Processing Neuroimaging and Behavioral Data with the Brain Imaging Data Structure (BIDS)

Praneeth Guduguntla[1, 2, 3], **Kayle S. Sawyer**[2, 3, 4], **Daniel Salz**[2, 3, 4], **Gordon J. Harris**[3], **Marlene Oscar-Berman**[2, 3, 4]

Cupertino High School, Cupertino, CA[1]; Boston University, Boston, MA[2]; Massachusetts General Hospital, Boston, MA[3];
VA Boston Healthcare System, Boston, MA[4]

INTRODUCTION

Frequent inconsistencies in Magnetic Resonance Imaging (MRI) aquisitions (e.g., functional MRI [fMRI], diffusion MRI [dMRI], etc.) can slow imaging processing workflows. Scanning procedures vary between subjects due to irregularities such as claustrophobia, bathroom breaks, and equipment malfunctions that inevitably perturb standardized acquisitions. Moreover, because neuroimaging data are acquired over months or years, naming conventions change, and new types of scans are collected, making inconsistencies. The BIDS data structure allows for improved reproducibility of neuroimaging results, easy accessibility of the data, and facility in workflow analyses.

DISCUSSION

Many neuroimaging datasets consist of irregular naming conventions, missing/corrupted files, or unclear directory structures. The BIDS dataset allows laboratories to curate data according to the BIDS specification, irradicating the possibilities of error.

CONCLUSION

BIDS datasets are being widely used as the industry-standard for shareable neuroimaging data. Open science allows laboratories to share valuable research data, promoting reproducability and validation of existing research results.

