

The Imaging Course: “Standardized” fMRI preprocessing using BIDS, MRIQC and fMRIPREP

Martin Szinte

Department of Experimental and Applied Psychology
Vrije Universiteit Amsterdam, Amsterdam, Netherlands

martin.szinte@gmail.com
www.martinszinte.net

Course overview

1. The Brain Imaging Data Structure
2. MRIQC: rapid quality check
3. FMRIprep: a robust fMRI preprocessing pipeline

The Brain Imaging Data Structure (BIDS)

The Brain Imaging Data Structure (BIDS)

Based on the volume of publication¹, tens of thousand participants are scanned per year. This results in complex datasets composed of many files and format.

The Brain Imaging Data Structure (BIDS)

Based on the volume of publication¹, tens of thousand participants are scanned per year. This results in complex datasets composed of many files and format.

The adoption of a common standard allows:

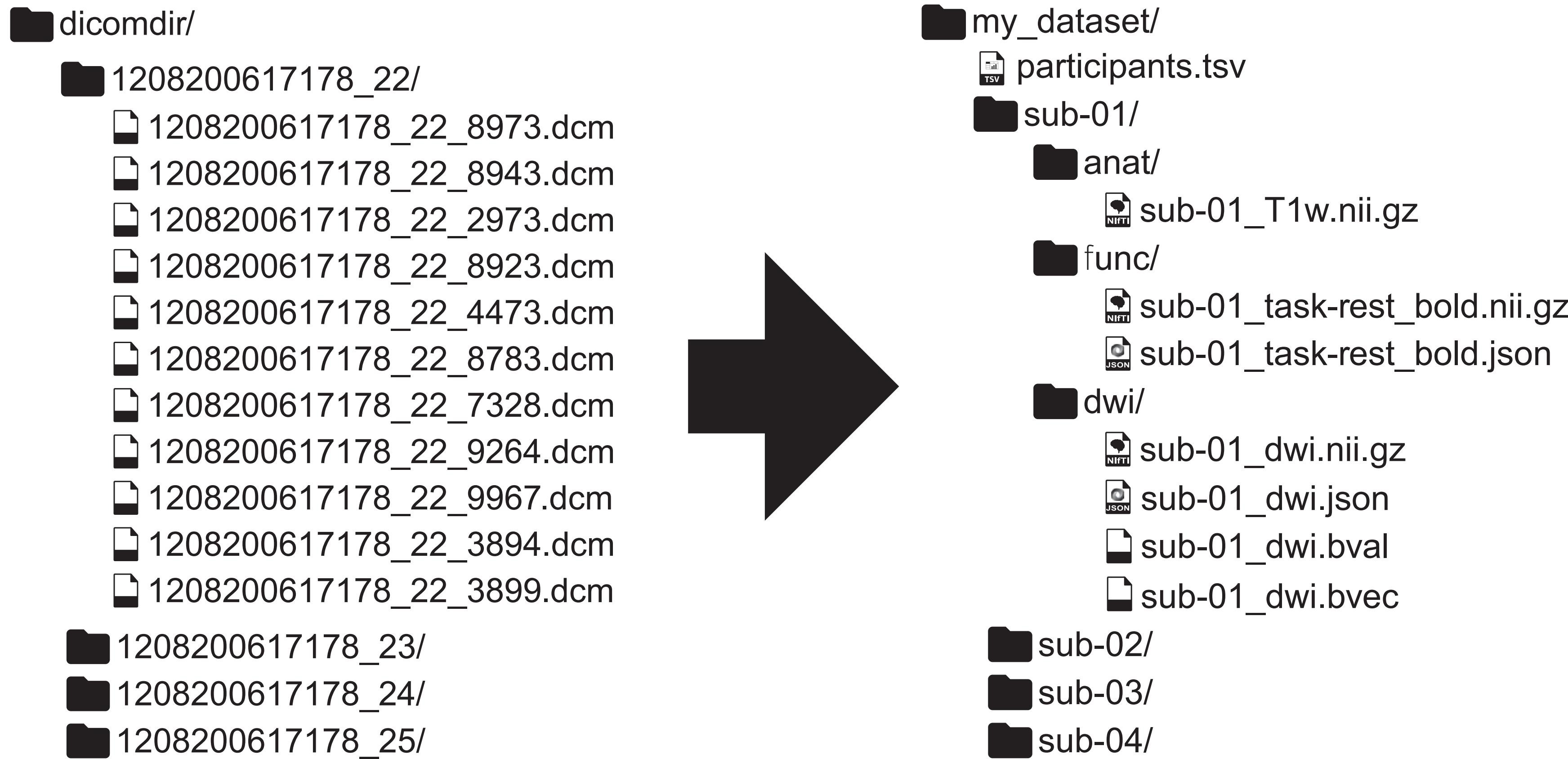
- data to remain usable within and across labs
- the use of automated data analysis software (BIDS Apps)
- the reduction of simple errors (standardisation and detection error softwares)
- better reproducibility of fMRI science

The Brain Imaging Data Structure (BIDS)

The BIDS format¹ is a way of arranging data and specifying metadata, with mandatory and optional (but regulated) aspects.

The Brain Imaging Data Structure (BIDS)

The BIDS format¹ is a way of arranging data and specifying metadata, with mandatory and optional (but regulated) aspects.



The Brain Imaging Data Structure (BIDS)

The BIDS format¹ is a way of arranging data and specifying metadata, with mandatory and optional (but regulated) aspects.

- The BIDS format is designed to standardize the raw data not the derivatives.
- The selected image format is NIfTI (nii.gz).
- Other data (e.g. events or behavioural data) must be saved in TSV format.
- Scan metadata must be given in JSON files with a key/value system.
- Required, recommended and optional scan metadata are defined here:
http://bids.neuroimaging.io/bids_spec.pdf
- Metadata can be defined at different level (session, subject or dataset).

The Brain Imaging Data Structure (BIDS)

Steps to create a BIDS compatible dataset:

- **Step 1:** convert raw data to NIfTI
- **Step 2:** define folder structure
 - create a subfolder per participant (sub-001, sub-002, sub-003, etc)
 - in each participant folder create 3 subfolders /anat (for anatomical scans), /func (for functional scans) and /fmap (for field map scans)
- **Step 3:** rename and move files
 - anat/sub-00X_T1w.nii.gz (T1 weighted file)
 - fmap/sub-00X_magnitude1.nii.gz (B0 magnitude)
 - fmap/sub-00X_phasediff.nii.gz (B0 phase difference)
 - func/sub-00X_task-YourTask_run-X_bold.nii.gz (Functional BOLD)
 - func/sub-00X_task-YourTask_run-X_events.tsv (Events)

The Brain Imaging Data Structure (BIDS)

Steps to create a BIDS compatible dataset:

- **Step 4:** create metadata json files
 - *dataset_description.json*
 - *task-MyTask_bold.json*
 - *sub-00X/fmap/sub-00X_phasediff.json*
- **Step 5:** validate the dataset (<https://bids-standard.github.io/bids-validator/>)

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 1:** convert raw data to NIfTI

/Spinoza_Course/BIDS/convert2niigz.py

convert2niigz.py

Goal of the script:

Convert PAR/REC to nifti (nii.gz) format

Input(s):

sys.argv[1]: raw data directory

sys.argv[2]: name of converter to use (parrec2nii or dcm2niix)

Output(s):

nifti files

To run:

```
ssh -Y compute-01
```

```
module load collections/default
```

```
cd /data1/projects/fMRI-course/Spinoza_Course/
```

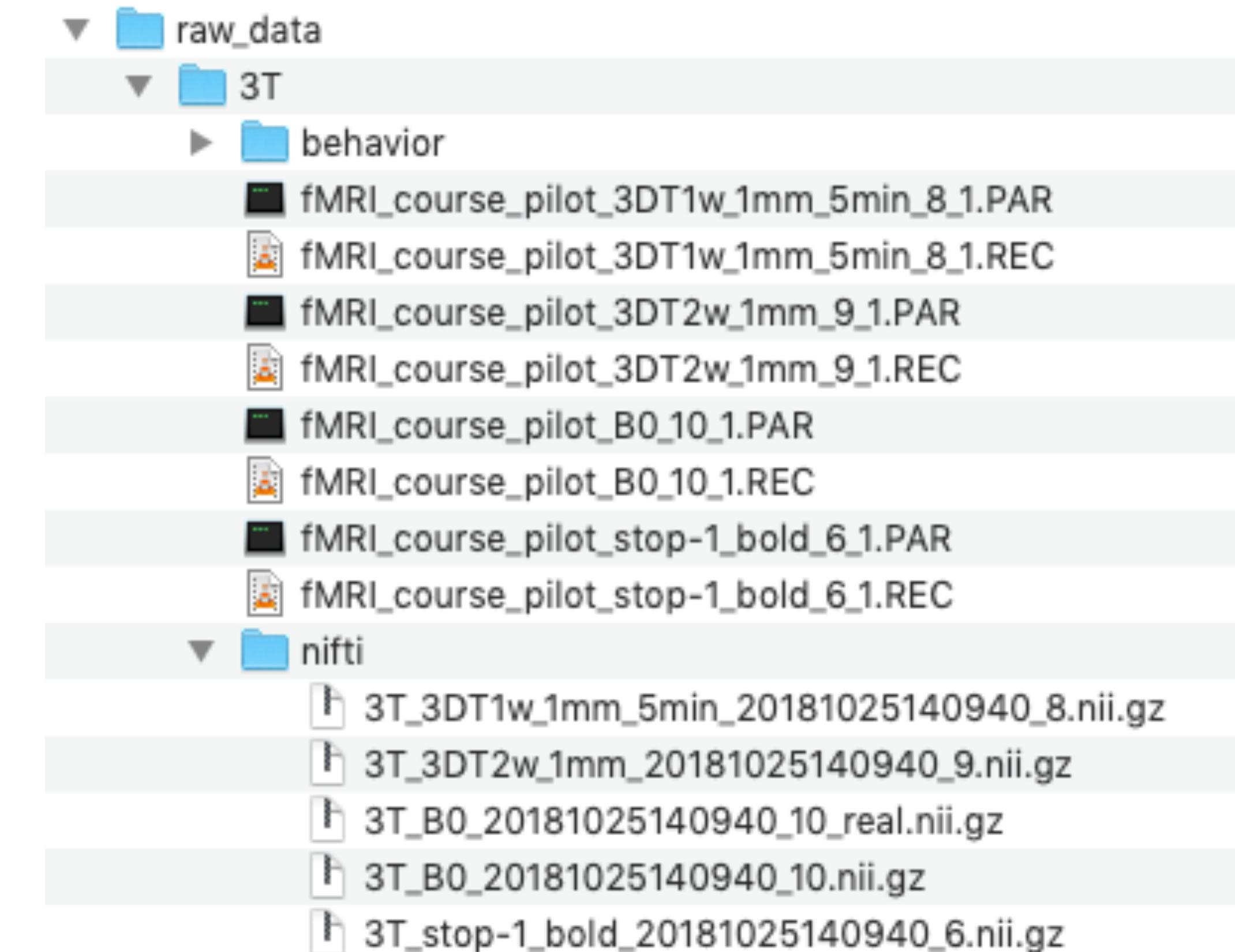
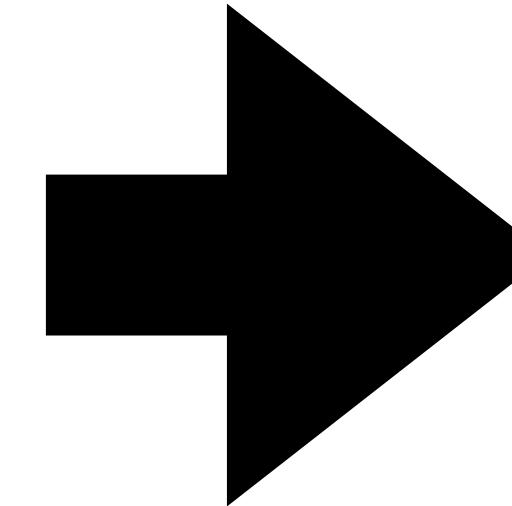
```
python convert2niigz.py [your raw data path] [dcm2niix or parrec2nii]
```

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 1:** convert raw data to NIfTI

/Spinoza_Course/BIDS/convert2niigz.py



The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 2 & 3:** define folder structure & rename and move files

/Spinoza_Course/BIDS/bids_generator.py

bids_generator.py

Goal of the script:

Convert data in BIDS format

Input(s):

sys.argv[1]: raw data folder

sys.argv[2]: bids folder

sys.argv[3]: subject bids number (e.g. sub-001)

sys.argv[4]: task name

Output(s):

BIDS files

To run:

