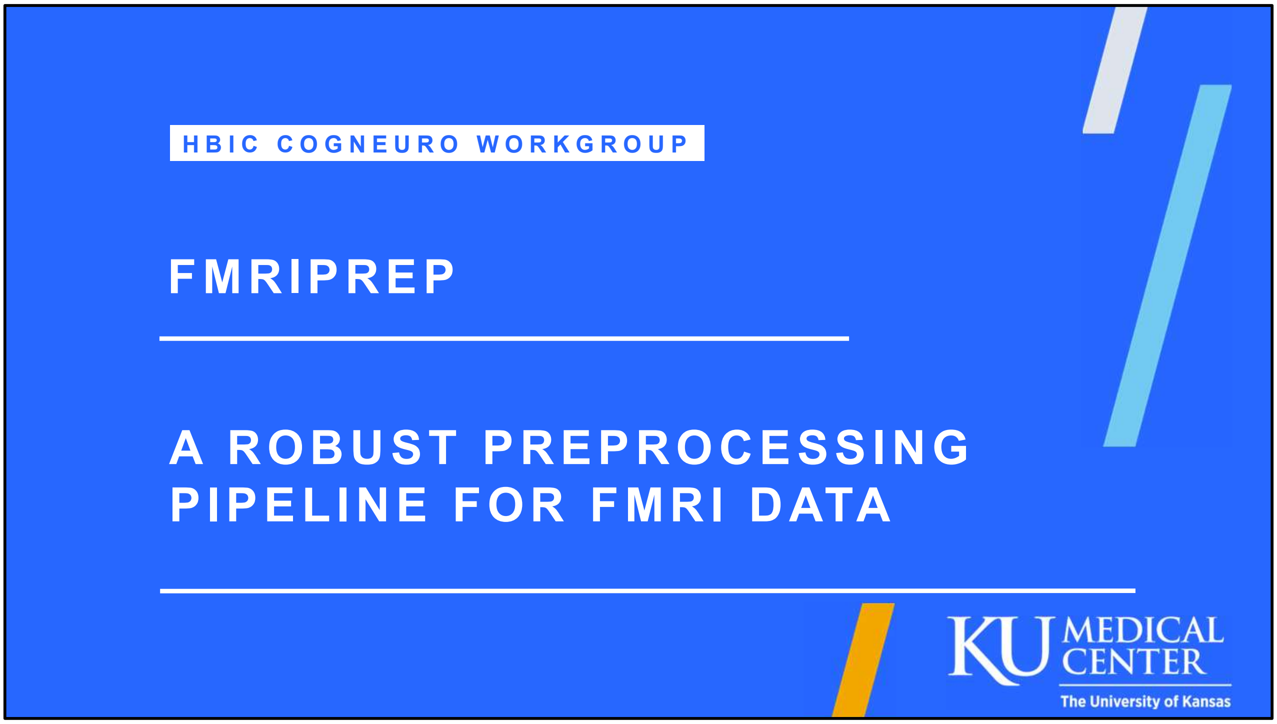


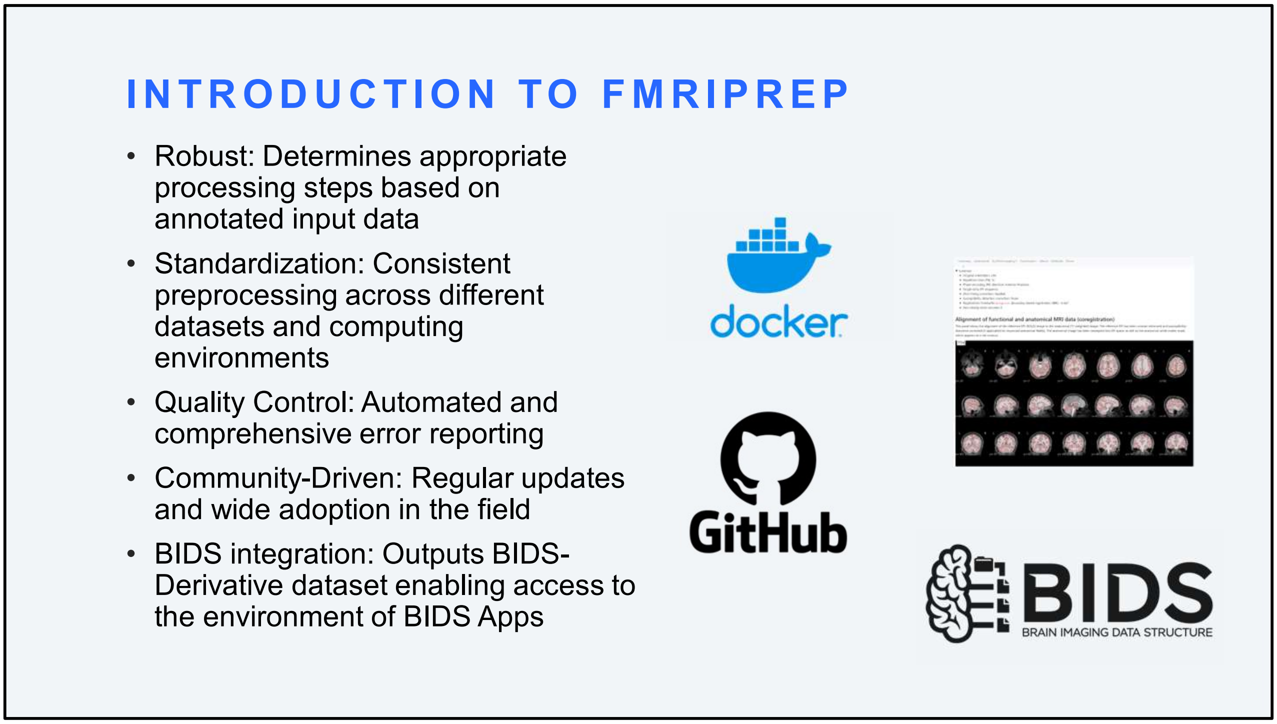
HBIC COGNEURO WORKGROUP

FMRIPREP

A ROBUST PREPROCESSING PIPELINE FOR FMRI DATA

KU MEDICAL CENTER
The University of Kansas

2024-06-20
revision 1.0



INTRODUCTION TO FMRIPREP

- Robust: Determines appropriate processing steps based on annotated input data
- Standardization: Consistent preprocessing across different datasets and computing environments
- Quality Control: Automated and comprehensive error reporting
- Community-Driven: Regular updates and wide adoption in the field
- BIDS integration: Outputs BIDS-Derivative dataset enabling access to the environment of BIDS Apps

Standardization

- It’s common when starting a new study to copy and edit the processing pipeline from a previous study. This can inadvertently carry over hard-coded values or other decisions that may have applied to one experiment but not another.
- Software packages and their dependencies vary over time and across environments – windows, mac, linux. fMRIPrep is distributed as a docker container, which is a self-contained environment with consistent versions of AFNI, FSL, Java, Python, etc.. If you use the same container to process data on two different computers at two different times, you will get the same output.

Quality Control

- fMRIPrep outputs comprehensive visual reports to help identify preprocessing issues
- It provides boilerplate preprocessing methods text that you can – and should – copy & paste directly into a manuscript. Accurate reporting of preprocessing decisions is a significant issue in fMRI, but fMRIPrep standardizes this for you.

