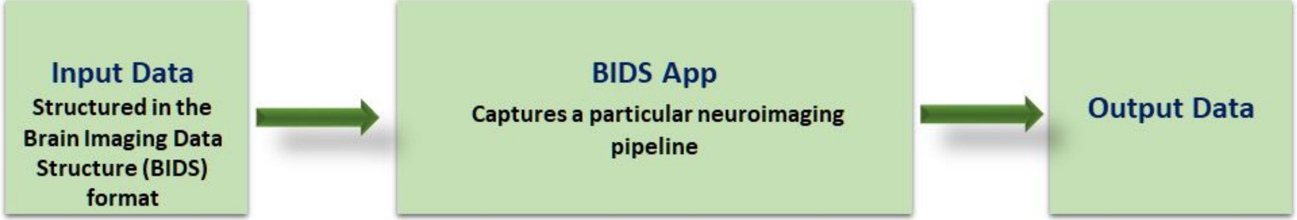
dcm2bids + fmriprep Tutorial

Cognition Lab Meeting - 11 July 2020

A look at arranging data in the BIDS format, and standardizing and automating fMRI Preprocessing

What are Brain Imaging Data Structure (BIDS) apps?

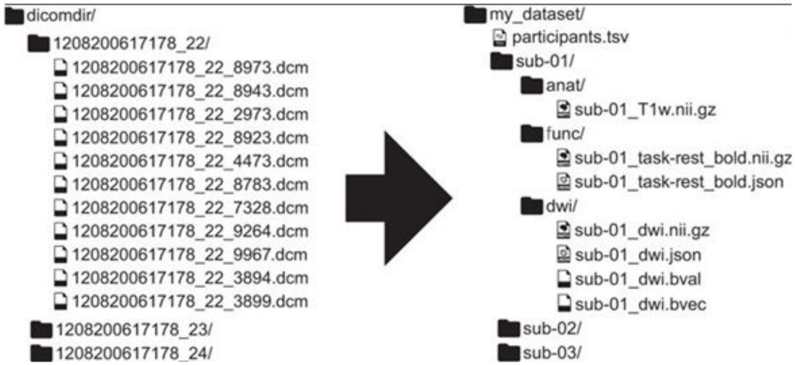
A BIDS App is an application which implements a neuroimaging pipeline, by taking a BIDS formatted dataset as input



Input Data
Structured in the Brain Imaging Data Structure (BIDS) format

BIDS App
Captures a particular neuroimaging pipeline

Output Data



```
dicomdir/
    1208200617178_22/
        1208200617178_22_8973.dcm
        1208200617178_22_8943.dcm
        1208200617178_22_2973.dcm
        1208200617178_22_8923.dcm
        1208200617178_22_4473.dcm
        1208200617178_22_8783.dcm
        1208200617178_22_7328.dcm
        1208200617178_22_9264.dcm
        1208200617178_22_9967.dcm
        1208200617178_22_3894.dcm
        1208200617178_22_3899.dcm
    1208200617178_23/
    1208200617178_24/
```

```
my_dataset/
    participants.tsv
    sub-01/
        anat/
            sub-01_T1w.nii.gz
        func/
            sub-01_task-rest_bold.nii.gz
            sub-01_task-rest_bold.json
        dwi/
            sub-01_dwi.nii.gz
            sub-01_dwi.json
            sub-01_dwi.bval
            sub-01_dwi.bvec
    sub-02/
    sub-03/
```

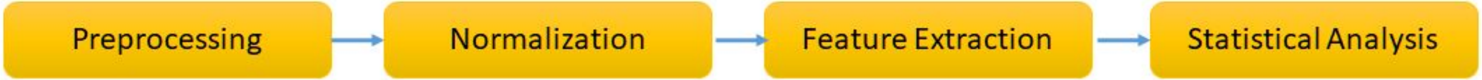
Example: Structuring a dataset into the BIDS format

- BIDS apps can be written in one **of several languages** like python, MATLAB, C++ etc. or a combination of languages.
- They come **packaged with all additional dependencies**, so the user need not download all the software required to run the pipeline separately.
- They are written in the form of **container images** that can be run on **Docker**.

BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods *Krzysztof J. Gorgolewsk et al. Plos Computational Biology 2017*

Why do we need BIDS apps?

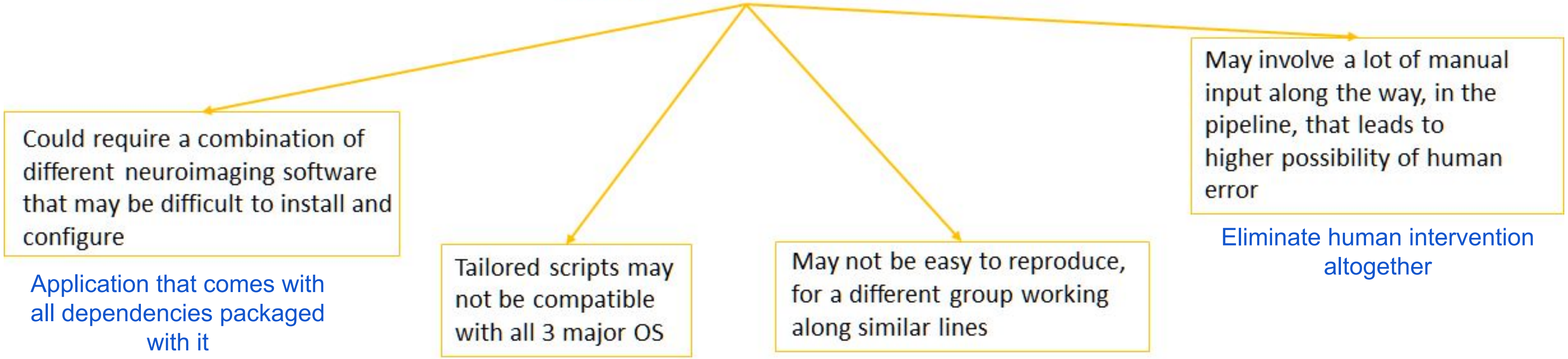
Outline of a Neuroimaging Pipeline



Preprocessing → Normalization → Feature Extraction → Statistical Analysis

- Different research groups generally come up with their own pipelines which incorporate these steps.
- These pipelines are generally tailored to the dataset they use, and are designed to meet some very specific requirements.

Issues with such an approach: (Solutions in blue)



Could require a combination of different neuroimaging software that may be difficult to install and configure

Application that comes with all dependencies packaged with it

Tailored scripts may not be compatible with all 3 major OS

Execution isn't dependent on host OS

May not be easy to reproduce, for a different group working along similar lines

Standardized

May involve a lot of manual input along the way, in the pipeline, that leads to higher possibility of human error

Eliminate human intervention altogether

Containers and the Docker Engine:

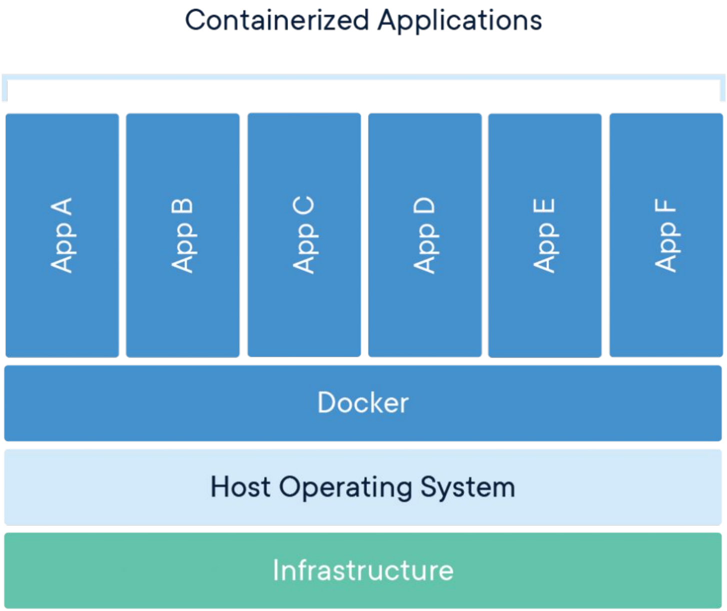
Solution: The need to separately install required external dependencies, incompatibility with different OS and low reproducibility can be solved by using **containers**.

BIDS apps are generally packaged as containers.

So what are containers?

Defn: ‘A **container** is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another.

Defn: ‘A **Docker container image** is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.’

