



Introduction to SCC & fMRIPrep

Mitchell Horn

mhorn@bu.edu

Research Computing Services
Information Services & Technology
Boston University

Resources

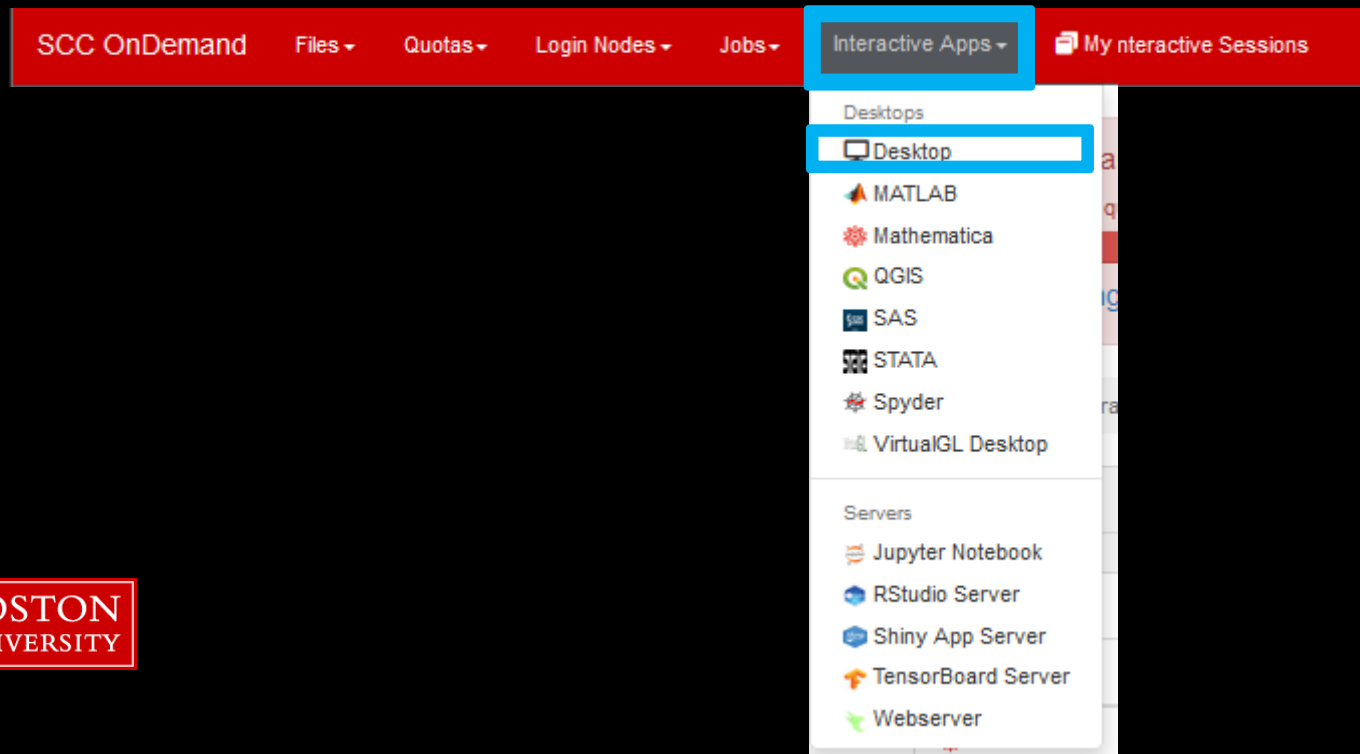
- Tutorial Slides available:
 - <http://rcs.bu.edu/examples/imaging>
- Questions, Problems, Need Help?
 - help@scc.bu.edu
 - mhorn@bu.edu

Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

Open an OnDemand session

1. Go to: scc-ondemand.bu.edu
2. Interactive Apps
3. Desktop



Open an OnDemand session

Desktop
This app will launch an interactive desktop on a compute node.

List of modules to load (space separated)

Working Directory

The directory to start in. (Defaults to home directory.)

Initial command to run

Number of hours

Number of cores

Number of gpus

Project

Extra qsub options

I would like to receive an email when the session starts

* The Desktop session data for this session can be accessed under the data root directory.

12 hours

1 core

project-ID

click launch!

5

Open an OnDemand session

Desktop (6994379) 1 core | Running

Host: [>_scc-bb3](#) Delete

Created at: 2022-08-29 11:12:24 EDT

Time Remaining: 19 hours and 53 minutes

Session ID: [afff80fb-ca1f-44fd-a440-0637da849e84](#)

Compression Image Quality

0 (low) to 9 (high) 0 (low) to 9 (high)

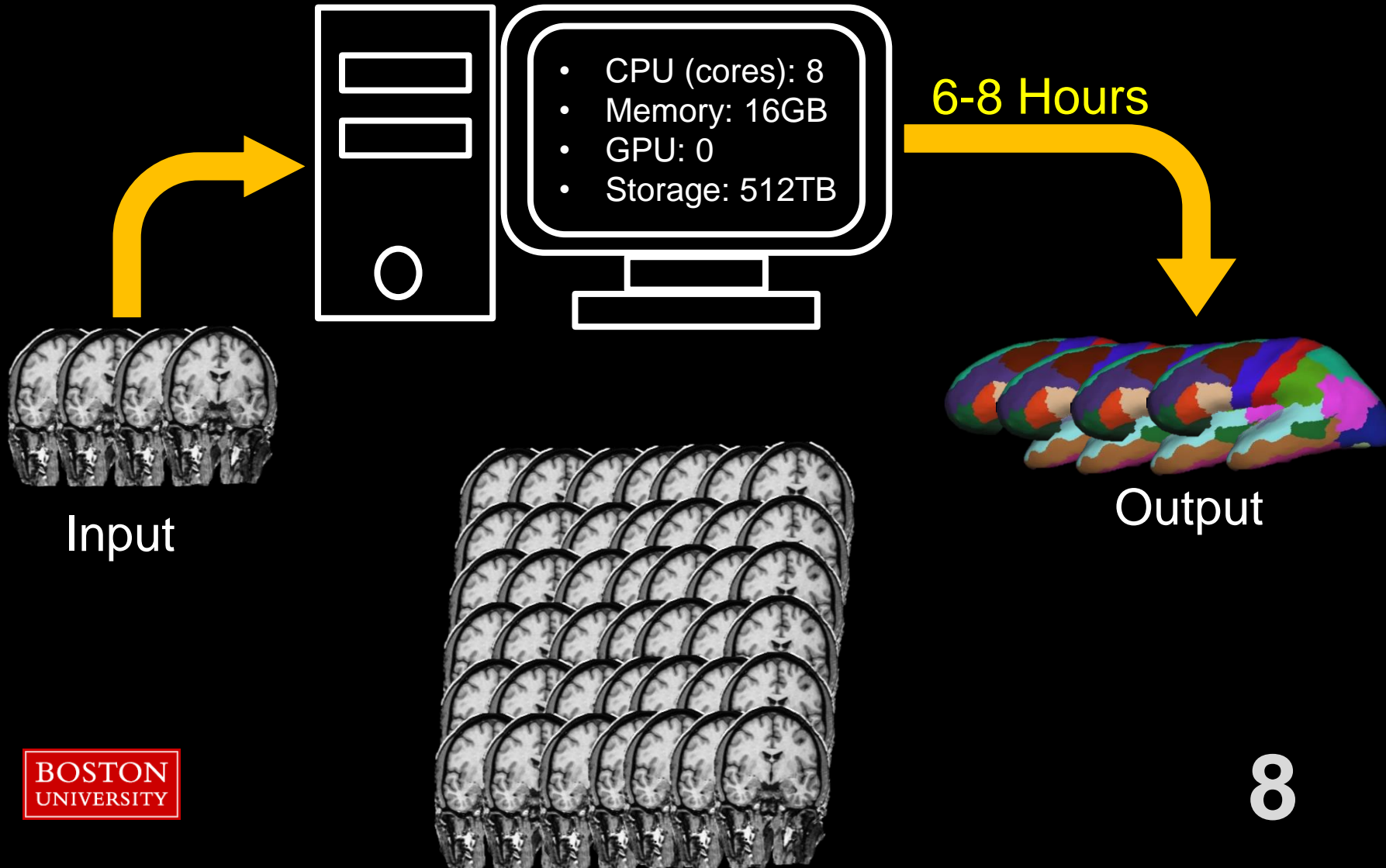
[Connect to Desktop](#) View Only (Share-able Link)

click Connect to Desktop!