ssh -Y compute-01

module load collections/default

cd /data1/projects/fMRI-course/Spinoza_Course/

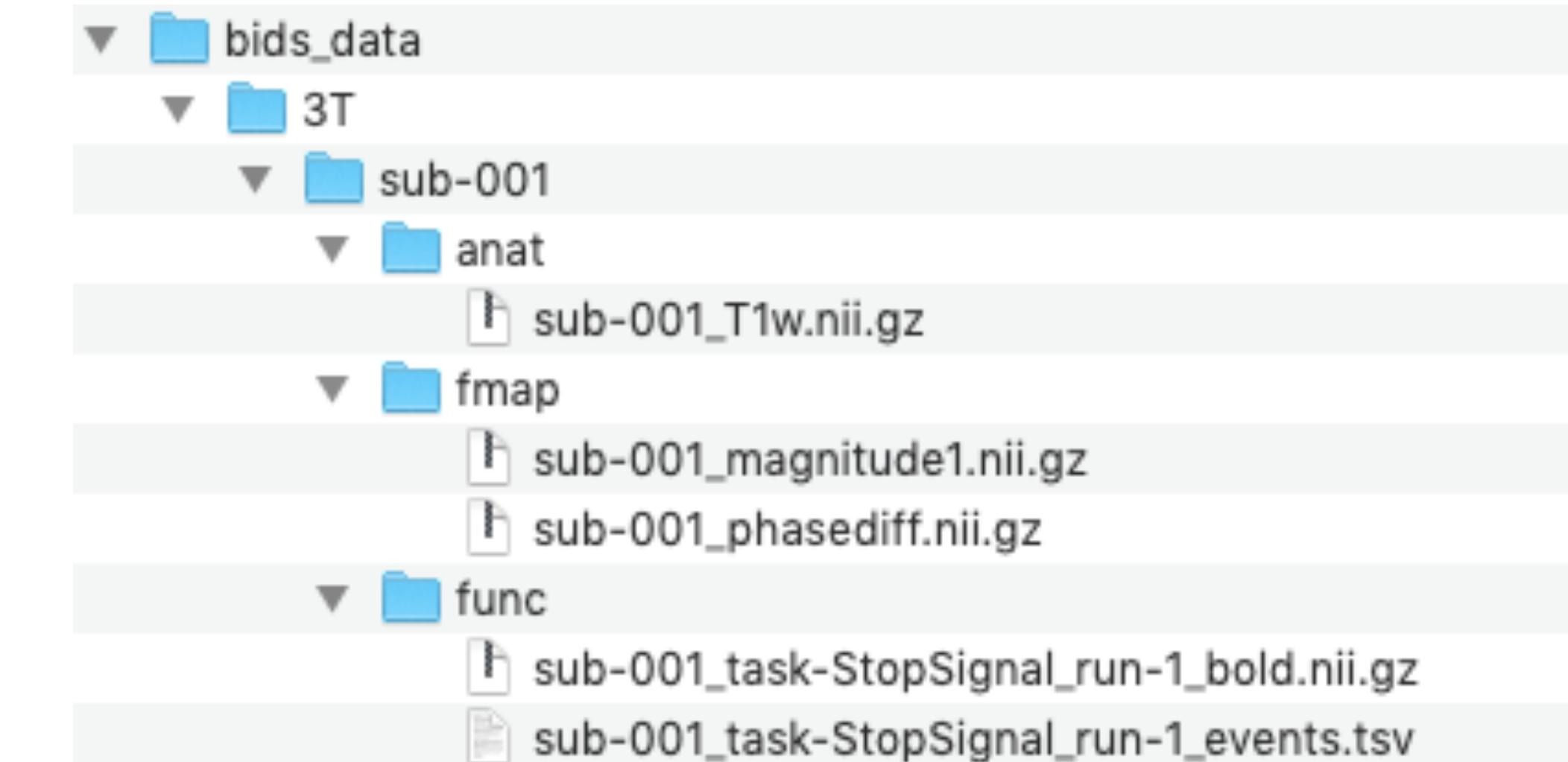
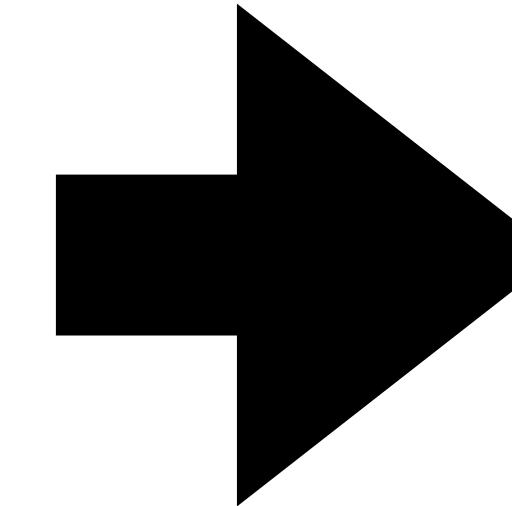
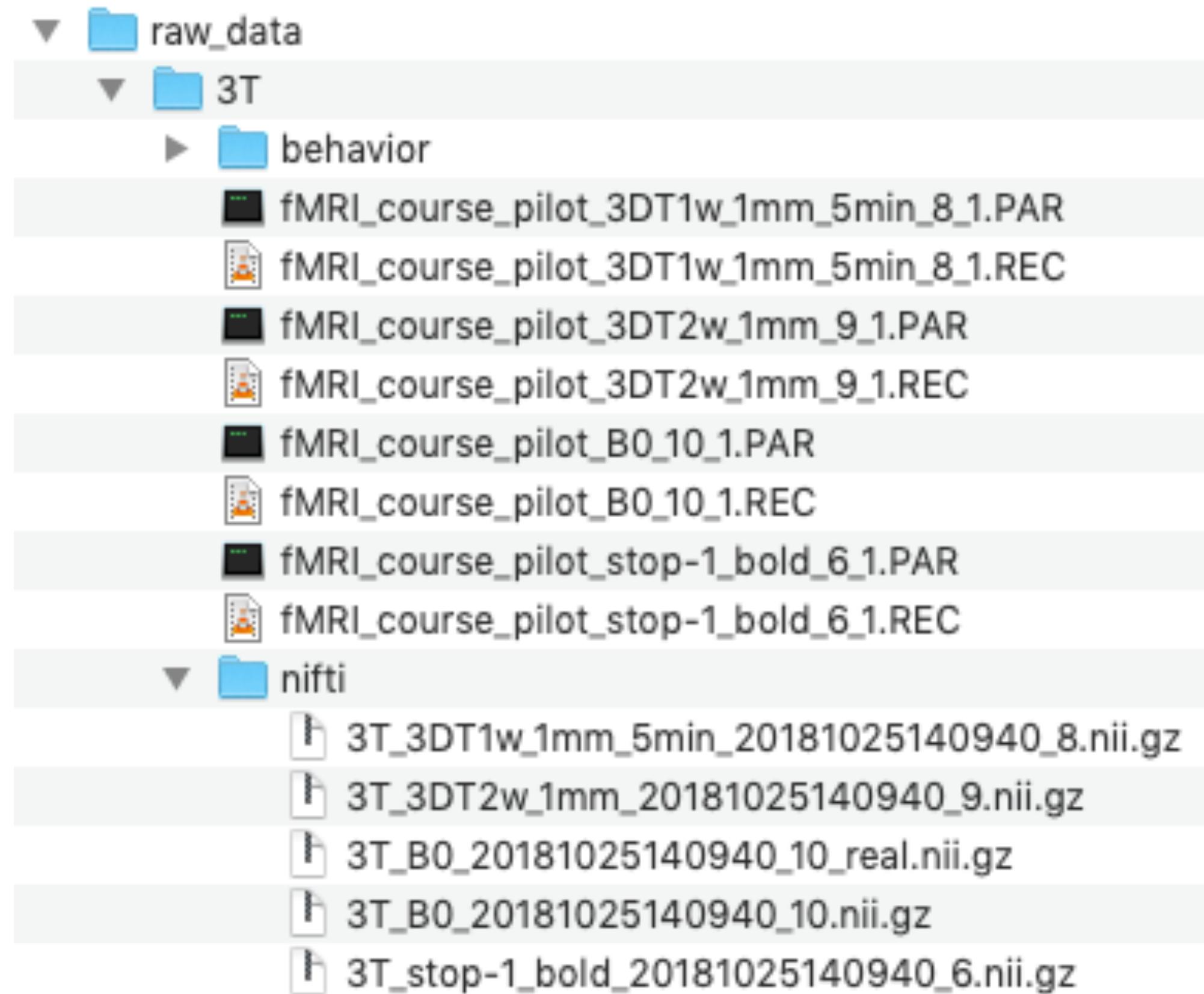
python bids_generator.py [raw data path] [bids data path] [subject name] [task name]

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 2 & 3:** define folder structure & rename and move files

/Spinoza_Course/BIDS/bids_generator.py



The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *dataset_description.json*

```
{"Name": "StopSignal task for fMRI experiment in 3T scanner",
"BIDSVersion": "1.1.1",
"Authors": ["Sara Jahfari, Martin Szinte"] }
```

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *dataset_description.json*

```
{"Name": "StopSignal task for fMRI experiment in 3T scanner",  
"BIDSVersion": "1.1.1",  
"Authors": ["Sara Jahfari, Martin Szinte"] }
```

Decide values yourself

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *task-MyTask_bold.json*

```
{"RepetitionTime": 0.700,  
"EchoTime": 0.030,  
"EchoTimeDifference": 0.00253,  
"EpiFactor": 53,  
"WaterFatShift": 14.042,  
"PhaseEncodingDirection": "j",  
"SenseFactor": 4,  
"SliceDirection": "z",  
"MultiBandFactor": 4,  
"EffectiveEchoSpacing": 0.00015193,  
"SliceTiming":  
[0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364, 0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364, 0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364],  
"SliceOrder": "ascending",  
"TaskName": "StopSignal"}
```

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *task-MyTask_bold.json*

```
{"RepetitionTime": 0.700,  
"EchoTime": 0.030,  
"EchoTimeDifference": 0.00253,  
"EpiFactor": 53,  
"WaterFatShift": 14.042,  
"PhaseEncodingDirection": "j",  
"SenseFactor": 4,  
"SliceDirection": "z",  
"MultiBandFactor": 4,  
"EffectiveEchoSpacing": 0.00015193,  
"SliceTiming":  
[0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364, 0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364, 0.0, 0.0636, 0.1273, 0.1909, 0.2545, 0.3182, 0.3818, 0.4455, 0.5091, 0.5727, 0.6364],  
"SliceOrder": "ascending",  
"TaskName": "StopSignal"}
```

Read in .par file of bold scan file
Read in .par file of B0 scan file
Ask your scanning assistant
Compute/decide value yourself

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *sub-00X/fmap/sub-00X_phasediff.json*

```
{ "EchoTime1": 0.030,  
"EchoTime2": 0.03253,  
"IntendedFor": ["func/sub-001_task-StopSignal_run-1_bold.nii.gz"] }
```

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 4:** create metadata json files (http://bids.neuroimaging.io/bids_spec.pdf)
 - *sub-00X/fmap/sub-00X_phasediff.json*

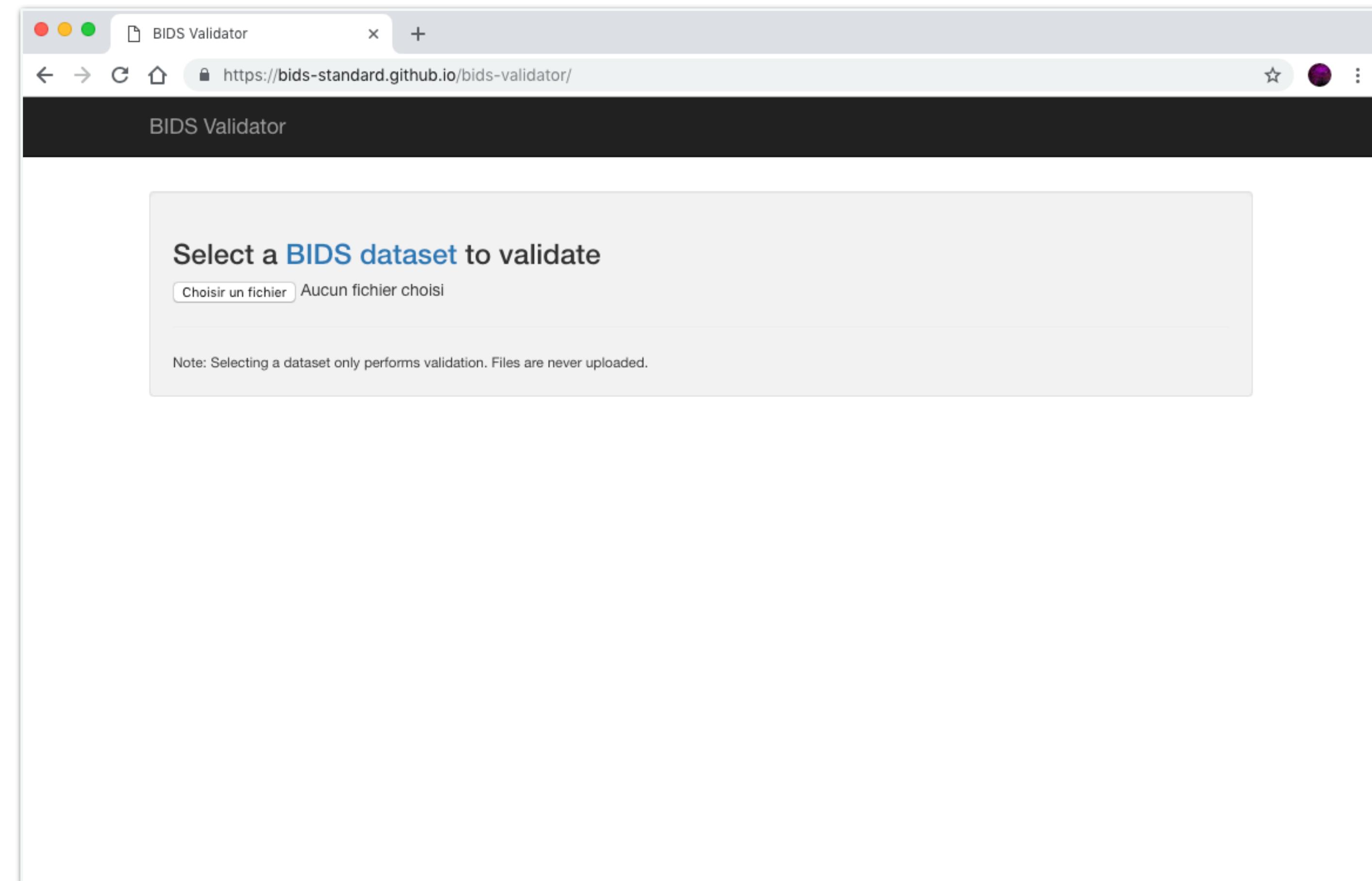
```
{ "EchoTime1": 0.030,  
"EchoTime2": 0.03253,  
"IntendedFor": ["func/sub-001_task-StopSignal_run-1_bold.nii.gz"] }
```

**Read in .par file of bold scan file
Compute/decide value yourself**

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 5:** validate the dataset (<https://bids-standard.github.io/bids-validator/>)



The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 5: validate the dataset (<https://bids-standard.github.io/bids-validator/>)**

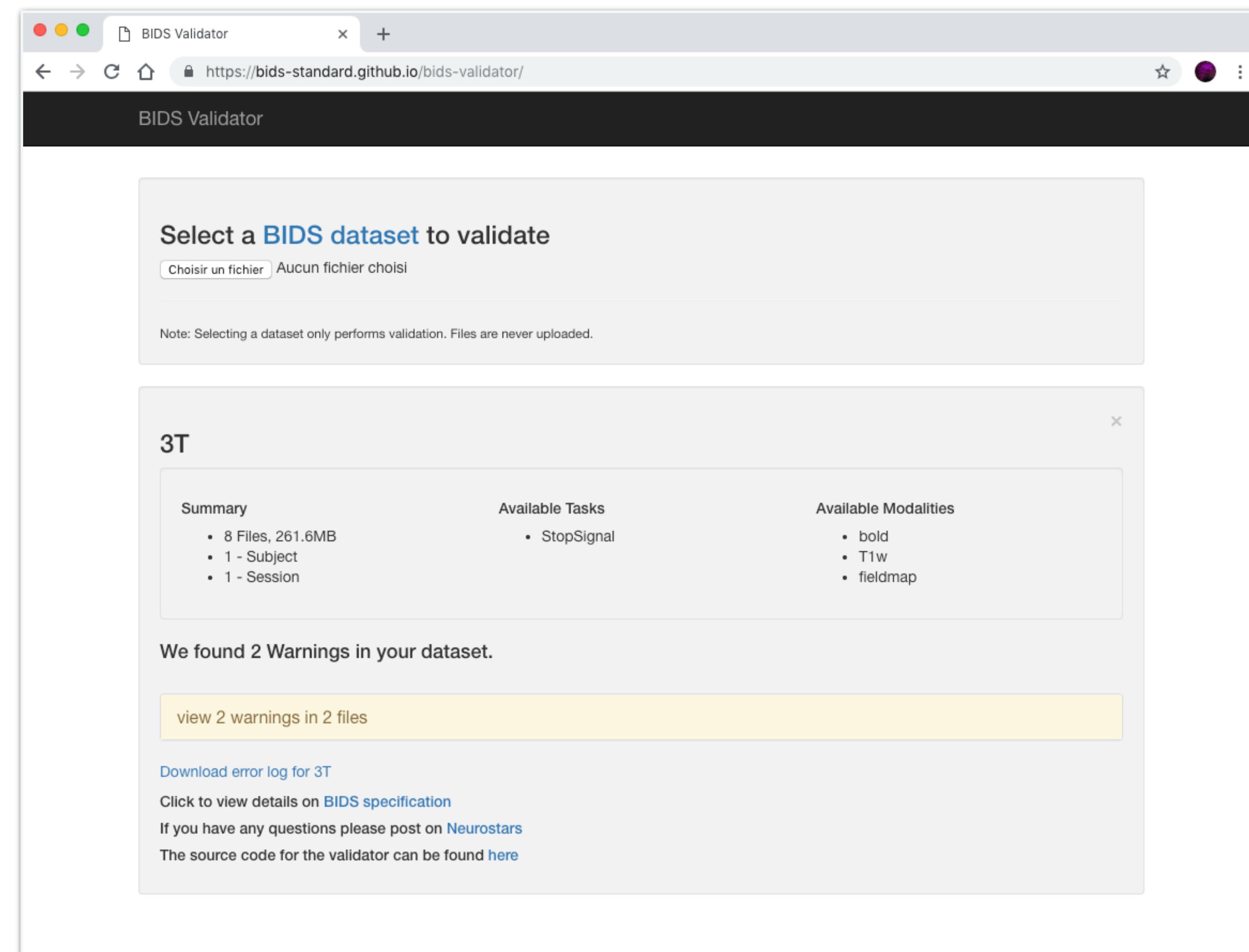
The screenshot shows the BIDS Validator interface. At the top, it says "Select a BIDS dataset to validate" and "Aucun fichier choisi". Below this, a note states: "Note: Selecting a dataset only performs validation. Files are never uploaded." A section for the dataset "3T" is shown, containing "Summary" (8 Files, 261.6MB), "Available Tasks" (StopSignal), and "Available Modalities" (bold, T1w, fieldmap). A message at the bottom says "Your dataset is not a valid BIDS dataset." Below this are buttons for "view 1 error in 1 file" (red background) and "view 1 warning in 1 file" (yellow background). At the very bottom, there are links for "Download error log for 3T", "Click to view details on BIDS specification", "If you have any questions please post on Neurostars", and "The source code for the validator can be found [here](#)".