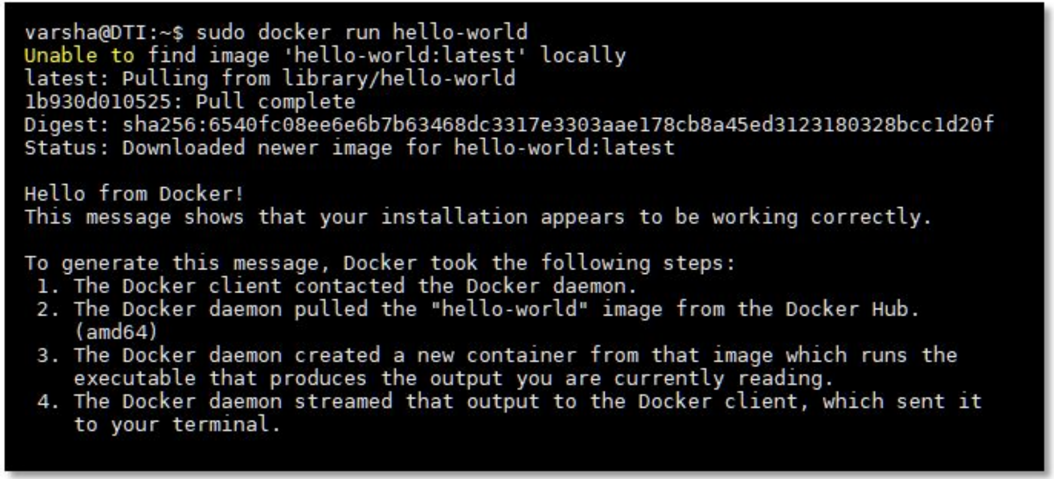


Container images turn into containers when they are run on the docker engine. Docker creates an abstraction over the OS layer, which allows these containers to run without having to adapt to a particular computing system.

Docker Installation:

Docker installation links :

https://docs.docker.com/install/linux/docker-ce/ubuntu/ - Ubuntu Installation
https://docs.docker.com/v17.09/docker-for-windows/install/#about-windows-containers-and-windows-server-2016 – Windows Installation

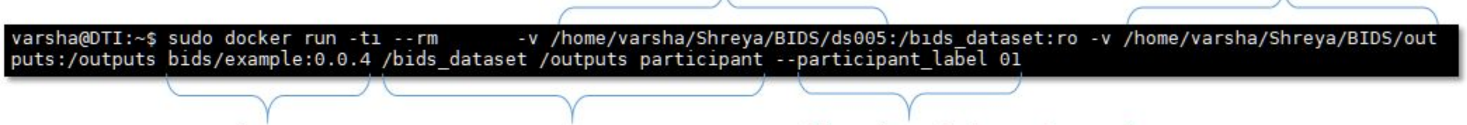


```
varsha@DTI:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
1b930d010525: Pull complete
Digest: sha256:6540fc08ee6e6b7b63468dc3317e3303aae178cb8a45ed3123180328bcc1d20f
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

- Install Docker, by following the steps in the provided links (fairly straightforward for both Windows and Ubuntu)
- Pull a hello-world image form the Docker Repository, and run it.
- The attached message should appear if the installation was successful.

Ref: http://reproducibility.stanford.edu/fmriprep-tutorial-running-the-docker-image/

Intro: Running a sample BIDS app



Input Data Path

Output Data Path

```
varsha@DTI:~$ sudo docker run -ti --rm      -v /home/varsha/Shreya/BIDS/ds005:/bids_dataset:ro -v /home/varsha/Shreya/BIDS/out
puts:/outputs bids/example:0.0.4 /bids_dataset /outputs participant --participant_label 01
```

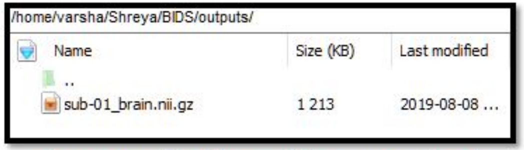
Sample BIDS app

Input, Output to BIDS app

Subjects from the input dataset for which BIDS app is to be run

For running this sample BIDS app, **sample BIDS structured** data (ds005) was downloaded from :

https://drive.google.com/drive/folders/0B2JWN60ZLkgkMGlUY3B4MXZlZW8



/home/varsha/Shreya/BIDS/outputs/

Name	Size (KB)	Last modified
..		
sub-01_brain.nii.gz	1 213	2019-08-08 ...

Corresponding Output file generated and stored in specified directory

Flags Being Used in this Command :

- **-ti :** Allows all further commands like mkdir (make directory), ls (list contents of a directory) to be used within the container
- **--rm (Clean Up) :** Removes the container, and all temporary files that were generated for running it after use
- **-v:** Used to specify an external volume to allow data saving, and data sharing between containers.

Docker uses the term **volumes** to refer to file systems that exist outside the container

https://bids-apps.neuroimaging.io/tutorial/ - refer for more details

Structuring Data into the BIDS format - dcm2bids

- **Dcm2bids structures the input data** (obtained from the scanner or downloaded from an external database) into a BIDS format.
- The tool **first converts data from the DICOM format to the NIfTI (.nii) format**, which is the format BIDS uses.
- The tool also **extracts header information from DICOM files**, and creates an **informative .json file with scan parameters**,corresponding to each available scan.
- This metadata corresponding to each scan is what BIDS apps use, to **eliminate human intervention**, as all information is available alongside the scan.

To those collecting fMRI data in the lab:

Opt for saving your raw data in the in the **DICOM format,** as it allows us to retain all neccessary parameters for fmriprep, post conversion to BIDS using **dcm2bids**.

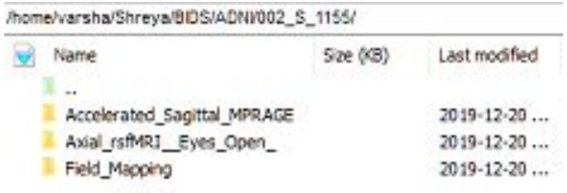
We have developed a **NIfTI to BIDS script** as well, but some parameters are sometimes lost in the DICOM to NIFTI conversion (e.g slice timing information, time of echo etc.)

Points to note for the slides that follow:

- All Steps are demonstrated on the Parker server, as this server has both dcm2bids and fmriprep installed.
- The steps will remain identical for use on a different server, as long as both softwares have been appropriately installed.
- All **command words are in blue font,** and the input arguments required are in italics.
- Some commands do not require any input arguments.
- **Comments are in green, preceded by the % sign,** and should not be written as a part of the command.
- An ADNI database subject has been used in the example that follows, but the same steps hold for any raw DICOM data.

dcm2bids Implementation Steps(without fieldmaps):

- **Transfer the DICOM folder, with all its subfolders intact, to the Parker server.** The DICOM folder will usually have 3 different subfolders, one each for structural, functional and fieldmap data as seen. In other cases, such a division might not be seen.



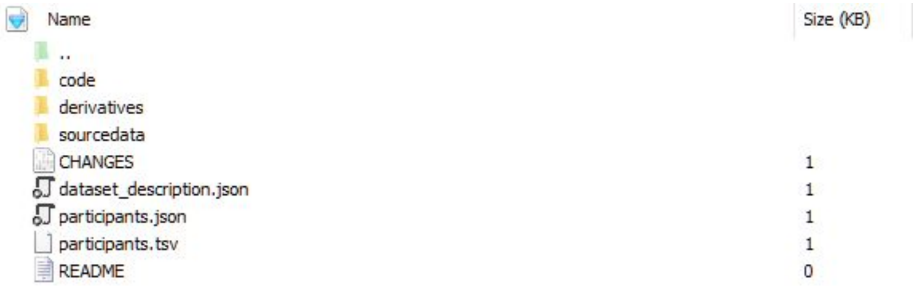
```
/home/varsha/Shreya/BIDS/ADNI/002_S_1155/
Name                              Size (KB)    Last modified
..
Accelerated_Sagittal_MPRAGE                    2019-12-20 ...
Axial_rsfMRI__Eyes_Open_                       2019-12-20 ...
Field_Mapping                                  2019-12-20 ...
```

- **Create a new folder to hold our data structured into the BIDS syntax, hereafter referred to as the BIDS folder.**
- **CD to this BIDS folder and run the dcm2bids_scaffold command.** This generates an empty file and folder structure to fill in the data in the BIDS format.

```
Commands:
cd /path_to_BIDS_folder/ (press enter)
dcm2bids_scaffold (press enter)

