dcm2bids + fmriprep

Usage Instructions for Different Databases

Please note the following:

- For details about downloading data or accessing pre-downloaded data from the ADNI, RADC and HCP databases, please refer this document: https://docs.google.com/document/d/1b5vqWGeCNhsIA9nXOTALSIkFpvUB7X2ypNAatcrJP4I/edit?usp=sharing
- 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 software 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.
- All commands contain only lowercase alphabets

Preprocessing without Fieldmap Correction:

(demonstrated with an ADNI subject example)

1. **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.

2. **Create a new folder to hold our data structured into the BIDS syntax, hereafter referred to as the BIDS folder.**
3. **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

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

4. **Next run the dcm2bids_helper command.** This command converts the files 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 .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

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)

5. **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.
    - These **keywords**, and the remaining parameters of the configuration file are highly likely to remain the same across ADNI subjects, given the way ADNI names its files. This will allow you to use the same config file across subjects.
    - However, perform a spot check across a few subjects to ensure this is correct for your data.
    - 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.

Config.json - Contents of the Configuration file for Example Subject.

*%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*"
                    }
        },
       ]
}
```

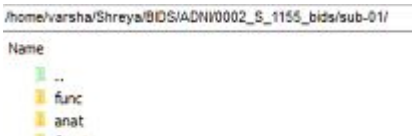
- **Now that the configuration file is ready, we will run the final command to arrange the NIfTI files into the BIDS syntax.** As the tmp_dcm2bids/helper folder was intended only to use as a reference for creating the configuration file, DELETE that folder before running the following command.

  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*

  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.).



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

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

- **Now, the BIDS folder contains our fMRI and T1 data in the BIDS syntax, as required by fmriprep. Now we can proceed to running the fmriprep command.**

  Provide full paths wherever required in the following 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 fielmaps (*%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. Then provide the full path to this output folder.
- 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.
- --fs-no-reconall indicates we do not want to use the Freesurfer option. This is because this severely extends the processing time, and its effects do not have a great impact on our research (used for surface level BOLD signal preprocessing).
- If you have fieldmaps, but choose to avoid fieldmap correction for some reason, 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, or just choose to avoid it in case of low TR in multiband scans, 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

Example Subject:

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*

Initially shows whether the data clears the BIDS validator (to see if the BIDS syntax is correct). Errors show up in red and stop the command from running successfully, Trivial warnings as the one below can be ignored.

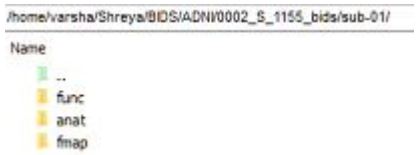
A summary of the files and modalities present in the BIDS folder are also provided, and can be used as a sanity check.

Incorporating Fieldmap Correction:

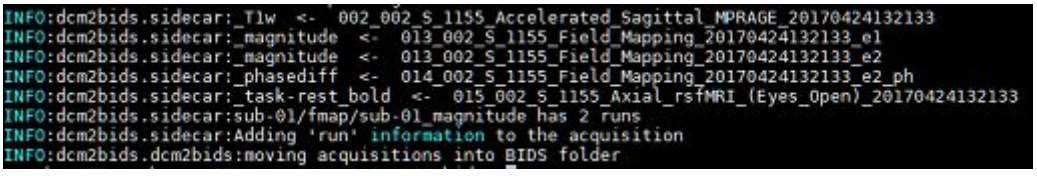
1. To incorporate fieldmaps into the BIDS structure, **add the following parameters** to the **configuration file** while using the **dcm2bids** command, and change the Sidecar Filenames to contain words that only the fieldmap files have, as seen from the helper folder. Then run the same dcm2bids steps as indicated in the previous section.

```
{
    "dataType": "fmap",
    "modalityLabel": "magnitude1",
    "criteria": {
      "SidecarFilename": "*013**e1*"
          }
  },
  {
    "dataType": "fmap",
    "modalityLabel": "magnitude2",
    "criteria": {
      "SidecarFilename": "*013**e2*"
          }
  },
  {
    "dataType": "fmap",
    "modalityLabel": "phasediff",
    "intendedFor": 0,
    "criteria": {
      "SidecarFilename": "*_ph*"
          }
  }
```

- This will result in the creation of a third folder within the subject folder within the BIDS syntax - the fmap folder.
- Then on running the **dcm2bids** command, it can be seen that the fieldmap files are also being retained in the BIDS structure.

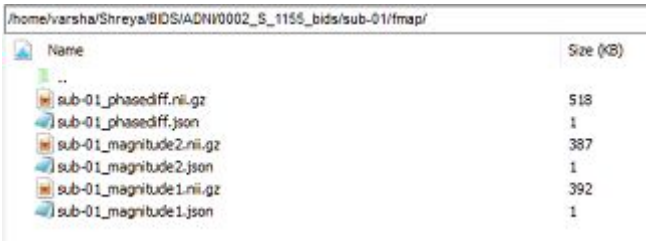


Messages displayed as the command is running (shown in the figure below) indicate the names of the NIfTI files that are being retained. In this case, we can see that all 5 files we need (1 struc., 1 func., and 3 fieldmap files have been retained)



- When using **GRE fieldmaps**, there should be 3 separate fieldmaps files created in the fmap subject folder - **2 magnitude images and 1 phase difference image**.

  The contents of the fmap folder are as follows (1 phase difference and 2 magnitude image files and all 3 corresponding .jsons)



- **The phasediff .json file only contains the Echo time corresponding to the second magnitude image.** This is what the EchoTime field of the phasediff .json originally looks like