Outline

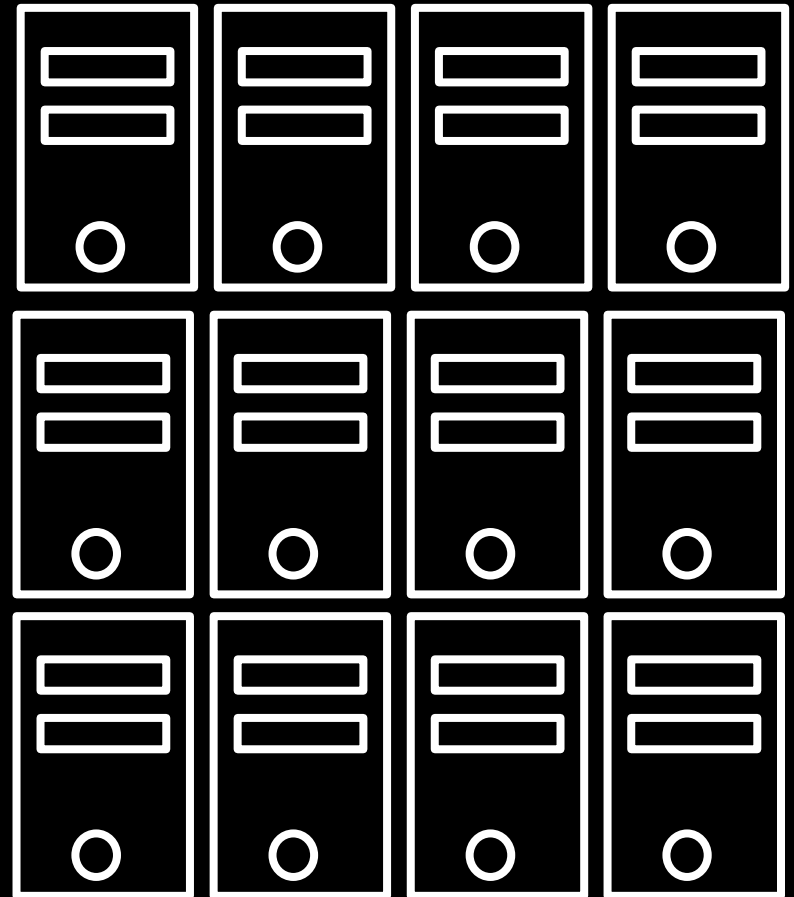
- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

What is the SCC?

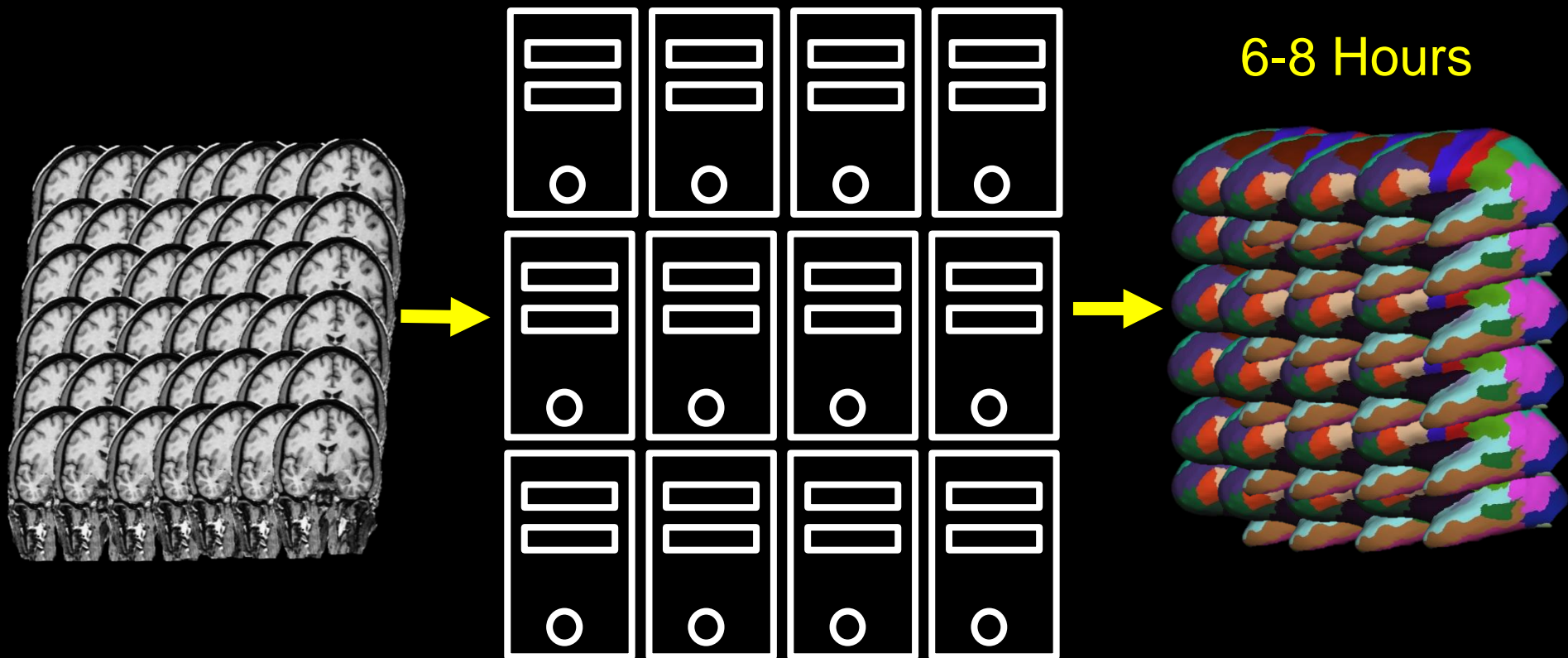


What is the SCC?

- > 900 Computers (nodes)
- > 25,000 CPUs (cores)
- > 100 GPUs
- > 9 PB (storage)