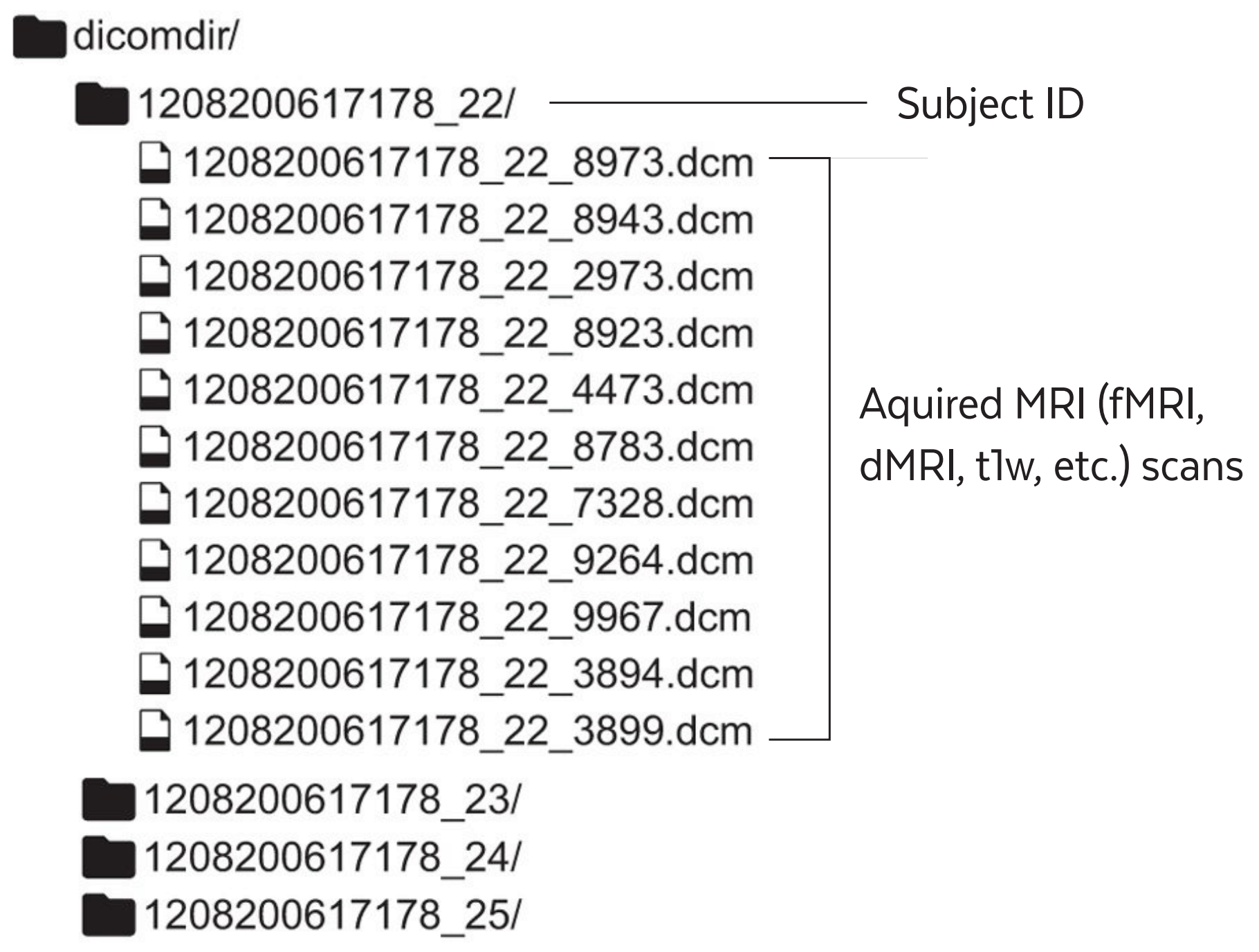
ACKNOWLEDGEMENTS

I would like to thank Dr. Kayle Sawyer, Dr. Daniel Salz, and Dr. Marlene Oscar-Berman for giving me the opportunity to contribute to their research.

NIH Grant: NIAAA R01-AA07112

METHODS AND RESULTS

MRI scans are initially stored in the DICOM (Digital Imaging and Communications in Medicine) file format