in "Sources", regardless of whether X was used to generate Y. Using paths specified relative to the dataset root is DEPRECATED.
VarianceExplained	OPTIONAL. Dictionary, where keys are the component's indices and the values, are the variance explained by each component in percentage.
Dropped	OPTIONAL. Dictionary, with Boolean values indicating if component was excluded in back projection
Scaled	OPTIONAL. Dictionary, with Boolean value indicating if component was scaled

Example

```
{
      “modelDescriptionPath”:

      “bids:mydataset:derivatives/ICA/model-ICA_description.json”,
      “ComponentsOrdered”: true,
      “ComponentsOrderdDescription”: “Ordered by variance explained,
             descending order”,
      “Sources”:
      [“bids:mydataset:derivatives/preprocessing/sub-01/func/sub-01_t
      ask-rest_desc-preproc_bold.nii.gz”]
}
```

3.8 model-<label>_param-mixing[_item-<item>]_<modality>map.tsv

Decomposition time courses are represented as a tabular data file, consistent with Raw-BIDS. For example, in fMRI the mixing matrix contains the independent component analysis time courses representing the fMRI hemodynamics for each component. In PET, the mixing matrix typically contains the independent component analysis participants representing each participant’s loading value (contribution) to each component. Column names in the dimensionality reduction model take the form of <item>. It is RECOMMENDED that <item> is a numeric identifier that SHOULD start at 1. If there is a natural ordering, indices should reflect that ordering. For example, the aCompCor component, which explains the most variance, should be named aCompCor01. These time courses may be viewed as weights along time for each spatial component. Depending on the data modality, size and temporal resolution, the .tsv files with the resulting mixing matrix can either be stored as is (for example, fMRI data) or might have to be compressed, e.g. .gzip, getting a .gz extension (for example, M/EEG data). In the example below, the number of rows is usually more than 100 rows.

Column name	**Description**
1	Loading values for every time point or participant, representing component 1
2	Loading values for every time point or participant, representing component 2
…	Holds loading values for components 3-19 for every time point or participant
…	…

Example:

1	2	3	…	…
0.75	0.12	0.05	…	…
0.21	0.62	0.19	…	…
0.48	0.29	0.39	…	…
…	…	…	…	…
0.09	0.28	0.43	…	…

3.9 model-<label>_param-mixing[_item-<item>]_<modality>map.json

This file describes the columns in the corresponding model-<label>_param-mixing[_item-<item>]_map.<ext> file further via providing additional metadata. It entails for example, information on the components' ordering. Following the inheritance principle, it is possible to define a single JSON sidecar that applies to all temporal tabular files within a model. In addition to the fields defined in Raw-BIDS section 4.2 (https://bids-specification.readthedocs.io/en/stable/02-common-principles.html#tabular-files), the following REQUIRED and OPTIONAL fields are defined:

Key Name	**Description**
Description	REQUIRED.
modelDescriptionPath	REQUIRED. Path to the model description metadata json describing the model used to derive this file.
ComponentsOrdered	REQUIRED. Boolean field to specify whether the components contained within the file are ordered (ie. 1 or 0 only) or not.
ComponentsOrderedDescription	RECOMMENDED. If the components are ordered, a free-form description of the ordering should be provided, e.g. “based on variance explained, descending order”.
ComponentsType	REQUIRED. String field. Indicates whether weight values refer to time points (typically the case for fMRI/EEG data types) or participants (typically the case for PET data type)
Sources	REQUIRED. A list of files with the paths specified using BIDS URIs; these files were directly used in the creation of this derivative data file. For example, if a derivative A is used in the creation of another derivative B, which is in turn used to generate

	C in a chain of A->B->C, C should only list B in "Sources", and B should only list A in "Sources". However, in case both X and Y are directly used in the creation of Z, then Z should list X and Y in "Sources", regardless of whether X was used to generate Y. Using paths specified relative to the dataset root is DEPRECATED.
VarianceExplained	OPTIONAL. Dictionary, where keys are the component's indices and the values, are the variance explained by each component in percentage.
Dropped	OPTIONAL. Dictionary, with Boolean values indicating if component was excluded in back projection
Scaled	OPTIONAL. Dictionary, with Boolean value indicating if component was scaled

Example:

```
{
      "modelDescriptionPath":

      "bids:mydataset:derivatives/ICA/model-ICA_description.json",
      "ComponentsOrdered": true,
      "ComponentsOrderedDescription": "Ordered by variance explained,
             descending order",
      "ComponentsType": "Timepoint",
      "Sources":
      ["bids:mydataset:derivatives/preprocessing/sub-01/func/sub-01_t
      ask-rest_desc-preproc_bold.nii.gz"]
}
```

3.10 <source_entities>_model-<label>_mask.<ext>

If the dimensionality reduction was performed within a subset of the data, e.g. within cortical gray matter voxels and/or a specific region, the respectively used mask SHOULD be provided as a RECOMMENDED file. This will most likely entail data with a focus on the spatial resolution, ie (f)MRI, PET, etc. and thus, the mask should be a binary 3D mask following the corresponding BIDS guidelines on masks. Importantly, the mask should only be stored together with the other results of the dimensionality reduction analysis if it was derived/computed during the respective analysis. If the mask was derived/computed in a different derivative pipeline and applied during dimensionality reduction, it will only be stored in the derivative directory of the respective pipeline. In either case, it SHOULD be indicated in the "sources" field of the metadata JSON sidecar files of the files obtained through the dimensionality reduction as per BIDS guidelines on masks.

```
<source_entities>[_space-<space>][_res-<label>][_den-<label>][_label-<label>][_
desc-<label>]_mask.nii.gz
```

Example

```
sub-01_ses-01_space-MNI152NLin2009cAsym_desc-brain_mask.nii.gz
```

3.11 <source_entities>_model-<label>_mask.json

This file specifies the metadata corresponding to the <source_entities>_model-<label>_mask.<ext> file. Refer to the BIDS guidelines on masks.

```
<source_entities>[_space-<space>][_res-<label>][_den-<label>][_label-<label>][_desc-<label>]_mask.json
```

Example

```
sub-01_ses-01_space-MNI152NLin2009cAsym_desc-brain_mask.json
```

```
{
      “Type”: “Brain”,
      “Sources”: [“sub-01/ses-01/anat/sub-01_ses-01_T1w.nii.gz”],

}
```

4. Examples

4.1 model level meta-data

The example below illustrates how meta-data at the model level is represented. Specifically, this example showcases how model-level data of pipeline containing two models, a spatial ICA and cortical gradients, are represented through the models.tsv and model-<model_id>_description.json files. The first lists the models with their respective model ID, the datatype they were applied to and a short-form description. The second entails detailed information concerning the models listed in models.tsv.

models.tsv

model_id	datatype	description
SpatialICA	func	Blind group ICA with 20 components
CorticalGradients	func	Diffusion-map embedding with 10 components.