The screenshot shows the BIDS Validator interface with a detailed error report. At the top, it says "Summary" (8 Files, 261.6MB), "Available Tasks" (StopSignal), and "Available Modalities" (bold, T1w, fieldmap). A message at the bottom says "Your dataset is not a valid BIDS dataset." Below this is a red box with the link "view 1 error in 1 file". A detailed error report follows:
Error 1: [Code 12] REPETITION_TIME_MISMATCH
Repetition time did not match between the scan's header and the associated JSON metadata file.
sub-001_task-StopSignal_run-1_bold.nii.gz 267543.965 KB | application/x-gzip
Location: 3T/sub-001/func/sub-001_task-StopSignal_run-1_bold.nii.gz
Reason: Repetition time defined in the JSON (0.9 sec.) did not match the one defined in the NIFTI header (0.7 sec.).
Below this is a yellow box with the link "view 1 warning in 1 file". At the very bottom, there are links for "Download error log for 3T", "Click to view details on BIDS specification", "If you have any questions please post on Neurostars", and "The source code for the validator can be found [here](#)".

The Brain Imaging Data Structure (BIDS)

Tutorial

- **Step 5:** validate the dataset (<https://bids-standard.github.io/bids-validator/>)



The Brain Imaging Data Structure (BIDS)

Now we can use any BIDS App

<http://bids-apps.neuroimaging.io/apps/>

Available BIDS Apps

BIDS-Apps/example	version 0.0.7	open bug issues 0	build passing	open bug pull requests 0	docker pulls 11k	439.5MB 23 layers
BIDS-Apps/freesurfer	version	open bug issues 0	build passing	open bug pull requests 0	docker pulls 10k	2.6GB 52 layers
BIDS-Apps/ndmg	version v0.1.0	open bug issues 0	build passing	open bug pull requests 0	docker pulls 7k	920.9MB 31 layers
BIDS-Apps/BROCCOLI	version v1.0.1	open bug issues 1	build passing	open bug pull requests 0	docker pulls 273	3GB 21 layers
BIDS-Apps/FibreDensityAndCrosssection	version v0.0.1	open bug issues 0	build passing	open bug pull requests 0	docker pulls 80	576.8MB 31 layers
BIDS-Apps/SPM	version v0.0.14	open bug issues 1	build passing	open bug pull requests 0	docker pulls 997	1.6GB 24 layers
poldracklab/mriqc	version 0.14.2	open bug issues 13	build passing	open bug pull requests 0	docker pulls 22k	2.8GB 38 layers
BIDS-Apps/QAP	Image not found	open bug issues 0	build passing	open bug pull requests 0	docker pulls 7	Image not found
BIDS-Apps/CPAC	version v1.0.1a_22	open bug issues 0	build passing	open bug pull requests 0	docker pulls 2k	1.6GB 45 layers
BIDS-Apps/hyperalignment	Image not found	open bug issues 0	build passing	open bug pull requests 0	docker pulls 3	Image not found
BIDS-Apps/mindboggle	version 0.0.4-1	open bug issues 2	build passing	open bug pull requests 0	docker pulls 394	1.9GB 81 layers
BIDS-Apps/MRtrix3_connectome	version 0.3.0	open bug issues 0	build passing	open bug pull requests 0	docker pulls 520	4.1GB 61 layers
BIDS-Apps/rs_signal_extract	version 0.1	open bug issues 0	build passing	open bug pull requests 0	docker pulls 76	240MB 17 layers
BIDS-Apps/aa	version v0.1.0	open bug issues 0	build passing	open bug pull requests 0	docker pulls 76	6.2GB 30 layers
BIDS-Apps/niak	version latest	open bug issues 1	build passing	open bug pull requests 0	docker pulls 118	2.7GB 103 layers
BIDS-Apps/oppni	version v0.7.0-1	open bug issues 1	build passing	open bug pull requests 0	docker pulls 142	2.9GB 41 layers
poldracklab/fmriprep	version 1.2.3	open bug issues 23	build passing	open bug pull requests 0	docker pulls 58k	0B 48 layers
BIDS-Apps/brainiak-srm	version latest	open bug issues 0	build failed	open bug pull requests 0	docker pulls 80	559.3MB 13 layers
BIDS-Apps/nipypipelines	version 0.3.0	open bug issues 0	build passing	open bug pull requests 0	docker pulls 86	478.1MB 20 layers
BIDS-Apps/HCPPipelines	version v3.17.0-18	open bug issues 0	build passing	open bug pull requests 0	docker pulls 655	4GB 31 layers
BIDS-Apps/MAGeTbrain	Image not found	open bug issues 0	build passing	open bug pull requests 0	docker pulls 185	Image not found
BIDS-Apps/tracula	version v6.0.0-4	open bug issues 0	build passing	open bug pull requests 0	docker pulls 410	3.4GB 57 layers
BIDS-Apps/baracus	Image not found	open bug issues 0	build passing	open bug pull requests 0	docker pulls 857	Image not found
BIDS-Apps/antsCorticalThickness	version v2.2.0-1	open bug issues 0	build passing	open bug pull requests 0	docker pulls 128	351.9MB 21 layers
BIDS-Apps/DPARSF	version v4.3.12	open bug issues 0	build passing	open bug pull requests 0	docker pulls 118	1.4GB 28 layers
BIDS-Apps/afni_proc	Image not found	open bug issues 0	build passing	open bug pull requests 0	docker pulls 63	Image not found
BIDS-Apps/rsHRF	version 1.0.0	open bug issues 0	build invalid response data	open bug pull requests 0	docker pulls 28	64.7MB 5 layers

MRIQC: rapid quality check

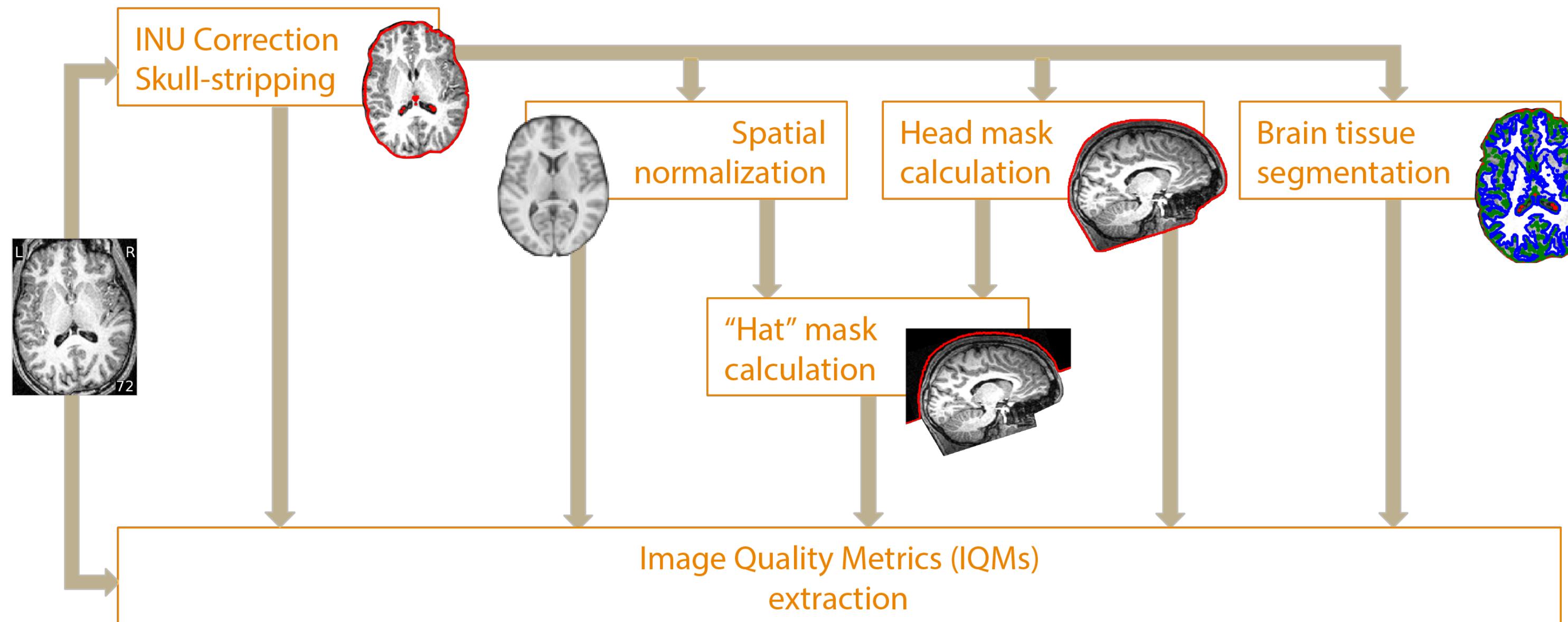
MRIQC will allow us to quickly evaluate the quality of the collected data (and the validity of our BIDS conversion).

MRIQC for MRI Quality Control is a tool for extracting quality measures of anatomical and functional data.

MRIQC returns individual and group-report html files.

MRIQC: rapid quality check

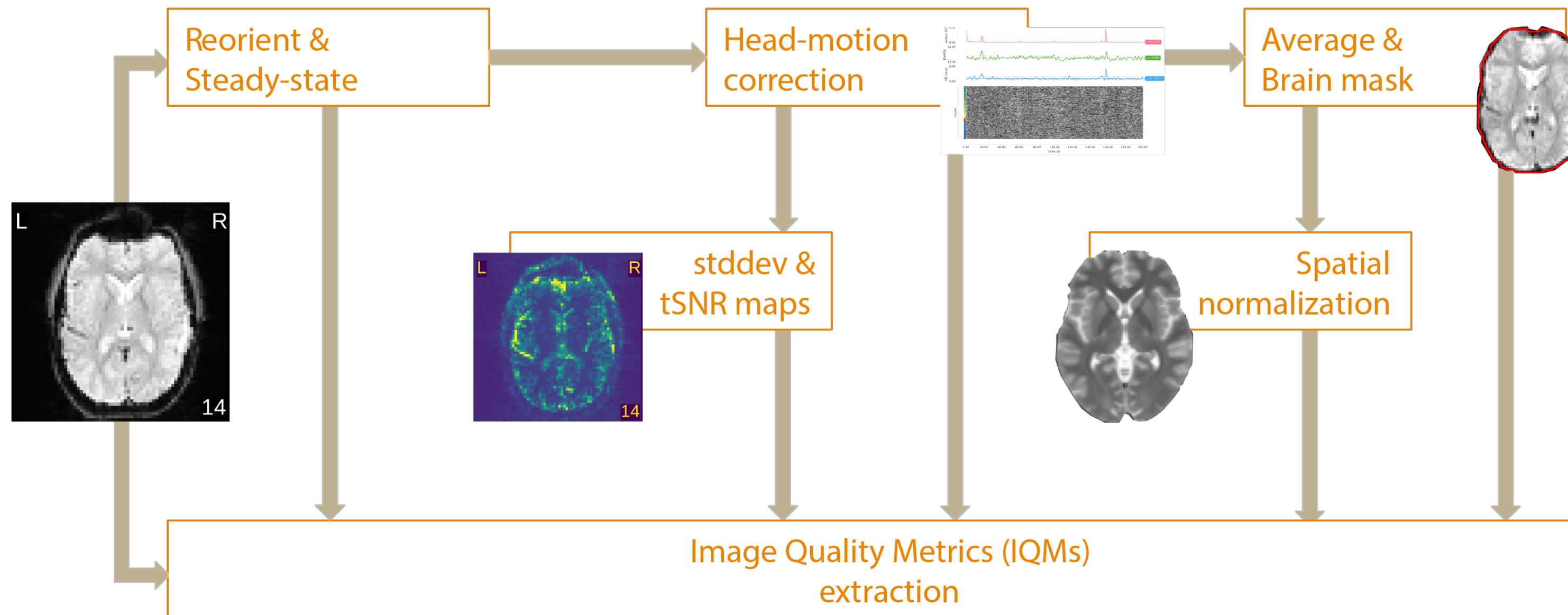
The anatomical workflow:



1. Conform (reorientations, revise data types) input data and read associated metadata.
2. Skull-stripping (AFNI).
3. Calculate head mask
4. Spatial Normalization to MNI (ANTs)
5. Calculate air mask above the nasal-cerebellum plane
6. Brain tissue segmentation (FAST).
7. Extraction of IQMs
8. Individual-reports generation

MRIQC: rapid quality check

The functional workflow:



1. Sanitize (revise data types and xforms) input data, read associated metadata and discard non-steady state frames.
2. Head Motion Correction
3. Skull-stripping of the time-series (AFNI)
4. Calculate mean time-series, and tSNR.
5. Spatial Normalization to MNI (ANTs)
6. Extraction of IQMs
7. Individual-reports generation

MRIQC: rapid quality check

Measures based on noise measurements

CJV	The coefficient of joint variation of GM and WM was proposed as objective function by Ganzetti et al. [30] for the optimization of INU correction algorithms. Higher values are related to the presence of heavy head motion and large INU artifacts.
CNR	The contrast-to-noise ratio [31] is an extension of the SNR calculation to evaluate how separated the tissue distributions of GM and WM are. Higher values indicate better quality.
SNR	MRIQC includes the signal-to-noise ratio calculation proposed by Dietrich et al. [32], using the air background as noise reference. Additionally, for images that have undergone some noise reduction processing, or the more complex noise realizations of current parallel acquisitions, a simplified calculation using the within tissue variance is also provided.
QI ₂	The second quality index of [12] is a calculation of the goodness-of-fit of a χ^2 distribution on the air mask, once the artifactual intensities detected for computing the QI ₁ index have been removed. The description of the QI ₁ is found below.

Measures based on information theory

EFC	The entropy-focus criterion [33] uses the Shannon entropy of voxel intensities as an indication of ghosting and blurring induced by head motion. Lower values are better.
FBER	The foreground-background energy ratio [14] is calculated as the mean energy of image values within the head relative the mean energy of image values in the air mask. Consequently, higher values are better.

Measures targeting specific artifacts

INU	MRIQC measures the location and spread of the bias field extracted estimated by the inu correction. The smaller spreads located around 1.0 are better.
QI ₁	The first quality index of [12] measures the amount of artifactual intensities in the air surrounding the head above the nasio-cerebellar axis. The smaller QI ₁ , the better.
WM2MAX	The white-matter to maximum intensity ratio is the median intensity within the WM mask over the 95% percentile of the full intensity distribution, that captures the existence of long tails due to hyper-intensity of the carotid vessels and fat. Values should be around the interval [0.6, 0.8].

Other measures

FWHM	The full-width half-maximum [34] is an estimation of the blurriness of the image using AFNI's 3dFWHMx. Smaller is better.
ICVs	Estimation of the icv of each tissue calculated on the FSL FAST's segmentation. Normative values fall around 20%, 45% and 35% for cerebrospinal fluid (CSF), WM and GM, respectively.
rPVE	The residual partial volume effect feature is a tissue-wise sum of partial volumes that fall in the range [5%-95%] of the total volume of a pixel, computed on the partial volume maps generated by FSL FAST. Smaller residual partial volume effects (rPVEs) are better.
SSTATs	Several summary statistics (mean, standard deviation, percentiles 5% and 95%, and kurtosis) are computed within the following regions of interest: background, CSF, WM, and GM.
TPMs	Overlap of tissue probability maps estimated from the image and the corresponding maps from the ICBM nonlinear-asymmetric 2009c template [35].