that is extracted into unique formats for various labs. The variance of data structures among labs can result in disorganized directory structures.



```
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/
  1208200617178_25/
```

Figure 1: Typical Dicom File Structure

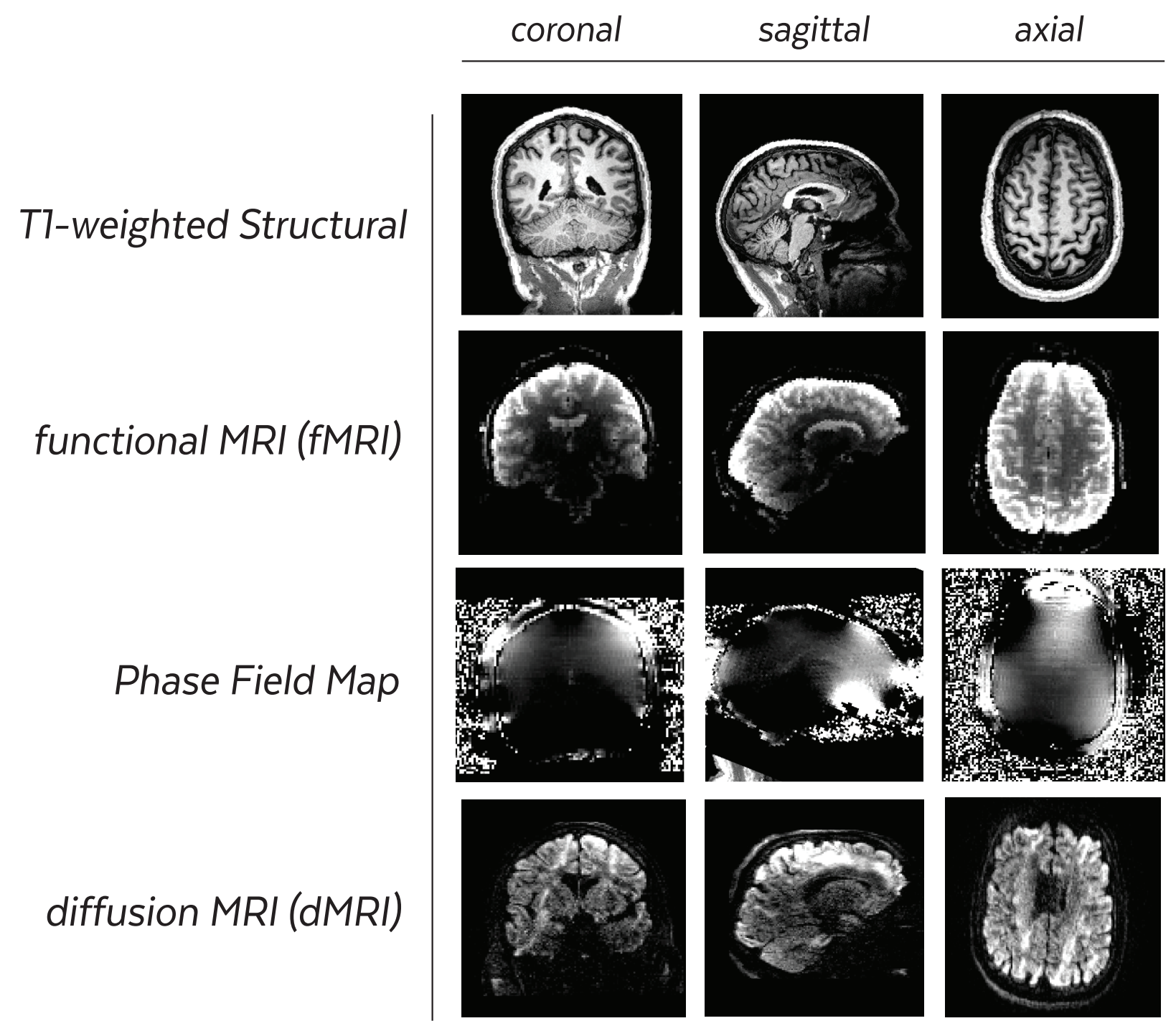
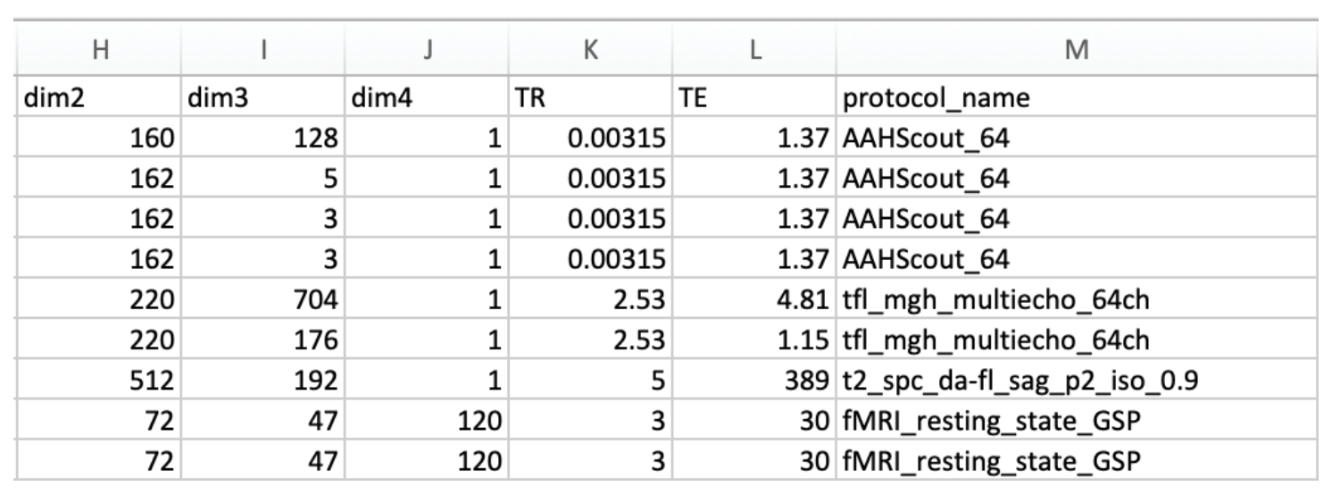