model-SpatialICA_bold.json


```
{
    "modelID": "SpatialICA",
    “modelType: “ICA”
    "Algorithm": "InfoMax",
    "AlgorithmParams": {
            "block less than 50485": 129,
            "stopCriteria": 1e-6,
            "weight": 0,
            "irate": 0.005,
            "maxSteps": 512,
            "annealDegrees": 60,
            "momentum": 0,
            "select posact": 0,
            "select sphering": true,
            "select bias": true
        }
    "Mask": "default&icv",
    "NDimensions": 20,
    "Software": "GIFT@v4.0.3.0",
    "SoftwareParameters": {
        "MDLEstimated": false,
        "stabilityType": "regular",
        "backReconAlgo": "GICA",
        "normalization": "z-scores",
        "GroupICAtype": "spatial",
        "PCAtype": "standard",
        "GroupPCAtype": "subject",
        "PCAsteps": 2,
        "numPCstep1": 30,
        "numPCstep2": 20,
        "autofill": false,
        "dataPreprocessingType": "demean",
        "extended": 0,
    }

}
```


model-CorticalGradients_bold.json


```
{
    "modelID": "CorticalGradients",
    “modelType: “Cortical Gradient”
    "Algorithm": "Diffusion embedding",
    "N_dimensions": 10,
    "Software": "mapalign@0.1",
    "SoftwareParameters": {
        "affinity": "cosine",
        "alpha": 0.5,
        "diffusion_time": 0,
        "alignment": "procrustes"
    }
```

4.2 Subject-level cortical gradients

The example below illustrates how whole-brain subject-specific dimensionality reduction-based networks in the form of gradients obtained via diffusion map embedding are represented according to this BEP. In more detail, the Pearson correlation of resting-state time series in volume format was computed between 1000 nodes and diffusion map embedding applied to the resulting correlation/affinity matrix, yielding 10 gradients in the form of spatial maps. Each of these gradients is represented by a corresponding image file, which are stacked across the 4th dimension in one .nii.gz, and an associated meta-date file in json. The model-related information is provided by the respective .tsv and .json files.

```
bids_dataset/derivative/
   preprocessing/
      sub-01/
        func/
          sub-01_task-rest_desc-preproc_bold.nii.gz
          sub-01_task-rest_desc-preproc_bold.json
          sub-01_task-rest_desc-confounds_timeseries.tsv
          sub-01_task-rest_desc-confounds_timeseries.json
   connectivity/
      sub-01/
        func/
          sub-01_task-rest_atlas-1k_desg.nii.gz
          sub-01_task-rest_atlas-1k_desg.tsv
          sub-01_task-rest_atlas-1k_desg.json
          sub-01_task-rest_measure-pearsoncorrelation_relmat.tsv
          sub-01_task-rest_measure-pearsoncorrelation_relmat.json
          sub-01_task-rest_measure-pearsoncorrelation_nodelabels.tsv
          sub-01_task-rest_measure-pearsoncorrelation_nodelabels.json
   cortical_gradientsDE/
      models.tsv
      model-corticalGradientsDE_bold.json
      sub-01/
        func/
          sub-01_task-rest_model-corticalGradientsDE_param-components_boldmap.nii.gz
          sub-01_task-rest_model-corticalGradientsDE_param-components_boldmap.json
```

The files under `/derivatives/preprocessing/sub-01/func` entail the preprocessed fMRI time series in volume format and obtained confounds, as well as the respective meta-data JSON files. The files under `/derivatives/connectivity/sub-01/func/` entail two general types: data regarding discrete segmentation used to downsample voxel-wise data to nodes and data regarding the correlation that was computed between the respective nodes. The first are described following the atlas BEP, ie BEP38, and the second following the connectivity and relationship matrix BEP, ie BEP17. The files under `/derivatives/cortical_gradients/`, ie models.tsv and model-corticalGradientsDE_bold.json, describe details concerning the model(s) used to conduct the dimensionality reduction and the files under `/derivatives/cortical_gradients/sub-01/func/` comprise the files obtained through the dimensionality reduction, ie the diffusion map embedding. Both are further outlined below, starting with the model-related files.

models.tsv

model_id	datatype	description
corticalGradientsDE	func	

model-CorticalGradientsDE_bold.json

```
{
    "modelID": "CorticalGradients",
    “modelType: “Cortical Gradient”
    "Algorithm": "Diffusion embedding",
    "NDimensions": 10,
    "Software": "mapalign@0.1",
    "SoftwareParameters": {
        "affinity": "cosine",
        "alpha": 0.5,
        "diffusion_time": 0,
        "alignment": "procrustes"
    }
}
```

Subsequently, the files obtained through the respective model are further explained.

sub-01_task-rest_model-corticalGradientsDE_param-components_boldmap.nii.gz

The image file contains the 10 gradients obtained through the applied model, ie 10 3D images stacked along the 4th dimension. Their ordering, among further information, is further explained in the respective metadata .json file below.

sub-01_task-rest_model-corticalGradientsDE_param-components_boldmap.json

```
{
      “modelDescriptionPath”:
      “bids:mydataset:derivatives/cortical_gradients/model-corticalGradientsDE_
      description.json”,
      “ComponentsOrdered”: true,
      “ComponentsOrderdDescription”: “Ordered based on eigenvalues,
             descending order”,
      “Sources”:
      [“bids:mydataset:derivatives/preprocessing/sub-01/func/sub-01_task-rest_d
      esc-preproc_bold.nii.gz”]
}
```