<https://doi.org/10.1371/journal.pone.0184661.t002>

MRIQC: rapid quality check

Tutorial

mriqc tmux.py

Goal of the script:

Run `frmiqc` on tmux of a server

Input(s) :

sys.argv[1]: main data directory to mount in singularity (e.g. data1)

sys.argv[2]: bids directory

sys.argv[3]: deriv directory

sys.argv[4]: temp directory

sys.argv[5]: bids subject name (e.g. sub-001)

sys.argv[6]: server nb of processor to use (e.g 4)

`sys.argv[7]`: your name to create your tmux session (e.g. student01)

Output (s) :

BIDS files

To run:

```
ssh -Y compute-01
```

```
module load collections/default
```

```
cd /data1/projects/fMRI-course/Spinoza Course/
```

```
python mriqc_tmux.py [main directory] [bids directory] [deriv directory] [temp directory]  
[subject] [processors] [your id]
```

MRIQC: rapid quality check

Tutorial

```
# imports modules
import sys
import os
import time

# inputs
main_dir = sys.argv[1]
bids_dir = sys.argv[2]
deriv_dir = sys.argv[3]
temp_dir = sys.argv[4]
sub = sys.argv[5]
nb_procs = int(sys.argv[6])
your_id = sys.argv[7]

# define singularity and fs licence
singularity_dir = '/packages/singularity_containers/poldracklab_mriqc_0.11.0-2018-06-05-442554ee49a6.img'
```

MRIQC: rapid quality check

Tutorial

```
# run singularity
singularity_cmd = "singularity run --bind /{main_dir}:/main_dir {dir} {source} {deriv_dir} participant --participant_label {sub} -w {temp} --n_procs {nb_procs:.0f} --verbose-reports --mem_gb 64 -m bold T1w --no-sub".format(
    main_dir = main_dir,
    dir = singularity_dir,
    source = bids_dir,
    deriv_dir = deriv_dir,
    sub = sub,
    temp = temp_dir,
    nb_procs = nb_procs,
    )

# define tmux session
session_name = "{id}_{sub}_mriqc".format(id = your_id, sub = sub)

# run singularity
print('run singularity on tmux {session_name}'.format(session_name = session_name))
print('to run manually >> {cmd}'.format(cmd = singularity_cmd))
print('to check on tmux type >> tmux a -t {session_name}'.format(session_name = session_name))

os.system("tmux new-session -d -s {session_name} '{cmd}'".format(session_name = session_name, cmd =
singularity_cmd))
time.sleep(2)
```

MRIQC: rapid quality check

The screenshot shows a web-based interface for the MRIQC: individual bold report. The title bar reads "MRIQC: individual bold report". The main content area has a heading "MRIQC: individual bold report" with a cursor pointing to it. Below this is a section titled "Summary" containing a bulleted list of metadata:

- Subject ID: 001_StopSignal_1.
- Date and time: 2018-11-02, 16:25.
- MRIQC version: 0.11.0.
- Workflow details:
 - Framewise Displacement was computed using AFNI `3dvolreg`
 - Framewise Displacement threshold was defined at 0.200000 mm

Below the summary is a section titled "Visual reports" with a sub-section "BOLD average" showing a 3x8 grid of brain slices. The slices are labeled from 0 to 15. The first row shows axial slices, the second row coronal slices, and the third row sagittal slices. Each slice includes "L" and "R" labels indicating left and right directions.

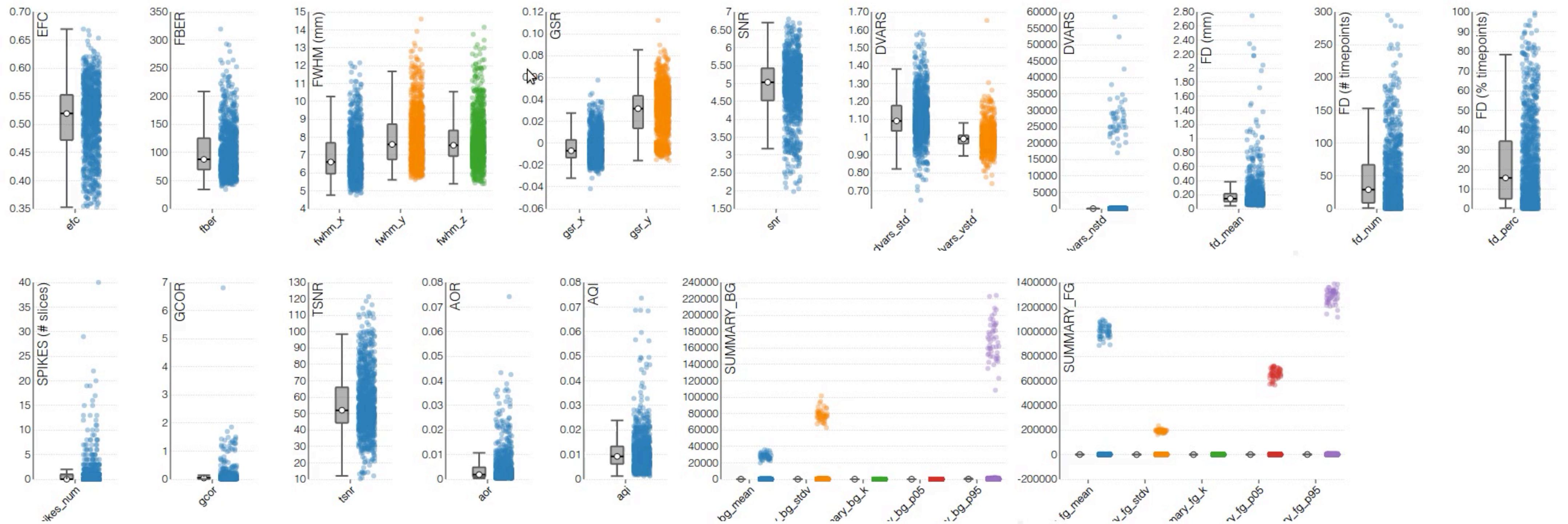
Esteban et al. (2017) *PLoS ONE*
mriqc.readthedocs.io

MRIQC: rapid quality check

MRIQC: group functional report

Summary

- Date and time: 2017-02-05, 12:27.
- MRIQC version: 0.9.0-rc2.



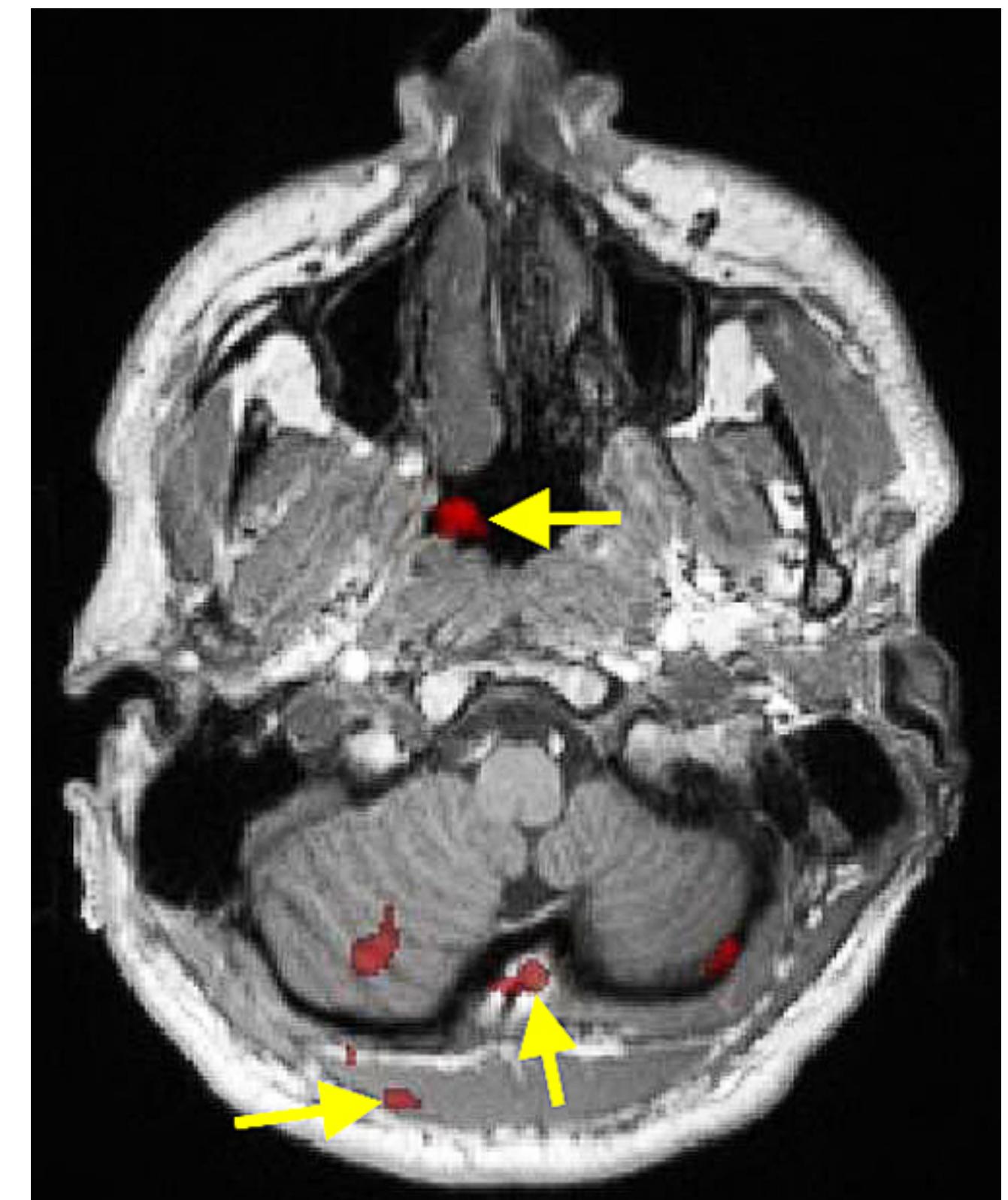
FMRIPREP: a robust fMRI preprocessing pipeline

FMRIPREP: a robust fMRI preprocessing pipeline

fMRI BOLD signal is typically mixed with many non-neural source of variability.

Preprocessing:

- identify these sources and reduces their effects.
- deal with imaging artefacts (slice-timing correction, head motion correction, susceptibility distortion correction).
- allows co-registration across runs and participants.



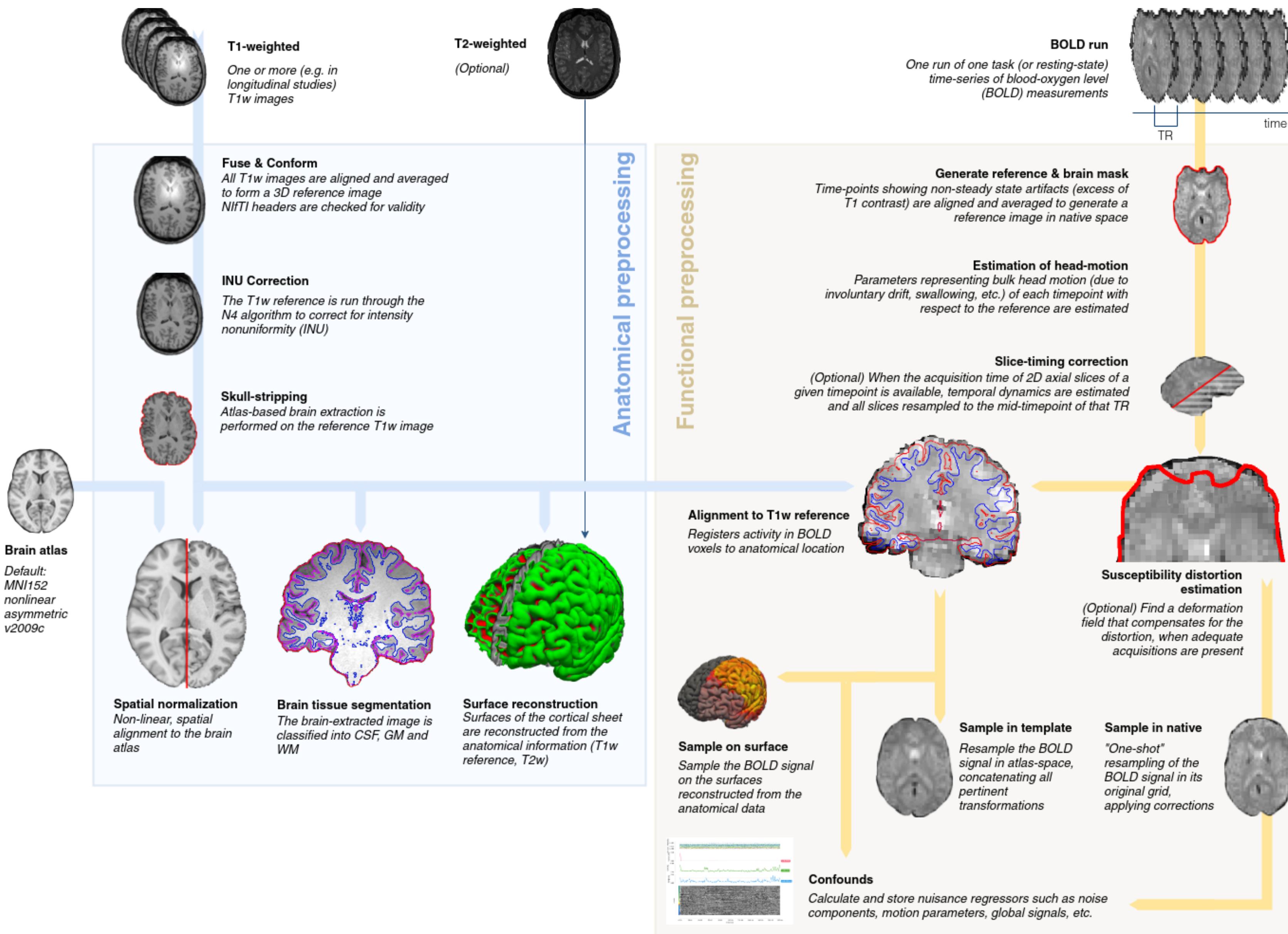
FMRIPREP: a robust fMRI preprocessing pipeline

FMRIPREP is a preprocessing workflow for task-based and resting-state fMRI.

FMRIPREP follows 3 principles:

1. **robustness** to idiosyncrasies of input
2. **quality and transparency** of the outcomes
3. **ease-of-use** with minimal manual intervention

FMRIPREP: a robust fMRI preprocessing pipeline



Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline

Tutorial

fmriprep_tmux.py

Goal of the script:

Run fmriprep on tmux of a server

Input(s) :

sys.argv[1]: main data directory to mount in singularity (e.g. data1)
sys.argv[2]: bids directory
sys.argv[3]: deriv directory
sys.argv[4]: temp directory
sys.argv[5]: bids subject name (e.g. sub-001)
sys.argv[6]: server nb of threads (e.g. 8)
sys.argv[7]: your name to create your tmux session (e.g. student01)

Output(s) :

Preprocessed data and confound regressors in deriv directory

To run:

```
ssh -Y compute-01
module load collections/default
cd /data1/projects/fMRI-course/fMRIcourse/
python fmriprep_tmux.py [main directory] [bids directory] [deriv directory]
                           [temp directory] [subject] [processors] [your id]
```

FMRIPREP: a robust fMRI preprocessing pipeline

Tutorial

```
# imports modules
import sys
import os
import time

# inputs
main_dir = sys.argv[1]
bids_dir = sys.argv[2]
deriv_dir = sys.argv[3]
temp_dir = sys.argv[4]
sub = sys.argv[5]
nb_thread = int(sys.argv[6])
your_id = sys.argv[7]

# define singularity and fs licence
singularity_dir = '/packages/singularity_containers/poldracklab_fmriprep_1.1.8-2018-10-04-8958de85c5c6.img'
fs_licence = '/data1/projects/fMRI-course/fMRIcourse/license.txt'
```