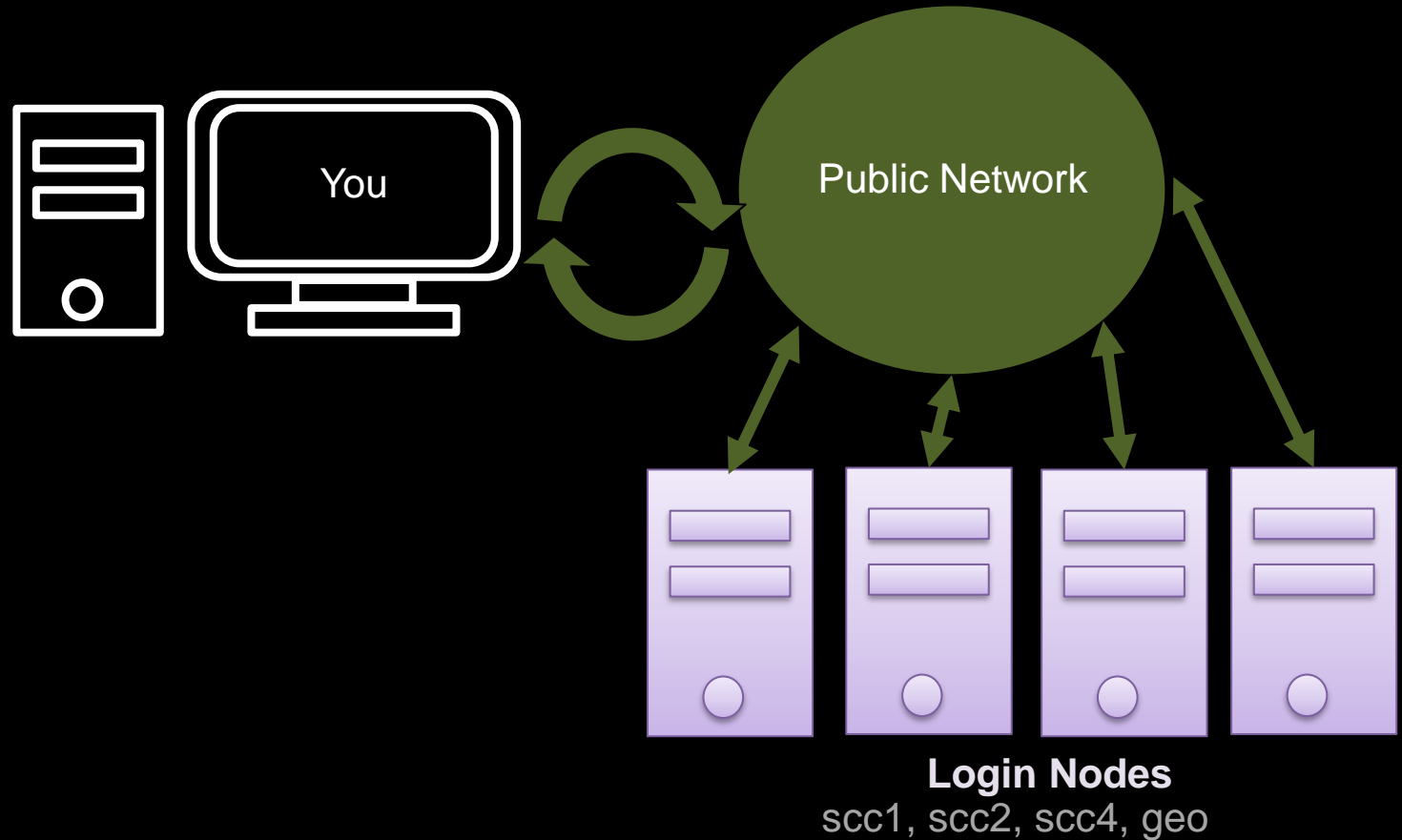
What is the SCC?



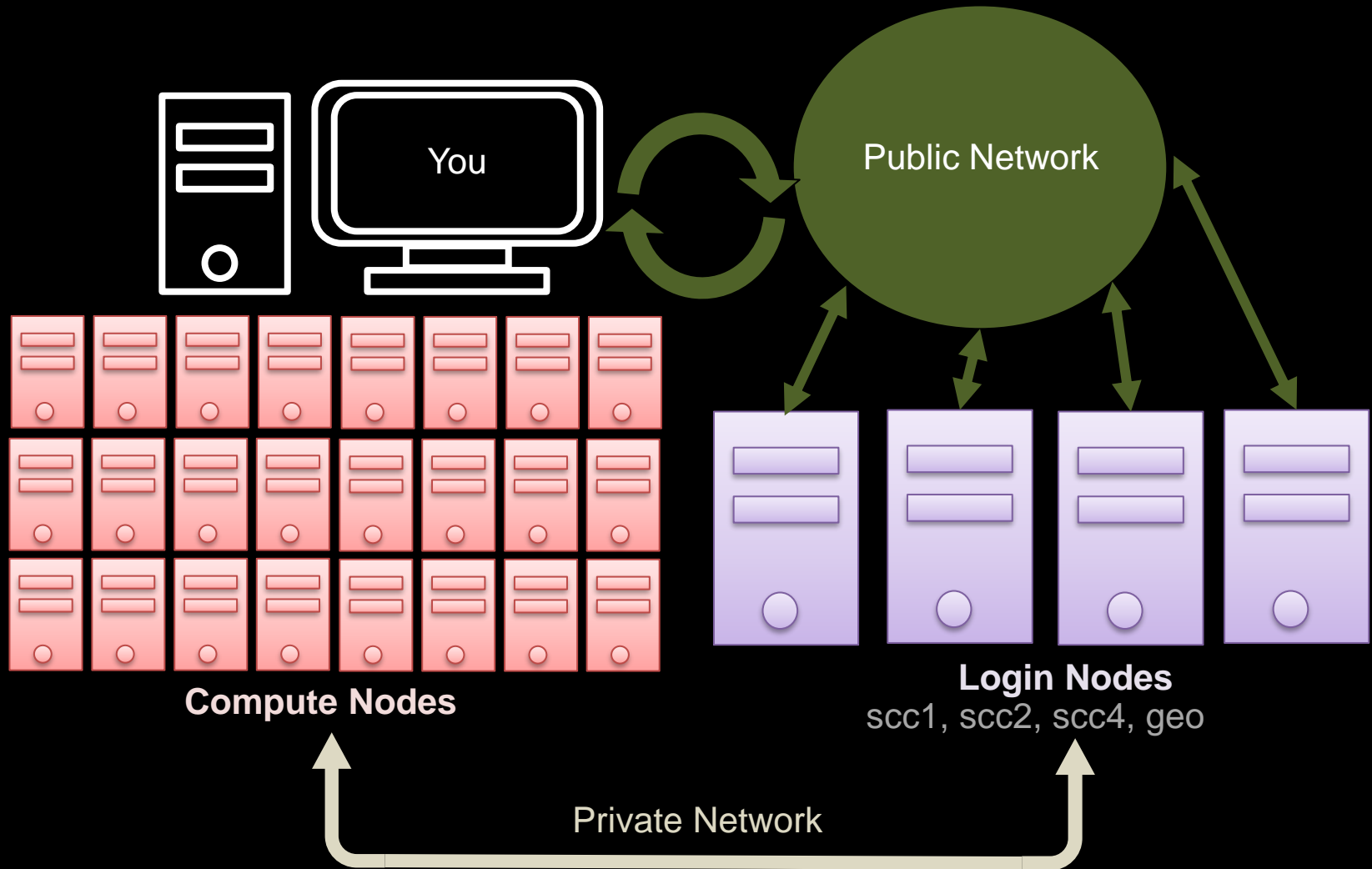
How to Connect to the SCC?