4.3 Group level ICA on fMRI - GIFT

The example below illustrates how subject-specific and group files obtained from a whole-brain ICA are represented according to this BEP. Please note that the examples below only show the first subject (sub-01), but additional subjects (e.g., sub-02 and sub-03 or more), and repetitions of sub-01 are needed but are left out to save space. In more detail, the GIFT toolbox was used to apply ICA to fMRI data in volume format, yielding ICA components and respective decomposed time courses (mixing matrix) at the subject level. Additionally, the corresponding files were averaged across subjects to obtain group-level data. Each of these components is represented by a corresponding image file, which are stacked across the 4th dimension in a file ending with .nii.gz, and an associated meta-data file in JSON (per subject and at the group level). The time courses are represented by a .tsv file within which each component is reflected by a column of the same item/number as in the .nii.gz and the number of rows is equal to the number of TRs in the source fMRI data. The model-related information is provided by the respective .tsv and .json files. The mask file is simply a preprocessing step to mask out noise. The desc-<type> denotes different types of GIFT outputs, where the group level most commonly uses stat-mean, stat-std and stat-tvalue, but also supports desc-agg and other outputs, such as outputs from intermediate PCA steps. At the subject level, GIFT may also output different types of results, including the most commonly used desc-cal, which has other less frequently used outputs, such as desc-backrecon, which is the raw ICA output (without calibration). GIFT may produce results for multiple sessions and if that is the case, one may simply add the ses-01.

```
bids_dataset/derivative/
   preprocessing/
       sub-01/
         func/
           sub-01_task-rest_desc-preproc_bold.nii.gz
           sub-01_task-rest_desc-preproc_bold.json
           sub-01_task-rest_desc-confounds_timeseries.tsv
           Sub-01_task-rest_desc-confounds_timeseries.json
           …
   GroupSpatialICA/
       models.tsv
       model-GroupSpatialICA_bold.json
       model-GroupSpatialICA_participants.tsv
       group/
         func/
          group_task-rest_model-GroupSpatialICA_param-components_stat-mean_boldmap.nii.gz
          group_task-rest_model-GroupSpatialICA_param-components_stat-mean_boldmap.json
          group_task-rest_model-GroupSpatialICA_param-components_stat-std_boldmap.nii.gz
          group_task-rest_model-GroupSpatialICA_param-components_stat-std_boldmap.json
          group_task-rest_model-GroupSpatialICA_param-components_stat-tvalue_boldmap.nii.gz
          group_task-rest_model-GroupSpatialICA_param-components_stat-tvalue_boldmap.json
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-mean_boldmap.tsv
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-mean_boldmap.json
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-std_boldmap.tsv
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-std_boldmap.json
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-tvalue_boldmap.tsv
          group_task-rest_model-GroupSpatialICA_param-mixing_stat-tvalue_boldmap.json
          group_task-rest_model-GroupSpatialICA_mask.nii.gz

       sub-01/
         func/

sub-01_task-rest_model-GroupSpatialICA_param-components_desc-backrecon_boldmap.nii.gz
          sub-01_task-rest_model-GroupSpatialICA_param-components_desc-backrecon_boldmap.json
          sub-01_task-rest_model-GroupSpatialICA_param-components_desc-cal_boldmap.nii.gz
          sub-01_task-rest_model-GroupSpatialICA_param-components_desc-cal_boldmap.json
          sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc-backrecon_boldmap.tsv
          sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc-backrecon_boldmap.json
          sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc-cal_boldmap.tsv
          sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc-cal_boldmap.json
```

The files under `/derivatives/preprocessing/sub-01/func` entail the preprocessed fMRI time series in volume format and obtained confounds, as well as the respective meta-data JSON files. The files under `/derivatives/GroupSpatialICA/` entail three general types: data regarding the applied model, i.e. models.tsv and model-GroupSpatialICA_bold.json, data obtained from the applied model on the subject level and group level. The files under `/derivatives/GroupSpatialICA/sub-01/func/` comprise the former and thus files obtained through the dimensionality reduction, ie the ICA and files under `/derivatives/GroupSpatialICA/group/func/` comprise the latter and thus files obtained via averaging the subject level files. All file types are further outlined below, starting with the model-related files.

models.tsv

model_id	datatype	description
GroupSpatialICA	func	Blind ICA example with 20 components

In the above file structure, an example of the `model-GroupSpatialICA_description.json` would be as follows:


```
{
    "model": "GroupSpatialICA",
    “modelType: “ICA”
    "Algorithm": "InfoMax",
    "AlgorithmParams": {
            "block less than 50485": 129,
            "stopCriteria": 1e-6,
            "weight": 0,
            "irate": 0.005,
            "maxSteps": 512,
            "annealDegrees": 60,
            "momentum": 0,
            "select posact": 0,
            "select sphering": true,
            "select bias": true
        }
    "Mask": "default&icv",
    "NDimensions": 20,
    "Software": "GIFT@v4.0.3.0",
    "SoftwareParameters": {
        "MDLEstimated": false,
        "stabilityType": "regular",
        "backReconAlgo": "GICA",
        "normalization": "z-scores",
        "GroupICAtype": "spatial",
        "PCAtype": "standard",
        "GroupPCAtype": "subject",
        "PCAsteps": 2,
        "numPCstep1": 30,
        "numPCstep2": 20,
        "autofill": false,
        "dataPreprocessingType": "demean",
        "extended": 0,
    }
}
```

Subsequently, the files obtained through the respective model, ie ICA, are further explained, starting with subject level files.

sub-01_task-rest_model-GroupSpatialICA_param-components_desc-cal_boldmap.nii.gz

The image file contains the 20 spatial ICA components obtained through the applied model, ie 20 3D images stacked along the 4th dimension. Their ordering, among further information, is further explained in the respective metadata .json file below. The calibrated description (desc-cal) is processed from the back reconstructed (desc-backrecon) spatial components.

sub-01_task-rest_model-GroupSpatialICA_param-components_desc-cal_boldmap.json

```
{
	"modelDescriptionPath":
	"bids:mydataset:derivatives/GroupSpatialICA/model-GroupSpatialICA_descrip
	tion.json",
	"ComponentsOrdered": true,
	"ComponentsOrderdDescription": "Ordered based on variance explained,
		descending order",
	"Sources":
	["bids:mydataset:derivatives/preprocessing/sub-01/func/sub-01_task-rest_d
	esc-preproc_bold.nii.gz"]
}
```

sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc_cal_boldmap.tsv

The example below shows the first row with the component ID number for the 20 independent components. The following rows show examples of calibrated ICA time courses at the subject level, with the intensities from the first TR for each component in the second row, the second TR in the third row, and the last TR's ICA intensities in the last row. Usually, for fMRI, there are more than 100 TRs. The calibrated description (desc-cal) is processed from the back reconstructed (desc-backrecon) time courses.

1	2	…	20
-0.1968	-1.0013	…	2.4545
-0.1820	-2.223	…	0.0072
…	…	…	…
-0.5305	0.3486	…	-1.3401

sub-01_task-rest_model-GroupSpatialICA_param-mixing_desc-cal_boldmap.json

```
{
	"modelDescriptionPath":
	"bids:mydataset:derivatives/GroupSpatialICA/model-GroupSpatialICA_description.json",
	"ComponentsOrdered": true,
	"ComponentsOrderdDescription": "Ordered based on variance explained,
		descending order",
	"Sources":
	["bids:mydataset:derivatives/preprocessing/sub-01/func/sub-01_task-rest_desc-preproc_bold.nii.gz"]
}
```

Finally, the files at the group level, obtained through averaging subject-level files, are further explained.

group_task-rest_model-GroupSpatialICA_stat-tvalue_mixing_boldmap.tsv

In the above file structure, an example of the group_task-rest_model-GroupSpatialICA_stat-tvalue_mixing_boldmap.tsv would be as follows: having numbers of rows as many as there are time points in the raw fMRI dataset. The first row contains the items for each of the 20 components that were used in this example.