Example Subject
 cd /home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/
 %the BIDS folder name is in yellow
 dcm2bids_scaffold
```

dcm2bids implementation - Conversion to NIfTI

Output:
An empty file structure will be seen within the BIDS folder as shown below



Name	Size (KB)
..	
code	
derivatives	
sourcedata	
CHANGES	1
dataset_description.json	1
participants.json	1
participants.tsv	1
README	0

- **Next run the dcm2bids_helper command.** This command converts the scans from **DICOM to NIfTI format,** and preserves all DICOM header information corresponding to each scan (manufacturer, slice timing, slice thickness, phase encoding direction etc.) in a **corresponding .json file.**
  - There is **one .json file** corresponding to each scan (.nii file).
  - The .json file can be opened up in any **text editor,** to read the contents as necessary.

Command:
dcm2bids_helper -d */path_to_DICOM_folder/* (press enter)

Note: The Path has to the outermost DICOM folder (which further contains the 3 func., struc., and fieldmap DICOM folders) has to be provided in the above command.

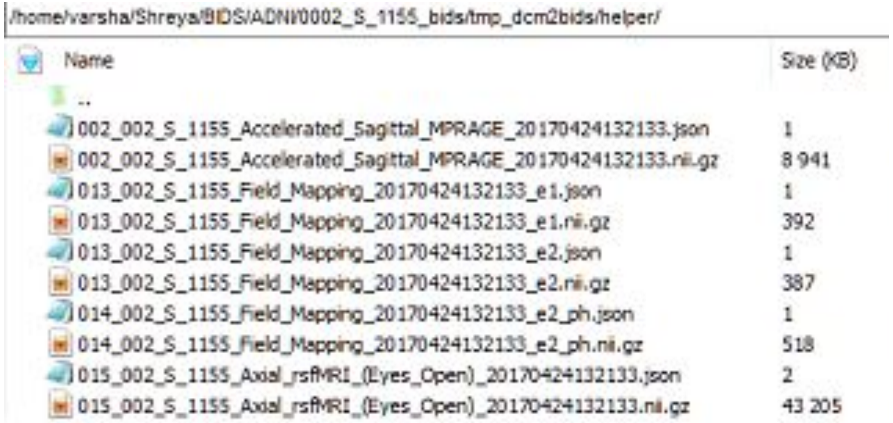
Example Subject:
dcm2bids_helper -d */home/varsha/Shreya/BIDS/ADNI/002_S_1155/*
%Outermost DICOM folder name highlighted above

dcm2bids implementation contd.

Output Message:
Example in:
/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/tmp_dcm2bids/helper

- This message indicates that the NIfTI converted files and corresponding .jsons are present in the folder labeled tmp_dcm2bids/helper, generated within the BIDS folder.

- The contents of the folder look as follows. Note that there is one NIfTI file for each of the structural and functional scans, and other fieldmap files (2 magnitude images with names ending in e1 and e2, and a phase difference image with name ending in e2_ph)



/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/tmp_dcm2bids/helper/

Name	Size (KB)
..	
002_002_S_1155_Accelerated_Sagittal_MPRAGE_20170424132133.json	1
002_002_S_1155_Accelerated_Sagittal_MPRAGE_20170424132133.nii.gz	8 941
013_002_S_1155_Field_Mapping_20170424132133_e1.json	1
013_002_S_1155_Field_Mapping_20170424132133_e1.nii.gz	392
013_002_S_1155_Field_Mapping_20170424132133_e2.json	1
013_002_S_1155_Field_Mapping_20170424132133_e2.nii.gz	387
014_002_S_1155_Field_Mapping_20170424132133_e2_ph.json	1
014_002_S_1155_Field_Mapping_20170424132133_e2_ph.nii.gz	518
015_002_S_1155_Axial_rsfMRI_(Eyes_Open)_20170424132133.json	2
015_002_S_1155_Axial_rsfMRI_(Eyes_Open)_20170424132133.nii.gz	43 205

dcm2bids implementation - Configuration File

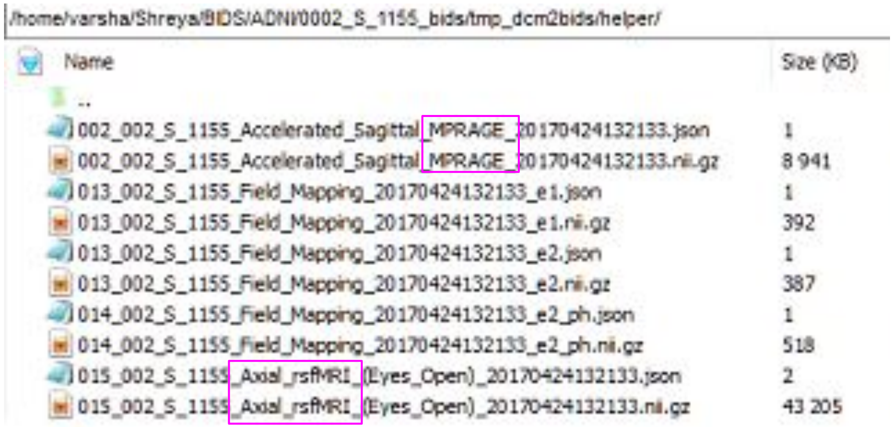
We now have access to NIfTI files. However, they are yet to be structured into the BIDS format. The first step to commence this procedure is to generate something called a configuration file.

- This file can be generated using **any text editor.**
- This file indicates to the program **which modalities you intend to retain**.
- It also provides the program with unique keywords associated **only with the names of scans of each particular modality (NOT shared with any other)** such that it can easily pick out the scans you wish to retain.
- To identify these **modality-specific keywords**, look at the **names of the scans generated by the above dcm2bids_helper** command.
- Once the config file is ready, save it by the name **config.json**, and **store it in the code subfolder** within the BIDS folder.

dcm2bids implementation - Config. File Example

%*The Modality-specific keywords identified from the names of files in the helper sub-folder have been highlighted. The *'s on either side indicate that the keyword occurs in the middle of the name.*

```
{
   "descriptions": [
      {
         "dataType": "func",
         "modalityLabel": "bold",
         "customLabels": "task-rest",
         "criteria": {
              "SidecarFilename":"*Axial_rsfMRI*"
                  }
      },
      {
         "dataType": "anat",
         "modalityLabel": "T1w",
         "criteria": {
              "SidecarFilename": "*MPRAGE*"
                  }
      },
     ]
}
```



/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/tmp_dcm2bids/helper/

Name	Size (KB)
..	
002_002_S_1155_Accelerated_Sagittal_MPRAGE_20170424132133.json	1
002_002_S_1155_Accelerated_Sagittal_MPRAGE_20170424132133.nii.gz	8 941
013_002_S_1155_Field_Mapping_20170424132133_e1.json	1
013_002_S_1155_Field_Mapping_20170424132133_e1.nii.gz	392
013_002_S_1155_Field_Mapping_20170424132133_e2.json	1
013_002_S_1155_Field_Mapping_20170424132133_e2.nii.gz	387
014_002_S_1155_Field_Mapping_20170424132133_e2_ph.json	1
014_002_S_1155_Field_Mapping_20170424132133_e2_ph.nii.gz	518
015_002_S_1155_Axial_rsfMRI_(Eyes_Open)_20170424132133.json	2
015_002_S_1155_Axial_rsfMRI_(Eyes_Open)_20170424132133.nii.gz	43 205

Dcm2bids_helper output

dcm2bids implementation - Final Step

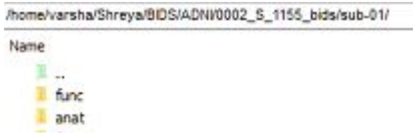
- **Now that the configuration file is ready, we will run the final command to arrange the NIfTI files into the BIDS syntax.** As the tmp_dcm2bids/helper folder was intended only to use as a reference for creating the configuration file, **DELETE that folder** before running the following command.

```
Command:
dcm2bids -d /path_to_DICOM_folder/ -p participant label -c
/path_to_configuration file/
% Participant label is the numerical ID you would like to assign to the
participant
%Path to config file should end with config.json

Example Subject:
dcm2bids -d /home/varsha/Shreya/BIDS/ADNI/002_S_1155/ -p 01 -c
/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/code/config.json
```

dcm2bids Output:

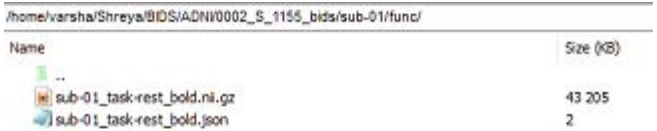
Output: Depending on the participant label you have provided, a subfolder titled sub - participant label will be created within the BIDS folder. In our example subject, the number we had used was one, consequently, the subfolder is labeled **sub - 01**.

This sub-folder will contain 2 folders, one each for func. and anat. (struc.).



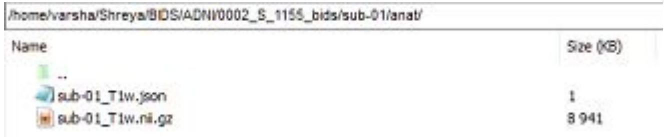
```
/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/sub-01/
Name
  ..
  func
  anat
```

The contents of the func folder are as follows (.nii fMRI scan + descriptive .json):



```
/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/sub-01/func/
Name                                   Size (KB)
  ..
  sub-01_task-rest_bold.nii.gz         43 205
  sub-01_task-rest_bold.json           2
```

The contents of the anat folder are as follows (.nii T1 scan + descriptive .json):