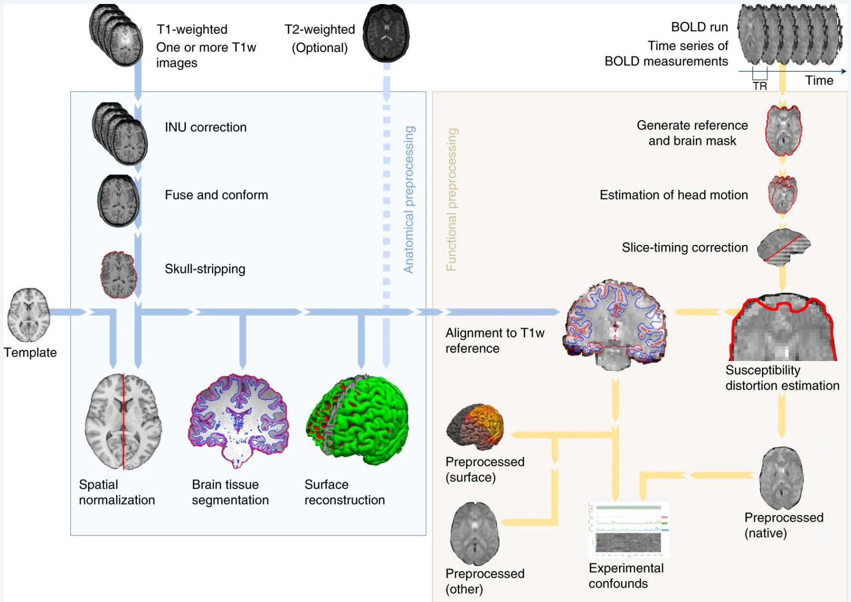
Community Driven

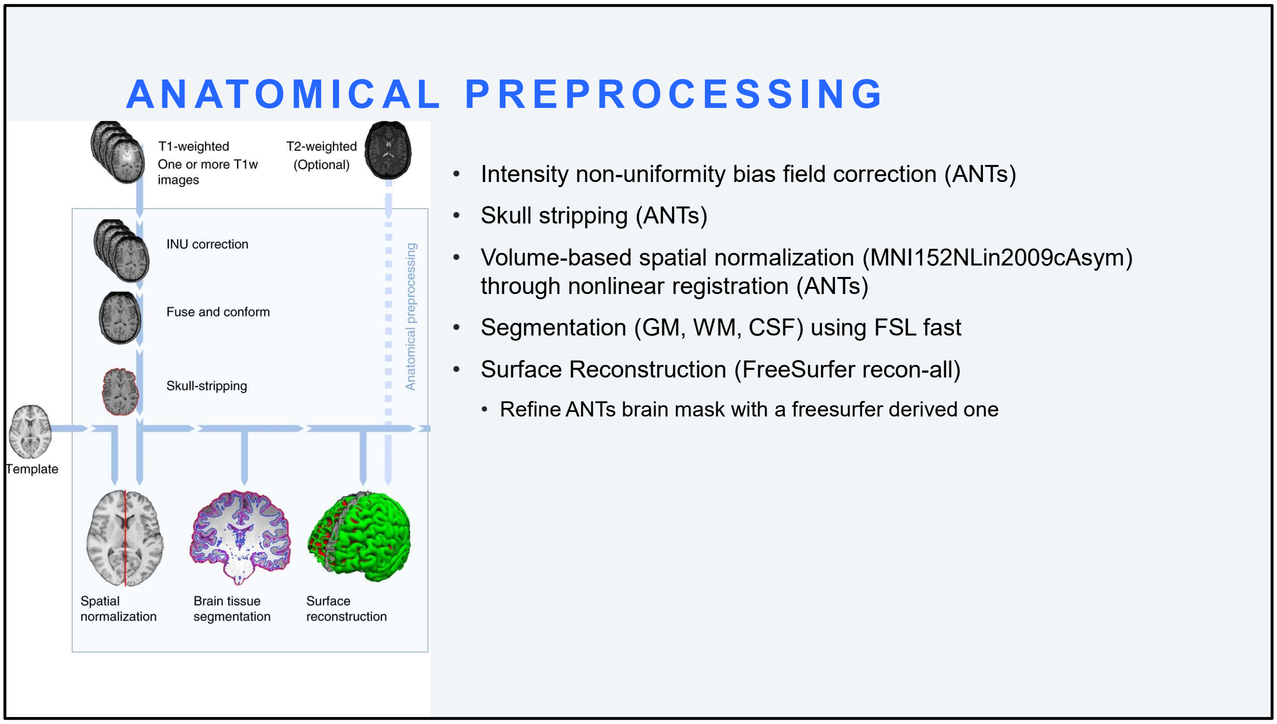
- This is part of a broader trend in open-source collaborative research methods. fMRIPrep has a public github that users have already used to expand its functionality to include support for multi-echo BOLD, Lesion masking, output formats, among other changes.

BIDS

- BIDS is the Brain Imaging Data Structure, which seeks to standardize how an imaging dataset is structured and documented. This not only removes a significant obstacle to data sharing and replicability, it also enables standardization of data processing and analysis tools. fMRIPrep is just one BIDS App in this ecosystem, and it relies on the documentation that you provide and characteristics of the data to determine the appropriate preprocessing pipeline.
- fMRIPrep outputs its data in a BIDS compliant way, allowing you to plug it directly into other BIDS Apps for nuisance variable regression, functional connectivity analysis, etc.