1	2	…	20
1.9785	-1.2646	…	1.9403
0.4579	-0.5089	…	1.4905
…	…	…	…
-1.5648	0.3993	…	-0.2300

4.4 Group level ICA on fMRI - FSL Melodic

Analogous as section 4.3, but is designed for FSL software. Each network is saved in its own *component-item-probmap2_boldmap.nii.gz file.

```
bids_dataset/derivative/
   preprocessing/
        sub-01/
          func/
            sub-01_task-rest_desc-preproc_bold.nii.gz
            sub-01_task-rest_desc-preproc_bold.json
            sub-01_task-rest_desc-confounds_timeseries.tsv
            sub-01_task-rest_desc-confounds_timeseries.json
   GroupSpatialICA/
        models.tsv
        model-GroupSpatialICA_bold.json
        model-GroupSpatialICA_participants.json
        group-01/
          func/
                group-01_task-rest_model-GroupSpatialICA_mask.nii.gz
                group-01_task-rest_model-GroupSpatialICA_mask.json
                group-01_task-rest_model-GroupSpatialICA_stat-mean_param-components_boldmap.nii.gz
                group-01_task-rest_model-GroupSpatialICA_stat-mean_param-mixing_boldmap.tsv
                group-01_task-rest_model-GroupSpatialICA_stat-std_param-components_boldmap.nii.gz
                group-01_task-rest_model-GroupSpatialICA_stat-std_param-mixing_boldmap.tsv
                group-01_task-rest_model-GroupSpatialICA_stat-tvalue_param-components_boldmap.nii.gz
                group-01_task-rest_model-GroupSpatialICA_stat-tvalue_param-mixing_boldmap.tsv
        sub-01/
          func/
                sub-01_task-rest_model-GroupSpatialICA_param-mixing_boldmap.tsv
                sub-01_task-rest_model-GroupSpatialICA_param-component_item-probmap1_boldmap.nii.gz
                sub-01_task-rest_model-GroupSpatialICA_param-component_item-probmap2_boldmap.nii.gz
                …
                sub-01_task-rest_model-GroupSpatialICA_param-component_item-probmapN_boldmap.nii.gz
                sub-01_task-rest_model-GroupSpatialICA_model.json
```

4.5 Single Subject ICA on EEG - EEGIFT

The example below illustrates how a single subject processed using EEGIFT (temporal ICA) may be represented according to this BEP. The bids_dataset/raw/sub-01/eeg/sub-01_task-rest_channels.tsv file entails a raw BIDS file referenced from columns in derivatives measures. The files under `/derivatives/preprocessing/sub-01/*` entail the preprocessed EEG data, as well as the respective meta-data JSON files. The files at the root of `/derivatives/TemporalICA/` entail models.tsv and model-<model-id>.json, describing the model used and the model-<model-id>_participants.tsv lists all included subjects. Under `/derivatives/TemporalICA/sub-01/eeg`, the result from the temporal ICA applied on EEG data, yielding ICA components, represented in .tsv files, labeled with param-components (time courses), and an associated metadata file in JSON sidecar. The topology map is represented by the values within each column of .tsv files labeled param-mixing, having a matrix where each row represents an EEG channel and has a header number (column), matching the component numbers in the param-components file. These ICA components (time-courses) and the mixing matrix should have desc-backrecon and desc-cal labels, depending on the level of post-processing. Unlike fMRI, EEG does not have a brain mask and of course, the resulting components are a product of temporal ICA as opposed to spatial ICA, which is commonly used for fMRI.

```
bids_dataset/raw/sub-01/eeg/sub-01_task-rest_channels.tsv
bids_dataset/derivatives/
   preprocessing/
       sub-01/
           eeg/
             sub-01_task-rest_desc-preproc_eeg.eeg
             sub-01_task-rest_desc-preproc_eeg.json
   TemporalICA/
       models.tsv
       model-TemporalICA_eeg.json
       model-TemporalICA_participants.tsv
       sub-01/
           eeg/
             sub-01_task-rest_model-TemporalICA_param-components_desc-backrecon_eegmap.tsv
             sub-01_task-rest_model-TemporalICA_param-components_desc-backrecon_eegmap.json
             sub-01_task-rest_model-TemporalICA_param-mixing_desc-backrecon_eegmap.tsv
             sub-01_task-rest_model-TemporalICA_param-mixing_desc-backrecon_eegmap.json
             sub-01_task-rest_model-TemporalICA_param-components_desc-cal_eegmap.tsv
             sub-01_task-rest_model-TemporalICA_param-components_desc-cal_eegmap.json
             sub-01_task-rest_model-TemporalICA_param-mixing_desc-cal_eegmap.tsv
             sub-01_task-rest_model-TemporalICA_param-mixing_desc-cal_eegmap.json
```

All file types are further outlined below, starting with the model-related files.

models.tsv

model_id	datatype	description
TemporalICA	eeg	Blind temporal ICA set to 20 components

model-TemporalICA_description.json

In the above file structure, an example of the model-TemporalICA_description.json would be as follows:

```
{
    "model": "TemporalICA",
    "modelType: "ICA"
    "Algorithm": "InfoMax",
    "AlgorithmParams": {
            "block less than 50485": 261,
            "stopCriteria": 1e-6,
            "weight": 0,
            "irate": 0.005,
            "maxSteps": 512,
            "annealDegrees": 60,
            "momentum": 0,
            "select posact": 0,
            "select sphering": true,
            "select bias": true
        }
    "NDimensions": 20,
    "Software": "GIFT@v4.0.3.0",
    "SoftwareParameters": {
        "MDLEstimated": false,
        "stabilityType": "regular",
        "backReconAlgo": "GICA",
        "normalization": "z-scores",
        "GroupICAtype": "temporal",
        "PCAtype": "standard",
        "GroupPCAtype": "subject",
        "PCAsteps": 2,
        "numPCstep1": 30,
        "numPCstep2": 20,
        "autofill": false,
        "dataPreprocessingType": "demean",
        "extended": 0,
    }

}
```

model-TemporalICA_participants.tsv

Simply a list of subjects included in the model and may look as following for a single subject.

participant_id
01

Subsequently, the files obtained through the respective model, ie temporal ICA, are further explained.

sub-01_task-rest_model-TemporalICA_param-components_desc-backrecon_eegmap.tsv
Since temporal ICA is the transpose of spatial ICA, an example of the sub-01_task-rest_model-TemporalICA_param-components_desc-backrecon_eegmap.tsv, in the file structure above, contains the time-courses, having numbers of rows as many as there are time points in the eeg dataset plus a header (first row). The header contains the item numbers for each of the 20 components that were used in this example.