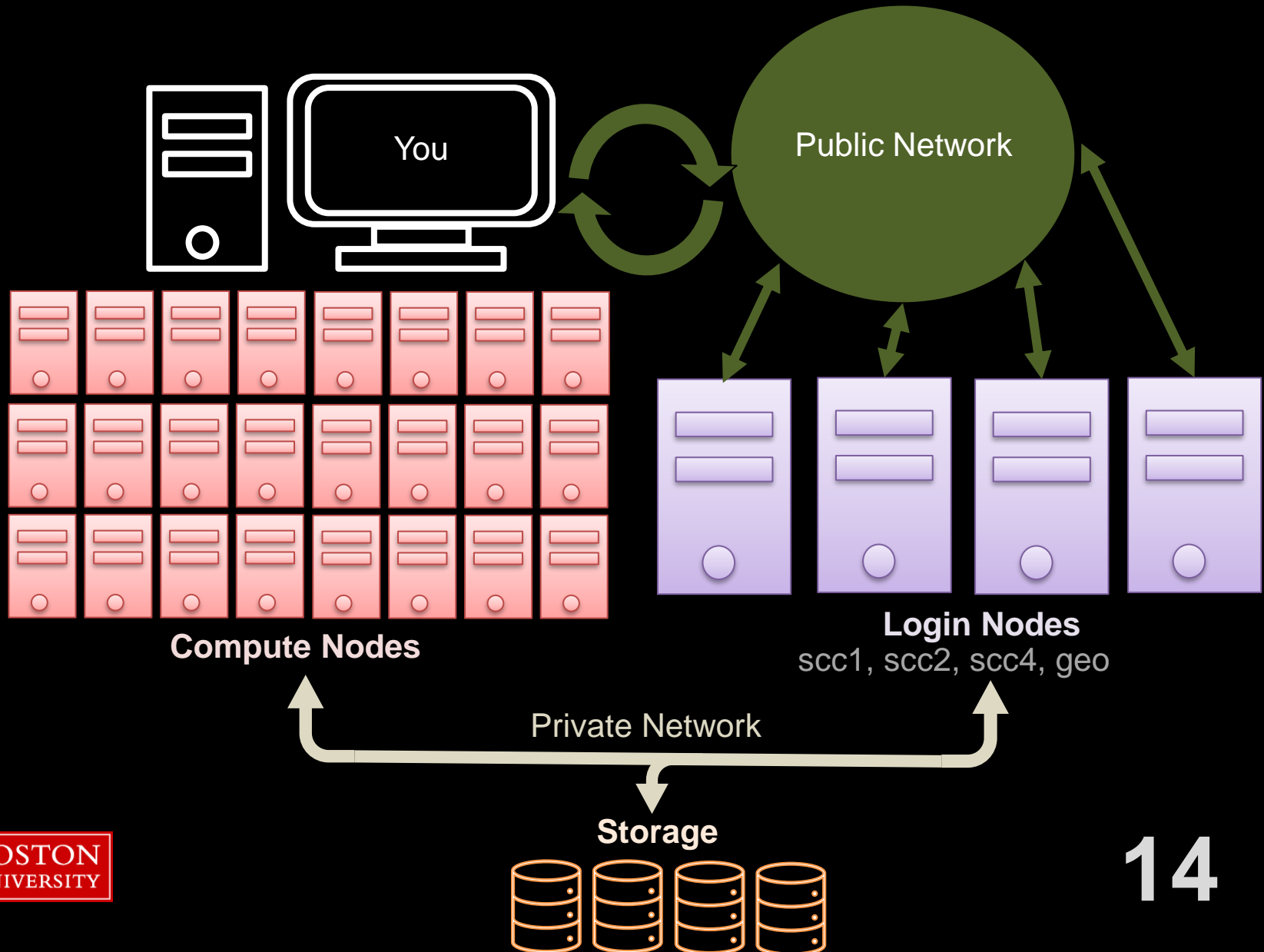
How to Connect to the SCC?



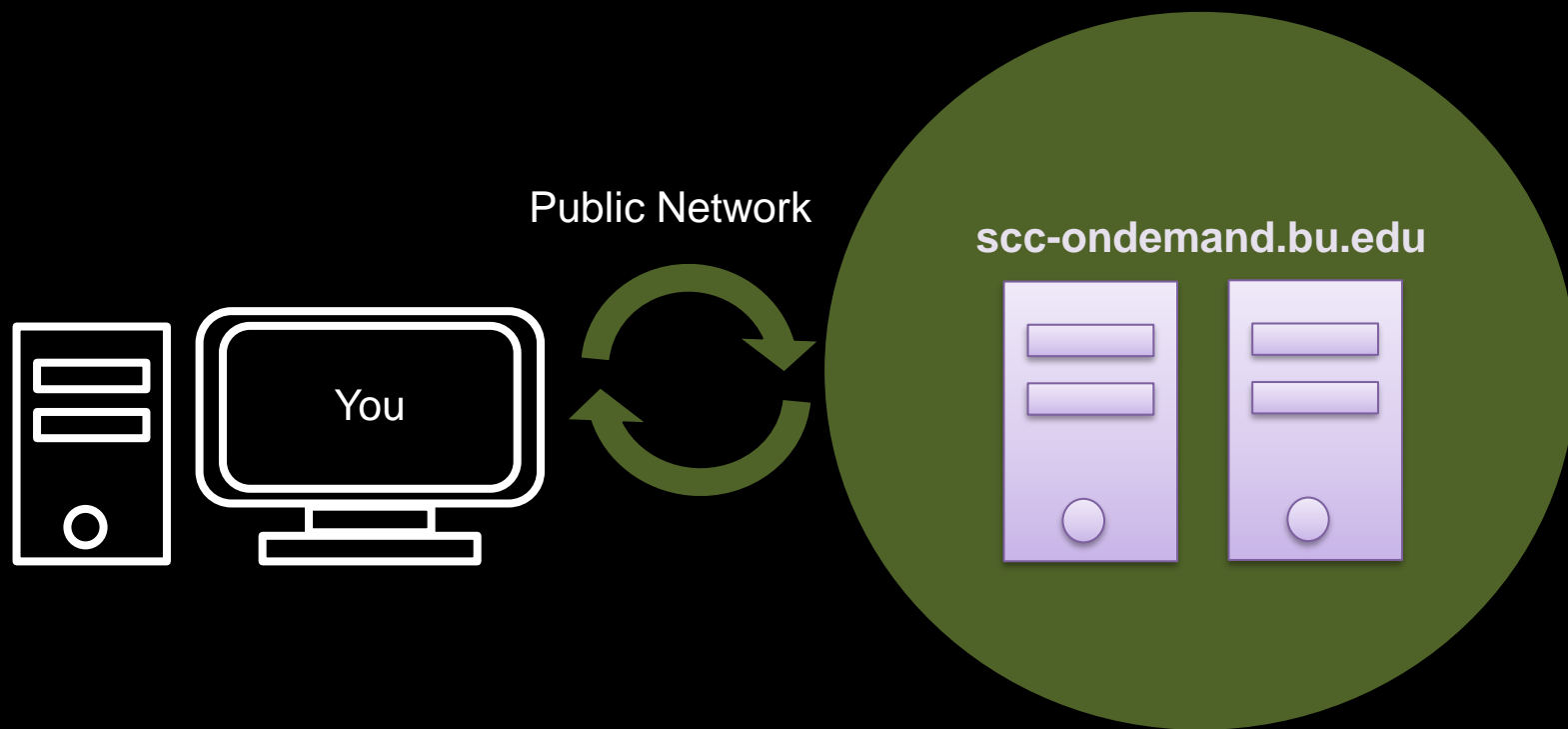
How to Connect to the SCC?



How to Connect to the SCC?



How to Connect to the SCC?