```
/home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/sub-01/anat/
Name                                   Size (KB)
  ..
  sub-01_T1w.json                      1
  sub-01_T1w.nii.gz                    8 941
```

Checking if your Data has been correctly structured into the BIDS format:

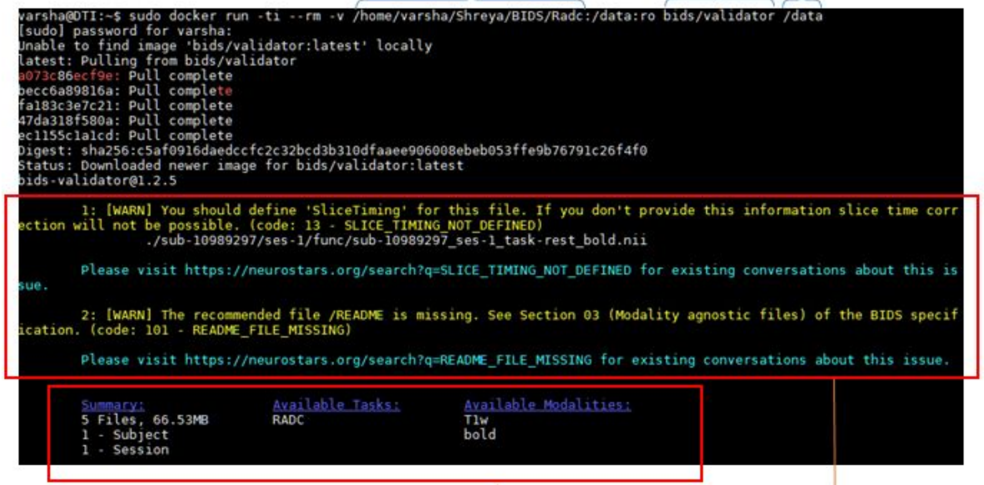
Two parallel methods

Running the validator from the Command Line

BIDS structured folder Path

Validator BIDS app

Input

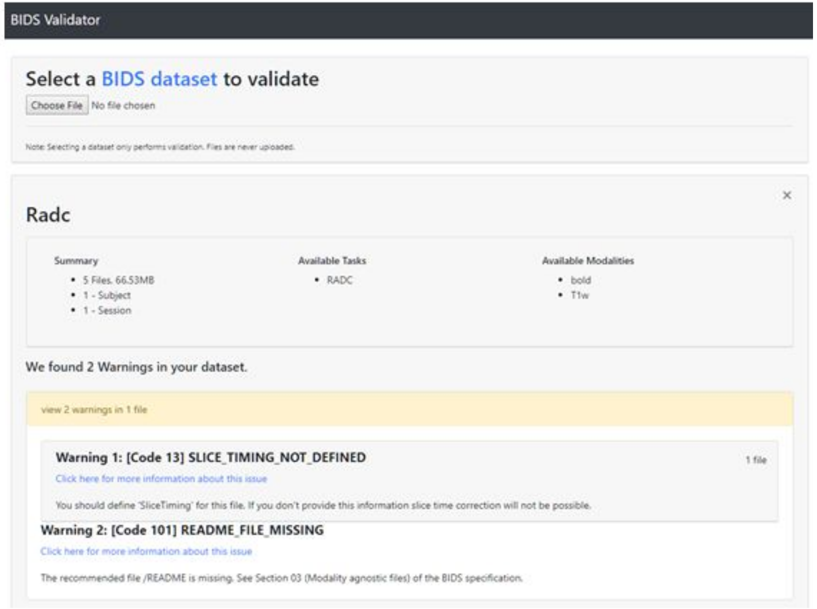


```
varsha@DTI:~$ sudo docker run -ti --rm -v /home/varsha/Shreya/BIDS/Radc:/data:ro bids/validator /data
[sudo] password for varsha:
Unable to find image 'bids/validator:latest' locally
latest: Pulling from bids/validator
a073c86ecf9e: Pull complete
becc6a89816a: Pull complete
fa183c3e7c21: Pull complete
47da318f580a: Pull complete
ec1155c1a1cd: Pull complete
Digest: sha256:c5af0916daedccfc2c32bcd3b310dfaaee906008ebeb053ffe9b76791c26f4f0
Status: Downloaded newer image for bids/validator:latest
bids-validator@1.2.5

        1: [WARN] You should define 'SliceTiming' for this file. If you don't provide this information slice time corr
ection will not be possible. (code: 13 - SLICE_TIMING_NOT_DEFINED)
                ./sub-10989297/ses-1/func/sub-10989297_ses-1_task-rest_bold.nii

        Please visit https://neurostars.org/search?q=SLICE_TIMING_NOT_DEFINED for existing conversations about this is
sue.

        2: [WARN] The recommended file /README is missing. See Section 03 (Modality agnostic files) of the BIDS specif
ication. (code: 101 - README_FILE_MISSING)

        Please visit https://neurostars.org/search?q=README_FILE_MISSING for existing conversations about this issue.

        Summary:                Available Tasks:        Available Modalities:
        5 Files, 66.53MB        RADC                    T1w
        1 - Subject                                     bold
        1 - Session
```

Summary of BIDS folder contents

Warnings about the structure
(Notifications about aspects to the structure that are not mandatory but preferable)

Using the online validator



BIDS Validator

Select a BIDS dataset to validate

Note: Selecting a dataset only performs validation. Files are never uploaded.

Radc

Summary
- 5 Files, 66.53MB
- 1 - Subject
- 1 - Session

Available Tasks
- RADC

Available Modalities
- bold
- T1w

We found 2 Warnings in your dataset.

view 2 warnings in 1 file

Warning 1: [Code 13] SLICE_TIMING_NOT_DEFINED 1 file

Click here for more information about this issue

You should define 'SliceTiming' for this file. If you don't provide this information slice time correction will not be possible.

Warning 2: [Code 101] README_FILE_MISSING

Click here for more information about this issue

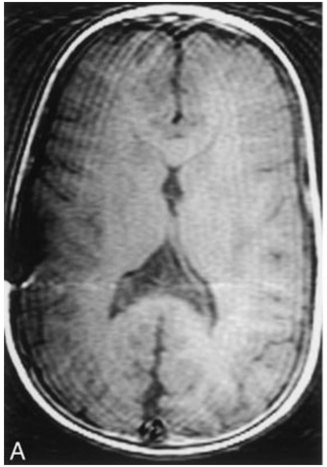
The recommended file /README is missing. See Section 03 (Modality agnostic files) of the BIDS specification.

http://bids-standard.github.io/bids-validator/ - Online Validator link

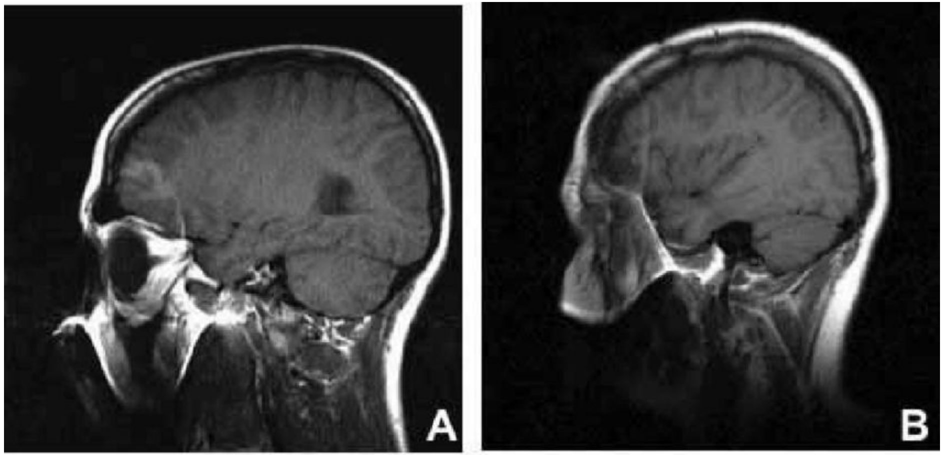
Visual Inspection of Unprocessed Data:

It’s a good idea to look at the data beforehand for severe defects as seen below, so that such scans can be eliminated without preprocessing.

T1 observable defect examples:



Extreme head motion causing a ringing effect



Susceptibility distortion caused due to dental implants - https://www.researchgate.net/figure/A-Sagittal-T1-weighted-MR-image-clearly-showing-distortion-around-jaw-from-gold-crown_fig1_24214125

Func. image possible defects:

- Parts of brain cut off
- Flipped scan (can be corrected)
- Whole brain is lit up

MRI-QC, a tool developed by Poldrack lab computes certain image quality measures to avoid the need for this inspection. In its current state however, the tool can take upto 1hr per scan

fmriprep BIDS app

A StandardizedPreprocessing Pipeline for fMRI Data

Developed by

the Poldrack Lab, Stanford University

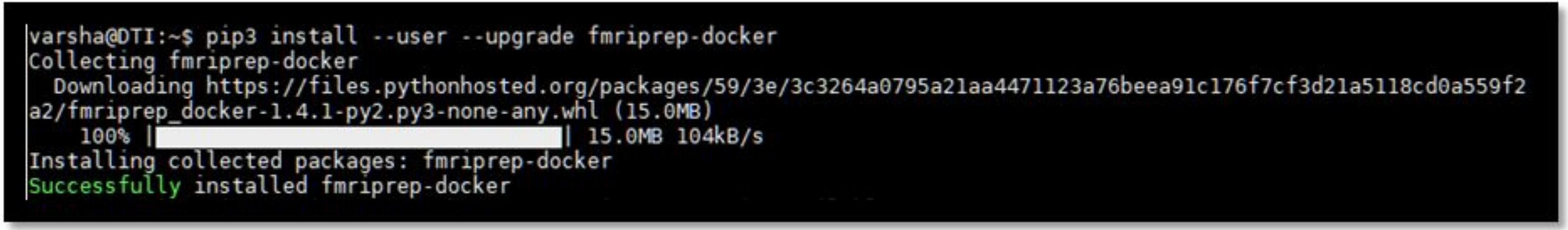
For use at the Center for Reproducible Neuroscience, and for Open Source Software Distribution

Running fmriprep on Docker:

Although fMRIprep can be run directly on Docker like the sample BIDS app in the previous example, the creators recommend using a **python wrapper function** called **fmriprep-docker** to make commands easier to write.

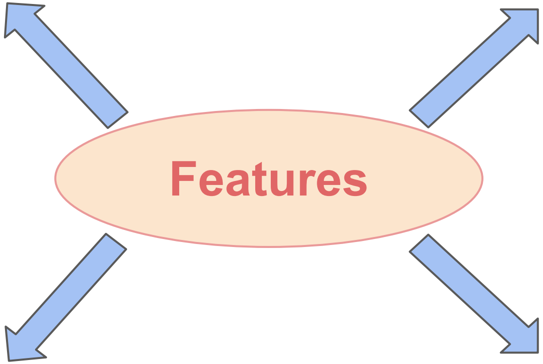
Prerequisites – Python and pip packages should be installed

Installation Command: pip install --user --upgrade fmriprep-docker