1	2	…	20
1.9785	-1.2646	…	1.9403
0.4579	-0.5089	…	1.4905
…	…	…	…

sub-01_task-rest_model-TemporalICA_param-components_desc-backrecon_eegmap.json

The sidecar to the .tsv file described in the section above holds the order of the components, among other information. Examples of content may be the following:

```
{
       "modelDescriptionPath":
       "bids:mydataset:derivatives/TemporalICA/model-TemporalICA_description.json",
       "ComponentsOrdered": true,
       "ComponentsOrderdDescription": "Ordered based on variance explained,
              descending order",
       "Sources":
       ["bids:mydataset:derivatives/preprocessing/sub-01/eeg/
             /sub-01_task-rest_desc-preproc_eeg.eeg"]
}
```

sub-01_task-rest_model-TemporalICA_param-mixing_desc-backrecon_eegmap.tsv

The file containing the EEG topology information is formatted as the matrix (table) below, where each column, having a header number, is coupled with the component item number. The table rows are the eeg channels according to the first two columns, where the channel_name should be synced with the name column in the raw BIDS file bids_dataset/raw/sub-01/eeg/<source_entities>_channels.tsv. In case <source_entities>_channels.tsv does not have information in the name column, the channel_order has to be matched with the row order of <source_entities>_channels.tsv, which is the channel index, in order to backtrack where the electrodes are positioned on the head to create a topology map for each component. The rows under the component item columns, represent the intensity from each channel, where each channel is its own row. An example, using 20 components with a 64 channel/electrode EEG-cap-profile, results in a table of 65 rows and 22 columns, looking similar to below:

channel_order	channel_name	1	2	…	20
1	Fp1	-0.6990	0.0776	…	0.0679
2	Fpz	0.6598	0.1169	…	0.1418
…	…	…	…	…	…
64	EYE	0.2283	-0.1763	…	0.0835

sub-01_task-rest_model-TemporalICA_param-mixing_desc-backrecon_eegmap.json

Sidecar to .tvs file described in the section above, having an example as follows:

```
{
      “modelDescriptionPath”:
      “bids:mydataset:derivatives/TemporalICA/model-TemporalICA_description.jso
      n”,
      “ComponentsOrdered”: true,
      “ComponentsOrderdDescription”: “Ordered based on variance explained,
             descending order”,
      “Sources”:
      [“bids:mydataset:derivatives/preprocessing/sub-01/eeg/
            /sub-01_task-rest_desc-preproc_eeg.eeg”]
}
```

Additional files with desc-cal suffix

Additional files with the desc-cal suffix are analogous to files with desc-backrecon suffix, with the only difference being that it represents data that is additionally post-processed after the desc-backrecon files.

4.6 Group ICA using Multisession EEG - EEGIFT:

The following example is analogous to section 4.5, with the exception that multiple sessions and multiple (NN) subjects are processed in a group model. Even though this example has multiple sessions, the analogy includes that the session files under derivatives/GroupTemporalICA/sub-01/ses-0X/eeg/, are basically the same as under section 4.5, only that we now have separate ICA results for each of the two sessions. This is in agreement with the original BIDS standard specifications. Further on, we will focus on what differentiates this section from section 4.5, which is that we have multiple files under folder derivatives/GroupTemporalICA/group-01/eeg/ (group folder). In the group folder, the files, including the entities ses-01, param-components and stat-mean, are the mean of the files with entities sub-0X, ses-01, param-components, desc-cal located across all single subject folders derivatives/GroupTemporalICA/sub-0X/ses-01/eeg/ within this model. Analogously the group folder files, including the entities ses-02, param-components and stat-mean, are the mean of the files with entities sub-0X, ses-02, param-components, desc-cal located in folders derivatives/GroupTemporalICA/sub-0X/ses-02/eeg/. Similarly, the group folder files having the ses-0X, param-components, and stat-std entities are analogous to how the files with entity stat-mean operated, with the difference in that its content is the standard deviation as opposed to the mean now. Again, the group folder files having the ses-0X, param-components and stat-tvalue entities are the t-scores from each subject instead of mean or standard deviations.
Under the group folder, we have only written about files having the entity param-components and in this folder, each file with such an entity has another file that matches its filename, except that the param-components entity is replaced with a param-mixing entity. All the files with param-mixing entities are calculated in an analogous way as the files with the param-components, only that they deal with time courses (param-mixing) as opposed to spatial maps (param-components). Finally the files under the group folder that are missing any entity with the prefix ses are calculated analogously as the files with the ses prefix, but with the difference that they encompass the results across all the sessions (in this case across session 1 and session 2), getting its results from all the subjects and sessions folders according with following paths: derivatives/GroupTemporalICA/sub-0X/ses-0Y/eeg, where X iterate over all subject numbers and Y iterates over all session numbers.
Since EEG time courses may be extensive, the tsv may be compressed with gzip.

```
bids_dataset/raw/sub-01/ses-01/eeg/sub-01_ses-01_task-rest_channels.tsv
bids_dataset/derivatives/
   preprocessing/
      sub-01/
           ses-01/
              eeg/
                sub-01_ses-01_task-rest_desc-preproc_eeg.eeg
                Sub-01_ses-01_task-rest_desc-preproc_eeg.json
           ses-02/
              eeg/
                sub-01_ses-02_task-rest_desc-preproc_eeg.eeg
                Sub-01_ses-02_task-rest_desc-preproc_eeg.json
      sub-NN/…
   GroupTemporalICA/
      models.tsv
      model-GroupTemporalICA_eeg.json
      model-GroupTemporalICA_participants.tsv
      group-01/
          eeg/
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.tsv
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.json
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.tsv.gz
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.json
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.tsv
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.json
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.tsv.gz
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.json
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.tsv
            ses-01_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.json
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.tsv.gz
            ses-01_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.tsv
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.tsv.gz
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.tsv
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.tsv.gz
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.tsv
            ses-02_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.json
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.tsv.gz
            ses-02_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.json
            task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.tsv
            task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.json
            task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.tsv.gz
            task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.json
            task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.tsv
            task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.json
            task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.tsv.gz
            task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.json
            task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.tsv
            task-rest_model-GroupTemporalICA_param-components_stat-tvalue__eegmap.json
            task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.tsv.gz
            task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.json
      sub-01/
       ses-01/
         eeg/
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-components_desc-backrecon_eegmap.tsv
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-components_desc-backrecon_eegmap.json
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-components_desc-cal_eegmap.tsv
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-components_desc-cal_eegmap.json
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-mixing_desc-backrecon_eegmap.tsv.gz
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-mixing_desc-backrecon_eegmap.json
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-mixing_desc-cal_eegmap.tsv.gz
          sub-01_ses-01_task-rest_model-GroupTemporalICA_param-mixing_desc-cal_eegmap.tsv.json
       ses-02/
         eeg/
          sub-01_ses-02_task-rest_model-GroupTemporalICA_param-components_desc-backrecon_eegmap.tsv
          sub-01_ses-02_task-rest_model-GroupTemporalICA_param-components_desc-backrecon_eegmap.json
          sub-01_ses-02_task-rest_model-GroupTemporalICA_param-components_desc-cal_eegmap.tsv
          sub-01_ses-02_task-rest_model-GroupTemporalICA_param-components_desc-cal_eegmap.json

        sub-01_ses-02_task-rest_model-GroupTemporalICA_param-mixing_desc-backrecon_eegmap.tsv.g
        z

        sub-01_ses-02_task-rest_model-GroupTemporalICA_param-mixing_desc-backrecon_eegmap.json
```

```
sub-01_ses-02_task-rest_model-GroupTemporalICA_param-mixing_desc-cal_eegmap.tsv.gz
sub-01_ses-02_task-rest_model-GroupTemporalICA_param-mixing_desc-cal_eegmap.tsv.json
```