Outline

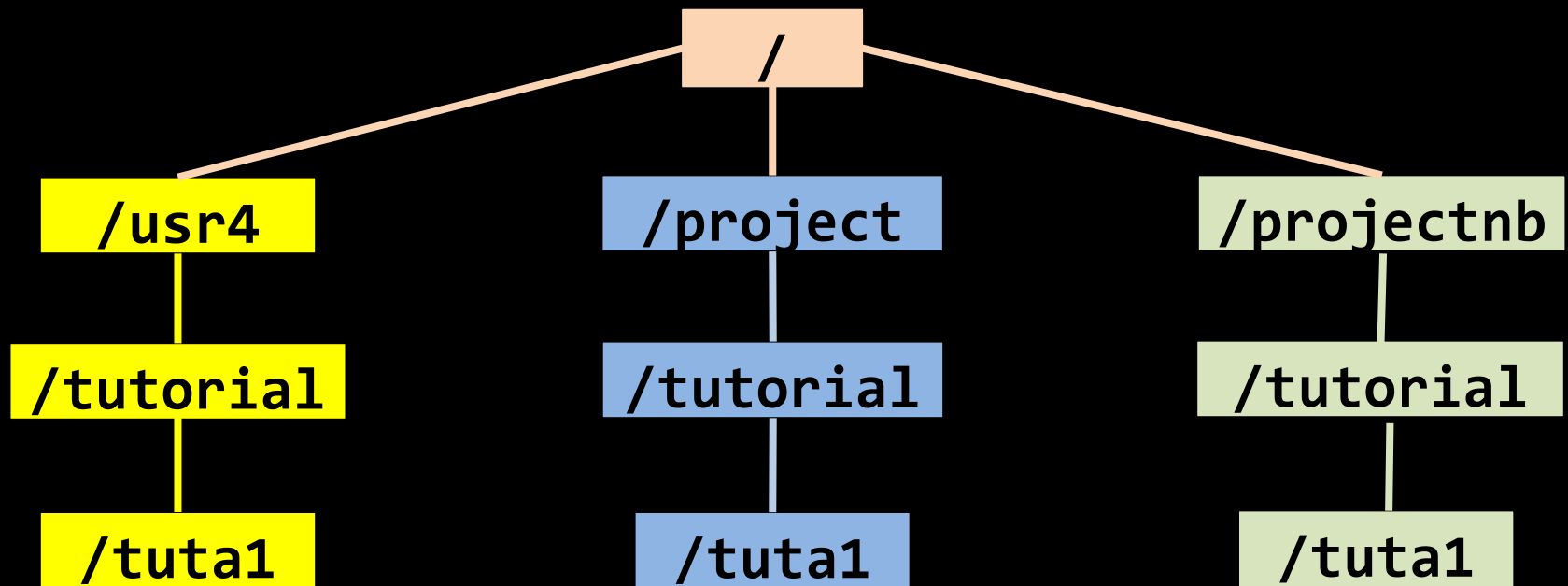
- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

Unix

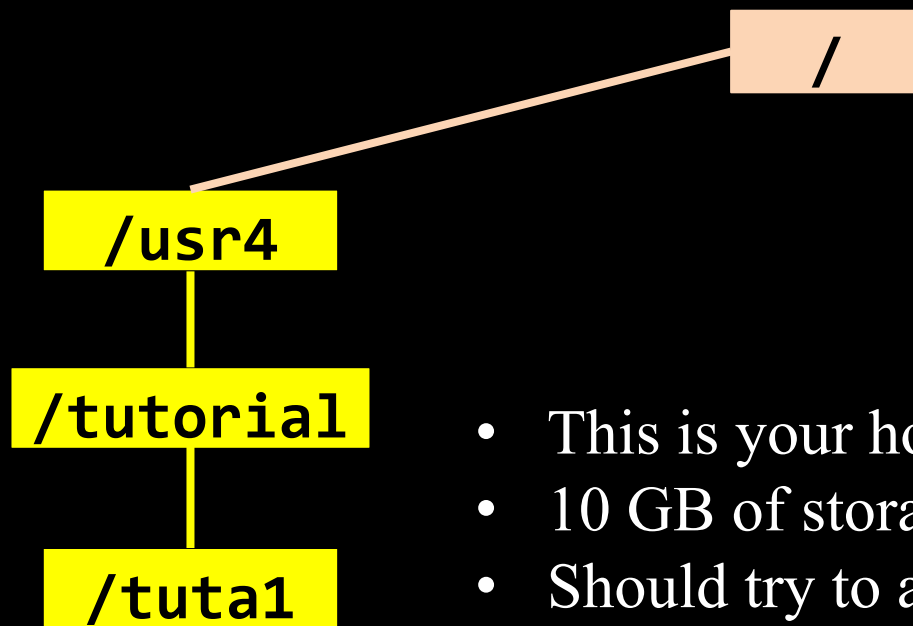
```
[tuta1@scc-v01 ~]$ pwd  
/usr4/tutorial/tuta1
```

Shows the directory you are in, with the full path

Unix

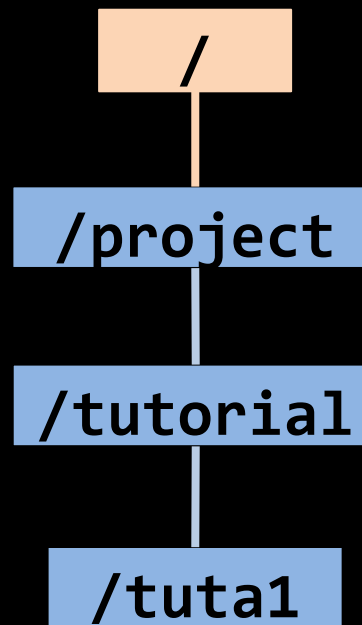


Unix



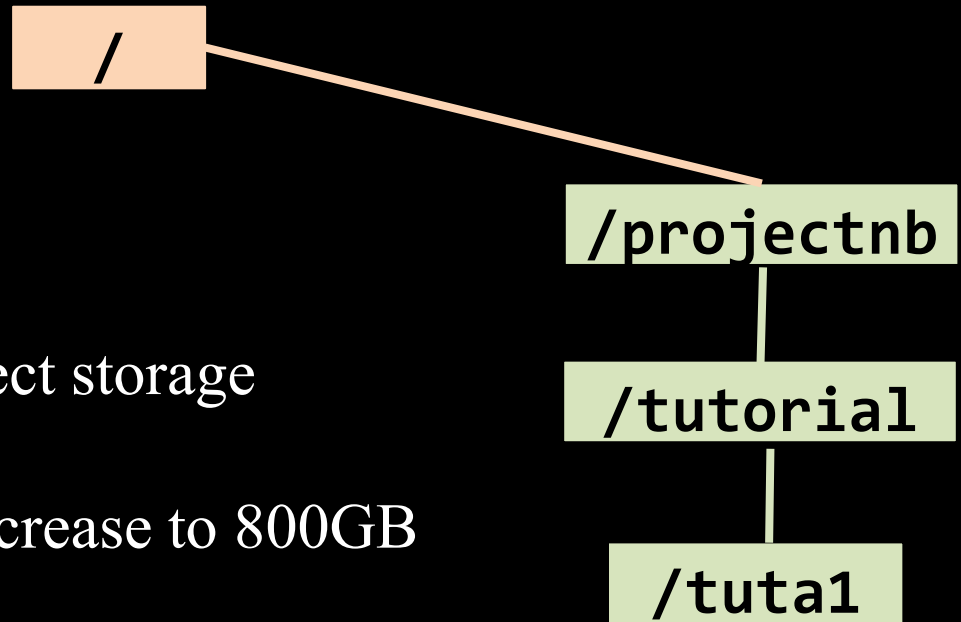
- This is your home directory
- 10 GB of storage
- Should try to avoid filling up this directory

Unix



- This is a backed-up project storage
- Backup in a separate location incase of physical damage
- Default 50GB of space, can increase to 200GB

Unix



- This is a NOT backed-up project storage
- Local snapshots
- Default 50GB of space, can increase to 800GB

Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

Unix

```
[tuta1@scc-v01 ~]$ cd /projectnb/ne742
```

cd = change directory

Unix

```
[tuta1@scc-v01 ~]$ mkdir mhorn
```



swap in your username

mkdir = make a directory

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```

ls = list contents

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn ne742 4096 Oct 28 16:03 mhorn
```

↑
owner

↑
group

↑
file/folder name

ls = list contents

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

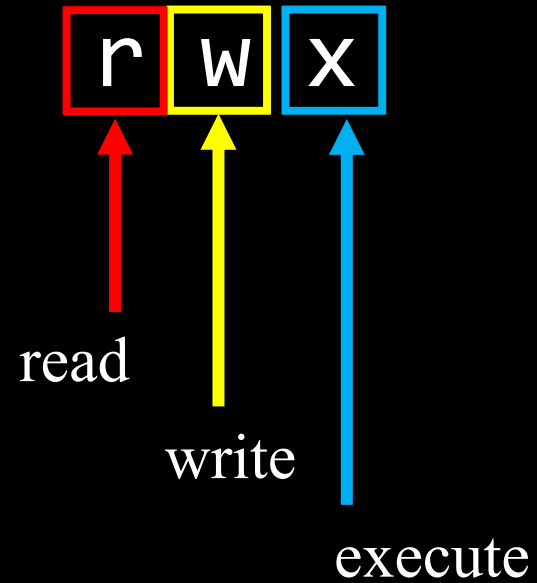
```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```

