

---

# **TAMU MRI Resources Documentation**

***Release 1.0***

**Joseph M. Orr**

**May 18, 2021**



---

## Contents

---

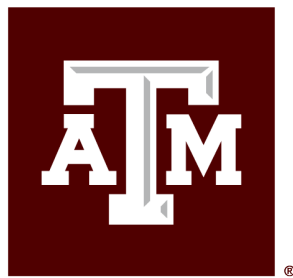
<b>1</b>	<b>About</b>	<b>3</b>
<b>2</b>	<b>Contact</b>	<b>5</b>
<b>3</b>	<b>Contents</b>	<b>7</b>
3.1	MRI Study IRB Procedures . . . . .	7
3.2	Scanning at TIC . . . . .	7
3.3	Neuroimaging Computing Resources . . . . .	8
3.4	Converting Raw Data . . . . .	10



<http://tamu-fmri.readthedocs.io/en/latest/>.

This site is meant to reserve as a comprehensive guide and repository for resources to support neuroimaging research in the Department of Psychological and Brain Science at Texas A&M University. Currently, there are guides for setting up a study at TIPS, available MRI sequences, other resources at TIPS, and computing resources.





TEXAS A&M UNIVERSITY

# Department of Psychological and Brain Sciences

Human neuroimaging at A&M is carried out at the TIPS facility on a Siemens Verio 3T Scanner. More information about the facility can be found here: [TIPS Facility](#).





## CHAPTER 2

---

### Contact

---

This resource was built by Joseph Orr with support from the Department of Psychological and Brain Sciences. Contact Joe at [joseph.orr@tamu.edu](mailto:joseph.orr@tamu.edu).



## 3.1 MRI Study IRB Procedures

- This page is currently under construction

### 3.1.1 IRB

At Texas A&M, MRI studies are reviewed by full board, which meets monthly. Applications are due about a month before the meeting. The schedule of meetings is [available here](#).

#### Personnel

Techs (i.e., Vidya) need to be added to study personnel.

#### Consent

The Research Compliance office has several templates for consent forms [available here](#). These templates don't contain language regarding data sharing, which is becoming the norm for the MRI community. There's an excellent discussion of data sharing [available here](#), including a template for a consent that allows for permissive data sharing. The Texas A&M IRB has approved permissive data sharing language. Data must be de-identified before sharing, including scrubbing the dicom headers for identifying information and elimination of identifying facial/dental features. There are easy to use tools available for doing so [anonymization tools](#).

## 3.2 Scanning at TIC

Procedures and tips for setting up a new study and running subjects at the TIC facility.

### 3.2.1 Safety Training

There are numerous safety training procedures for anyone involved with scanning at TIC. TAMU has a safety training here: <https://ehsdtraining.tamu.edu/>

### 3.2.2 Study Setup

TIPS has a form (posted to the fMRI Group Team Drive) that needs to be filled out prior to scanning in order to establish billing of scan hours. This form is currently submitted to Vidya Sridhar. The form requests information about scanning sequences needed, funding, IRB approval, research staff.

### 3.2.3 Running a study at TIC

#### Information for participants

<https://aggiemap.tamu.edu/?bldg=1904>

TIPS is located at 800 Raymond Stotzer Parkway (University Blvd) next to the new Veterinary Medicine Building Complex. The imaging center is back down the hallway to the left of the main entrance.

#### Directions from Psychology Building to the TIPS building

```
<iframe src="https://www.google.com/maps/d/embed?mid=1Mdg9esVRu665PAy8KOAZjcwYc1LGKP1" width="640" height="480"></iframe>
```

## 3.3 Neuroimaging Computing Resources

This page gives an overview of the resources available for neuroimaging computing and data storage.

### 3.3.1 Data Storage Server

Once a scan is completed, it is transferred to the Department's storage server (see your PI or Hugh McCann for access). This server had to be mounted on your local computer. Once the server is mounted on your computer you can transfer the files to your local workstations or to a computing cluster. Both brazos and terra are accessible through globus. You must install globus on your local computer and set it up as an endpoint. As part of the set-up, you must add access to mounted drives (e.g., the storage share or GoogleDrive), as shown [in these instructions](#).