In the above file structure most files are analogous with section 4.5, including the files

```
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.tsv,
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-mean_eegmap.json,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.tsv.gz,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-mean_eegmap.json.
model-GroupTemporalICA_participants.tsv
```

Similarly, this section (4.6) is related with section 4.3 for the following files:

```
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.tsv,
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-std_eegmap.json,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.tsv.gz,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-std_eegmap.json,
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.tsv,
ses-0X_task-rest_model-GroupTemporalICA_param-components_stat-tvalue_eegmap.json,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.tsv.gz,
ses-0X_task-rest_model-GroupTemporalICA_param-mixing_stat-tvalue_eegmap.json
```

In the above file structure, an example of the model-GroupTemporalICA_description.json in the EEGIFT case would be as follows:


```
{
      "model": "InfoMax-attempt2",
      "Software": "GIFT@v4.0.3.0",
      “modelType”: “ICA”,
      "Algorithm": "InfoMax",
      "SoftwareParameters": {
            "MDLEstimated": false,
            "stabilityType": "regular",
            "backReconAlgo": "GICA",
            "normalization": "z-scores",
            "GroupICAtype": "Temporal",
            "PCAtype": "standard",
            "CroupPCAtype": "subject",
            "PCAsteps": 2,
            "numPCstep1": 0,
            "numPCstep2": 0,
            "autofill": false,
            "dataPreprocessingType": "demean",
            "extended": 0,
            }
      "AlgorithmParams": {
            "block less than 204800": 129,
            "stopCriteria": "1e-6",
            "weight": 0,
            "irate": 0.005,
            "maxSteps": 512,
            "annealBetween0to1": 0.9,
            "annealDegrees": 60,
            "momentum": 0,
            "select posact": 0,
            "select sphering": true,
            "select bias": true
      }
```

}

With models.tsv containing descriptions of the various models run on the dataset and looking like:

model_id	datatype	description
GroupTemporalICA	eeg	Group analysis with 2 sessions, using temporal ICA to reduce data into 20 dimensions
…	…	…

4.7 Group level ICA on PET and anat

In the case of data types anat and in most cases for data types PET, sources are 3D images of several subjects. These source 3D images are assumed to have already undergone pre-processing (see BEP011 and BEP023 for anat and PET preprocessing derivatives, respectively).

The outputs of group-level ICA on PET and anat data will be the same as for the func single-subject example, but they will have to be stored at the group level only and not in the sub-<label> folders. Note that the mixing matrix for this kind of data does not contain a time series but only subject loadings.

Group level ICA PET example:
An example is provided for a group-level ICA analysis on pre-processed PET data across 10 control subjects. The ICA analysis is run with an InfoMax algorithm, setting the number of components to 4. The software used is GIFT, which uses a mask to denoise the ICA process, delineating the voxels into components. The 4 extracted components are all deemed of interest by the experimenter (i.e. no “noise” components the experimenter deems should be dropped).

The directory structure for this example would look like:

```
bids_dataset/derivatives/
      SUVprep/

            descriptions.tsv
            sub-01/pet/
                  sub-01_desc-preproc_pet.nii.gz
                  sub-01_desc-preproc_pet.json
                  sub-01_desc-proc_meas-suv.nii.gz
                    sub-01_desc-proc_meas-suv.json
            sub-02/pet/
                  sub-02_desc-preproc_pet.nii.gz
                  sub-02_desc-preproc_pet.json
                  sub-02_desc-proc_meas-suv_pet.nii.gz
                  sub-02_desc-proc_meas-suv_pet.json
            …
      GroupICA/
            models.tsv
            model-sptaialICAPET_pet.json
            model-spatialICAPET_participants.tsv
            group-01/pet/
                  group-01_model-spatialICAPET_param-components_desc-backrecon_pe
            tmap.nii.gz
                  group-01_model-spatialICAPET_param-components_desc-backrecon_pet
            map.json
                  group-01_model-spatialICAPET_param-mixing_desc-backrecon_petmap.tsv
                  group-01_model-spatialICAPET_param-mixing_desc-backrecon_petmap.json
                  group-01_model-spatialICAPET_mask.nii.gz
                  group-01_model-spatialICAPET_mask.json
```

Where the model.tsv (required) file would look like this:

model_id	datatype	description
spatialICAPET	pet	

The model_description.json would look like:

```
{
      “modelID”: “spatialICAPET”,
      “modelType”: “ICA”
      “Algorithm”: “InfoMax”,
      “AlgorithmParams”: {
                  "block less than 50485": 129,
                  "stopCriteria": 1e-6,
                  "weight": 0,
                  "irate": 0.005,
                  "maxSteps": 512,
                  "annealDegrees": 60,
                  "momentum": 0,
                  "select posact": 0,
                  "select sphering": true,
                  "select bias": true
            },
      “NDimensions”: 4,
      “Software”: “GIFT@v4.0.3.0”,
}
```

The model-spatialICAPET_participants.tsv file would look like:

participant_id
01
02
03
04
05
06
07
08
09
10

The `group-01_model-spatialICAPET_param-components_petmap.nii.gz` contains the spatial maps (components), generated by the software. In this case, it is represented by a 4D image, where the 4th dimension is components.

The `group-01_model-spatialICAPET_param-components_petmap.json` describes the components in the `group-01_model-spatialICAPET_param-components_petmap.nii.gz` file and would look like this:

```
{
"modelDescriptionPath":
"bids_dataset/derivatives/GroupICA/model-sptaialICAPET_description.json",
"ComponentsOrdered": true,
"ComponentsOrderedDescription": "Ordered by variance explained,
                                 descending order",
"Sources":
["bids_dataset/derivatives/SUVprep/sub-01/pet/sub-01_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-02/pet/sub-02_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-03/pet/sub-03_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-04/pet/sub-04_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-05/pet/sub-05_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-06/pet/sub-06_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-07/pet/sub-07_meas-suv_desc-pro_petc.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-08/pet/sub-08_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-09/pet/sub-09_meas-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-10/pet/sub-10_meas-suv_desc-proc_pet.nii.gz"]
}
```

In the above file structure, an example of the `group-01_model-spatialICAPET_param-mixing_desc-backrecon_petmap.tsv` would be as follows: having numbers of rows as many as there are PET subjects for 4 components. The first row with 1,2,3,4 are the component items:

1	2	3	4
1.9785	-1.2646	0.9221	1.9403
0.4579	-0.5089	0.6199	1.4905
…	…	…	…
-1.5648	0.3993	-1.6223	-0.2300