```
varsha@DTI:~$ pip3 install --user --upgrade fmriprep-docker
Collecting fmriprep-docker
  Downloading https://files.pythonhosted.org/packages/59/3e/3c3264a0795a21aa4471123a76beea91c176f7cf3d21a5118cd0a559f2
a2/fmriprep_docker-1.4.1-py2.py3-none-any.whl (15.0MB)
    100% |████████████████████████████████| 15.0MB 104kB/s
Installing collected packages: fmriprep-docker
Successfully installed fmriprep-docker
```

Fmriprep:



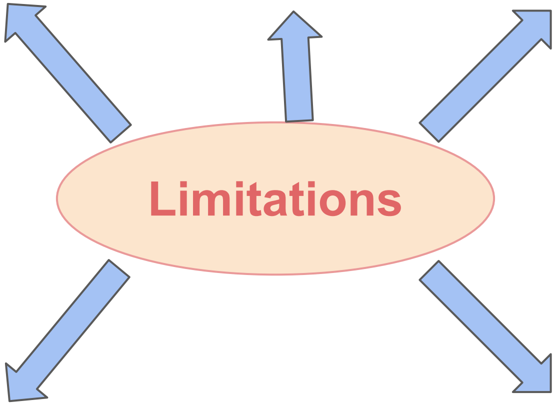
Leverages the BIDS syntax to extract required parameters from scan metadata, to **avoid manual intervention.**

Accounts for missing features like slice timing information or fieldmaps. Will skip step/ provide alternatives.

Extremely generalized and consequently **agnostic to future analysis.**
E.g: Does not include spatial and temporal filtering, Denoising etc. which might be specifically important for functional connectivity analysis

Consists of subworkflows which are assembled into an appropriate configuration, easy to detect which step causes the error.

Fmriprep limitations:



Limitations

Data has to be structured into the BIDS syntax, which might be difficult when DICOM files aren’t available.

Cannot be used for narrow field of view scans - e.g scans with only the visual cortex etc. can’t be processed by fmriprep

Output might need further tailoring for specific analysis - Tools like fMRIdenoise can be used to eliminate noise prior to functional connectivity analysis

Cannot be used for non-human species - rodent/primate data

No inbuilt Quality check for raw data - could use MRI-QC before to measure image quality measures like Contrast to noise ratio (b/w GM and WM), SNR within each tissue type, framewise displacement etc.

Running the fmriprep command:

Command:

```
sudo fmriprep-docker /path to BIDS arranged subject folder/ /path to output folder/
--participant_label label_name --fs-license-file /path to freesurfer license/ --fs-no-reconall
--ignore fieldmaps (%optional --ignore slicetiming --output-spaces MNI152NLin6Asym:res-2
)
```

- Fmriprep-docker command access requires adminstrative control, hence it must be run with the sudo command.
- Provide the **full path** to the BIDS folder you created in the preceding steps.
- Create an output folder to store the fmriprep output files, and then provide the full path to it.
- Provide the same participant label you had provided while using dcm2bids.
- It is mandatory to provide a path to a Freesurfer license even if we aren't using the Freesurfer option. In the parker server, this license is stored in the following path - /home/varsha/Shreya/BIDS/freesurfer.txt. You can also generate a new one at https://surfer.nmr.mgh.harvard.edu/registration.html, and transfer it to the server in use, and provide the full path to its location.

Running the **fmriprep** command (contd.):

Command:

```
sudo fmriprep-docker /path to BIDS arranged subject folder/ /path to output folder/
--participant_label label_name --fs-license-file /path to freesurfer license/ --fs-no-reconall
--ignore fielmaps (%optional --ignore slicetiming --output-spaces MNI152NLin6Asym:res-2 )
```

- --fs-no-reconall indicates we do not want to use the Freesurfer option. Freesurfer is used to reconstruct the brain surface so that the BOLD signal from the surface of the brain can be read correctly. This step severely extends the processing time (~8hrs/subject), and its effects do not have a great impact on our research.
- If you have fieldmaps, but choose to avoid fieldmap correction, indicate it by using the --ignore fielmaps flag.
- If you do not have access to slice timing information in the .json file of the fMRI scan, add the --ignore slicetiming flag
- IMP: If you want the preprocessed data to be generated in a format that is conducive to all the parcellation scripts in the lab - MNI 2mm - add the flag --output-spaces MNI152NLin6Asym:res-2

Fmriprep commands with & without fieldmaps:

Without Fieldmap Correction:

```
sudo fmriprep-docker /home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/
/home/varsha/Shreya/BIDS/ADNI/ADNI_out/ --participant_label 01 --fs-license-file
/home/varsha/Shreya/BIDS/freesurfer.txt --fs-no-reconall  --ignore fieldmaps --output-spaces
MNI152NLin6Asym:res-2
```

For details on how to extract fieldmaps from DICOMs using dcm2bids, refer to pages 6 of the dcm2bids documentation

With Fieldmap Correction, when GRE Fieldmaps are present (only change is that the --ignore fieldmaps flag has been removed. Fmriprep automatically detects fieldmaps and performs the correction) :

```
sudo fmriprep-docker /home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/
/home/varsha/Shreya/BIDS/ADNI/ADNI_out/ --participant_label 01 --fs-license-file
/home/varsha/Shreya/BIDS/freesurfer.txt --fs-no-reconall  --output-spaces
MNI152NLin6Asym:res-2
```

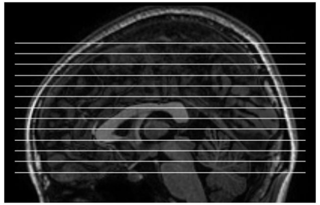
Fmriprep is only equipped to handle **GRE fieldmaps**. If you have spin-echo fieldmaps, consider using FSL's topup tool

With Fieldmap Correction, when fieldmaps are absent. Uses fmriprep's custom method However, not conducive with the MNI152Lin6Asyms Output space, thus remove flag:

```
sudo fmriprep-docker /home/varsha/Shreya/BIDS/ADNI/0002_S_1155_bids/
/home/varsha/Shreya/BIDS/ADNI/ADNI_out/ --participant_label 01 --fs-license-file
/home/varsha/Shreya/BIDS/freesurfer.txt --fs-no-reconall  --use-syn-sdc
```

Slice Timing Information:

- fMRI data is collected in 2D slices, which are then stacked together to build a volume.



- There are 2 forms of slice aquisition - **sequential (**one slice at a time) or **multiband** (multiple slices acquired in a go).

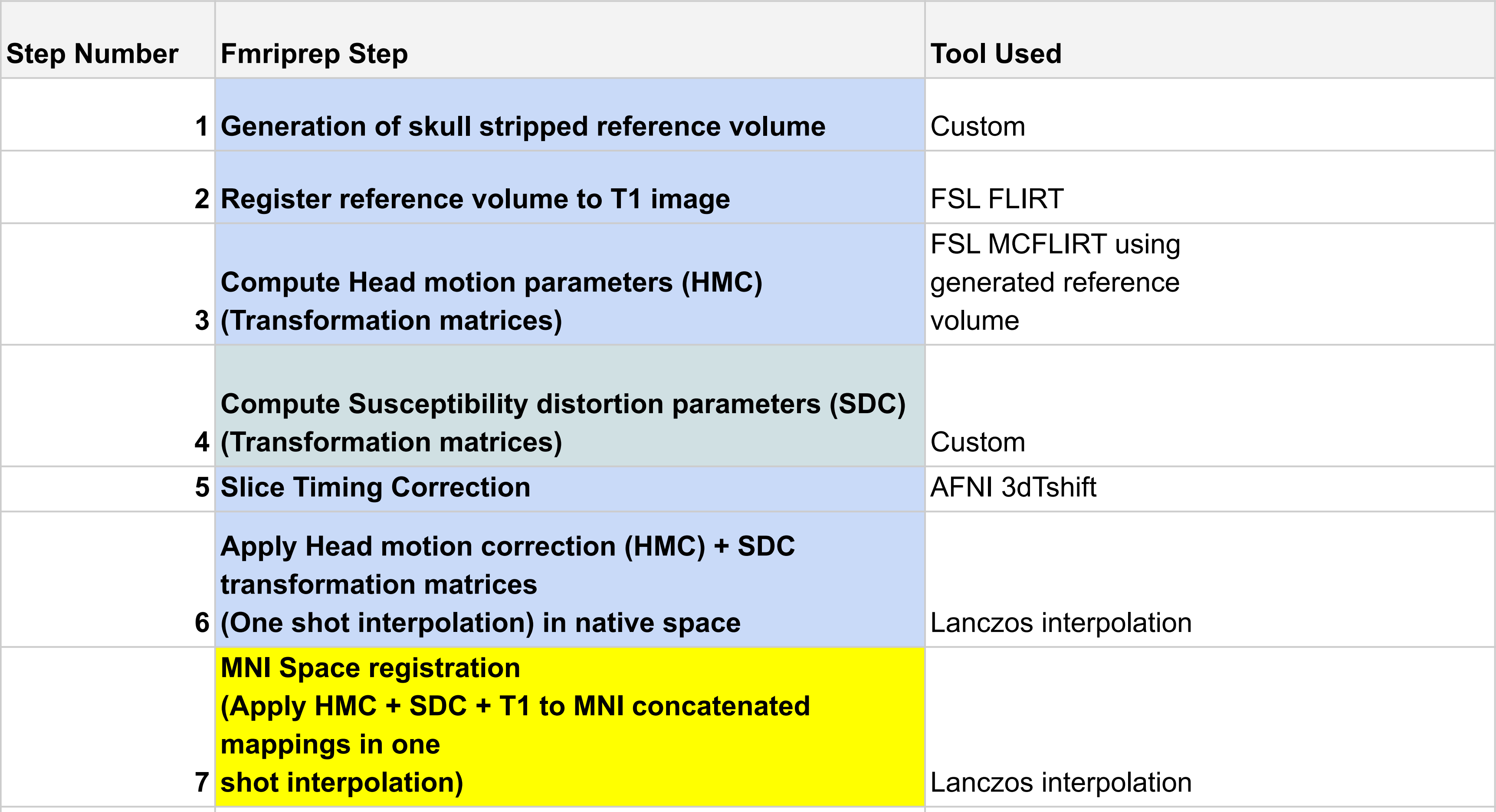
- Additionally, within sequential acquisition, slices can be acquired in an **ascending/descending** (1,23/ 321) or **interleaved manner** (1,3,5,7;2,4,6,8).

- **Fmriprep is equipped to deal with all forms of slice acquisition,** with no extra parameter to be specified.

- It extracts the **slice timing information from the .json file** corresponding the the functional file, with provides the time at which each slice in a volume was acquired.

- If this file is absent/ it is a multiband acquistion where slice timing need not be performed, use the **--ignore slicetiming flag** in the command

Steps performed within the **fmriprep** pipeline:



Step Number	Fmriprep Step	Tool Used
1	**Generation of skull stripped reference volume**	Custom
2	**Register reference volume to T1 image**	FSL FLIRT
3	**Compute Head motion parameters (HMC) (Transformation matrices)**	FSL MCFLIRT using generated reference volume