FMRIPREP METHOD



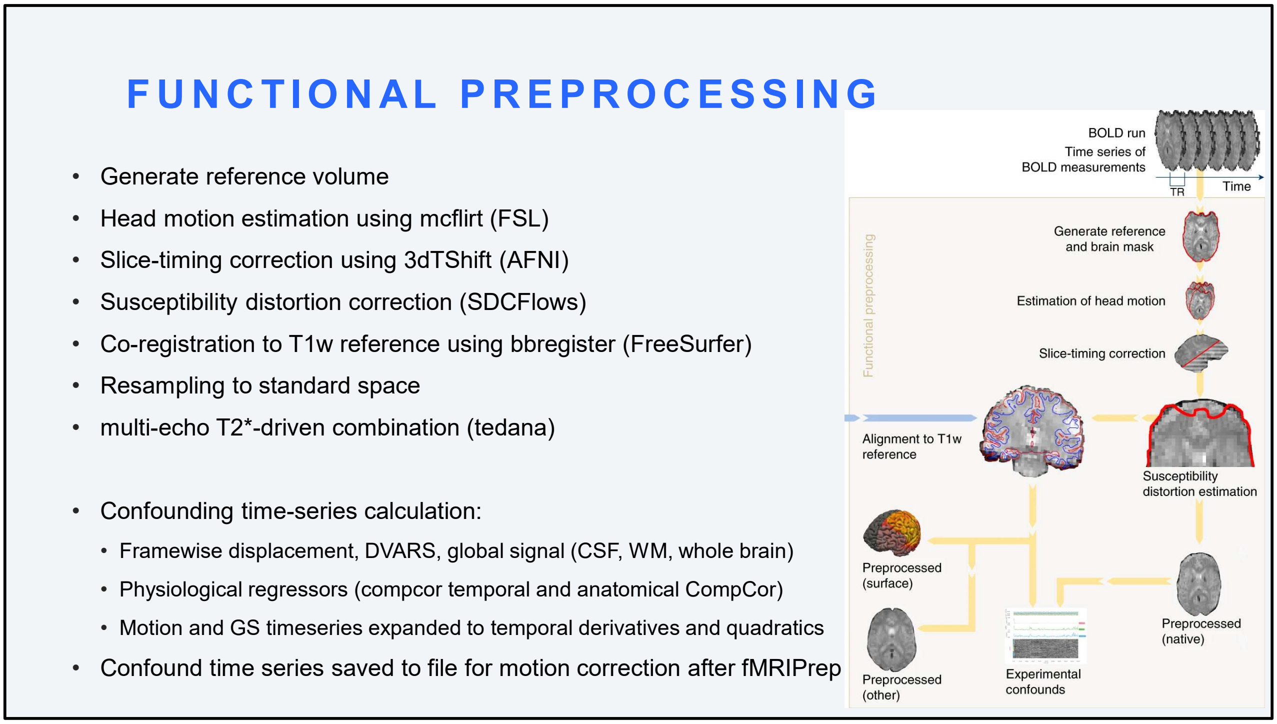


ANATOMICAL PREPROCESSING

- Intensity non-uniformity bias field correction (ANTs)
- Skull stripping (ANTs)
- Volume-based spatial normalization (MNI152NLin2009cAsym) through nonlinear registration (ANTs)
- Segmentation (GM, WM, CSF) using FSL fast
- Surface Reconstruction (FreeSurfer recon-all)
  - Refine ANTs brain mask with a freesurfer derived one

See https://fmriprep.org/en/stable/workflows.html

- Construct an average image by conforming all T1w images to RAS orientation and a common voxel size, and averaging them into a single reference template
- Correct each T1w image for intensity non-uniformity (INU) with ANTs' N4BiasFieldCorrection
- Skull-strip the T1w reference image using ANTs' antsBrainExtraction.sh (atlas-based extraciton)
- Spatial normalization to standard spaces using ANTs' antsRegistration in a multiscale, mutual-information based, nonlinear registration scheme.
- Brain tissue segmentation using FSL fast
- Surface reconstruction from T1w/T2w structural images using FreeSurfer:
  - Basic reconstruction (autorecon1) with T1w and T2w structural images; skip skull-stripping
  - Import the previously calculated brainmask from ANTs
  - Resume reconstruction using the T2w image to assist in finding the pial surface, if available.
- Replace the original brain mask (from antsBrainExtraction.sh) with a refined one that derives from the Freesurfer aseg.mgz file



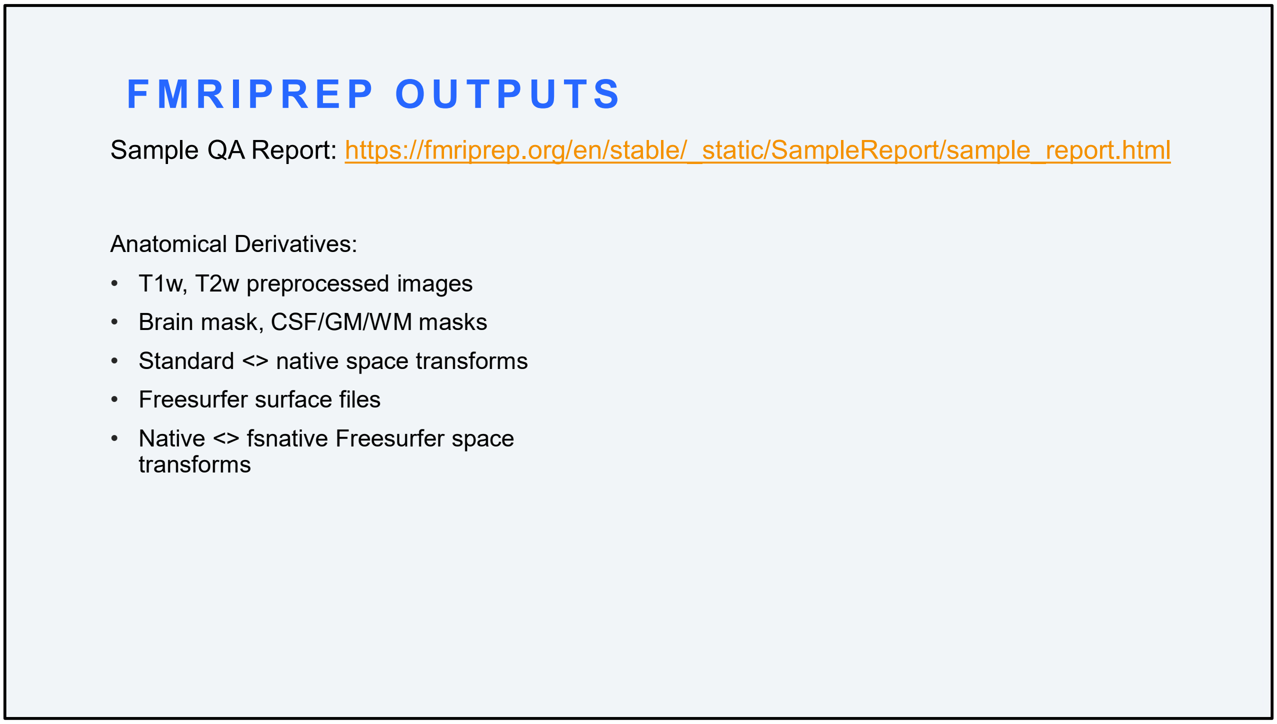
FUNCTIONAL PREPROCESSING

- Generate reference volume
- Head motion estimation using mcflirt (FSL)
- Slice-timing correction using 3dTShift (AFNI)
- Susceptibility distortion correction (SDCFlows)
- Co-registration to T1w reference using bbregister (FreeSurfer)
- Resampling to standard space
- multi-echo T2*-driven combination (tedana)

- Confounding time-series calculation:
  - Framewise displacement, DVARS, global signal (CSF, WM, whole brain)
  - Physiological regressors (compcor temporal and anatomical CompCor)
  - Motion and GS timeseries expanded to temporal derivatives and quadratics
- Confound time series saved to file for motion correction after fMRIPrep

See https://fmriprep.org/en/stable/workflows.html

- Generate reference volume
- Head motion estimation (transformation matrices + six rotation/translation parameters) using mcflirt (FSL), before slice-timing correction
- Slice-timing correction, realigned to the middle of each TR using AFNI 3dTShift
- Susceptibility distortion correction using acquired fieldmaps to correct for phase-encoding shifts in EPI (SDCFlows)
- Boundary-based co-registration to T1w reference (?h.white surfaces from recon-all) using bbregister (FreeSurfer), configured with six degrees of freedom. Alternatively, FSL flirt with the BBR cost function is run if FreeSurfer is disabled.
- Resampling to standard space. All transformations (head-motion, SDC, co-registration to T1w, anatomical-to-standard) concatenated and applied in one step
- Tedana T2* workflow to optimally combine multi-echo data. T2* and S0 estimated by nonlinear regression (curvefit)
- framewise displacement (FD) and DVARS calculated using Nipype following definitions by Power et al. 2014
- Global signals extracted within CSF, WM, and whole-brain masks
- Physiological regressors for component-based noise correction (*CompCor*)
- Temporal derivatives and quadratic terms for head-motion and global signal