The `group-01_model-spatialICAPET_param-mixing_desc-backrecon_petmap.json` file would look like this, with the meta-data of the 4 components, all deemed of interest by the experimenter:

```
{
"modelDescriptionPath":
"bids_dataset/derivatives/GroupICA/model-sptaialICAPET_description.json",
"ComponentsOrdered": true,
"ComponentsOrderdDescription": "Ordered by variance explained,
                               descending order",
"Components Type": "Participants"
"Sources":
["bids_dataset/derivatives/SUVprep/sub-01/pet/sub-01_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-02/pet/sub-02_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-03/pet/sub-03_meas-suvkm-suv_desc-proc_pet.nii.gz",
```

```
"bids_dataset/derivatives/SUVprep/sub-04/pet/sub-04_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-05/pet/sub-05_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-06/pet/sub-06_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-07/pet/sub-07_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-08/pet/sub-08_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-09/pet/sub-09_meas-suvkm-suv_desc-proc_pet.nii.gz",
"bids_dataset/derivatives/SUVprep/sub-10/pet/sub-10_meas-suvkm-suv_desc-proc_pet.nii.gz"]
}
```

The `group-01_model-spatialICAPET_mask.nii.gz` (OPTIONAL) file would be the output mask file, , indicating the voxels where the dimensionality reduction took place, generated by the software.

The `group-01_model-spatialICAPET_petmap.json` (OPTIONAL

Legacy

Group level ICA anat example:

To be implemented**The ICA of anatomical files are analogous with the PET processing, receiving following file structure.**

```
bids_dataset/derivative/
   preprocessing/
       sub-01/
         anat/
           sub-01_desc-preproc_T1w.nii.gz
           sub-01_desc-preproc_T1w.json
       sub-02/
         anat/
           sub-02_desc-preproc_T1w.nii.gz
           sub-02_desc-preproc_T1w.json
           …
   GroupSpatialICAt1w/
       models.tsv
       model-GroupSpatialICAt1w_description.json
       model-GroupSpatialICAt1w_participants.tsv
       group/
         anat/
          group-01_model-GroupSpatialICAt1w_param-components_desc-backrecon_T1wmap.nii.gz
          group-01_model-GroupSpatialICAt1w_param-components_desc-backrecon_T1wmap.json
          group-01_model-GroupSpatialICAt1w_param-mixing_desc-backrecon_T1wmap.tsv
          group-01_model-GroupSpatialICAt1w_param-mixing_desc-backrecon_T1wmap.json
          group-01_model-GroupSpatialICAt1w_mask.nii.gz
          group-01_model-GroupSpatialICAt1w_mask.json
```

The content of these files would be analogous to what shown in the Group level ICA pet example above.

```
models.tsv
group-01/
     anat/
       model-spatialICAanat/
             model_description.json
             group-01_model-spatialICAanat_param-components_mapmfp.nii.gz
             group-01_model-spatialICAanat_param-mixing_mapmfp.tsv
             group-01_model-spatialICAanat_mask.nii.gz
             group-01_model-spatialICAanat_mapmfp.json
```

4.8 Group level ICA on fMRI - variation of GIFT

Analogous as section 4.3, using a slightly different file name approach, but does not show where to save the individual ICA files.

```
bids_dataset/derivative/
   preprocessing/
       sub-01/
         func/
           sub-01_task-rest_desc-preproc_bold.nii.gz
           sub-01_task-rest_desc-preproc_bold.json
           sub-01_task-rest_desc-confounds_timeseries.tsv
           sub-01_task-rest_desc-confounds_timeseries.json
   GroupSpatialICAnoOutliers/
       models.tsv
       model-GroupSpatialICAnoOutliers.json
       group-01/
```

```
    func/
       model-GroupSpatialICAnoOutliers/
       group-01_task-rest_model-GroupSpatialICANoOutliers_mapmfp.json
       group-01_task-rest_model-GroupSpatialICANoOutliers_mask.nii.gz
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-mean_param-components_mapmfp.ni
       i.gz
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-mean_param-mixing_mapmf
p.tsv
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-std_param-components_mapm
       fp.nii .gz
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-std_param-mixing_mapmfp
.tsv
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-tvalue_param-components_map
       mfp.nii.gz
       group-01_task-rest_model-GroupSpatialICANoOutliers_stat-tvalue_param-mixing_map
mfp.tsv
       group-01_task-rest_model-GroupSpatialICANoOutliers_desc-subjects_mapmfp.tsv
sub-01/
    func/
       sub-01_task-rest_model-GroupSpatialICANoOutliers_param-components_mapmfp.nii.gz
       sub-01_task-rest_model-GroupSpatialICANoOutliers_param-components_mapmfp.json
       sub-01_task-rest_model-GroupSpatialICANoOutliers_param-mixing_mapmfp.tsv
       sub-01_task-rest_model-GroupSpatialICANoOutliers_param-mixing_mapmfp.json
```

4.9 Single Subject ICA with Component Classification with ICA-AROMA

Open questions:

1. When the decomposition (MELODIC) and classification (AROMA) are different, what should the model be named?
    a. I went with model-AROMA instead of MELODIC, even though AROMA is just the classification step.
2. With classification approaches, there may be many component-wise metrics that would be more readable in a TSV than in the JSON. This applies to VarianceExplained as well.

```
fmripost-aroma/
  dataset_description.json
  models.tsv
  model-AROMA_bold.json
  sub-01.html
  sub-01/
    func/
      sub-01_task-rest_model-AROMA_desc-aroma_metrics.json
      sub-01_task-rest_model-AROMA_desc-aroma_metrics.tsv
      sub-01_task-rest_model-AROMA_desc-confounds_timeseries.tsv
      sub-01_task-rest_model-AROMA_space-MNI152NLin6Asym_res-2_param-components_boldmap.nii.gz
      sub-01_task-rest_model-AROMA_space-MNI152NLin6Asym_res-2_param-mixing_timeseries.tsv
      sub-01_task-rest_space-MNI152NLin6Asym_res-2_desc-denoised+aggr_bold.nii.gz
      sub-01_task-rest_space-MNI152NLin6Asym_res-2_desc-denoised+nonaggr_bold.nii.gz
      sub-01_task-rest_space-MNI152NLin6Asym_res-2_desc-denoised+orthaggr_bold.nii.gz
```

4.10 Single Subject ICA with Component Classification with tedana

Open questions:

1. ICA typically follows PCA. How should the model be defined in that case?
    a. I (Taylor Salo) went with tedana+PCA for now, but could change to PCA only.

```
fmripost-tedana/
  dataset_description.json
  models.tsv
  model-tedana+PCA_bold.json  # Describes the PCA used to whiten the data before ICA
  model-tedana_bold.json
  sub-01.html
  sub-01/
    func/
      # PCA derivatives
      sub-01_task-rest_model-tedana+PCA_param-components_stat-z_boldmap.nii.gz
      sub-01_task-rest_model-tedana+PCA_param-mixing_timeseries.tsv
      sub-01_task-rest_model-tedana+PCA_metrics.tsv
      sub-01_task-rest_model-tedana+PCA_metrics.json
      sub-01_task-rest_model-tedana+PCA_desc-averagingWeights_components.nii.gz
      sub-01_task-rest_model-tedana+PCA_stat-F_desc-S0fit_boldmap.nii.gz
      sub-01_task-rest_model-tedana+PCA_stat-F_desc-T2fit_boldmap.nii.gz

      # ICA derivatives
      sub-01_task-rest_model-tedana_param-components_boldmap.nii.gz
      sub-01_task-rest_model-tedana_param-mixing_timeseries.tsv
      sub-01_task-rest_model-tedana_param-mixing_desc-orth_timeseries.tsv
      sub-01_task-rest_model-tedana_param-components_stat-z_boldmap.nii.gz
      sub-01_task-rest_model-tedana_metrics.tsv
      sub-01_task-rest_model-tedana_metrics.json
      sub-01_task-rest_model-tedana_desc-crosscomponent_metrics.json
      sub-01_task-rest_model-tedana_param-components_desc-accepted_boldmap.nii.gz

sub-01_task-rest_model-tedana_param-components_stat-z_desc-accepted_stat-z_boldmap.nii.gz

      # ICA derivatives related to classification (outside BEP scope probably)
      sub-01_task-rest_model-tedana_desc-averagingWeights_components.nii.gz
      sub-01_task-rest_model-tedana_statustable.tsv
      sub-01_task-rest_model-tedana_decisiontree.json
      sub-01_task-rest_model-tedana_stat-F_desc-S0fit_boldmap.nii.gz
      sub-01_task-rest_model-tedana_stat-F_desc-T2fit_boldmap.nii.gz

      # Other derivatives (multi-echo-related and outside BEP scope)
      sub-01_task-rest_desc-optcom_bold.nii.gz
      sub-01_task-rest_desc-confounds_timeseries.tsv
      sub-01_task-rest_desc-denoised_bold.nii.gz
      sub-01_task-rest_S0map.nii.gz
      sub-01_task-rest_T2starmap.nii.gz
      sub-01_task-rest_desc-adaptiveGoodSignal_mask.nii.gz
      sub-01_task-rest_desc-limited_S0map.nii.gz
      sub-01_task-rest_desc-limited_T2starmap.nii.gz
      sub-01_task-rest_desc-optcom+accepted_bold.nii.gz
      sub-01_task-rest_desc-optcom+rejected_bold.nii.gz
      sub-01_task-rest_desc-optcom+whitened_bold.nii.gz
      sub-01_task-rest_desc-rmse_boldmap.nii.gz
      sub-01_task-rest_registry.json
```

Legacy

Spatial maps

The spatial maps, which may represent either the components or mixing matrix, can be represented EITHER as a 4D image, where the 4th dimension is components, OR as a series of 3D images, which are enumerated using the `item-` entity. Mixing matrices may be omitted for algorithms that are temporal decompositions with no spatial component, such as CompCor variants.

Whereas in the case of spatial decomposition, the spatial map represents the components, and in the case of temporal decomposition the spatial maps represent the mixing matrix.

Spatial decomposition fMRI example:

```
<source_entities>_model-<label>_param-components_map.nii.gz
```

Temporal decomposition example:

```
<source_entities>_model-<label>_param-mixing_map.tsv
```

```
<pipeline_name>/
  models.tsv
  [models.json]
  sub-<label>/
     <datatype>/
       model-<label>/
         <source_entities>_model-<label>_param-<mixing|components>_map.<ext>
         <source_entities>_model-<label>_param-<mixing|components>_map.<ext>
         <source_entities>_model-<label>_map.json
         [<source_entities>_model-<label>_mask.<ext>]
         model_description.json
```

```
models.tsv
sub-01/
    func/
        model-cortical gradientsDE/
             model_description.json
             sub-01_task-rest_model-corticalgradientsDE_param-mixing_map.tsv
             sub-01_task-rest_model-corticalgradientsDE_param-components_map.tsv
             sub-01_task-rest_model-corticalgradientsDE_map.json
```

```
models.tsv
group-01/
      func/
            model-GroupSpatialICA/
                  model_description.json
                  task-rest_model-GroupSpatialICA_map.json
                  task-rest_model-GroupSpatialICA_desc-defaulticv_mask.nii.gz

              task-rest_model-GroupSpatialICA_stat-mean_param-components_map.ni
              i.gz

              task-rest_model-GroupSpatialICA_stat-mean_param-mixing_map.tsv
```

```
              task-rest_model-GroupSpatialICA_stat-std_param-components_map.nii
              .gz
                  task-rest_model-GroupSpatialICA_stat-std_param-mixing_map.tsv

              task-rest_model-GroupSpatialICA_stat-tvalue_param-components_map.
              nii.gz
                  task-rest_model-GroupSpatialICA_stat-tvalue_mixing_map.tsv
                  task-rest_model-GroupSpatialICA_desc-subjects_map.tsv
sub-01/
     func/
             model-GroupSpatialICA/
                  sub-01_task-rest_model-GroupSpatialICA_param-mixing_map.tsv

sub-01_task-rest_model-GroupSpatialICA_param-components_map.nii.gz
                  sub-01_task-rest_model-GroupSpatialICA_model.json
```

```
models.tsv
sub-01/
    model-temporalICA/
      model_description.json
      sub-01_task-rest_model-temporalICA_stat-mean_param-components_map.<ext>
      sub-01_task-rest_model-temporalICA_stat-mean_param-mixing_map.<ext>
      sub-01_task-rest_model-temporalICA_stat-mean_mask.nii.gz
      sub-01_task-rest_model-temporalICA_stat-mean_map.json

    ses-01/
      eeg/
         model-temporalICA/
            sub-01_ses-01_task-rest_run-1_model-temporalICA_param-mixing_map.<e
      eg>
            sub-01_ses-01_task-rest_run-1_model-temporalICA_param-components_ma
      p.<eeg>
            sub-01_ses-01_task-rest_run-1_model-temporalICA_map.json
     ses-02/
       eeg/
         model-temporalICA/
            sub-01_ses-02_task-rest_run-1_model-temporalICA_param-mixing_map.<e
            eg>
            sub-01_ses-02_task-rest_run-1_model-temporalICA_param-components_ma
            p.<eeg>   sub-01_ses-02_task-rest_run-1_model-temporalICA_map.json
```

5. 2024/10/15 BIDS Maintainers Meeting Notes

1. Split subject-level decomposition PR and group definition PR. The combination of these independent features should make group-level decompositions possible.
2. Group definition needs to take into account more than just subject (i.e., groups can be built from specific files).
3. Tool developers are not fans of using (or at least not relying entirely on) SourceFiles because their users want something easier to use, like a TSV.
    a. Chris: Allow model-<label>_participants.tsv to have one line per file rather than one line per participant. Maybe call it “spec.tsv”
4. Is the metadata compliant with BIDS-Prov?
5. New item entity should be its own PR.