4	**Compute Susceptibility distortion parameters (SDC) (Transformation matrices)**	Custom
5	**Slice Timing Correction**	AFNI 3dTshift
6	**Apply Head motion correction (HMC) + SDC transformation matrices (One shot interpolation) in native space**	Lanczos interpolation
7	**MNI Space registration (Apply HMC + SDC + T1 to MNI concatenated mappings in one shot interpolation)**	Lanczos interpolation

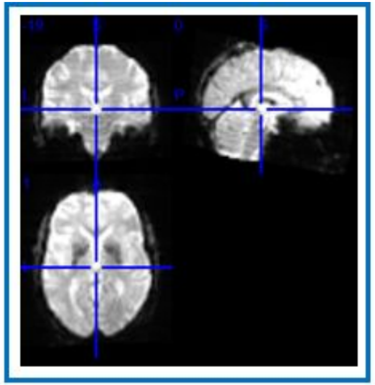
fmriprep Output Files:

1. **Visual Quality Assessment(VQA) tool:** One HTML file per subject that allows the user a thorough visual assessment of the quality of processing. *(Explanatory images in the following slides are from the generated HTML file for a sample subject)*

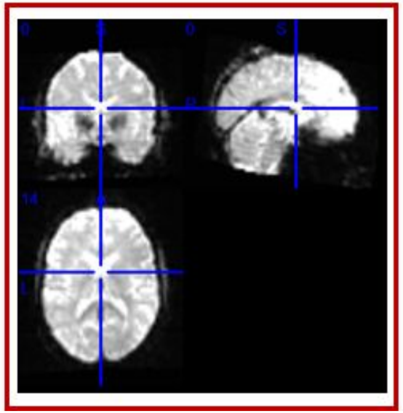
2. **Preprocessed Imaging Data:** This contains various derivatives of the original structural and functional images after various stages of preprocessing procedures have been applied. *(Snapshots of intermediate and final files to follow)*

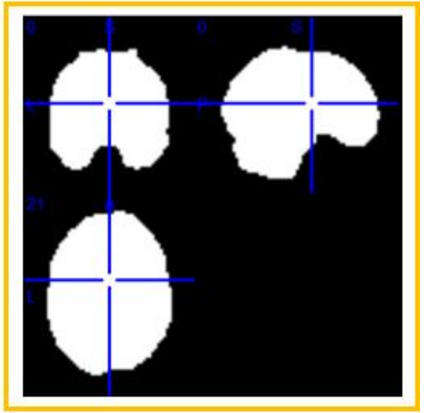
Functional Output files- 3 files:

MIddle Volume used as reference



Final Preprocessed File

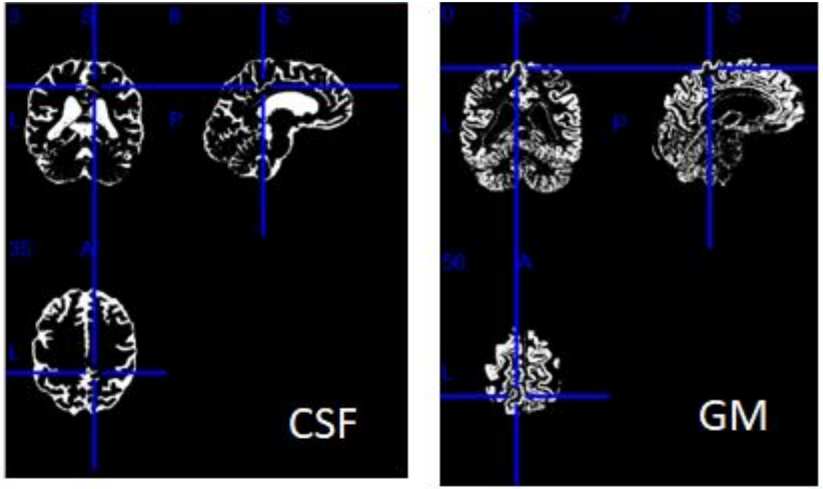




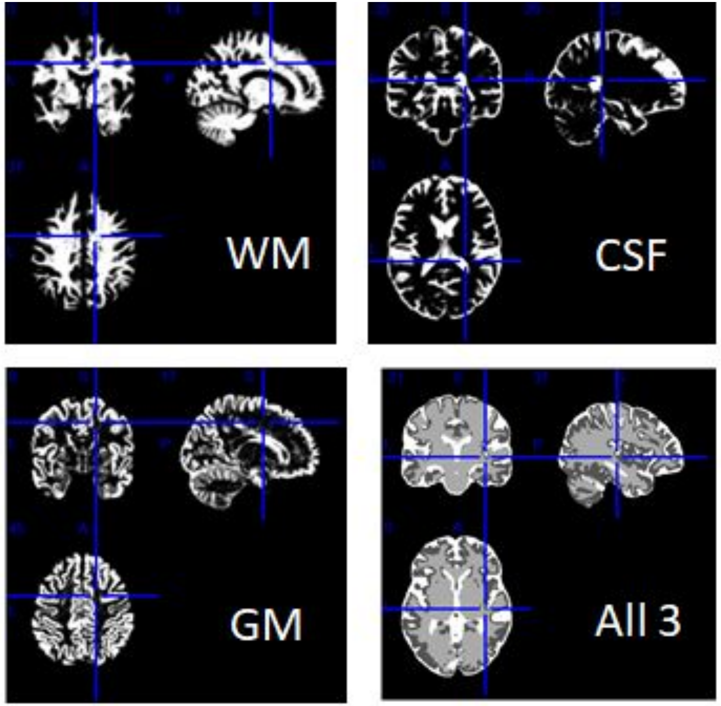
Functional Brain mask

- **Final fMRI output is not skull stripped,** thus this file can be used with external tools if skull stripping is needed (BET from FSL)

Structural Output files - part 1:

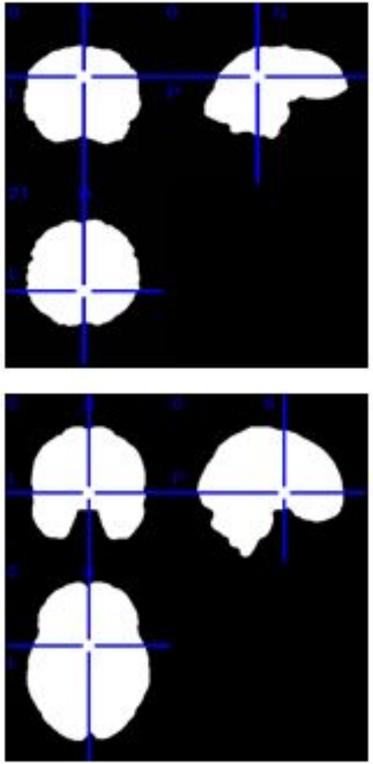