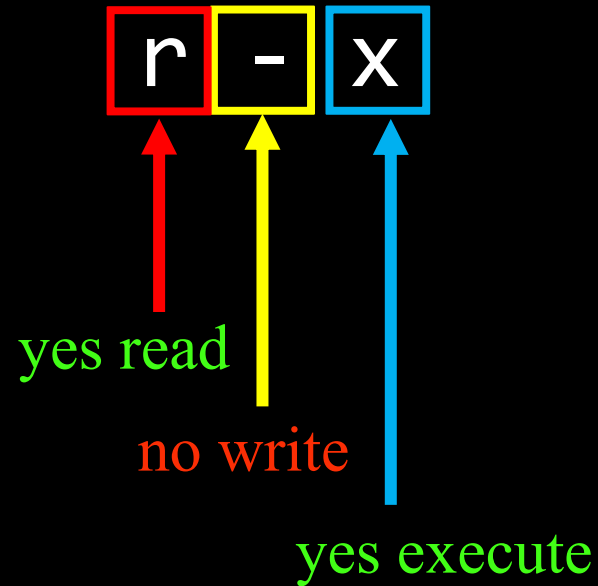
↑
permissions

ls = list contents

Unix



Unix



Unix

`drwxr-xr-x`

The diagram shows the Unix permission string `drwxr-xr-x` with three colored boxes highlighting different parts: a red box around `drwx`, a yellow box around `r-x`, and a blue box around `r-x`. Below each box is an arrow pointing to the text labels `user`, `group`, and `other` respectively.

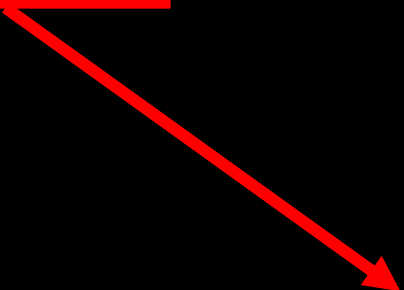
user group other

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```



drwxr-xr-x

Object Type

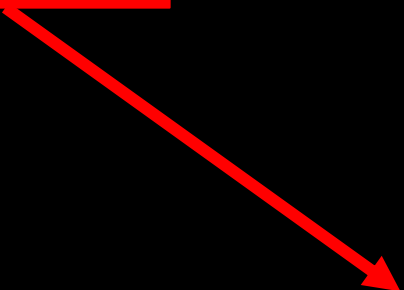
- d = directory
- - = file

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```



drwxr-xr-x

User/Owner Permissions:

- mhorn is owner
- currently all enabled

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn ne742 4096 Oct 28 16:03 mhorn
```



drwxr-xr-x

Group Permissions:

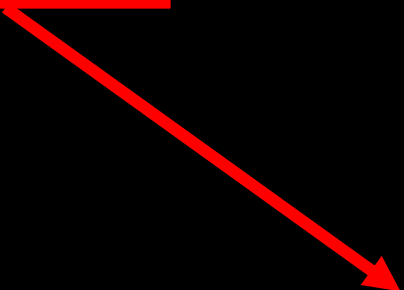
- ne742 is group
- read and execute are enabled
- write is not enabled

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```



drwxr-xr-x

Other/World Permissions:

- read and execute are enabled
- write is not enabled

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxr-xr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```



How can we change permissions?

Unix

```
[tuta1@scc-v01 ~]$ chmod g+w mhorn
```



To group add write permissions

Unix

```
[tuta1@scc-v01 ~]$ ls -l
```

```
total 1
```

```
drwxrwxr-x 3 mhorn  ne742 4096 Oct 28 16:03 mhorn
```



yes write

Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

Software on the SCC

```
[tuta1@scc-v01 tuta1]$ module avail | less
```

ctrl+z to quit “less”

Software on the SCC

```
[tuta1@scc-v01 tuta1]$ module avail dcm2niix
```

ctrl+z to quit “less”

Software on the SCC

```
[tuta1@scc-v01 tuta1]$ module load dcm2nix
```

Module system to load software packages

Software on the SCC

```
[tuta1@scc-v01 tuta1]$ module list
```

Currently Loaded Modules:

```
1) dcm2niix/1.0.20220720
```

Module system to load software packages

Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

Batch Jobs

- Up to this point, we've been doing things *Interactively*
 - request resources
 - connect to the remote session
 - program via interface (terminal &/or GUI's)

- Can process *Non-Interactively*
 - easier to scale
 - flexible - submit and forget
 - more efficient resource management
 - can submit large number (batches) of jobs

Batch Jobs

- Submit from command line

qsub *[options]* **command** *[arguments]*

- Submit with a jobscript file:

qsub *script.qsub*

Batch Jobs

■ qsub options

General Directives		Directives to request SCC resources	
Directive	Description	Directive	Description
-l h_rt=hh:mm:ss	Hard run time limit in <i>hh:mm:ss</i> format. The d	-l h_rt=hh:mm:ss	Hard run time limit in <i>hh:mm:ss</i> format. The default is 12 hours.
-P project_name	Project to which this jobs is to be assigned. Th associated with any Med.Campus project.	-l mem_per_core=#G	Request a node that has at least this amount of memory per core. Recommended choices are: 3G, 4G, 6G, 8G, 12G, 16G, 18G and 28G
-N job_name	Specifies the job name. The default is the scrip	-pe omp N	Request multiple slots for Shared Memory applications (OpenMP, pthread). This option can also be used to reserve a larger amount of memory for the application. <i>N</i> can vary. Currently, to request multiple cores on SCC's shared nodes, we recommend to select 1-4, 8, 16, 28, or 36 cores.
-o outputfile	File name for the stdout output of the job.	-pe mpi_#_tasks_per_node N	Select multiple nodes for an MPI job. Number of tasks can be 4, 8, 12, 16, or 28 and <i>N</i> must be a multiple of this value. See Running Parallel Batch Jobs for more information.
-e errfile	File name for the stderr output of the job.	-t N	Submit an Array Job with <i>N</i> tasks. <i>N</i> can be up to 75,000. For more information see Array Jobs
-j y	Merge the error and output stream files into a s	-l cpu_arch=ARCH	Select a processor architecture (broadwell, ivybridge, cascadelake...). See Technical Summary for all available choices.
-m b e a s n	Controls when the batch system sends email t job begins (b), ends (e), is aborted (a), is susp	-l cpu_type=TYPE	Select a processor type (X5670, X5675, Gold-6132 etc.) See Technical Summary for all available choices.
-M user_email	Overwrites the default email address used to s	-l gpus=G	Requests a node with GPUs. <i>G</i> is the number of GPUs. See GPU Computing for more information.
-V	All current environment variables should be exp	-l gpu_type=GPUMODEL	To see the current list of available GPU models, run <i>qgpus</i> command. See GPU Computing for more information.
-v env=value	Set the runtime environment variable <i>env</i> to <i>v</i>	-l gpu_c=CAPABILITY	Specify minimum GPU capability. Current choices for <i>CAPABILITY</i> are 3.5, 5.0, 6.0, 7.0, and 8.6
-hold_jid job_list	Setup job dependency list. <i>job_list</i> is a con names which must complete before this job ca Usage for more information.	-l gpu_memory=#G	Request a node with a GPU that has 12G, 16G, 24G, 32G, 48G of memory.
		-l avx	Request a node that supports AVX and newer CPU instructions . A small number of modules require support for these instructions.

Batch Jobs

- jobscript file

Script Interpreter