FMRIPREP OUTPUTS

Sample QA Report: https://fmriprep.org/en/stable/_static/SampleReport/sample_report.html

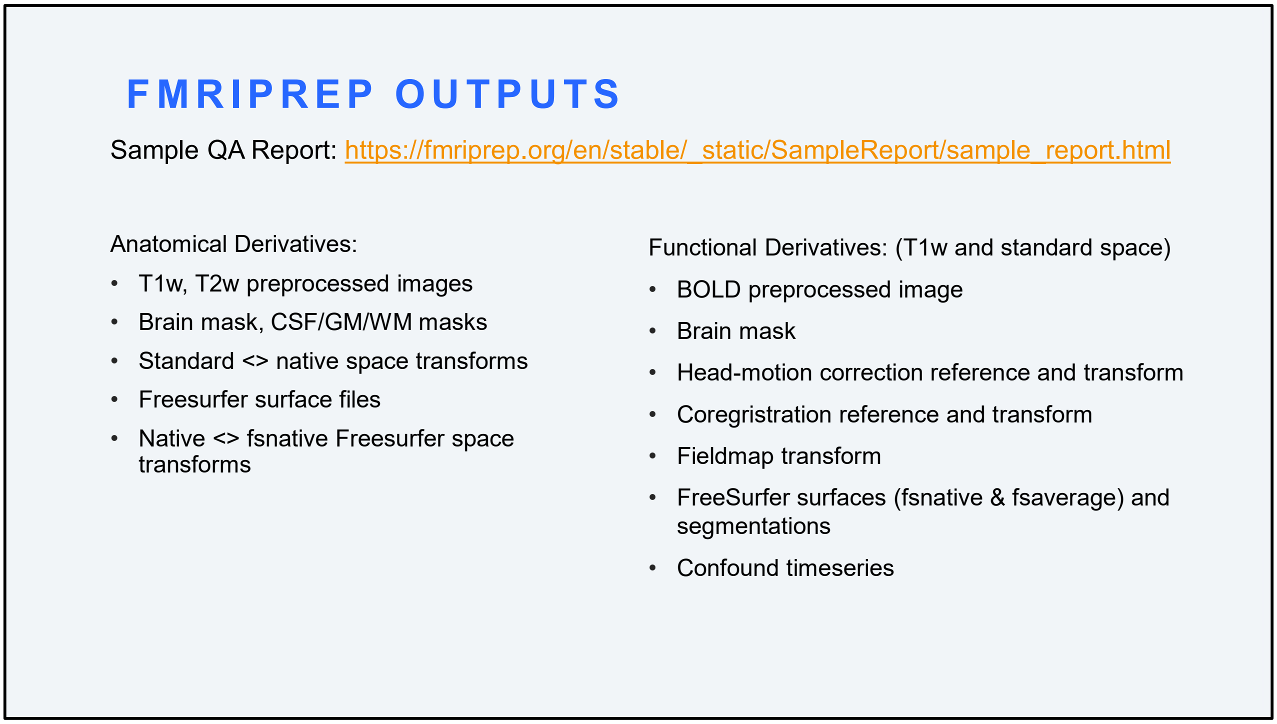
Anatomical Derivatives:

- T1w, T2w preprocessed images
- Brain mask, CSF/GM/WM masks
- Standard <> native space transforms
- Freesurfer surface files
- Native <> fsnative Freesurfer space transforms

https://fmriprep.org/en/stable/_static/SampleReport/sample_report.html
https://fmriprep.org/en/stable/outputs.html#anatomical-derivatives

```
sub-<subject_label>/
  anat/
    sub-<subject_label>[_space-<space_label>]_desc-preproc_[T1w, T2w].nii.gz
    sub-<subject_label>[_space-<space_label>]_desc-brain_mask.nii.gz
    sub-<subject_label>[_space-<space_label>]_dseg.nii.gz
    sub-<subject_label>[_space-<space_label>]_label-[CSF, GM, WM]_probseg.nii.gz
    sub-<subject_label>_from-MNI152NLin2009cAsym_to-T1w_mode-image_xfm.h5
    sub-<subject_label>_from-T1w_to-MNI152NLin2009cAsym_mode-image_xfm.h5
    sub-<subject_label>_hemi-[LR]_white.surf.gii
    sub-<subject_label>_hemi-[LR]_midthickness.surf.gii
    sub-<subject_label>_hemi-[LR]_pial.surf.gii
    sub-<subject_label>_hemi-[LR]_desc-reg_sphere.surf.gii
    sub-<subject_label>_hemi-[LR]_space-fsLR_desc-reg_sphere.surf.gii
    sub-<subject_label>_hemi-[LR]_space-fsLR_desc-msmsulc_sphere.surf.gii
    sub-<subject_label>_from-fsnative_to-T1w_mode-image_xfm.txt
    sub-<subject_label>_from-T1w_to-fsnative_mode-image_xfm.txt
```

Freesurfer derivatives saved to <output_dir>/sourcedata/freesurfer



FMRIPREP OUTPUTS

Sample QA Report: https://fmriprep.org/en/stable/_static/SampleReport/sample_report.html

Anatomical Derivatives:

- T1w, T2w preprocessed images
- Brain mask, CSF/GM/WM masks
- Standard <> native space transforms
- Freesurfer surface files
- Native <> fsnative Freesurfer space transforms

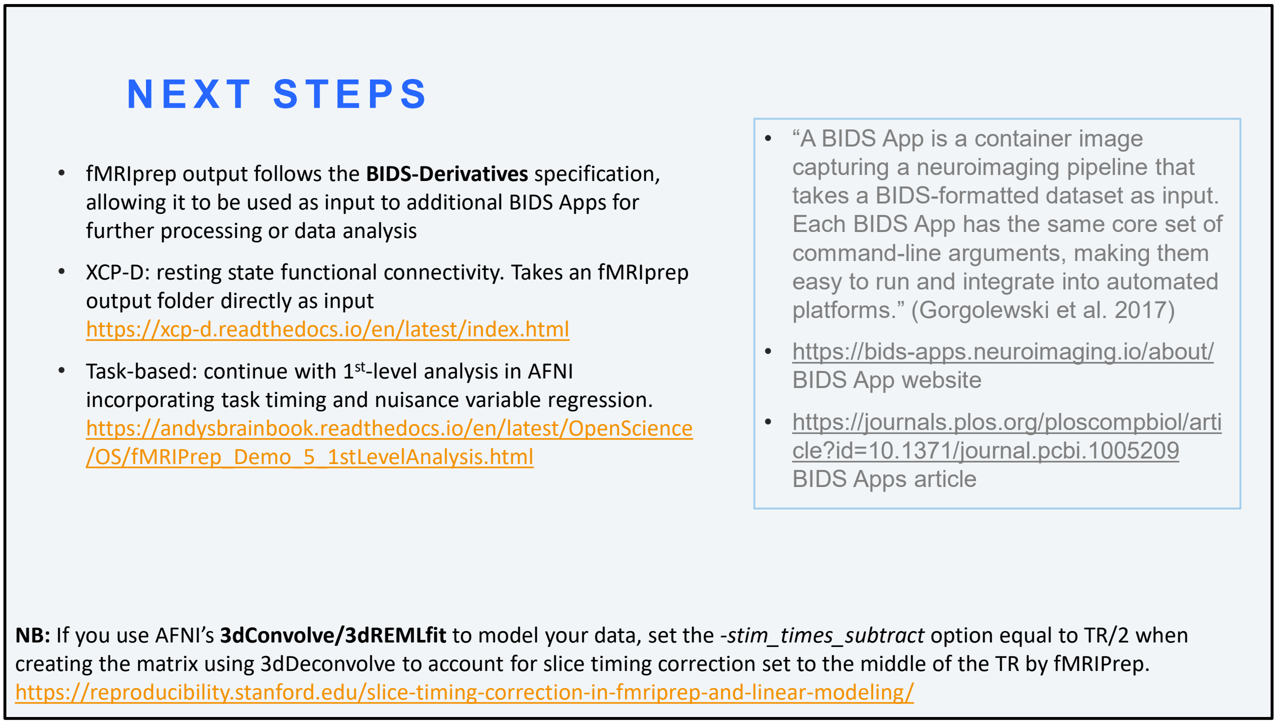
Functional Derivatives: (T1w and standard space)