Subject's Grey Matter and white matter prior to MNI152 Registration

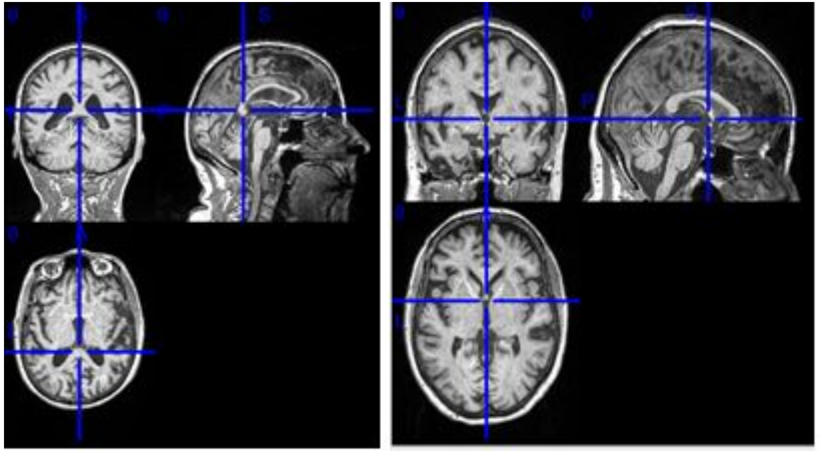


Subject's Grey Matter, white matter and CSF after registration to the MNI152 space

Structural Output files - part 2:



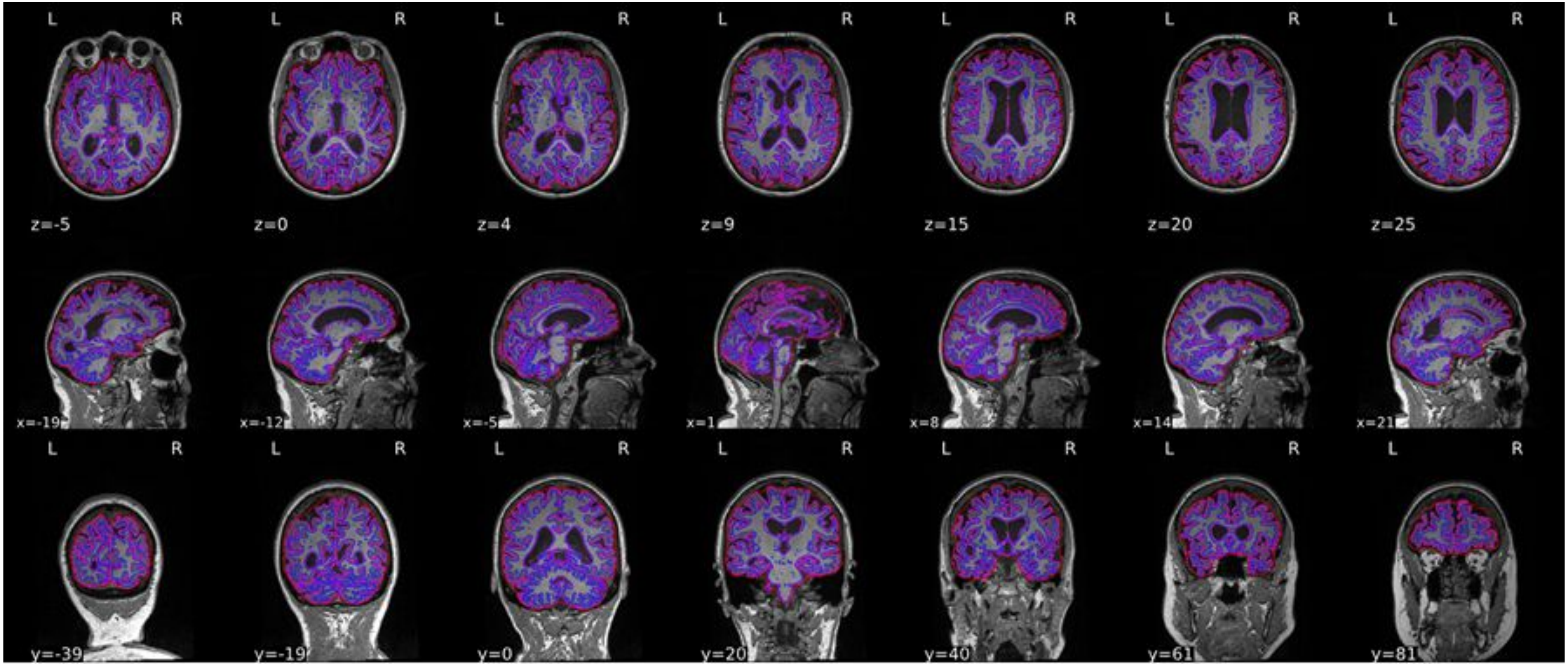
Subject’s Brain contour before and registration to the MNI152 space



Preprocessed T1 image, prior to and after MNI152 Registration

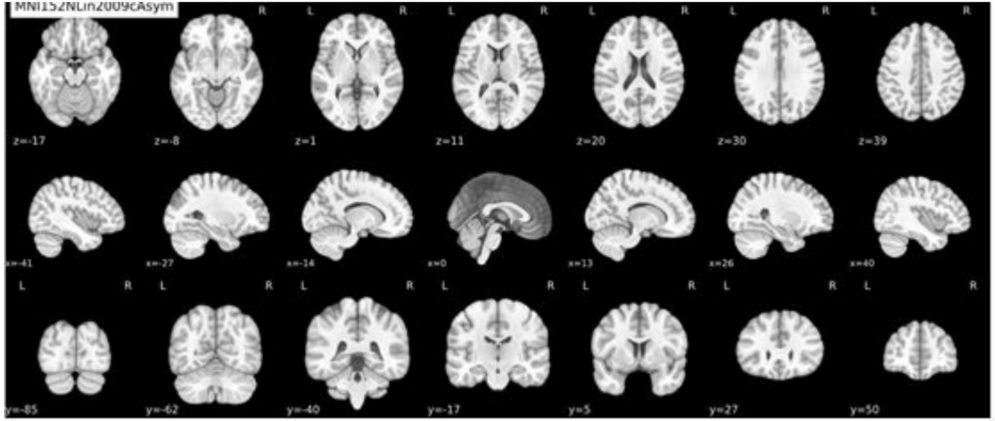
Checking the VQA Reports:

Check if the tissue segmentation and brain extraction has been performed properly.

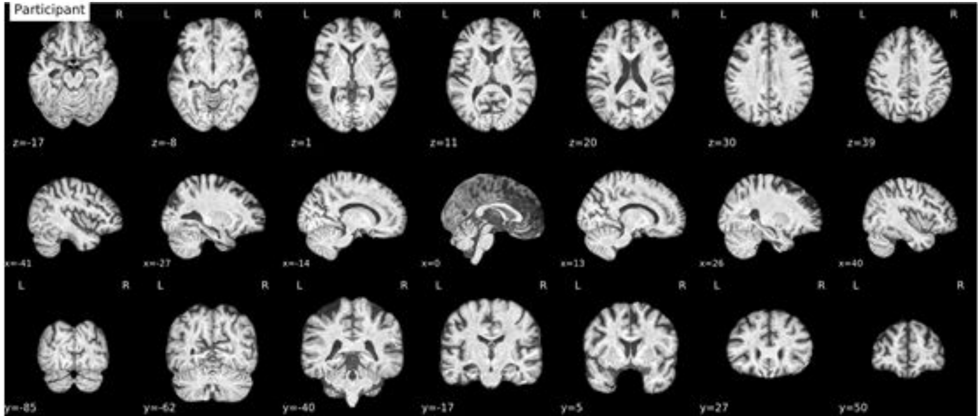


Checking the VQA Reports:

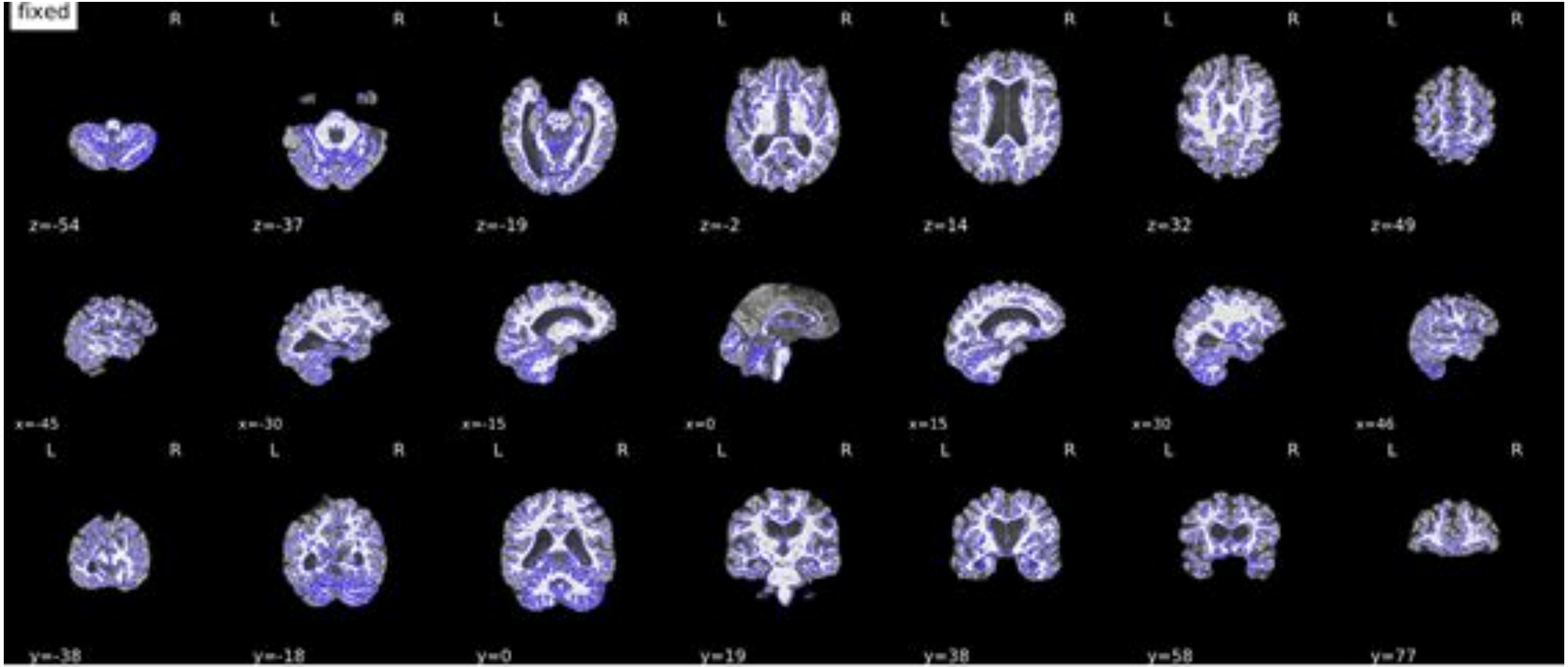
Check if the normalization has been done properly - does the registered brain look symmetric?



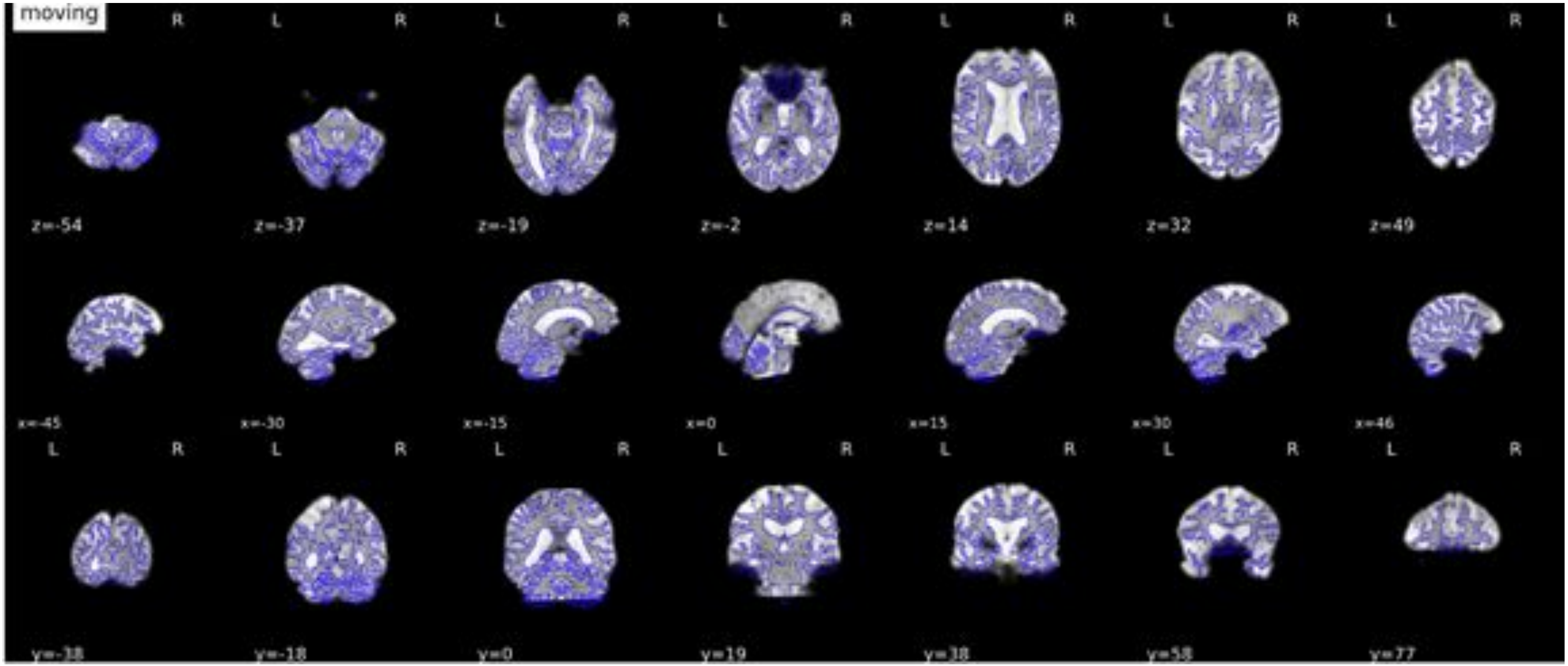
MNI Template



Subject brain registered onto MNI templa

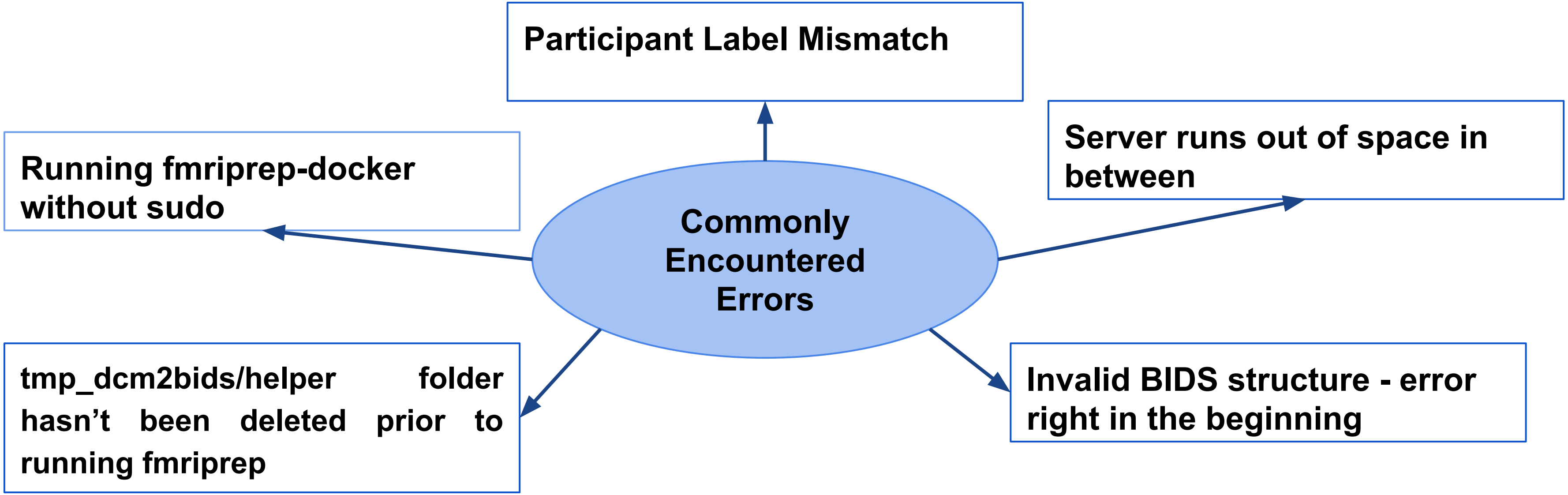


T1w preprocessed image, with white matter boundaries marked



BOLD image co-registered onto the T1 image

Troubleshooting:



Commonly Encountered Errors

Participant Label Mismatch

Running fmriprep-docker without sudo

Server runs out of space in between

tmp_dcm2bids/helper folder hasn’t been deleted prior to running fmriprep

Invalid BIDS structure - error right in the beginning

For more details, please refer to this documentation:
https://docs.google.com/document/d/1d5750a3RG245S5ME4fcds9z_lGkMTYqAxF4jlXwJ3sc/edit?usp=sharing