```
#!/bin/bash -l
```

```
# Time Limit
```

```
#$ -l h_rt=12:00:00
```

```
# Project Name
```

```
#$ -P ne742
```

```
# Job Name
```

```
#$ -N testjob
```

```
# Load Modules
```

```
module load python/3.10.12
```

```
# Run the program
```

```
python myscript.py
```

Scheduler Directives

Commands to execute

Batch Jobs

Directives to request SCC resources	
Directive	Description
<code>-l h_rt=hh:mm:ss</code>	Hard run time limit in <i>hh:mm:ss</i> format. The default is 12 hours.
<code>-l mem_per_core=#G</code>	Request a node that has at least this amount of memory per core. Recommended choices are: 3G, 4G, 6G, 8G, 12G, 16G, 18G and 28G
<code>-pe omp N</code>	Request multiple slots for Shared Memory applications (OpenMP, pthread). This option can also be used to reserve a larger amount of memory for the application. <i>N</i> can vary. Currently, to request multiple cores on SCC's shared nodes, we recommend to select 1-4, 8, 16, 28, or 36 cores.
<code>-pe mpi_#_tasks_per_node N</code>	Select multiple nodes for an MPI job. Number of tasks can be 4, 8, 12, 16, or 28 and <i>N</i> must be a multiple of this value. See Running Parallel Batch Jobs for more information.
<code>-t N</code>	Submit an Array Job with <i>N</i> tasks. <i>N</i> can be up to 75,000. For more information see Array Jobs
<code>-l cpu_arch=ARCH</code>	Select a processor architecture (broadwell, ivybridge, cascadelake...). See Technical Summary for all available choices.
<code>-l cpu_type=TYPE</code>	Select a processor type (X5670, X5675, Gold-6132 etc.) See Technical Summary for all available choices.
<code>-l gpus=G</code>	Requests a node with GPUs. <i>G</i> is the number of GPUs. See GPU Computing for more information.
<code>-l gpu_type=GPUMODEL</code>	To see the current list of available GPU models, run <i>qgpus</i> command. See GPU Computing for more information.
<code>-l gpu_c=CAPABILITY</code>	Specify minimum GPU capability. Current choices for <i>CAPABILITY</i> are 3.5, 5.0, 6.0, 7.0, and 8.6
<code>-l gpu_memory=#G</code>	Request a node with a GPU that has 12G, 16G, 24G, 32G, 48G of memory.
<code>-l avx</code>	Request a node that supports AVX and newer CPU instructions . A small number of modules require support for these instructions.

Batch Jobs

- **All Purpose Nodes:**
 - can run single-processor jobs and parallel jobs ($\leq 720\text{h}$)
- **Whole Node** queues (8, 16, 28, 36 cores):
 - must request whole node ($\leq 240\text{h}$)
- **GPU** nodes:
 - must request GPU(s) ($\leq 48\text{h}$)
- **MPI** queues:
 - must request multiple nodes ($\leq 120\text{h}$)
- **VirtualGL** nodes
 - for interactive graphics jobs

Batch Jobs

```
[tuta1@scc-v01 tuta1]$ qstat -u tuta1
```



swap in your username

Check status of job

Batch Jobs

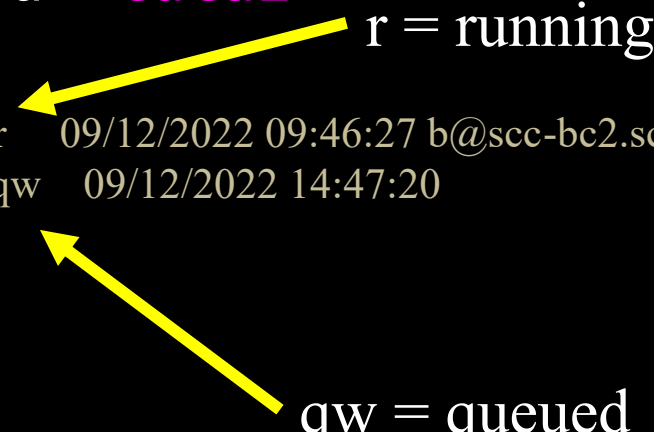
Check status of running job

```
[tuta1@scc-v01 tuta1]$ qstat -u tuta1
```

7238828	0.14071	ood-deskto	tuta1	r	09/12/2022 09:46:27	b@scc-bc2.scc.bu.edu	1
7243957	0.00000	bet.qsub	tuta1	qw	09/12/2022 14:47:20		1

r = running

qw = queued



Batch Jobs

Check report of **completed jobs**

```
[tuta1@scc-v01 tuta1]$ qacct -j 596557
```

```
qname p100
hostname scc-c11.scc.bu.edu
group ozymandias
owner mhorn
project ne742
jobname myjob
jobnumber 551947
qsub_time Wed Sep 6 20:08:56 2017
start_time Wed Sep 6 20:09:37 2017
end_time Wed Sep 6 23:32:29 2017
granted_pe NONE
slots 1
failed 0
exit_status 0
cpu 11232.780
mem 611514.460
io 14.138
iow 0.000
maxvmem 71.494G
arid undefined
```

Batch Jobs

- The job should run soon and produce an output file:

```
[tuta1@scc1 tut]$ cat bet.qsub.o1041461
```

- There will also be an error file which should be empty:

```
[tuta1@scc1 tut]$ cat bet.qsub.e1041461
```

Get Example Data

1.

```
cd /projectnb/ne742/students/$USER
```

2.

```
cp /project/scv/examples/imaging/ne742/Data.zip $PWD
```

3.

```
unzip Data.zip
```

Submit Single FMRIPrep Job

4.

```
cd Data/scripts
```

5.

```
qsub fmriprep.qsub
```

Submit Multi-FMRIPrep Job

6.

```
qsub fmriprep_m.qsub subjects.txt
```


Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- **BIDS**
- fMRIPrep

BIDS

- Brain Imaging Data Structure

Why?

- To standardize organization of complex data
- Have software develop BIDS apps to take in datasets
- Avoid lab-specific pipelines
- Facilitate validating datasets and curation
- Facilitate data sharing

BIDS

- Brain Imaging Data Structure

What?

- Three main file types in a BIDS dataset:
 - .JSON – contain metadata as key:value pairs
 - .TSV – contain tables of metadata
 - .NII.GZ – raw data files for fMRI and MRI data

BIDS