```
"EchoNumber": 2,
"EchoTime": 0.00738,
```

  The EchoTime field from the .json of the first magnitude image needs to be manually added to the phase diff .json as “EchoTime1”: value1, and the existing EchoTime field needs to be renamed as “EchoTime2”: value2, as seen in the following image.

```
"EchoTime1": 0.00492,
"EchoTime2": 0.00738,
"RepetitionTime": 0.571,
```

- Confirm that the **phasediff.json** file has an **“IntendedFor:”** field which specifies which functional file the fieldmaps are intended to correct.

2. **Running fmriprep with fieldmap correction:**

   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 (*%optional* --ignore slicetiming --use-syn-sdc --output-spaces MNI152NLin6Asym:res-2)

- If there is an **fmap folder** in the BIDS structure, fmriprep will automatically spot it and perform fieldmap correction. There is no separate flag for performing fieldmap correction, if the fmap folder is present and structured correctly.
- As the command begins running, a message saying ‘No fieldmaps found’ will show up if fmriprep was unable to detect fieldmaps. This generally indicates that the fmap folder is either not structured correctly, or the **phasediff** .json file is missing the
- If you wish to do fieldmap correction in the absence of fieldmap data fmriprep’s custom algorithm, add --use-syn-sdc. However, this flag is not conducive with the --output-spaces MNI152NLin6Asym:res-2 flag, and hence only one can be used at a time.

Slice Timing Correction:

- When using dcm2bids, slice timing information is automatically extracted from the DICOM files and stored within the “SliceTiming:” parameter in the functional .json file.
- If an additional **--ignore slicetiming** field is not provided, **fmriprep automatically performs slice timing correction. No Separate Command is required.**
- Fmriprep is equipped to perform slice timing correction for both **multiband** and **sequentially acquired data** in the same manner, as long as the slice timing parameter is present in the functional .json file.

RADC:

- A script to convert RADC NIfTI data into the BIDS format is presently underway. Once that is done, **the need to use dcm2bids for RADC data is entirely eliminated**.
- The **fmriprep command** can then be applied in the following format (refer ADNI section for explanations about the command details)

  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 *(%optional* --ignore fielmaps --ignore slicetiming --use-syn-sdc --output-spaces MNI152NLin6Asym:res-2)

  RADC specific details:

- RADC has sequential interleaved slice acquisition which means that the slices are collected in the order 1,3,5,7…. and then 2,4,6,8,... is the total number of slices is odd, and vice versa if the number of slices is even. This information will be incorporated into the NIfTI to BIDS format script, in .json file generation.
- RADC has GRE fieldmaps - 2 Magnitude and 1 Phase image. This means that fmriprep will be able to use the fieldmap data RADC provides to perform the correction, thereby preventing the need to use the --use-syn-sdc flag which uses fmriprep's custom susceptibility distortion correction algorithm.

HCP:

- HCP is only available for download in the NIfTI format. A script to convert HCP NIfTI data into the BIDS format can be designed (REQUIRED?). Once that is done, **the need to use dcm2bids for HCP data is entirely eliminated**.
- The **fmriprep command** can then be applied in the following format:

  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 *(%optional* --ignore fielmaps --ignore slicetiming --use-syn-sdc --output-spaces MNI152NLin6Asym:res-2)

  HCP specific details:

- HCP uses multiband acquisition, which means more than one slice of a volume is acquired at the same time. Consequently, the Time of Repition (TR) for HCP data is very low (0.67s). Therefore HCP does not recommend performing slice timing correction as a part of preprocessing. This can be ensured in fmriprep by using the --ignore slicetiming flag.
- HCP provides spin-echo fieldmaps that fmriprep is not compatible with. Therefore,

the fieldmap correction can be ensured to be off using the --ignore fieldmaps flag. Fmriprep's custom algorithm can also be used with the --use-syn-sdc flag.

sudo fmriprep-docker /home/varsha/Shreya/BIDS_final/RADC_nifti2bids/ */home/varsha/Shreya/BIDS_final/RADC_nifti2bids_out/* --participant_label *01* --fs-license-file */home/varsha/Shreya/BIDS/freesurfer.txt* --fs-no-reconall --ignore slicetiming --output-spaces MNI152NLin6Asym:res-2

sudo fmriprep-docker /home/varsha/Shreya/BIDS_final/RADC_dcm2bids/ */home/varsha/Shreya/BIDS_final/RADC_dcm2bids_out/* --participant_label *01* --fs-license-file */home/varsha/Shreya/BIDS/freesurfer.txt* --fs-no-reconall --ignore slicetiming --output-spaces MNI152NLin6Asym:res-2