FMRIPREP: a robust fMRI preprocessing pipeline

Tutorial

```
# run singularity
singularity_cmd = "singularity run --bind /{main_dir}:/main_dir {dir} {source} {deriv_dir} participant --
participant_label {sub} -w {temp} --output-space T1w template fsaverage --nthreads {nb_thread:.0f} --use-syn-sdc --low-mem
--fs-license-file {fs_licence}" .format(
    main_dir = main_dir,
    dir = singularity_dir,
    source = bids_dir,
    deriv_dir = deriv_dir,
    sub = sub,
    temp = temp_dir,
    nb_thread = nb_thread,
    fs_licence = fs_licence,
)

# define tmux session
session_name = "{id}_{sub}_fmriprep".format(id = your_id, sub = sub)

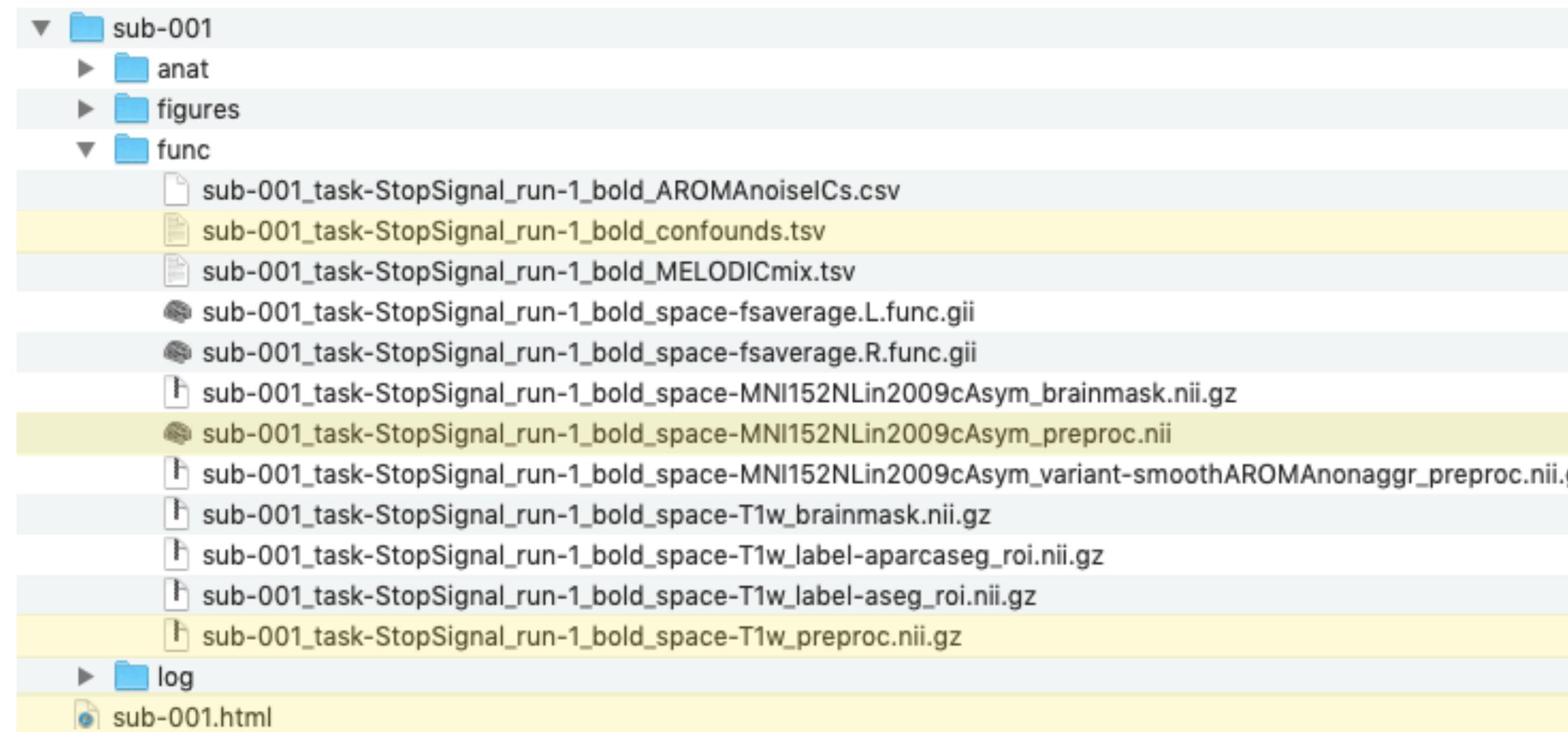
# run singularity
print('run singularity on tmux {session_name}'.format(session_name = session_name))
print('to check type >> tmux a -t {session_name}'.format(session_name = session_name))
print('to run manually >> {cmd}'.format(cmd = singularity_cmd))

os.system("tmux new-session -d -s {session_name} '{cmd}'".format(session_name = session_name, cmd = singularity_cmd))
time.sleep(2)
```

FMRIPREP: a robust fMRI preprocessing pipeline

FMRIPREP will then produce 2 classes of outputs:

- preprocessed data (*data after corrections, resampled onto a target space*)
- confounds (*additional time series, motion parameters, CompCor, tCor, etc*)



FMRIPREP: a robust fMRI preprocessing pipeline

The screenshot shows a web browser window displaying the FMRIPREP summary report for subject sub-001. The title bar reads "sub-001.html#Summary". The address bar shows the URL "file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#Summary". The page content includes a navigation menu with links to "Summary", "Anatomical", "Fieldmaps", "Functional", "About", "Methods", and "Errors". The main content area is titled "Summary" and contains a bulleted list of processing details:

- Subject ID: sub-001
- Structural images: 1 T1-weighted (+ 1 T2-weighted)
- Functional series: 1
 - Task: StopSignal (1 run)
- Resampling targets: T1w, fsaverage, MNI152NLin2009cAsym
- FreeSurfer reconstruction: Pre-existing directory

Anatomical

Anatomical Conformation

- Input T1w images: 1
- Output orientation: RAS
- Output dimensions: 250x256x256
- Output voxel size: 1mm x 1mm x 1mm
- Discarded images: 0

Brain mask and brain tissue segmentation of the T1w

This panel shows the template T1-weighted image (if several T1w images were found), with contours delineating the detected brain mask and brain tissue segmentations.

Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline

The screenshot shows a web browser window with the title bar "sub-001.html#Summary". The address bar displays "Fichier | file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#Summary". The page content includes a navigation menu with links to "Summary", "Anatomical", "Fieldmaps", "Functional", "About", "Methods", and "Errors". The main content area is titled "Summary" and contains a bulleted list of processing details:

- Subject ID: sub-001
- Structural images: 1 T1-weighted (+ 1 T2-weighted)
- Functional series: 1
 - Task: StopSignal (1 run)
- Resampling targets: T1w, fsaverage, MNI152NLin2009cAsym
- FreeSurfer reconstruction: Pre-existing directory

Anatomical

Anatomical Conformation

- Input T1w images: 1
- Output orientation: RAS
- Output dimensions: 250x256x256
- Output voxel size: 1mm x 1mm x 1mm
- Discarded images: 0

Brain mask and brain tissue segmentation of the T1w

This panel shows the template T1-weighted image (if several T1w images were found), with contours delineating the detected brain mask and brain tissue segmentations.

Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline

The screenshot shows a web browser window with the title bar "sub-001.html#Summary". The address bar displays "Fichier | file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#Summary". The page content includes a navigation menu with links to "Summary", "Anatomical", "Fieldmaps", "Functional", "About", "Methods", and "Errors". The main content area is titled "Summary" and contains a bulleted list of processing details:

- Subject ID: sub-001
- Structural images: 1 T1-weighted (+ 1 T2-weighted)
- Functional series: 1
 - Task: StopSignal (1 run)
- Resampling targets: T1w, fsaverage, MNI152NLin2009cAsym
- FreeSurfer reconstruction: Pre-existing directory

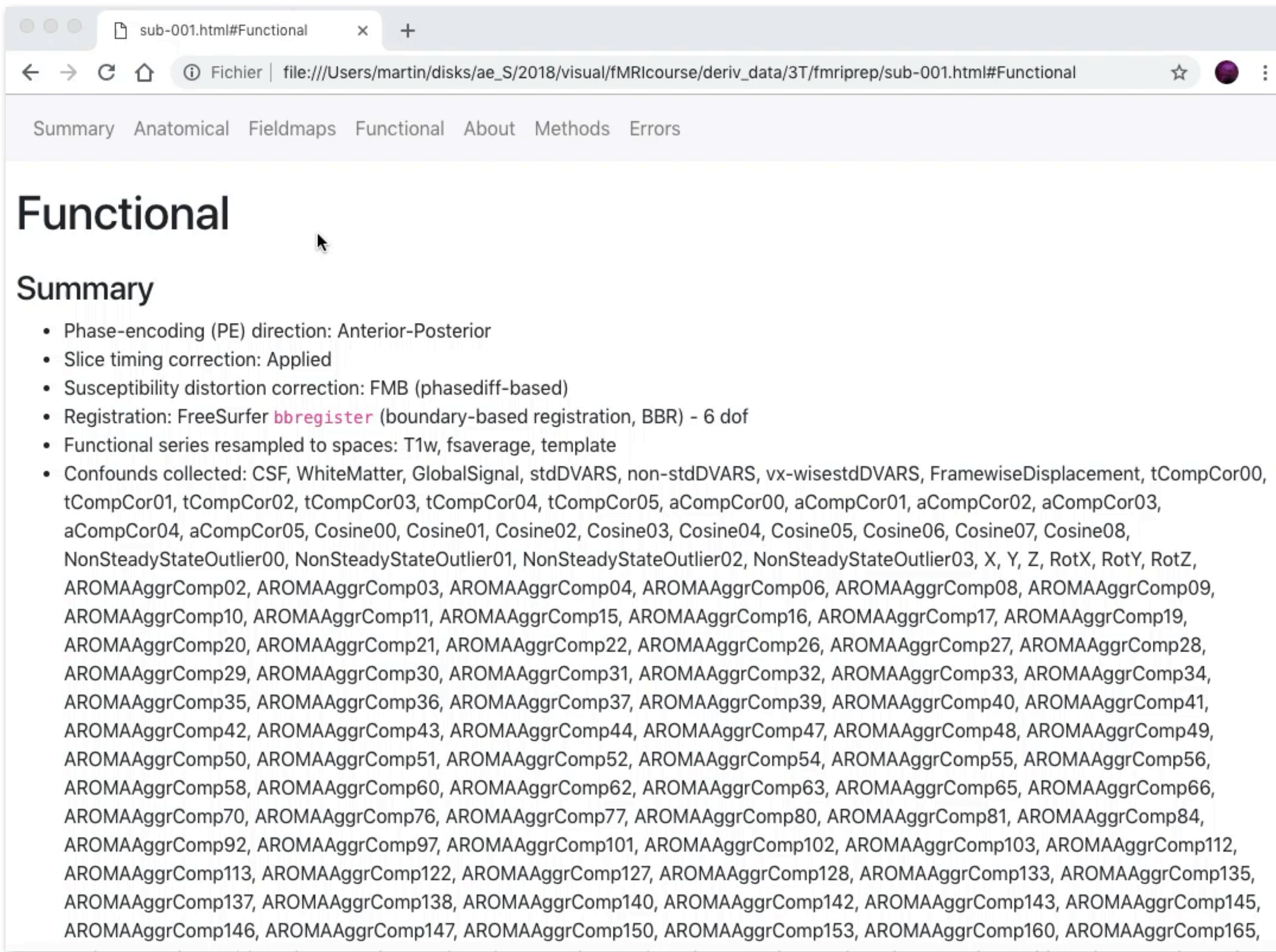
Below this, there is a section titled "Anatomical" with a sub-section "Anatomical Conformation" containing a bulleted list:

- Input T1w images: 1
- Output orientation: RAS
- Output dimensions: 250x256x256
- Output voxel size: 1mm x 1mm x 1mm
- Discarded images: 0

Finally, there is a section titled "Brain mask and brain tissue segmentation of the T1w" with a descriptive text: "This panel shows the template T1-weighted image (if several T1w images were found), with contours delineating the detected brain mask and brain tissue segmentations."

Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline

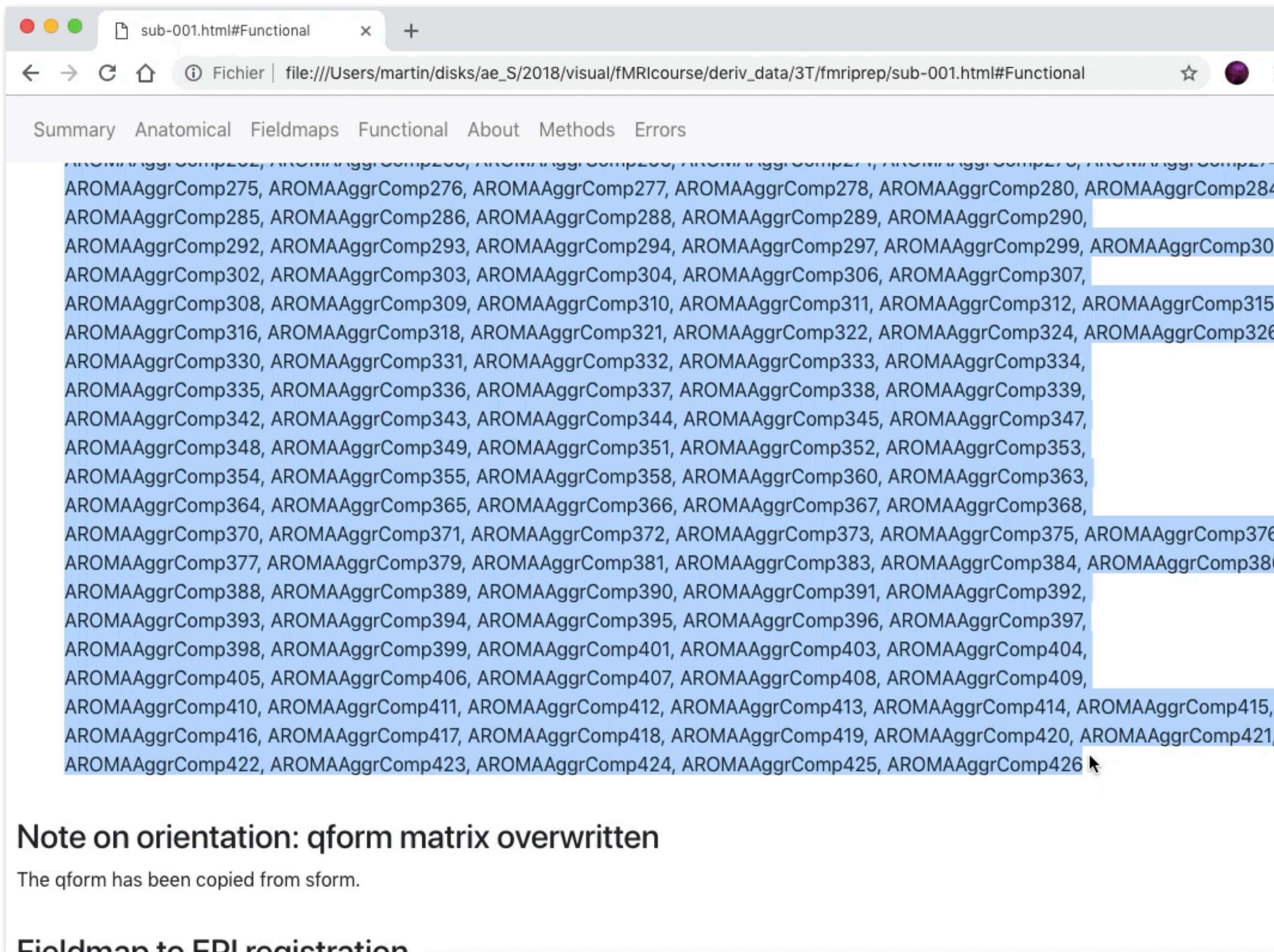


The screenshot shows a web browser window with the title "sub-001.html#Functional". The address bar displays "Fichier | file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#Functional". Below the address bar is a navigation bar with links to "Summary", "Anatomical", "Fieldmaps", "Functional", "About", "Methods", and "Errors". The main content area is titled "Functional" and contains a "Summary" section. The "Summary" section lists various preprocessing steps and confounds collected:

- Phase-encoding (PE) direction: Anterior-Posterior
- Slice timing correction: Applied
- Susceptibility distortion correction: FMB (phasediff-based)
- Registration: FreeSurfer `bbregister` (boundary-based registration, BBR) - 6 dof
- Functional series resampled to spaces: T1w, fsaverage, template
- Confound collected: CSF, WhiteMatter, GlobalSignal, stdDVARS, non-stdDVARS, vx-wisestdDVARS, FrameworkDisplacement, tCompCor00, tCompCor01, tCompCor02, tCompCor03, tCompCor04, tCompCor05, aCompCor00, aCompCor01, aCompCor02, aCompCor03, aCompCor04, aCompCor05, Cosine00, Cosine01, Cosine02, Cosine03, Cosine04, Cosine05, Cosine06, Cosine07, Cosine08, NonSteadyStateOutlier00, NonSteadyStateOutlier01, NonSteadyStateOutlier02, NonSteadyStateOutlier03, X, Y, Z, RotX, RotY, RotZ, AROMAAggrComp02, AROMAAggrComp03, AROMAAggrComp04, AROMAAggrComp06, AROMAAggrComp08, AROMAAggrComp09, AROMAAggrComp10, AROMAAggrComp11, AROMAAggrComp15, AROMAAggrComp16, AROMAAggrComp17, AROMAAggrComp19, AROMAAggrComp20, AROMAAggrComp21, AROMAAggrComp22, AROMAAggrComp26, AROMAAggrComp27, AROMAAggrComp28, AROMAAggrComp29, AROMAAggrComp30, AROMAAggrComp31, AROMAAggrComp32, AROMAAggrComp33, AROMAAggrComp34, AROMAAggrComp35, AROMAAggrComp36, AROMAAggrComp37, AROMAAggrComp39, AROMAAggrComp40, AROMAAggrComp41, AROMAAggrComp42, AROMAAggrComp43, AROMAAggrComp44, AROMAAggrComp47, AROMAAggrComp48, AROMAAggrComp49, AROMAAggrComp50, AROMAAggrComp51, AROMAAggrComp52, AROMAAggrComp54, AROMAAggrComp55, AROMAAggrComp56, AROMAAggrComp58, AROMAAggrComp60, AROMAAggrComp62, AROMAAggrComp63, AROMAAggrComp65, AROMAAggrComp66, AROMAAggrComp70, AROMAAggrComp76, AROMAAggrComp77, AROMAAggrComp80, AROMAAggrComp81, AROMAAggrComp84, AROMAAggrComp92, AROMAAggrComp97, AROMAAggrComp101, AROMAAggrComp102, AROMAAggrComp103, AROMAAggrComp112, AROMAAggrComp113, AROMAAggrComp122, AROMAAggrComp127, AROMAAggrComp128, AROMAAggrComp133, AROMAAggrComp135, AROMAAggrComp137, AROMAAggrComp138, AROMAAggrComp140, AROMAAggrComp142, AROMAAggrComp143, AROMAAggrComp145, AROMAAggrComp146, AROMAAggrComp147, AROMAAggrComp150, AROMAAggrComp153, AROMAAggrComp160, AROMAAggrComp165,

Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



The screenshot shows a web browser window with the title "sub-001.html#Functional". The address bar displays "Fichier | file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#Functional". The page content is a processing log for FMRIPREP, specifically for subject "sub-001". The log lists numerous steps, each starting with "AROMAAggrComp" followed by a number (e.g., 202, 203, 204, ..., 426). Many of these step names are highlighted in blue, likely indicating they have been executed or are currently running.