```
BIDS/
├── sourcedata
│   └── <DICOMS go here>
├── sub-101
│   └── ses-001
│       ├── anat
│       │   ├── sub-001_ses-001_run-01_T1w.json
│       │   └── sub-001_ses-001_run-01_T1w.nii.gz
│       └── func
│           ├── sub-001_ses-001_run-01_BOLD.json
│           └── sub-101_ses-001_run-01_BOLD.nii.gz
├── derivatives
│   └── fmriprep
├── README
├── dataset_description.json
└── participants.tsv
```

BIDS Compliance

- For BIDS apps to successfully recognize and import data, you can verify organization
- Not always necessary
- <http://bids-standard.github.io/bids-validator/>

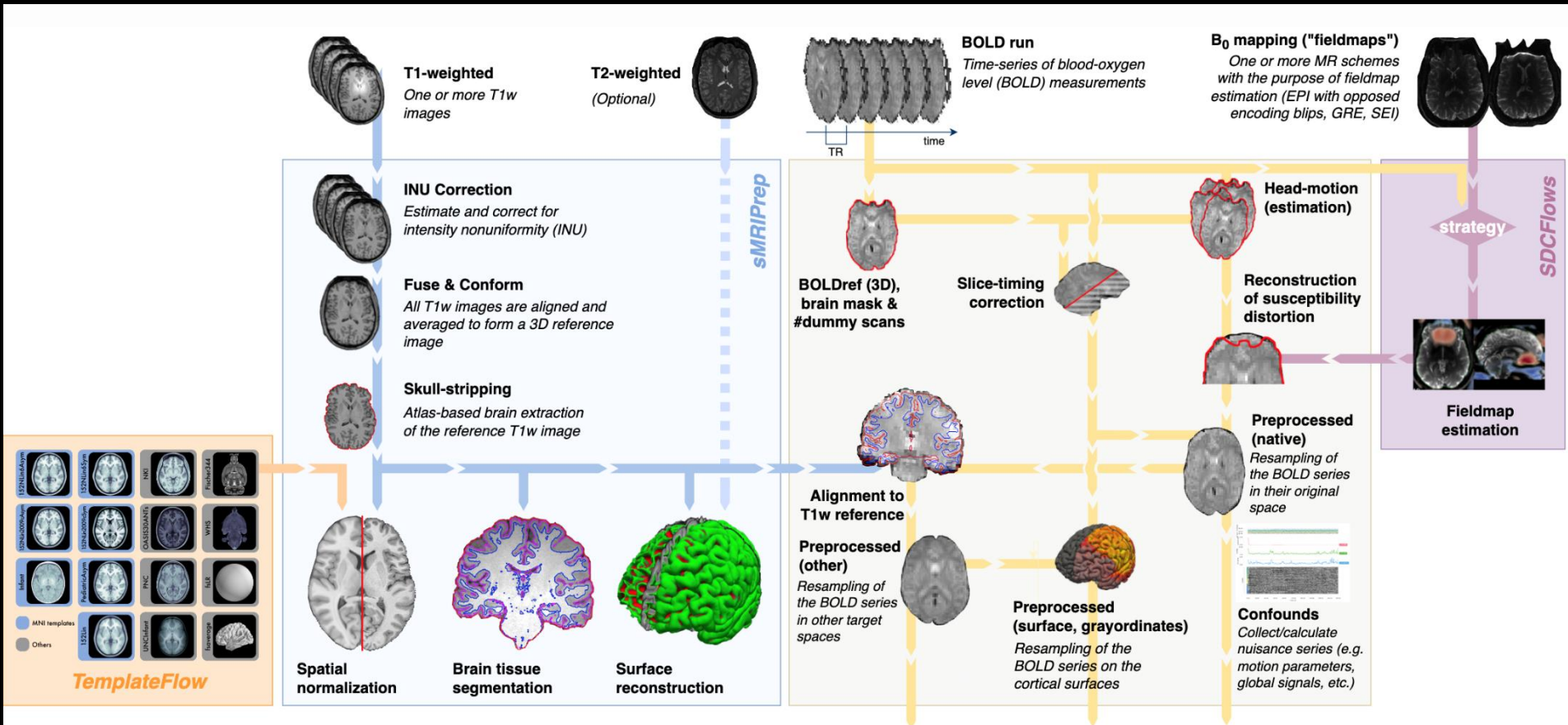
Outline

- SCC
- File Structure
- File Permissions
- Software
- Batch Jobs
- BIDS
- fMRIPrep

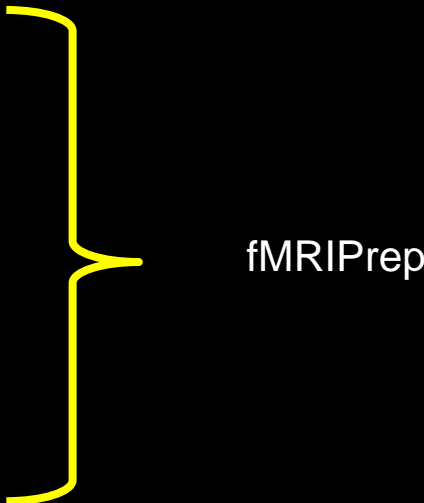
fMRIPrep

- Workflow taking principal input images and generates a standardized preprocessed output for analysis
- Uses BIDS standard for input and output
- Customize preprocessing via one command line with arguments
- Generate visual outputs for basic QC analysis

What's fMRIPrep



fMRIPrep

- Adapts multiple software packages into a single standardized process.
 - FreeSurfer
 - FSL
 - AFNI
 - ANTS
 - Connectomewb
 - ICA-AROMA
 - Python
- 
- fMRIPrep

fMRIPrep

```
fmriprep BIDS BIDS/derivatives participant [options]
```



command to run fmriprep

fMRIPrep

```
fmriprep BIDS BIDS/derivatives participant [options]
```



input directory

fMRIPrep

```
fmriprep BIDS BIDS/derivatives participant [options]
```



output directory

fMRIPrep

fmriprep BIDS BIDS/derivatives **participant** [options]



analysis level

fMRIPrep

fmriprep BIDS BIDS/derivatives participant [options]



tons of options!

fMRIPrep

■ Some Useful Options

■ Performance control

- `--nprocs` : number of cpus for entire process
- `--omp-nthreads` : number of threads per process
- `--mem` : upper bound memory limit for entire process

■ Workflow configurations

- `--output-spaces` : list spaces i.e. (MNI251Lin6Asym, MNI152Lin2009cAsym)
- `--bold2t1w-init` : DOF when registering BOLD to T1w (6=def, options 6, 9, 12)
- `--dummy-scans` : number of nonsteady-state volumes
- `--output-spaces` : choose your standardized spaces
- `-w` : define where the working directory will save output

■ Estimating Confounds options

- `--use-aroma` : add ICA_AROMA to your processing steam
- `--fd-spike-threshold` : framewise displacement threshold for flagging frame as an outlier
- `--dvars-spike-threshold` : DVARS threshold for flagging frame as an outlier

■ FreeSurfer options

- `--fs-license-file` : path to FreeSurfer license key
- `--fs-subjects-dir` : path to existing FreeSurfer directory to import
- `--fs-no-reconall` : disable FreeSurfer preprocessing

fMRIPrep

■ Some Useful Options

■ Performance control

- **--nprocs** : **number of cpus for entire process**
- **--omp-nthreads** : **number of threads per process**
- **--mem** : **upper bound memory limit for entire process**

■ Workflow configurations

- **--output-spaces** : list spaces i.e. (MNI251Lin6Asym, MNI152Lin2009cAsym)
- **--bold2t1w-init** : DOF when registering BOLD to T1w (6=def, options 6, 9, 12)
- **--dummy-scans** : number of nonsteady-state volumes
- **--output-spaces** : choose your standardized spaces
- **-w** : **define where the working directory will save output**

■ Estimating Confounds options

- **--use-aroma** : add ICA_AROMA to your processing steam
- **--fd-spike-threshold** : framewise displacement threshold for flagging frame as an outlier
- **--dvars-spike-threshold** : DVARS threshold for flagging frame as an outlier

■ FreeSurfer options

- **--fs-license-file** : **path to FreeSurfer license key**
- **--fs-subjects-dir** : path to existing FreeSurfer directory to import
- **--fs-no-reconall** : disable FreeSurfer preprocessing

fMRIPrep

```
fmriprep BIDS BIDS/derivatives participant \  
--participant-label $subjid \  
-w /scratch/$USER/fmriprep \  
--nprocs 8 \  
--omp-nthreads 2 \  
--mem 64 \  
--fs-license-file /share/pkg.8/freesurfer/7.4.1/install/license.txt
```

- - command
- - required arguments
- - option flags
- - option arguments

Additional BU RCS Web Resources

- Research Computing Support Pages
<http://www.bu.edu/tech/support/research/>
- Technical Summary of SCC Resources
<http://www.bu.edu/tech/support/research/computing-resources/tech-summary/>
- SCC Updates – Latest SCC News
<http://www.bu.edu/tech/support/research/whats-happening/updates/>
- Code Examples for Popular Software Packages
<http://scv.bu.edu/examples/>

1. Please open a web browser and:
2. Google “rcs tutorial evaluation”
3. Click 1st link
http://scv.bu.edu/survey/tutorial_evaluation.html
4. Fill out the survey!