- BOLD preprocessed image
- Brain mask
- Head-motion correction reference and transform
- Coregristration reference and transform
- Fieldmap transform
- FreeSurfer surfaces (fsnative & fsaverage) and segmentations
- Confound timeseries

https://fmriprep.org/en/stable/_static/SampleReport/sample_report.html
https://fmriprep.org/en/stable/outputs.html#functional-derivatives

```
sub-<subject_label>/
  func/
    sub-<subject_label>_task-<task_label>_run-<run_index>_space-
<space_label>_desc-brain_mask.nii.gz
    _space-<space_label>_desc-preproc_bold.nii.gz
    _desc-hmc_boldref.nii.gz
    _from-orig_to_boldref_mode-image_desc-hmc_xfm.nii.gz
    _desc-coreg_boldref.nii.gz
    _from-boldref_to-T1w_mode-image_desc-coreg_xfm.txt
    _from-boldref_to-TOPUP_mode-image_xfm.nii.gz
    _from-boldref_to-auto000XX_mode-image_xfm.txt
    _space-T1w_desc-aparcaseg_dseg.nii.gz
    _space-T1w_desc-aseg_dseg.nii.gz
    _hemi-[LR]_space-<space_label>_bold.func.gii
    _desc-confounds_timeseries.tsv
    _desc-confounds_timeseries.json
```

https://fmriprep.org/en/stable/outputs.html#confound-regressors-description



NEXT STEPS

- fMRIprep output follows the **BIDS-Derivatives** specification, allowing it to be used as input to additional BIDS Apps for further processing or data analysis
- XCP-D: resting state functional connectivity. Takes an fMRIprep output folder directly as input https://xcp-d.readthedocs.io/en/latest/index.html
- Task-based: continue with 1st-level analysis in AFNI incorporating task timing and nuisance variable regression. https://andysbrainbook.readthedocs.io/en/latest/OpenScience/OS/fMRIPrep_Demo_5_1stLevelAnalysis.html

- "A BIDS App is a container image capturing a neuroimaging pipeline that takes a BIDS-formatted dataset as input. Each BIDS App has the same core set of command-line arguments, making them easy to run and integrate into automated platforms." (Gorgolewski et al. 2017)
- https://bids-apps.neuroimaging.io/about/ BIDS App website
- https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005209 BIDS Apps article

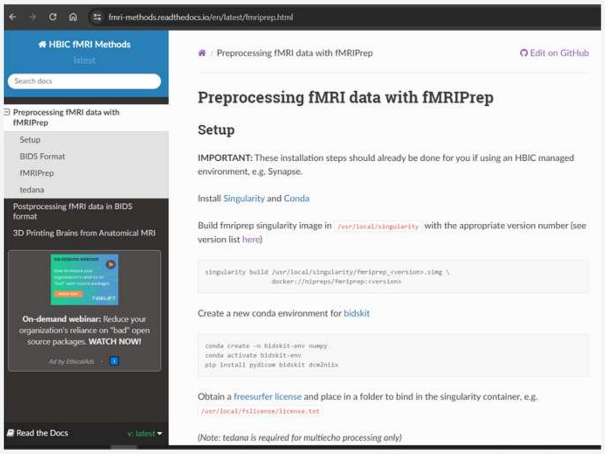
NB: If you use AFNI's **3dConvolve/3dREMLfit** to model your data, set the *-stim_times_subtract* option equal to TR/2 when creating the matrix using 3dDeconvolve to account for slice timing correction set to the middle of the TR by fMRIPrep. https://reproducibility.stanford.edu/slice-timing-correction-in-fmriprep-and-linear-modeling/

XCP-D: https://xcp-d.readthedocs.io/en/latest/index.html
XCP-D picks up right where fMRIprep ends, directly consuming the outputs of fMRIPrep. XCP-D leverages the BIDS and NiPreps frameworks to automatically generate denoised BOLD images, parcellated time series, functional connectivity matrices, and quality assessment reports.
XCP-D is designed for resting-state or pseudo-resting-state functional connectivity analyses. XCP-D derivatives may be useful for seed-to-voxel and ROI-to-ROI functional connectivity analyses, as well as decomposition-based methods, such as ICA or NMF.

For task-based designs, continue with 1st and 2nd level GLMs in your preferred fMRI software package

GETTING STARTED

- Guide: https://fmri-methods.readthedocs.io/en/latest/fmriprep.html
- Install Docker/Singularity and fMRIprep container *not necessary if running on Synapse*
- Convert dataset to BIDS format
- Run fMRIprep container on data


fmri-methods.readthedocs.io/en/latest/fmriprep.html
HBIC fMRI Methods
latest
Search docs
Preprocessing fMRI data with fMRIPrep
Setup
BIDS Format
fMRIPrep
tedana
Postprocessing fMRI data in BIDS format
3D Printing Brains from Anatomical MRI
On-demand webinar: Reduce your organization's reliance on "bad" open source packages. WATCH NOW!
Ad by EthicalAds
Read the Docs
v: latest
Preprocessing fMRI data with fMRIPrep
Edit on GitHub

Preprocessing fMRI data with fMRIPrep

Setup

IMPORTANT: These installation steps should already be done for you if using an HBIC managed environment, e.g. Synapse.