Summary Anatomical Fieldmaps Functional About Methods Errors

AROMAAggrComp202, AROMAAggrComp203, AROMAAggrComp204, AROMAAggrComp205, AROMAAggrComp206, AROMAAggrComp207, AROMAAggrComp208, AROMAAggrComp209, AROMAAggrComp210, AROMAAggrComp211, AROMAAggrComp212, AROMAAggrComp213, AROMAAggrComp214, AROMAAggrComp215, AROMAAggrComp216, AROMAAggrComp217, AROMAAggrComp218, AROMAAggrComp219, AROMAAggrComp220, AROMAAggrComp221, AROMAAggrComp222, AROMAAggrComp223, AROMAAggrComp224, AROMAAggrComp225, AROMAAggrComp226, AROMAAggrComp227, AROMAAggrComp228, AROMAAggrComp229, AROMAAggrComp230, AROMAAggrComp231, AROMAAggrComp232, AROMAAggrComp233, AROMAAggrComp234, AROMAAggrComp235, AROMAAggrComp236, AROMAAggrComp237, AROMAAggrComp238, AROMAAggrComp239, AROMAAggrComp240, AROMAAggrComp241, AROMAAggrComp242, AROMAAggrComp243, AROMAAggrComp244, AROMAAggrComp245, AROMAAggrComp246, AROMAAggrComp247, AROMAAggrComp248, AROMAAggrComp249, AROMAAggrComp250, AROMAAggrComp251, AROMAAggrComp252, AROMAAggrComp253, AROMAAggrComp254, AROMAAggrComp255, AROMAAggrComp256, AROMAAggrComp257, AROMAAggrComp258, AROMAAggrComp259, AROMAAggrComp260, AROMAAggrComp261, AROMAAggrComp262, AROMAAggrComp263, AROMAAggrComp264, AROMAAggrComp265, AROMAAggrComp266, AROMAAggrComp267, AROMAAggrComp268, AROMAAggrComp269, AROMAAggrComp270, AROMAAggrComp271, AROMAAggrComp272, AROMAAggrComp273, AROMAAggrComp274, AROMAAggrComp275, AROMAAggrComp276, AROMAAggrComp277, AROMAAggrComp278, AROMAAggrComp279, AROMAAggrComp280, AROMAAggrComp281, AROMAAggrComp282, AROMAAggrComp283, AROMAAggrComp284, AROMAAggrComp285, AROMAAggrComp286, AROMAAggrComp287, AROMAAggrComp288, AROMAAggrComp289, AROMAAggrComp290, AROMAAggrComp291, AROMAAggrComp292, AROMAAggrComp293, AROMAAggrComp294, AROMAAggrComp295, AROMAAggrComp296, AROMAAggrComp297, AROMAAggrComp298, AROMAAggrComp299, AROMAAggrComp300, AROMAAggrComp301, AROMAAggrComp302, AROMAAggrComp303, AROMAAggrComp304, AROMAAggrComp305, AROMAAggrComp306, AROMAAggrComp307, AROMAAggrComp308, AROMAAggrComp309, AROMAAggrComp310, AROMAAggrComp311, AROMAAggrComp312, AROMAAggrComp313, AROMAAggrComp314, AROMAAggrComp315, AROMAAggrComp316, AROMAAggrComp317, AROMAAggrComp318, AROMAAggrComp319, AROMAAggrComp320, AROMAAggrComp321, AROMAAggrComp322, AROMAAggrComp323, AROMAAggrComp324, AROMAAggrComp325, AROMAAggrComp326, AROMAAggrComp327, AROMAAggrComp328, AROMAAggrComp329, AROMAAggrComp330, AROMAAggrComp331, AROMAAggrComp332, AROMAAggrComp333, AROMAAggrComp334, AROMAAggrComp335, AROMAAggrComp336, AROMAAggrComp337, AROMAAggrComp338, AROMAAggrComp339, AROMAAggrComp340, AROMAAggrComp341, AROMAAggrComp342, AROMAAggrComp343, AROMAAggrComp344, AROMAAggrComp345, AROMAAggrComp346, AROMAAggrComp347, AROMAAggrComp348, AROMAAggrComp349, AROMAAggrComp350, AROMAAggrComp351, AROMAAggrComp352, AROMAAggrComp353, AROMAAggrComp354, AROMAAggrComp355, AROMAAggrComp356, AROMAAggrComp357, AROMAAggrComp358, AROMAAggrComp359, AROMAAggrComp360, AROMAAggrComp361, AROMAAggrComp362, AROMAAggrComp363, AROMAAggrComp364, AROMAAggrComp365, AROMAAggrComp366, AROMAAggrComp367, AROMAAggrComp368, AROMAAggrComp369, AROMAAggrComp370, AROMAAggrComp371, AROMAAggrComp372, AROMAAggrComp373, AROMAAggrComp374, AROMAAggrComp375, AROMAAggrComp376, AROMAAggrComp377, AROMAAggrComp378, AROMAAggrComp379, AROMAAggrComp380, AROMAAggrComp381, AROMAAggrComp382, AROMAAggrComp383, AROMAAggrComp384, AROMAAggrComp385, AROMAAggrComp386, AROMAAggrComp387, AROMAAggrComp388, AROMAAggrComp389, AROMAAggrComp390, AROMAAggrComp391, AROMAAggrComp392, AROMAAggrComp393, AROMAAggrComp394, AROMAAggrComp395, AROMAAggrComp396, AROMAAggrComp397, AROMAAggrComp398, AROMAAggrComp399, AROMAAggrComp400, AROMAAggrComp401, AROMAAggrComp402, AROMAAggrComp403, AROMAAggrComp404, AROMAAggrComp405, AROMAAggrComp406, AROMAAggrComp407, AROMAAggrComp408, AROMAAggrComp409, AROMAAggrComp410, AROMAAggrComp411, AROMAAggrComp412, AROMAAggrComp413, AROMAAggrComp414, AROMAAggrComp415, AROMAAggrComp416, AROMAAggrComp417, AROMAAggrComp418, AROMAAggrComp419, AROMAAggrComp420, AROMAAggrComp421, AROMAAggrComp422, AROMAAggrComp423, AROMAAggrComp424, AROMAAggrComp425, AROMAAggrComp426

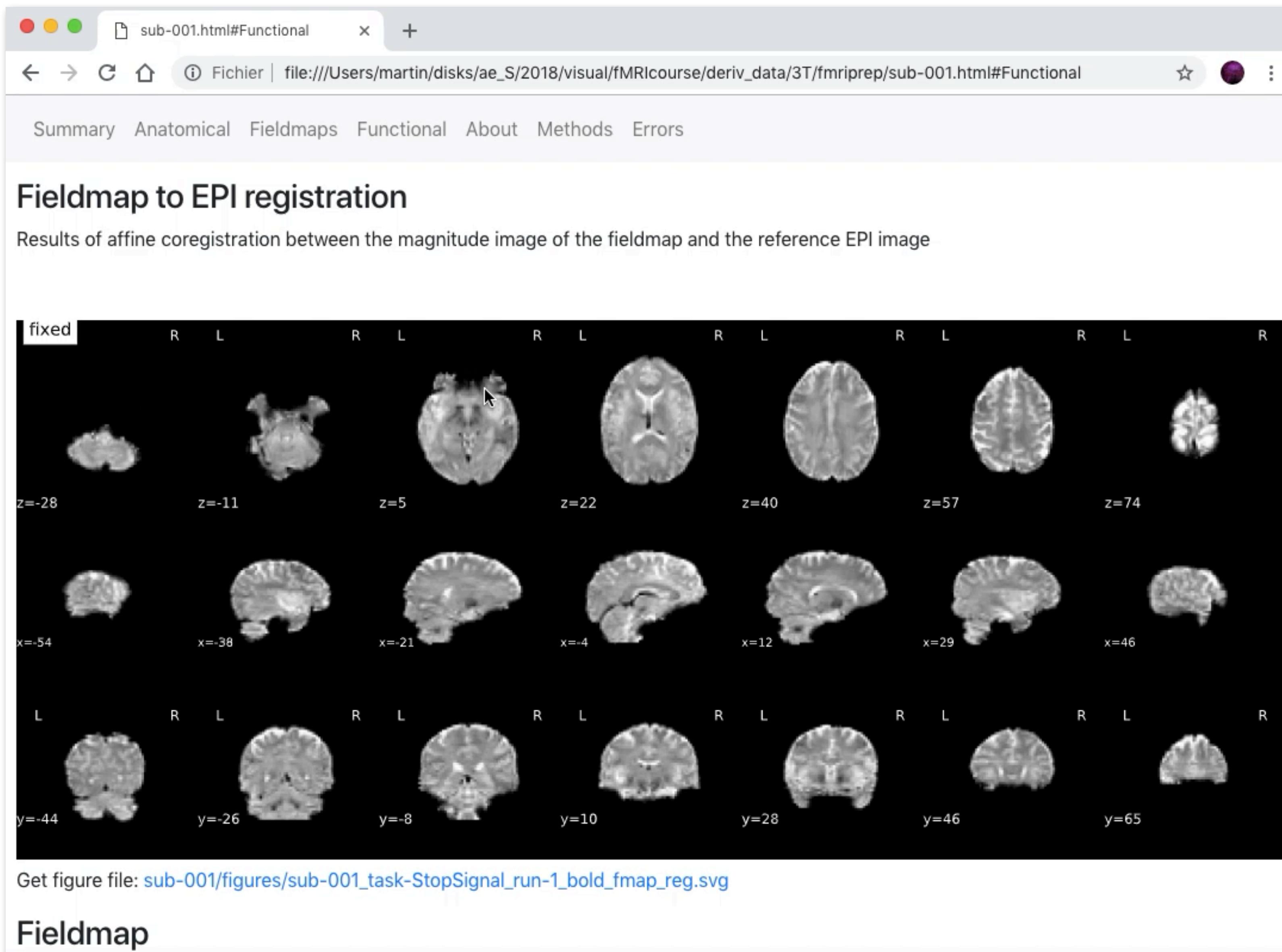
Note on orientation: qform matrix overwritten

The qform has been copied from sform.

Fieldmap to FPI registration

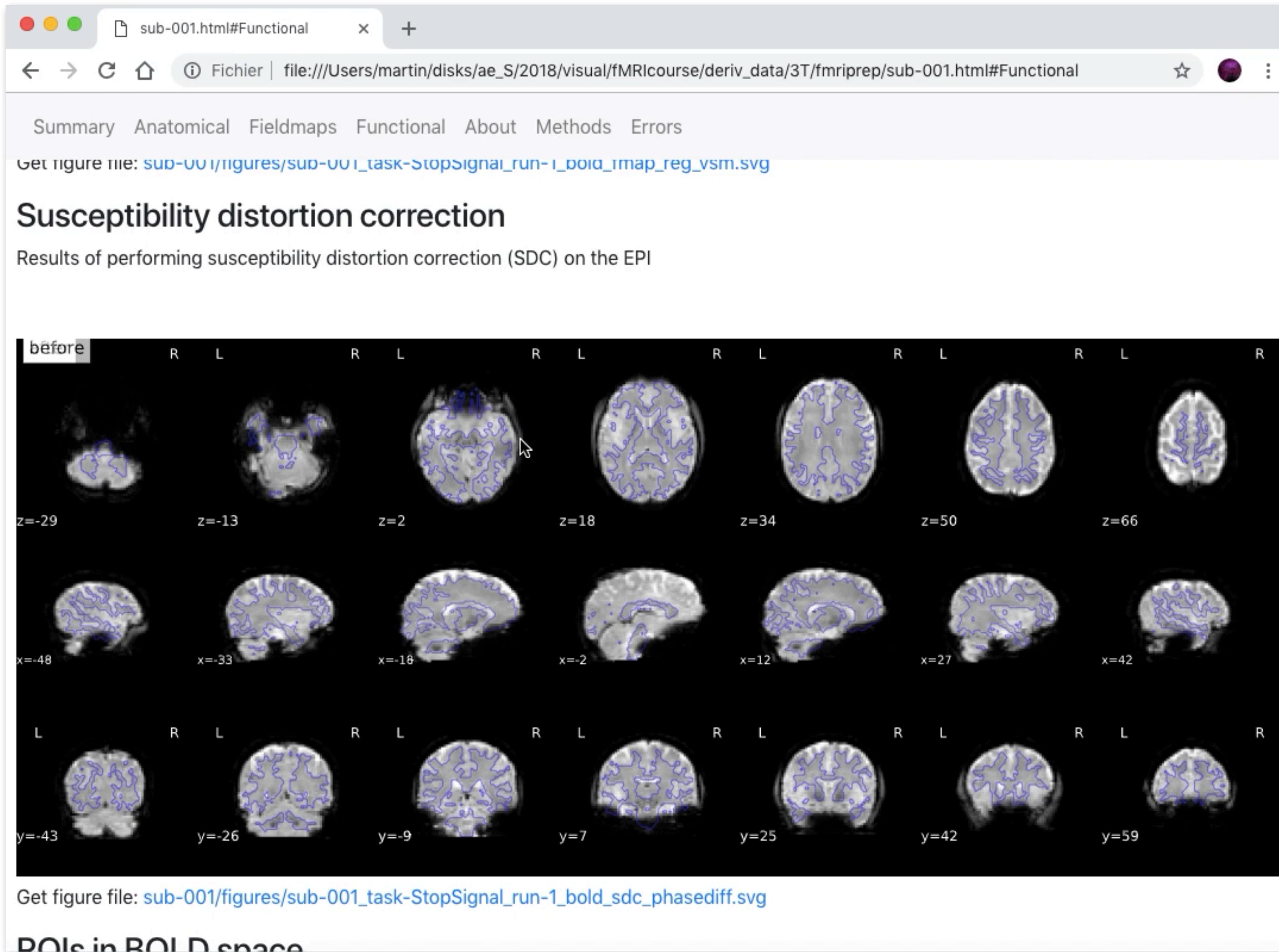
Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