Figure 2: MRI Imaging Modalities

In our laboratory, the neuroimaging data were acquired from adult individuals with a history of Alcohol Use Disorder (AUD) and demographically similar people without a history of AUD. A BIDS dataset converter, HeuDiConv, was used to transform the dataset. HeuDiConv uses a custom 'heuristic' Python file to unpack the DICOMs into NIfTI (Neuroimaging Informatics Technology Initiative) scans and organizes those scans into the standardized directory structure with metadata stored in JSON (a text based format for data storage) files. In order to write the 'heuristic', we first wrote a custom Python script to determine all unique scan types in the project.



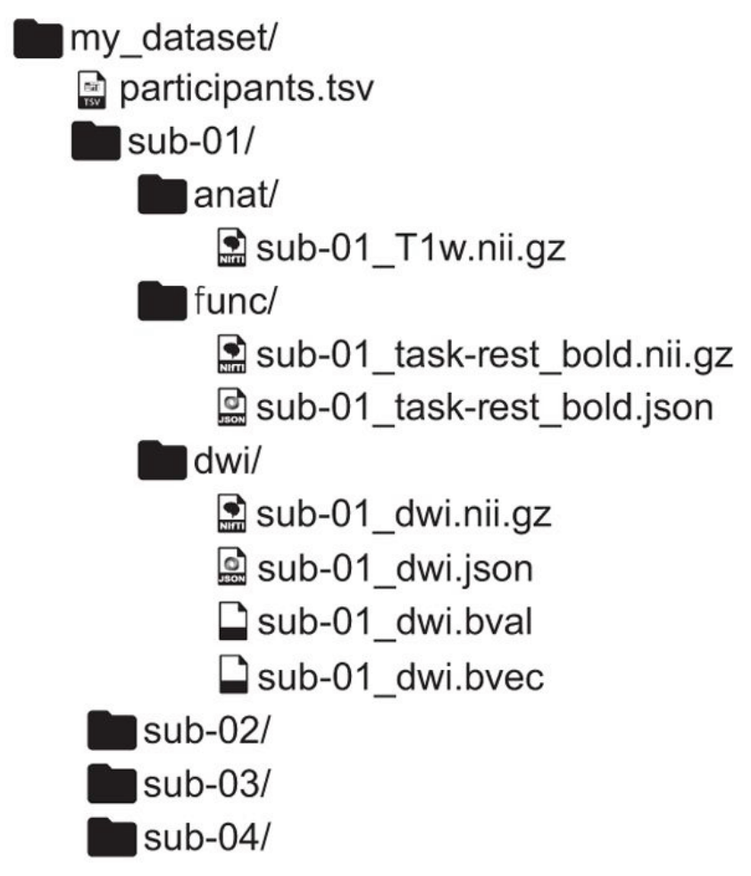
H	I	J	K	L	M
dim2	dim3	dim4	TR	TE	protocol_name
160	128	1	0.00315	1.37	AAHScout_64
162	5	1	0.00315	1.37	AAHScout_64
162	3	1	0.00315	1.37	AAHScout_64
162	3	1	0.00315	1.37	AAHScout_64
220	704	1	2.53	4.81	tfl_mgh_multiecho_64ch
220	176	1	2.53	1.15	tfl_mgh_multiecho_64ch
512	192	1	5	389	t2_spc_da-fl_sag_p2_iso_0.9
72	47	120	3	30	fMRI_resting_state_GSP
72	47	120	3	30	fMRI_resting_state_GSP

Figure 3: TSV file with metadata for all scans in the project

```
import os

def create_key(template, outtype=('nii.gz',), annotation_classes=None):
    if template is None or not template:
        raise ValueError('Template must be a valid format string')
    return template, outtype, annotation_classes

def infotodict(seqinfo):

    #scout scans
    aahscout =
create_key('sub-{subject}/{session}/anat/sub-{subject}_{session}_acq-aascout_run-{item}_localizer')

    #t1w
    mprage_multi =
create_key('sub-{subject}/{session}/anat/sub-{subject}_{session}_acq-MPRAGE_run-{item}_T1w')
    mprage_rms =
create_key('sub-{subject}/{session}/anat/sub-{subject}_{session}_acq-MPRAGErms_run-{item}_T1w')
    t1w = create_key('sub-{subject}/{session}/anat/sub-{subject}_{session}_run-{item}_T1w')
    t1w_acq = create_key('sub-{subject}/{session}/anat/sub-{subject}_{session}_acq-ep2d_run-{item}_T1w')
```

```
    for s in seqinfo:
        if("AAHScout_64" == s.series_description) or ("AAScout" == s.series_description) or ("AAHScout_64chan" == s.series_description):
            info[aahscout].append(s.series_id)
        if(("MEMPRAGE" in s.protocol_name) or ("tfl_mgh_multiecho_" in s.series_description)) and ("RMS" not in s.series_description):
            info[mprage_multi].append(s.series_id)
        if(("MEMPRAGE" in s.protocol_name) or ("tfl_mgh_multiecho_" in s.series_description)) and ("RMS" in s.series_description):
            info[mprage_rms].append(s.series_id)
        if("t1w" in s.protocol_name):
            info[t1w].append(s.series_id)
        if("ep2d_t1w_A-P" == s.series_description):
            info[t1w_acq].append(s.series_id)
        if("FLAIR" in s.protocol_name) or ("t2_spc_da-fl" in s.protocol_name):
            info[t2_flair].append(s.series_id)
        if("T2_CUBE" in s.protocol_name):
            info[t2_cube].append(s.series_id)
        if("T2_TSE_Axial_HiRes" in s.protocol_name) or ("*tse2d1_18" in s.sequence_name):
            info[t2w].append(s.series_id)
        if("t2_swi3d_tra_p2_1.2mm_384" in s.protocol_name) or("t2_fl3d_tra_p2_swi_highres" in s.protocol_name):
            info[swi].append(s.series_id)
```

Figure 4: 'heuristic ' file for conversion from DICOM to NIfTI
The 'heurisitic' file provides HeuDiConv with instructions for creating the BIDS directory. Using the list of all unique scan types, we wrote file names for each type according to the BIDS specification. As HeuDiConv parses through the DICOM directory and unpacks the files into NIfTI formats, it assigns each of the new files a name using the keys provided by the 'heuristic' file.



```
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/
  sub-04/
```

Figure 5: Example of Valid BIDS dataset

Our laboratory measured the research participants' brain activity (using fMRI), and their behavioral choices, while they were responding to a task (Monetary Reward Task) in the MRI scanner. The participants' responses to the task and the time markers at which they responded were stored in TSV (Tab Separated Values) files. These data typically would be stored in an external directory; however, with the BIDS format, the data were stored directly alongside the fMRI scan.