Install Singularity and Conda

Build fmriprep singularity image in `/usr/local/singularity` with the appropriate version number (see version list here)

```
singularity build /usr/local/singularity/fmriprep_<version>.simg \
                  docker://nipreps/fmriprep:<version>
```

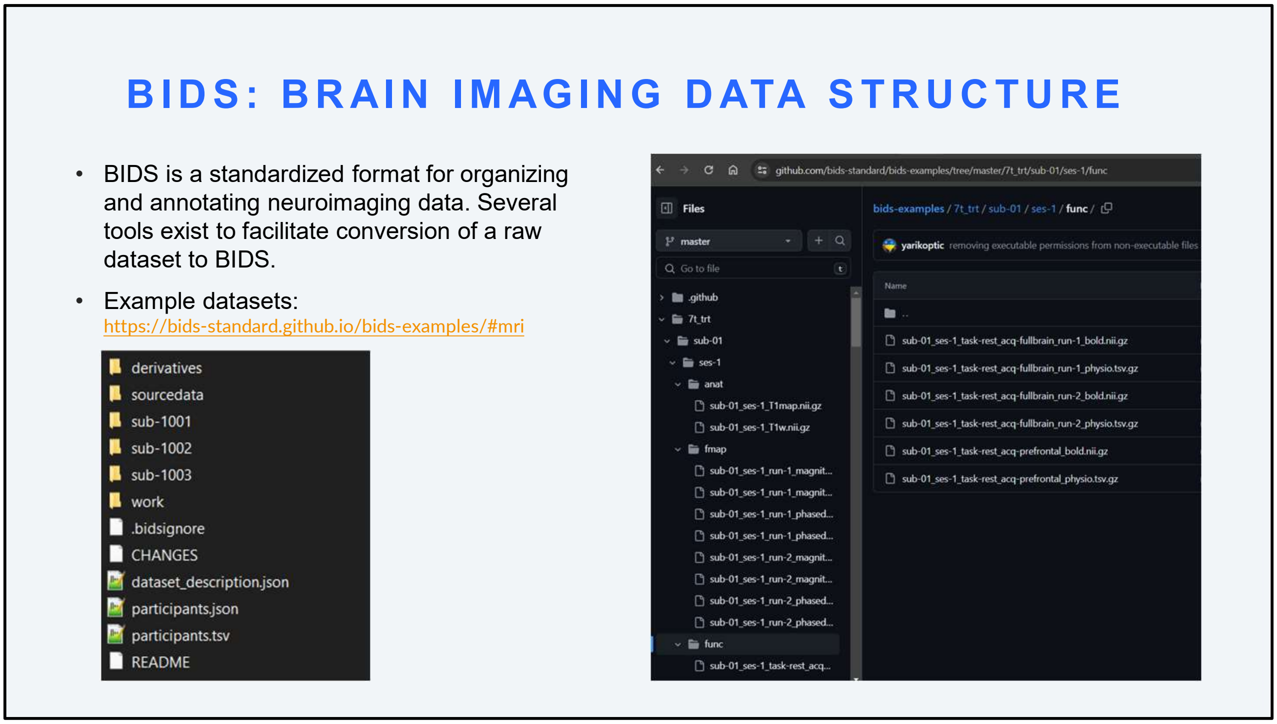
Create a new conda environment for bidskit

```
conda create -n bidskit-env numpy
conda activate bidskit-env
pip install pydicom bidskit dcm2niix
```

Obtain a freesurfer license and place in a folder to bind in the singularity container, e.g. `/usr/local/fslicense/license.txt`

(Note: tedana is required for multiecho processing only)

fMRIprep and bidskit are installed on Synapse and available for use.
Synapse access guide:
https://hbic-computing.readthedocs.io/en/latest/synapse_access.html



BIDS: BRAIN IMAGING DATA STRUCTURE

- BIDS is a standardized format for organizing and annotating neuroimaging data. Several tools exist to facilitate conversion of a raw dataset to BIDS.
- Example datasets: https://bids-standard.github.io/bids-examples/#mri

```
derivatives
sourcedata
sub-1001
sub-1002
sub-1003
work
.bidsignore
CHANGES
dataset_description.json
participants.json
participants.tsv
README
```

```
github.com/bids-standard/bids-examples/tree/master/7t_trt/sub-01/ses-1/func

Files
master
Go to file
> .github
v 7t_trt
  v sub-01
    v ses-1
      v anat
          sub-01_ses-1_T1map.nii.gz
          sub-01_ses-1_T1w.nii.gz
      v fmap
          sub-01_ses-1_run-1_magnit...
          sub-01_ses-1_run-1_magnit...
          sub-01_ses-1_run-1_phased...
          sub-01_ses-1_run-1_phased...
          sub-01_ses-1_run-2_magnit...
          sub-01_ses-1_run-2_magnit...
          sub-01_ses-1_run-2_phased...
          sub-01_ses-1_run-2_phased...
      v func
          sub-01_ses-1_task-rest_acq...

bids-examples / 7t_trt / sub-01 / ses-1 / func /
yarikoptic removing executable permissions from non-executable files
Name
..
sub-01_ses-1_task-rest_acq-fullbrain_run-1_bold.nii.gz
sub-01_ses-1_task-rest_acq-fullbrain_run-1_physio.tsv.gz
sub-01_ses-1_task-rest_acq-fullbrain_run-2_bold.nii.gz
sub-01_ses-1_task-rest_acq-fullbrain_run-2_physio.tsv.gz
sub-01_ses-1_task-rest_acq-prefrontal_bold.nii.gz
sub-01_ses-1_task-rest_acq-prefrontal_physio.tsv.gz
```

BIDS Resources

https://reproducibility.stanford.edu/bids-tutorial-series-part-1a/
https://bids-specification.readthedocs.io/en/stable/introduction.html
https://bids-standard.github.io/bids-starter-kit/tutorials/annotation.html
https://bids-standard.github.io/bids-examples/#mri
https://andysbrainbook.readthedocs.io/en/latest/OpenScience/OS/BIDS_Overview.html
https://bids.neuroimaging.io/

BIDS enforces a standardized folder hierarchy, file naming conventions, and metadata.

A single project folder may contain: a *sourcedata* folder with raw data, *sub-XXXX* folders containing annotated unprocessed data readable by BIDS Apps, a *derivatives* folder containing preprocessed or analyzed data, a *dataset_description.json* file detailing authors, license, funding, doi, etc., and *participants.json* and *participants.tsv* files that list participant characteristics e.g. age, group, condition.

BIDSKIT

- https://github.com/jmtyszka/bidskit/blob/master/docs/QuickStart.md
- Create a folder for your dataset containing the single folder ‘sourcedata,’ containing raw data organized by participant, session, and scan run.

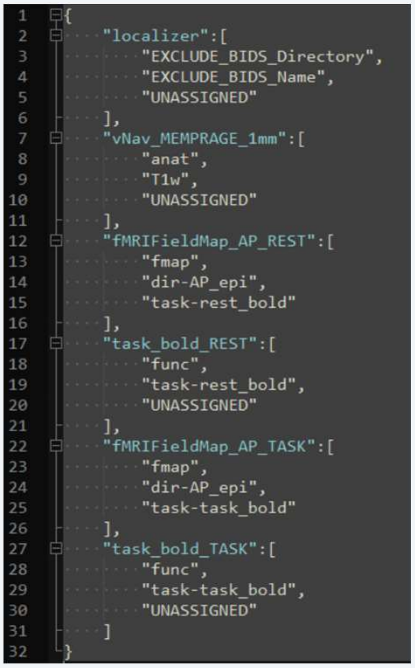
```
myproject/sourcedata/1001/T1/1/[DICOM Images]
                             /2/[DICOM Images]
                         /T2/1/[DICOM Images]
                    /1002/T1/1/[DICOM Images]
                             /2/[DICOM Images]
```

- run bidskit within the project folder. This will scan `sourcedata` and create a translator text file for you to edit before performing the actual BIDS conversion.

```
conda activate bidskit-env
cd /path/myproject
bidskit
```