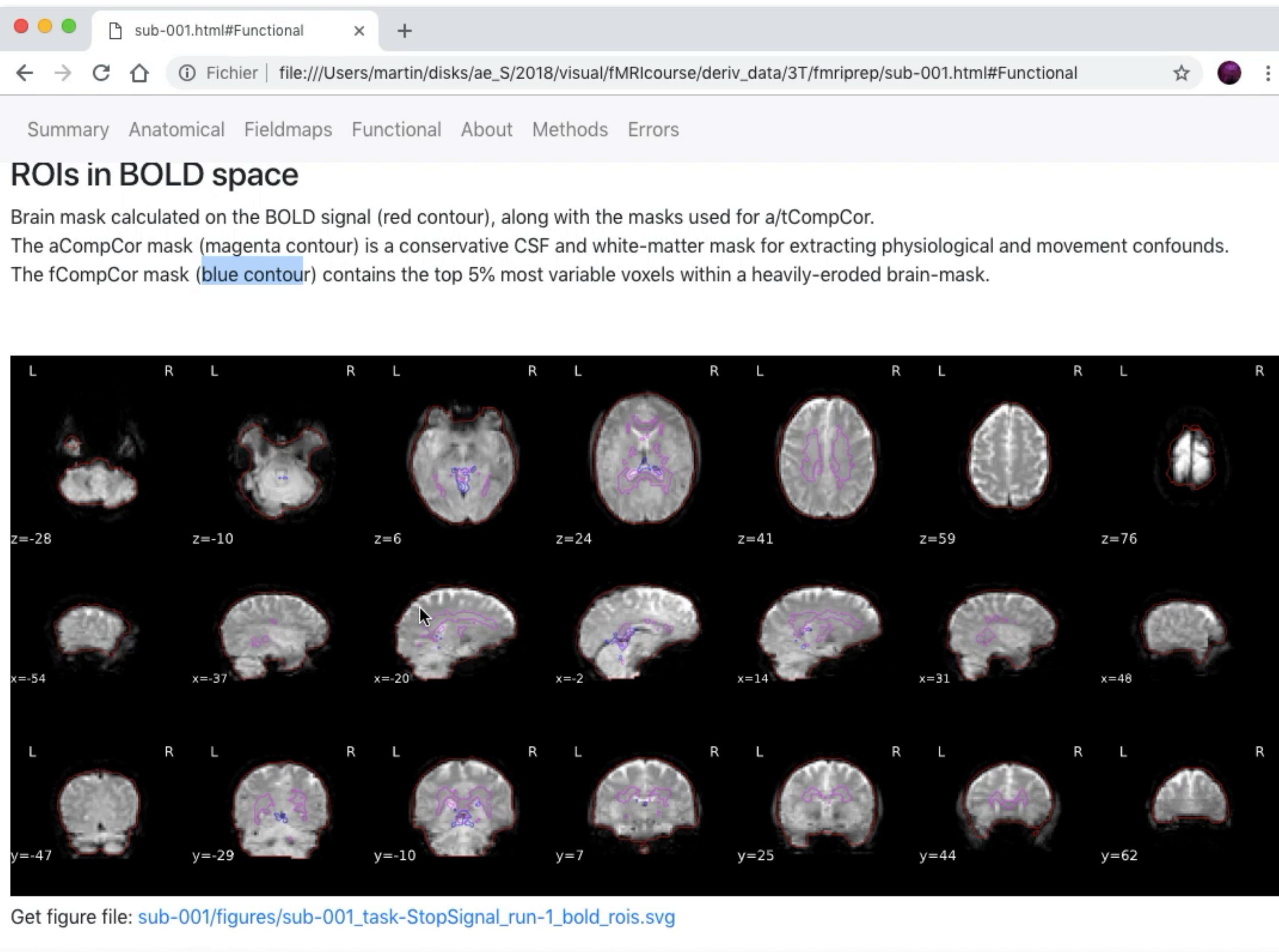
Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



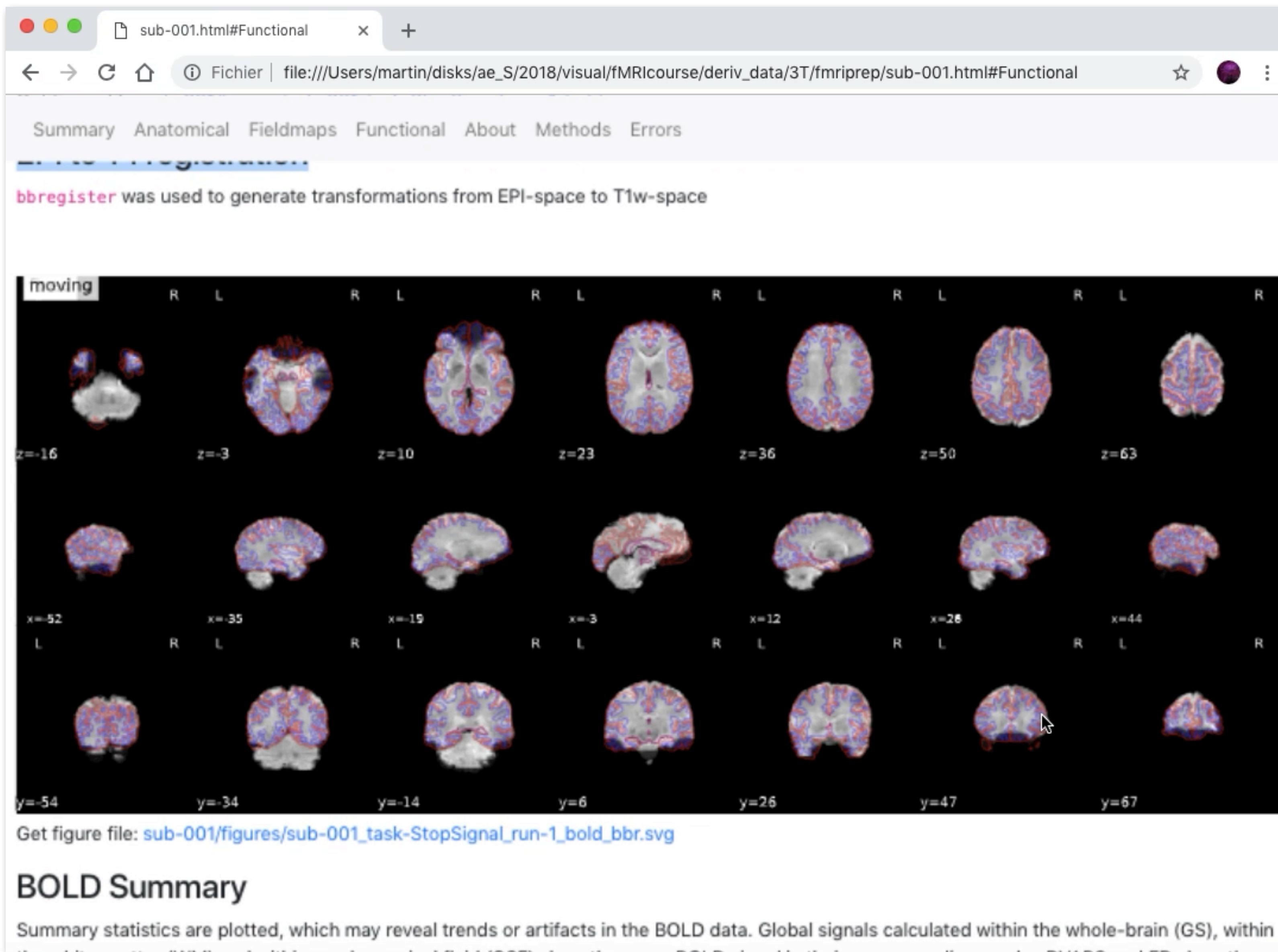
Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



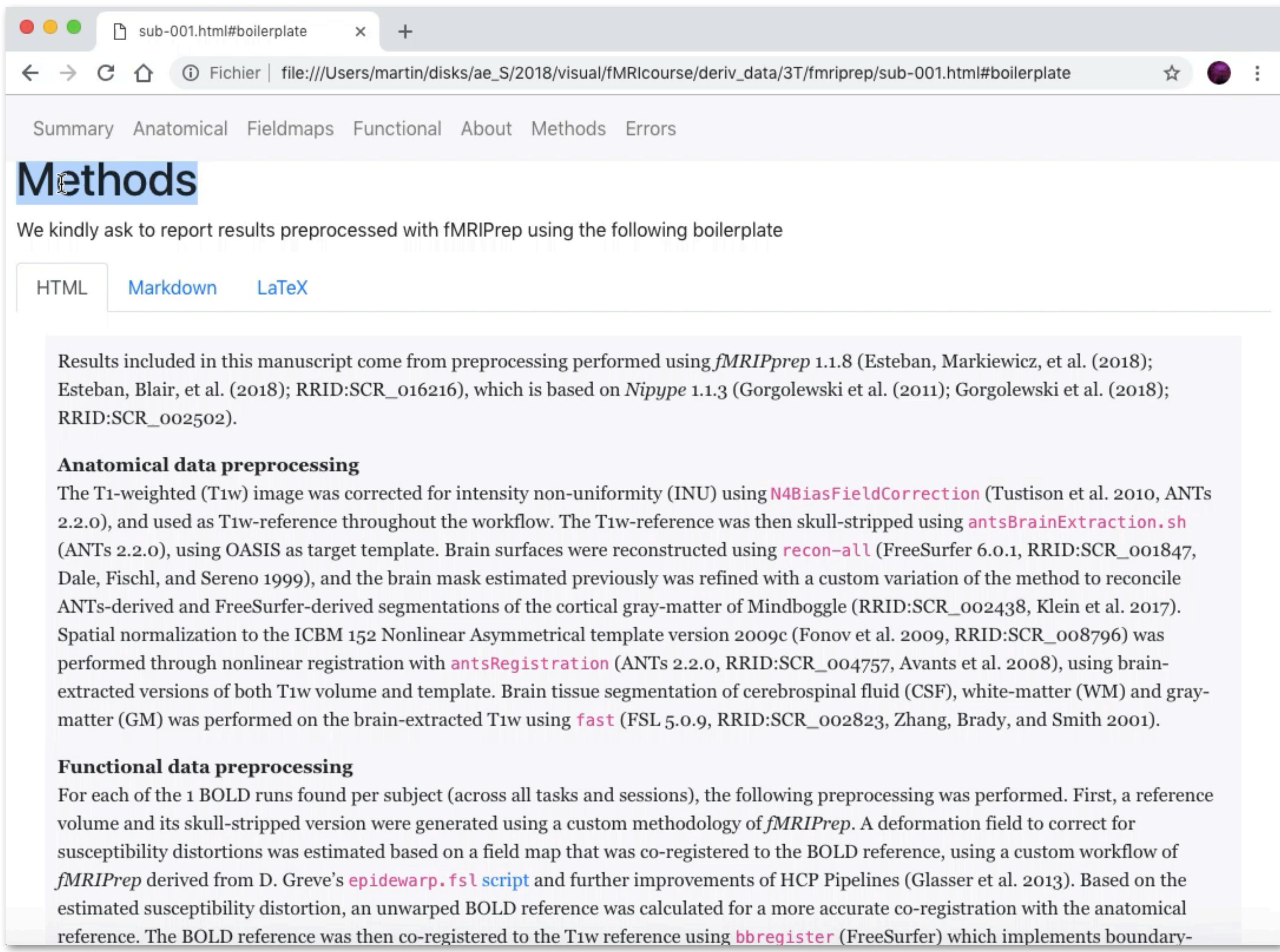
Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline



The screenshot shows a web browser window with the title "sub-001.html#boilerplate". The URL in the address bar is "file:///Users/martin/disks/ae_S/2018/visual/fMRIcourse/deriv_data/3T/fmriprep/sub-001.html#boilerplate". The page content is the "Methods" section of the fMRIPREP boilerplate. It includes a navigation bar with links to "Summary", "Anatomical", "Fieldmaps", "Functional", "About", "Methods", and "Errors". Below the navigation bar, the "Methods" section is highlighted with a blue background. A sub-section titled "We kindly ask to report results preprocessed with fMRIPrep using the following boilerplate" is shown. Under this, there are three tabs: "HTML" (selected), "Markdown", and "LaTeX". A text box contains the following text:

Results included in this manuscript come from preprocessing performed using *fMRIPrep* 1.1.8 (Esteban, Markiewicz, et al. (2018); Esteban, Blair, et al. (2018); RRID:SCR_016216), which is based on *Nipype* 1.1.3 (Gorgolewski et al. (2011); Gorgolewski et al. (2018); RRID:SCR_002502).

Anatomical data preprocessing

The T1-weighted (T1w) image was corrected for intensity non-uniformity (INU) using [N4BiasFieldCorrection](#) (Tustison et al. 2010, ANTs 2.2.0), and used as T1w-reference throughout the workflow. The T1w-reference was then skull-stripped using [antsBrainExtraction.sh](#) (ANTs 2.2.0), using OASIS as target template. Brain surfaces were reconstructed using [recon-all](#) (FreeSurfer 6.0.1, RRID:SCR_001847, Dale, Fischl, and Sereno 1999), and the brain mask estimated previously was refined with a custom variation of the method to reconcile ANTs-derived and FreeSurfer-derived segmentations of the cortical gray-matter of Mindboggle (RRID:SCR_002438, Klein et al. 2017). Spatial normalization to the ICBM 152 Nonlinear Asymmetrical template version 2009c (Fonov et al. 2009, RRID:SCR_008796) was performed through nonlinear registration with [antsRegistration](#) (ANTs 2.2.0, RRID:SCR_004757, Avants et al. 2008), using brain-extracted versions of both T1w volume and template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) was performed on the brain-extracted T1w using [fast](#) (FSL 5.0.9, RRID:SCR_002823, Zhang, Brady, and Smith 2001).

Functional data preprocessing

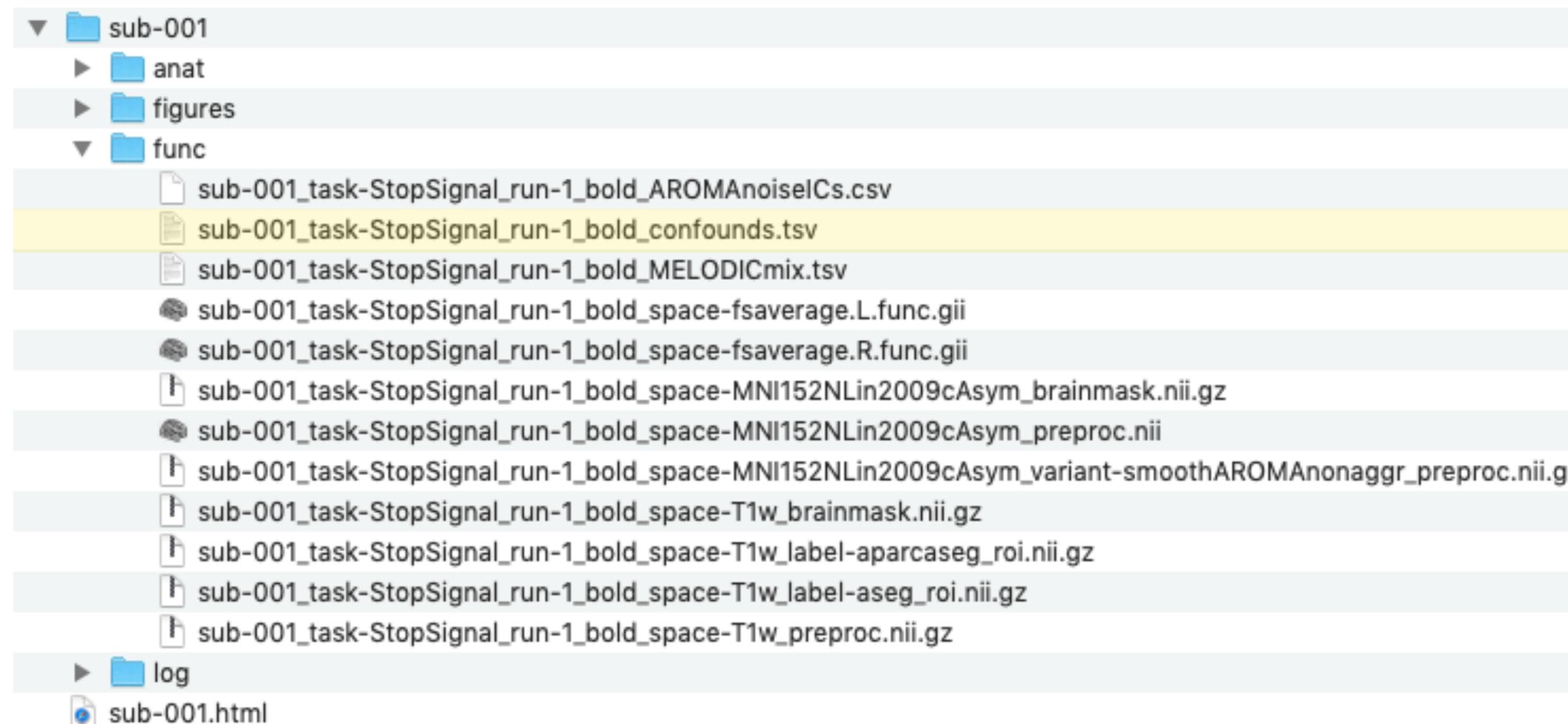
For each of the 1 BOLD runs found per subject (across all tasks and sessions), the following preprocessing was performed. First, a reference volume and its skull-stripped version were generated using a custom methodology of *fMRIPrep*. A deformation field to correct for susceptibility distortions was estimated based on a field map that was co-registered to the BOLD reference, using a custom workflow of *fMRIPrep* derived from D. Greve's [epidewarp.fsl script](#) and further improvements of HCP Pipelines (Glasser et al. 2013). Based on the estimated susceptibility distortion, an unwarped BOLD reference was calculated for a more accurate co-registration with the anatomical reference. The BOLD reference was then co-registered to the T1w reference using [bbregister](#) (FreeSurfer) which implements boundary-