[Terra Globus Instructions](#)

### 3.3.2 Using a computing cluster

[https://campuscluster.illinois.edu/user\\_info/doc/beginner.html](https://campuscluster.illinois.edu/user_info/doc/beginner.html)

### 3.3.3 ViDaL Cluster

The Vidal cluster supports high memory jobs, graphics intensive processes (P100 GPUs), and data requiring added security or legal protection (e.g., HIPAA, FERPA). Information on accessing the cluster is available [here](#).

### 3.3.4 Terra Supercomputer

The terra cluster is available to members of TAMU through the High Performance Research Computing (HPRC) Center. HPRC offers frequent [training classes and workshops](#) on a variety of computing topics for beginners and experienced users.

The [terra user guide](#) provides a lot of information of using terra. The only fMRI software currently installed as a module on terra is FSL, but you can request the installation of other software by [emailing the HPRC Help](#).

#### Remote Visualization

The HPRC Wiki has a great page on [remote visualization](#).

#### Loading software resources

Software on terra is managed with the `modules` system. To see what software is available on brazos, use `module avail`. If you have an idea of what software you want to load, you can use `module spider <name>` to see what versions are available and how to load them. If you don't specify a version, the most recent version is loaded by default.

To load a specific version you need to give the version number, e.g., `:: module load fsl/5.0.10`

By loading a module you are configuring paths and loading any dependent software. Once you load a software module, the executables should be available, e.g.,

```
$ which fsl
/usr/bin/which: no fsl in (/home/joseph.orr/bin:/usr/lib64/qt-3.3/bin:/usr/
↪local/bin:/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin/brazos)
$ module load fsl
$ which fsl
/apps/psyc/fsl/5.0.11/bin/fsl
```

#### Container packages

There is a growing trend in software to use *containers*, which are “lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.” Containers are useful for running software that require lots of library dependencies that might lead to compatibility issues with software already on a system. **Docker** is a leader in the field of container packaging, and there are already several docker container images available for neuroimaging software, e.g., `fmriprep` and `mriqc`. However, docker containers have issues with security on HPC systems, but there is a workaround called *singularity*. *singularity* is available on terra for running container images.

A lot of MRI software is available as a container through [neurodocker](#).

You should download the container images to your user or group scratch directory.

#### Submitting jobs

When you login to the cluster, you are on the login node, which is only meant for simple processes like editing text files or copying a small number of files. More complicated jobs should be submitted to the compute nodes with the job manager.

Terra uses `slurm` for managing jobs, an open source package which is used on most academic computing clusters. The terra user guide has a lot of documentation on submitting a variety of different types of jobs. A job is submitted

with `sbatch` and monitored with `squeue`. If you need to actively monitor or interface with a job, you can start an interactive job with the wrapper `srun`. This will queue the resources needed on a compute node.

FSL has a built in program for submitting jobs to slurm called `fsl_sub` that actually uses `sbatch` to submit your job to slurm. However, many packages will self-submit so you won't need to use `fsl_sub`. A list of the self-submitting programs is found [here](#), and includes `feat`. Because many of these self-submitting jobs do some organizational procedures on the login node before submitting, you may need to call them from an interactive node.

### submitting container jobs to slurm

To submit a container job to slurm, you must create a submission script as discussed above. The submission script will call singularity as follows:

```
singularity run <container>.simg <container commands>
```

AFNI (and possibly other containers) can be run interactively, meaning you can pull up the typical GUI and work with that (use singularity shell). To do this however, you will need to run the container as an interactive job so that it isn't running on the login node. You will need to make a slurm submission script, but instead of calling it with `sbatch` you can use `srun` instead. In your submission script you will need to specify the required walltime, RAM, nCPUS, etc.

## 3.4 Converting Raw Data

### 3.4.1 DICOM Conversion

Data is saved from the scanner in a raw format called DICOM. On the TIPS scanner the files have the ending .IMA, but these are a form of DICOM. As DICOM images cannot be read by most software packages, we need to convert them to a standard format for neuroimaging software. The current standard is NIFTI (.nii). The most common way to perform a DICOM-to-NIFTI conversion is a package developed by Chris Rorden called *dcm2nii*. The current version of *dcm2nii* is called *dcm2niix*.