- After annotating the `protocol_translator.json` file, run bidskit a second time to apply the conversion

```
bidskit
```



```
{
    "localizer":[
        "EXCLUDE_BIDS_Directory",
        "EXCLUDE_BIDS_Name",
        "UNASSIGNED"
    ],
    "vNav_MEMPRAGE_1mm":[
        "anat",
        "T1w",
        "UNASSIGNED"
    ],
    "fMRIFieldMap_AP_REST":[
        "fmap",
        "dir-AP_epi",
        "task-rest_bold"
    ],
    "task_bold_REST":[
        "func",
        "task-rest_bold",
        "UNASSIGNED"
    ],
    "fMRIFieldMap_AP_TASK":[
        "fmap",
        "dir-AP_epi",
        "task-task_bold"
    ],
    "task_bold_TASK":[
        "func",
        "task-task_bold",
        "UNASSIGNED"
    ]
}
```

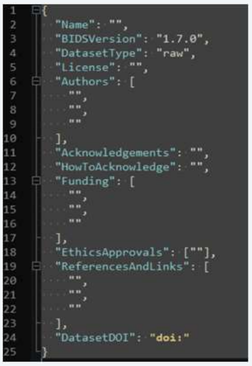
BIDSKIT https://github.com/jmtyszka/bidskit/blob/master/docs/QuickStart.md
Within sourcedata, create a folder for each participant. Within the participant folder, create a folder for each session (i.e., timepoint T1/T2/T3, or modality MRI/EEG). Within the session folder, create a folder for each scanning run that contains the DICOM images. If you do not want to organize by session, use the *–no-sessions* flag at the end of your command (i.e., *bidskit –no-sessions*) in both the first and second bidskit calls.

The first-pass conversion by *bidskit* will create a translator text file that you will edit to aid in creating the BIDS-compliant structure. This step only needs to be done once per project. After activating the bidskit-env conda environment, you can run *bidskit* from the data directory (i.e., the folder that contains the sourcedata subfolder).

Navigate to /path/myproject/code and open Protocol_Translator.json for editing in a text editor. Make use of BIDS documentation to aid in editing this file to assign appropriate BIDS purpose directory names (anat, func, fmap, etc.) and BIDS-compliant filename suffixes. The blue text contains the Series Description names from the DICOMs. These must be unique for each run or BIDSKIT will not be able to distinguish between runs. e.g., if you have two filedmaps that apply to two different runs, label them accordingly.

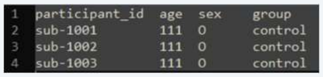
ADDITIONAL CONFIGURATION

dataset_description.json



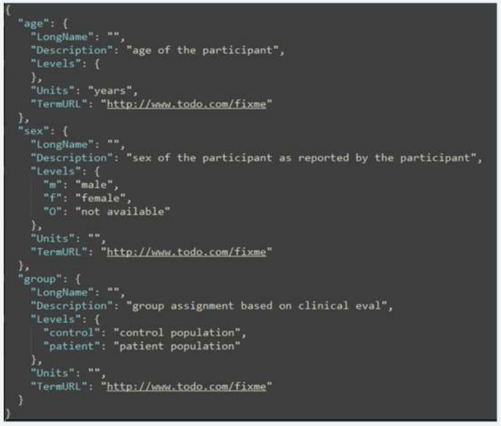
```
{
  "Name": "",
  "BIDSVersion": "1.7.0",
  "DatasetType": "raw",
  "License": "",
  "Authors": [
    "",
    "",
    ""
  ],
  "Acknowledgements": "",
  "HowToAcknowledge": "",
  "Funding": [
    "",
    "",
    ""
  ],
  "EthicsApprovals": [""],
  "ReferencesAndLinks": [
    "",
    "",
    ""
  ],
  "DatasetDOI": "doi:"
}
```

participants.tsv



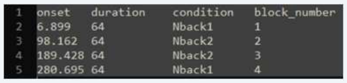
```
participant_id  age  sex   group
sub-1001        111  0     control
sub-1002        111  0     control
sub-1003        111  0     control
```

participants.json



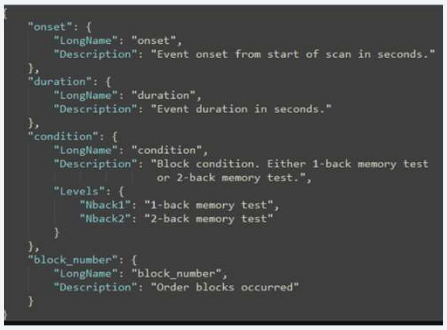
```
{
  "age": {
    "LongName": "",
    "Description": "age of the participant",
    "Levels": {
    },
    "Units": "years",
    "TermURL": "http://www.todo.com/fixme"
  },
  "sex": {
    "LongName": "",
    "Description": "sex of the participant as reported by the participant",
    "Levels": {
      "m": "male",
      "f": "female",
      "O": "not available"
    },
    "Units": "",
    "TermURL": "http://www.todo.com/fixme"
  },
  "group": {
    "LongName": "",
    "Description": "group assignment based on clinical eval",
    "Levels": {
      "control": "control population",
      "patient": "patient population"
    },
    "Units": "",
    "TermURL": "http://www.todo.com/fixme"
  }
}
```

sub-XXXX_task-TASK_events.tsv



```
onset   duration    condition   block_number
6.899   64          Nback1      1
98.162  64          Nback2      2
189.428 64          Nback2      3
280.695 64          Nback1      4
```

sub-XXXX_task-TASK_events.json

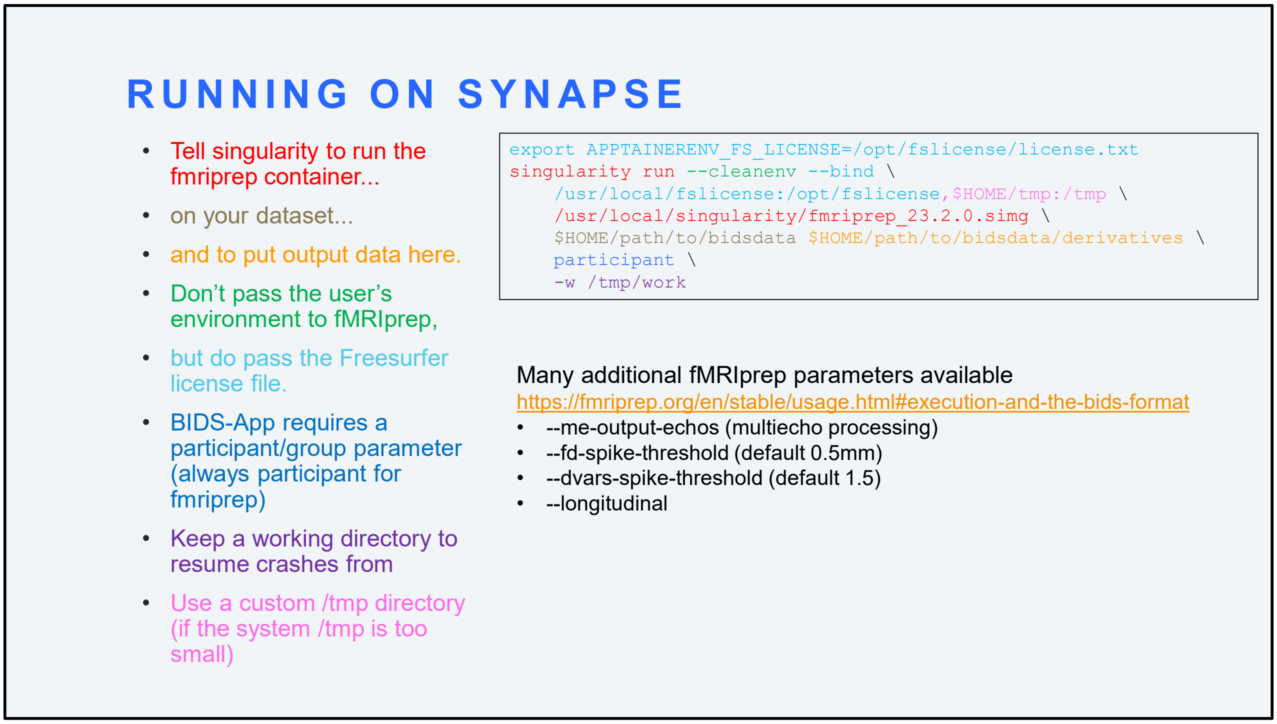


```
{
    "onset": {
        "LongName": "onset",
        "Description": "Event onset from start of scan in seconds."
    },
    "duration": {
        "LongName": "duration",
        "Description": "Event duration in seconds."
    },
    "condition": {
        "LongName": "condition",
        "Description": "Block condition. Either 1-back memory test
                        or 2-back memory test.",
        "Levels": {
            "Nback1": "1-back memory test",
            "Nback2": "2-back memory test"
        }
    },
    "block_number": {
        "LongName": "block_number",
        "Description": "Order blocks occurred"
    }
}
```