Esteban et al. (2018) *BioRxiv*
<http://fmriprep.readthedocs.io/>

FMRIPREP: a robust fMRI preprocessing pipeline

FMRIPREP will then produce 2 classes of outputs:

- preprocessed data (*data after corrections, resampled onto a target space*)
- confounds (*additional time series, motion parameters, PCA, ICA, CompCor, tCor, etc*)



FMRIPREP: a robust fMRI preprocessing pipeline

CSF	WhiteMatter	GlobalSignal	stdDVARS	non-stdDVARS	vx-wisestDVARS	FramewiseDisplacement	tCompCor00	tCompCor01	tCompCor02	tCompCor03	tCompCor04	tCompCor05
1865.5356	1413.4355	1293.4099	n/a	n/a	n/a	n/a	0	0	0	0	0	0
1739.3878	1401.8833	1258.9131	2.48462486	53.25794983	3.03803492	0.27152478436661	-0.1082883426	-0.01837308219999998	0.0853874057	0.029846082200000002	-0.0067984279	0.0824982101999999
1722.0355	1400.7476	1256.5485	1.11643708	23.93083572	1.02100766	0.0561268467	-0.0962955867	0.0266242004	-0.01226994689999999	-0.0131689247	-0.0029203742	0.128968778
1713.3002	1399.7977	1254.8053	1.10935032	23.77893257	0.96454	0.0839705547	-0.0381954336	0.0213888733	-0.041015965	0.0802395858	-0.0267013348	0.1158796731000001
1707.505	1398.6218	1253.6846	1.02736163	22.02150345	0.95201927	0.065083961	-0.0162271078	-0.008052339	0.0004476625	0.0735789927	-0.0157581044	0.024136448
1712.104	1398.0255	1252.6759	1.21943057	26.13850403	0.96918058	0.0546204000000001	-0.049221327	0.0172719917	0.0652598849	-0.0431586834	-0.03719791529999995	0.03497709740000004
1701.9543	1397.6012	1251.9409	1.1558181	24.7749691	0.96723455	0.0521067700000001	-0.0126860001	0.0075293976	-0.066532533	0.0331369389	-0.02206503410000002	0.040943938
1702.156	1398.4846	1252.0835	1.01854622	21.83254623	0.94425833	0.0917145699999998	-0.0201527147	-0.0189338176	0.0077397086	0.05392402509999995	-0.0205005553	0.06422364
1704.3052	1399.0449	1251.8899	1.06918848	22.91806221	0.97007698	0.0820899799999999	-0.0198662491	-0.01482642580000001	0.0782737758	0.0492015564	-0.0640090183	0.04437962929999994
1701.7227	1398.4667	1249.7021	1.18893182	25.48476219	0.96828435	0.0700657600000001	-0.0337773219	0.05536338769999995	-0.00586236930000001	-0.0471678096	-0.0552511633	0.045968886
1695.9918	1397.1357	1249.528	1.13029492	24.22788048	0.953152	0.04604244000000004	0.01721477060000002	0.03070063009999997	-0.0286988822	0.0561166249	-0.0377335395	-0.009879338
1696.2263	1397.0487	1249.5428	1.01035142	21.65689087	0.93157488	0.04037528	0.0223632622	0.0026352743	0.041530693	0.0729481213	-0.0497291419	-0.002384706
1698.8152	1398.071000000001	1250.1428	1.11271334	23.85101891	0.95970273	0.0424108500000003	-0.0246379247	0.0508953779	0.01642898280000002	-0.0451004172	-0.0628150208	0.025956230
1696.001	1398.3983	1250.4415	1.06363034	22.79892349	0.94743145	0.0707419	0.0031233684	0.0363684963	-0.0342496896	0.0135994328	0.00657603579999994	0.015928792
1697.6713	1398.5614	1251.0082	1.12956345	24.21220016	0.95072162	0.0976543000000003	-0.0037223138	0.0415226629	0.0157615695	0.0557991508	-0.0677190604	0.1008625033000001
1705.7054	1398.0486	1248.293999999999	1.0700388	22.93628883	0.97264636	0.06719562	-0.0318814621	0.050379619	0.0805498347	0.0428753297	-0.0580737934	0.101555408
1693.6549	1398.0017	1244.4111	1.3408947	28.74208832	1.02265978	0.06468251	0.0162090188	0.0800771082	-0.0281045861	0.0004778933	0.0093743996	0.03113418049999997
1687.0986	1397.942	1244.6506	1.09320486	23.4328537	0.95818901	0.1113806899999996	0.0792578137	0.0682480686	3.56228E-05	0.0226748018	-0.01587992600000003	-0.047335743
1686.2446	1397.8307	1246.075	1.02005315	21.86484528	0.94802666	0.0748841399999997	0.083662758	0.0384677314	0.0800149433	0.0410376471	-0.04434057370000004	-0.02245020479999998
1690.4603	1398.6678	1247.8354	1.14709926	24.58808136	0.99370128	0.10118756	0.0515718332	0.047728219	0.0253551646	-0.0780267545	-0.00492338780000001	-0.040550728
1689.6221	1398.1027	1248.8009	1.11240733	23.84445953	0.98441565	0.0940689499999997	0.0589209111	0.039505689890000004	-0.02453099089999997	0.02396222029999997	-0.0208285144	-0.004225549
1687.5603	1399.2957	1250.2439	0.9997316	21.42925453	0.93952823	0.0568365520000001	0.0631292309	0.00702900780000001	0.0020844344	0.045070448	-0.0542344929	-0.06536963
1697.0363	1399.1079	1251.8442	1.09106481	23.38698196	0.97277755	0.0524654519999994	0.0014515266	0.02158532779999998	0.0637665467999999	-0.0344160541	-0.0411298171	-0.067275434
1692.4015	1400.7449	1252.4409	1.08159387	23.18397331	0.95605707	0.03531519300000016	0.0047937316	0.0255778892	-0.0412400589	-0.0376521635	-0.0078319476	-0.0994220525999999
1699.7062	1400.087000000002	1253.8607	1.13369179	24.3006916	0.94998127	0.06160961300000035	-0.0325668785	0.027903935	-0.0143267505	0.0002823237	0.0087766712	-0.074507087
1715.8259	1401.842	1253.4563	1.12305498	24.07269096	0.98207134	0.0591921500000001	-0.0839799939999999	0.03088809339999997	0.0386365091	0.00814393	0.0156512653	0.067868617
1703.1566	1402.2036	1249.5548	1.30889976	28.05627632	1.00599384	0.0881499500000002	-0.05216973809999995	0.0721846886	-0.01520332699999999	0.0045072985	0.0770222904	-0.0123548
1692.1411	1398.9689	1248.4814	1.17841411	25.25931358	0.99848378	0.15933636	0.01631172439999998	0.0765372433	-0.01164606669999999	0.011472185	0.005383381	0.002232916
1694.6073	1395.5304	1245.8844	1.07011294	22.93787956	0.96369348	0.13688522	0.03643170979999996	0.04283612150000005	0.0703542311	0.100625094	0.028364775	0.020745008
1690.6445	1398.4669	1246.1017	1.33602965	28.63780594	1.03114557	0.0682995199999999	0.0367339048	0.0654293758	0.0137661504	-0.0774649511	0.02083730169999998	-0.041785721
1684.7263	1397.5526	1247.7566	1.18362772	25.37106895	0.99935275	0.1122395899999996	0.0941370662	0.0374871213	-0.0160950803	0.0237387225	-0.0195663099	-0.027994314
1684.4802	1397.2258	1250.549	1.03783631	22.2460289	0.95816493	0.099361864	0.0862572189	-0.0151925993	0.0096474709	-0.0044176653	-0.0043209001	-0.060897257
1698.7334	1398.6064	1254.1326	1.1008395	23.5965023	0.98773438	0.0790021560000003	-0.0028815493	-0.0368487053	0.0491079345	-0.0812200078000001	0.0056182794	-0.02956311399999998

FMRIPREP: a robust fMRI preprocessing pipeline

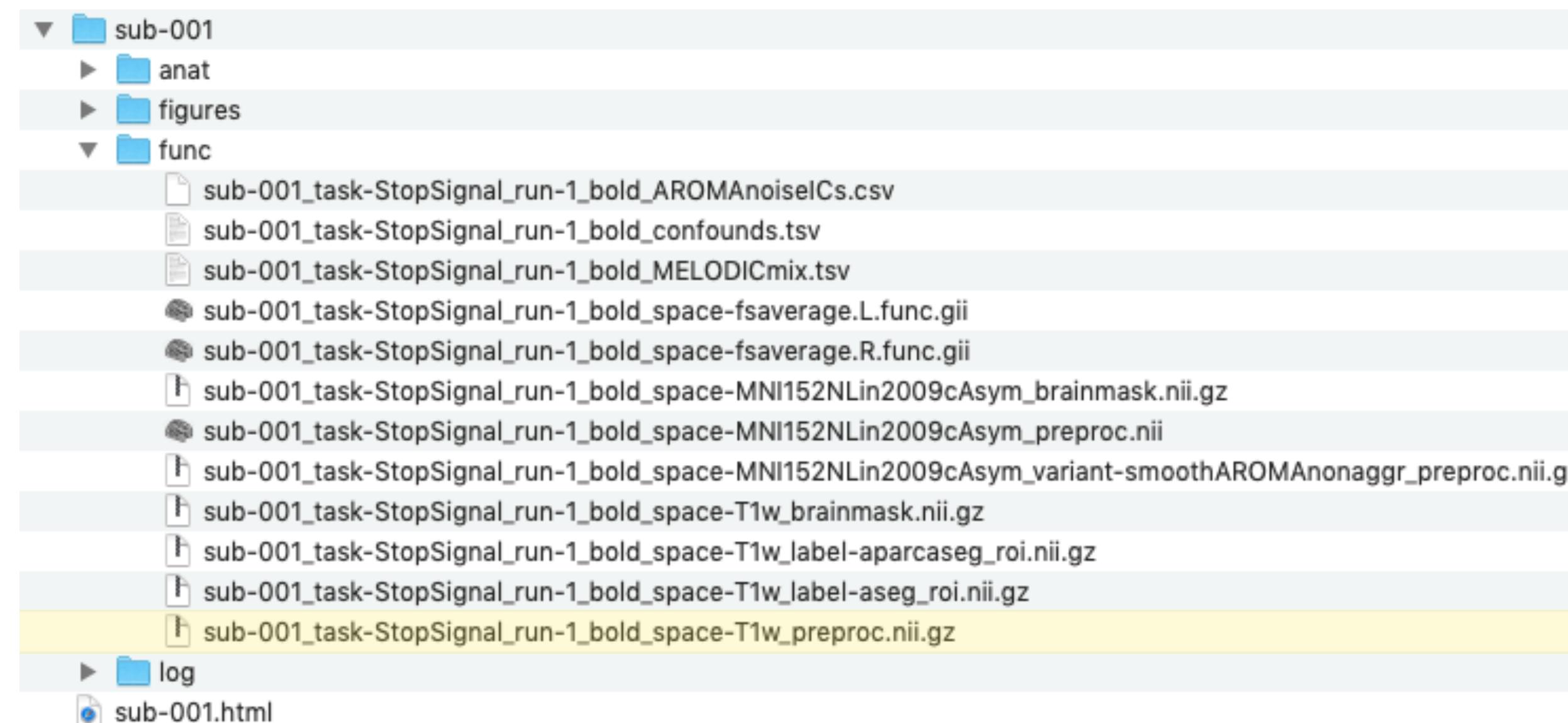
Columns represent the different confounds:

- csf and white_matter: average signal inside CSF and WM masks across time
- global_signal corresponds to the mean time series within the brain mask
- dvars and std_dvars relate to the derivative of root mean square variance over voxels
- framewise_displacement is a quantification of the estimated bulk-head motion
- trans_x, trans_y, trans_z, rot_x, rot_y, rot_z: motion-correction parameters
- anatomical (a_comp_cor_XX) and temporal (t_comp_cor_XX) variants of *CompCor*

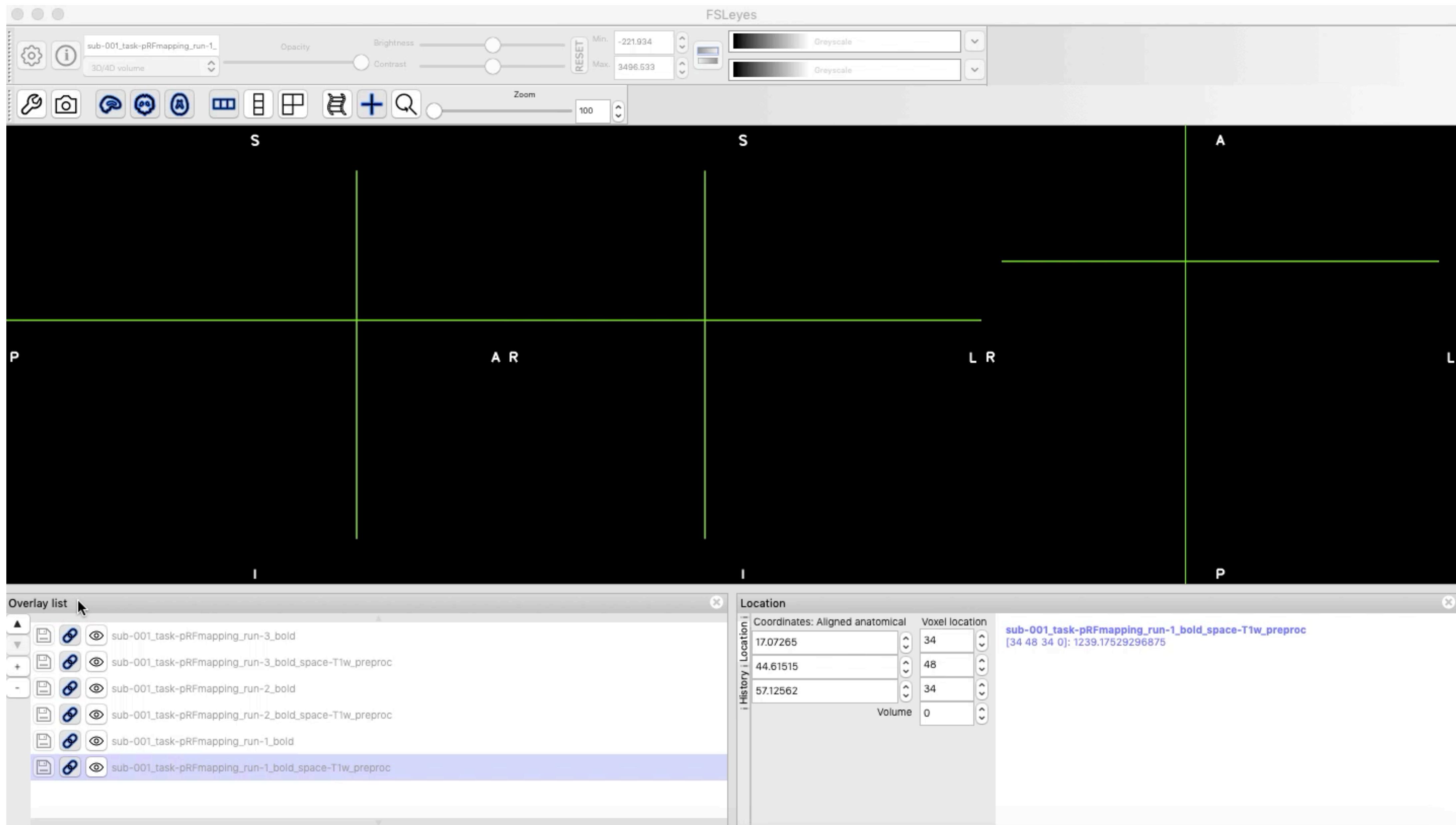
FMRIPREP: a robust fMRI preprocessing pipeline

FMRIPREP will then produce 2 classes of outputs:

- preprocessed data (*data after corrections, resampled onto a target space*)
- confounds (*additional time series, motion parameters, PCA, ICA, CompCor, tCor, etc*)



FMRIPREP: a robust fMRI preprocessing pipeline



Questions?

More questions:

martin.szinte@gmail.com