To convert your DICOM image you can simply call *dcm2niix* as follows: *dcm2niix <dicomdir>*, where *<dicomdir>* is the directory containing the DICOM images for a given scan.

### 3.4.2 BIDS Format

Brain Imaging Data Structure or *BIDS* is an attempt at creating a standardized organization structure for neuroimaging experiments. One benefit of BIDS is that every scan has a .JSON file accompanying the NIFTI files which contains descriptive information about the scan, such as scanning parameters. Most of this information come straight from the DICOM file headers, but this information is usually lost when converting to NIFTI format. BIDS also has specifications for behavioral data, esp. timing files used for fMRI. One of the main purposes of BIDS is to make sharing data as easy as possible. There are also data preprocessing and analysis tools now available that can run pipelines on your data with little need for user input, as the specification contains most of the critical information about settings. Examples are: *fmriprep* and *mriqc*.

There are several tools for converting your DICOM images to BIDS format. I am most familiar with the *bidskit* utility, but the *heudiconv* tool is also a popular tool for DICOM to BIDS conversion.

*bidskit* is current installed on brazos as a container (see below). *heudiconv* is installed to the *anaconda3* module.

## BIDSKIT

`bidskit` is run in a two step process. The first step converts the DICOM to NIFTI format and the second step organizes the converted files in the BIDS stucture.

To run `bidskit` call:

```
singularity run --clean-env /apps/psyc/containers/bidskit/<version> -i <dicomdir> -o
↪<bidsdir>
```

Where `<version>` is the version of `bidskit` you want to use, `dicomdir` is the path to the directory with all of your subject's dicom data, and `bidsdir` is the path of the directory where you would like to put the bids formatted output.

The first pass performs the actual dicom to nifti conversion so it takes a long time. The first pass conversion will create new translator dictionary (`Protocol_Translator.json`) in the root DICOM folder. This has been prefilled with the protocol series names from the DICOM header of all unique series detected in the original DICOM files. The command will also create the new BIDS directory containing a single temporary conversion directory containing Nifti images and JSON sidecars for all series in the source DICOM folder. You will need to edit the translator dictionary so that each scan has the necessary information from the bids specifications. Here is an example of a study with structural, rest, functional, and diffusion images as well as distortion correction fieldmap-type images:

```
{
  "Localizer": [
    "EXCLUDE_BIDS_Directory",
    "EXCLUDE_BIDS_Name",
    "UNASSIGNED"
  ],
  "rsBOLD_MB_1": [
    "func",
    "task-rest_acq-MB_run-01_bold",
    "UNASSIGNED"
  ],
  "T1w": [
    "anat",
    "T1w",
    "UNASSIGNED"
  ],
  "LearnRun1": [
    "func",
    "task-learn_acq-MB_run-01_bold",
    "UNASSIGNED"
  ],
  "LearnRun2": [
    "func",
    "task-learn_acq-MB_run-02_bold",
    "UNASSIGNED"
  ],
  "learn_distcorrAP": [
    "fmap",
    "dir-AP_epi.nii.gz",
    ["task-learn_acq-MB_run-02_bold", "dwi_96dir_PA"]
  ],
  "learn_distcorrPA": [
    "fmap",
    "dir-PA_epi.nii.gz",
    ["task-learn_acq-MB_run-01_bold", "dwi_96dir_AP"]
  ],
  "cmrr_mbep2d_diff_directions96_AtoP": [
```

(continues on next page)

(continued from previous page)

```
        "dwi",
        "dwi_96dir_AP",
        "UNASSIGNED"
    ],
    "cmrr_mbep2d_diff_directions96_PtoA": [
        "dwi",
        "dwi_96dir_PA",
        "UNASSIGNED"
    ]
}
```

## HEUDICONV

Information on heudiconv can be found here: [heudiconv](#). To use it you need to load the modules for `anaconda3` and `dcm2niix`. To run heudiconv you need to create a heuristic file. There is documentation on the file requirements at the website above. If you are using the CMRR multiband sequences:

```
srunk heudiconv -d '{subject}/{session}/**/*.IMA' -s TEST -ss sess1 -f /fdata/scratch/
→ joseph.orr/t3/tools/bin/cmrr_heuristic_vts.py
```