dataset_description.json and the participants.json *.tsv files do not need to be completed before running fMRIprep, but are useful means of recording dataset characteristics and necessary before sharing data.

If you have task-based designs, you will need to complete any `/path/myproject/sub-XXXX/ses-X/func/*_events.tsv` template files with task timing information.

The .json sidecar files are intended to be human and machine readable annotations of codes and variable names used in the accompanying .tsv files.



RUNNING ON SYNAPSE

- Tell singularity to run the fmriprep container...
- on your dataset...
- and to put output data here.
- Don’t pass the user’s environment to fMRIprep,
- but do pass the Freesurfer license file.
- BIDS-App requires a participant/group parameter (always participant for fmriprep)
- Keep a working directory to resume crashes from
- Use a custom /tmp directory (if the system /tmp is too small)

```
export APPTAINERENV_FS_LICENSE=/opt/fslicense/license.txt
singularity run --cleanenv --bind \
    /usr/local/fslicense:/opt/fslicense,$HOME/tmp:/tmp \
    /usr/local/singularity/fmriprep_23.2.0.simg \
    $HOME/path/to/bidsdata $HOME/path/to/bidsdata/derivatives \
    participant \
    -w /tmp/work
```

Many additional fMRIprep parameters available
https://fmriprep.org/en/stable/usage.html#execution-and-the-bids-format

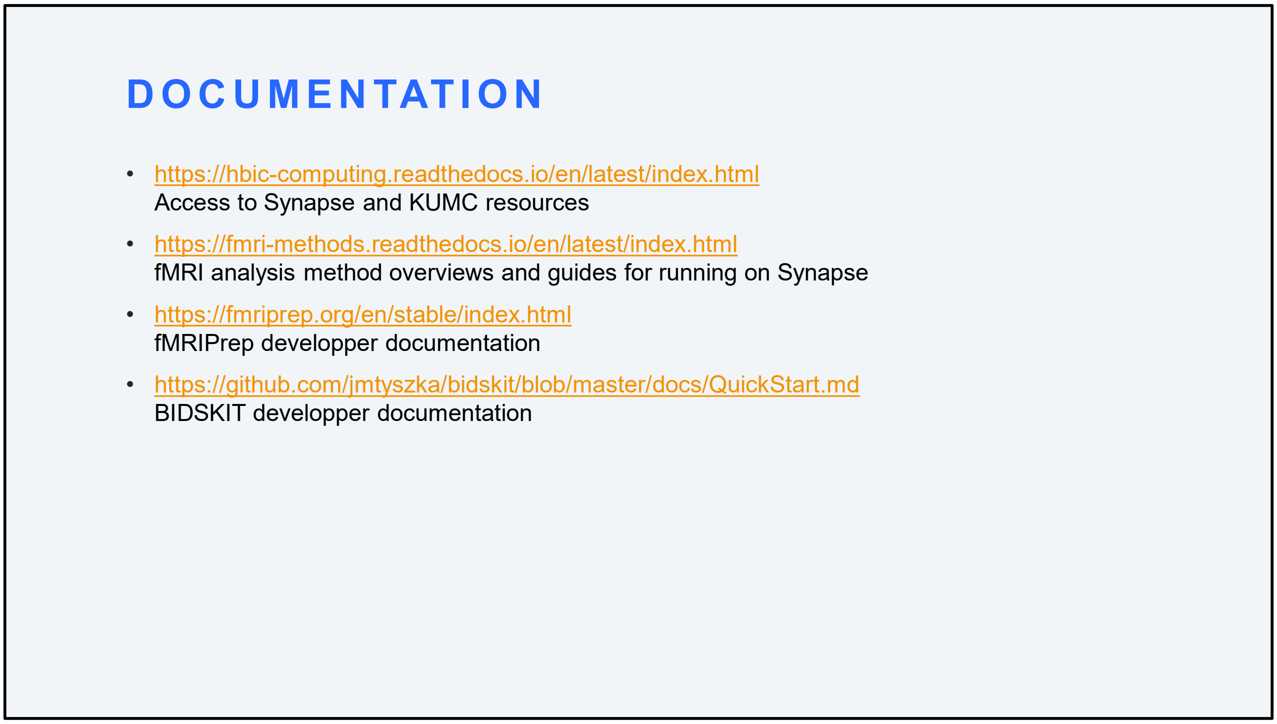
- --me-output-echos (multiecho processing)
- --fd-spike-threshold (default 0.5mm)
- --dvars-spike-threshold (default 1.5)
- --longitudinal

Use the --cleanenv flag to avoid passing environment variables, like the user’s FSL installation instead of the container’s FSL.

fMRIprep does not come with a FreeSurfer license – pass it to the container by creating an APPTAINERENV_FS_LICENSE variable. The option --bind /usr/local/fslicense:/opt/fslicense will map the contents of /usr/local/fslicense on the host (where the file actually resides) to the path /opt/fslicense within the container (where fMRIPrep will look for it, that is, the path specified by FS_LICENSE).

fMRIPrep makes extensive use of a temporary directory, /tmp for intermediate files. But /tmp may not have sufficient disk space allotted, causing the processing job to halt and error before completing. To remedy this, map a different location on the host system that is unrestricted in size to /tmp, e.g. $HOME/tmp:/tmp. ATTENTION: Clear this custom tmp directory regularly when you no longer need the intermediate working files!

If fMRIPrep aborts partway through, you can rerun it from where it stopped by using the -w option and passing the work directory from the failed attempt.



DOCUMENTATION

- https://hbic-computing.readthedocs.io/en/latest/index.html
  Access to Synapse and KUMC resources
- https://fmri-methods.readthedocs.io/en/latest/index.html
  fMRI analysis method overviews and guides for running on Synapse
- https://fmriprep.org/en/stable/index.html
  fMRIPrep developper documentation
- https://github.com/jmtyszka/bidskit/blob/master/docs/QuickStart.md
  BIDSKIT developper documentation