Analysis Workflow

As opposed to the tedious processes of manual inputting unpacked data files, with BIDS compatible applications such as fMRIprep or dMRIprep, analyzing imaging data can be streamlined.

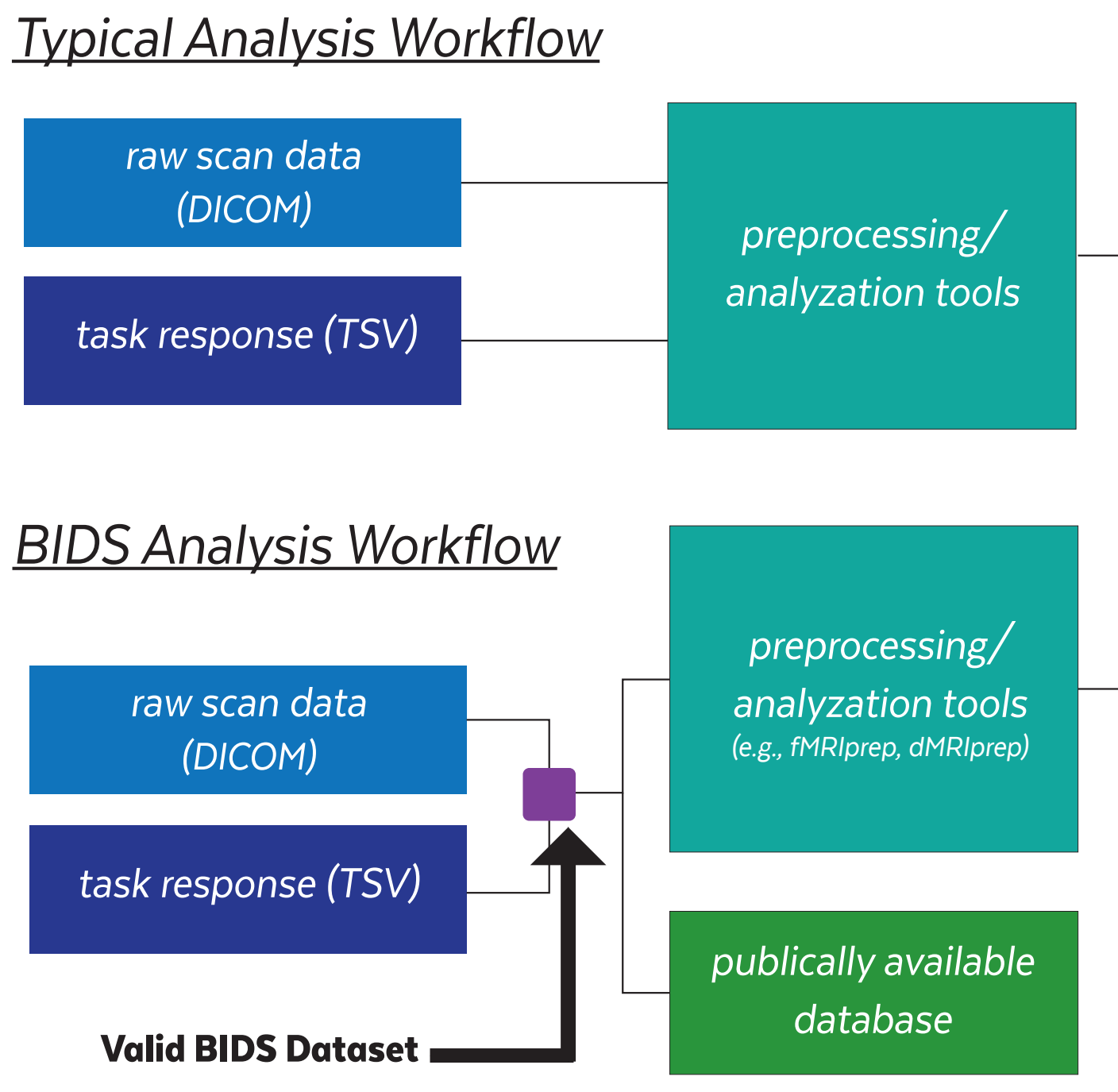


Figure 6: Comparison between a typical analysis workflow and an analysis workflow with a valid BIDS dataset