Using Minerva (HPC) for preprocessing data using fMRIPrep

Saren H. Seeley

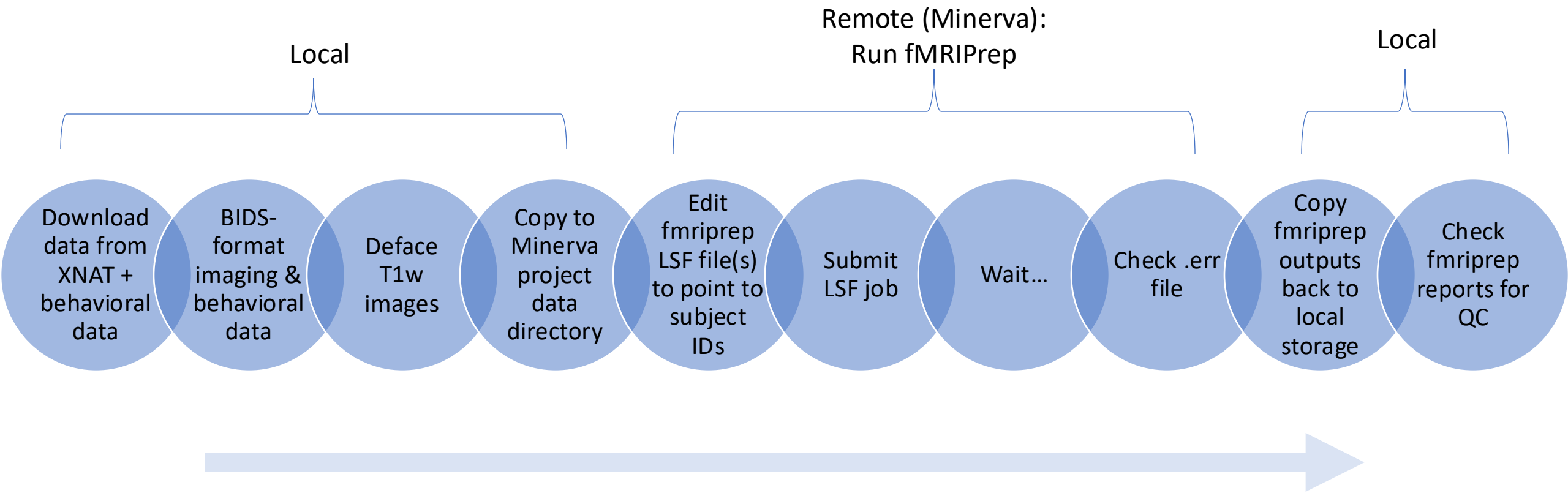
April 21, 2022

Minerva

- HPC scientific computing resource at Mount Sinai

- The supercomputer Minerva was upgraded in 2020, and utilizes 14,304 Intel Gold 8168 24C, 2.7 GHz compute cores (48 cores per node with two sockets in each node), 286 nodes with 192 GB of memory per node, 65.7 terabytes of total memory, 350 terabytes of solid-state storage and nearly 30 petabytes of spinning storage accessed via IBM's Spectrum Scale/General Parallel File System (GPFS) for a total of 1.2 petaflops of compute power.

- **See https://labs.icahn.mssm.edu/minervalab/ for Minerva documentation and training dates/slides**


Local
Remote (Minerva):
Run fMRIPrep
Local
Download data from XNAT + behavioral data
BIDS-format imaging & behavioral data
Deface T1w images
Copy to Minerva project data directory
Edit fmriprep LSF file(s) to point to subject IDs
Submit LSF job
Wait…
Check .err file
Copy fmriprep outputs back to local storage
Check fmriprep reports for QC

Workflow

Minerva cluster @ Mount Sinai

Chimera Partition:

- 3x **login nodes** - Intel 8168 24C, 2.7GHz - **384 GB** memory
- Compute nodes -
  - 275 **regular memory nodes** - Intel 8168 24C, 2.7GHz - 48 cores per node - **192 GB**/node)
  - 37 **high memory nodes** - Intel 8168/8268, 2.7/2.9GHz - **1.5 TB** mem
  - **GPU nodes:**
    - 12 -Intel 6142, 2.6GHz - 384 GB memory - 4x V100-**16 GB** GPU
    - 8 - Intel 8268, 2.9 GHz - 384 GB memory - 4x A100- **40 GB** GPU
      - 1.8 TB SSD per node
    - 2 - Intel 8358,2.6GHz - 2 TB memory - 4x A100- **80 GB** GPU
      - 7 TB SSD per node, NVlink

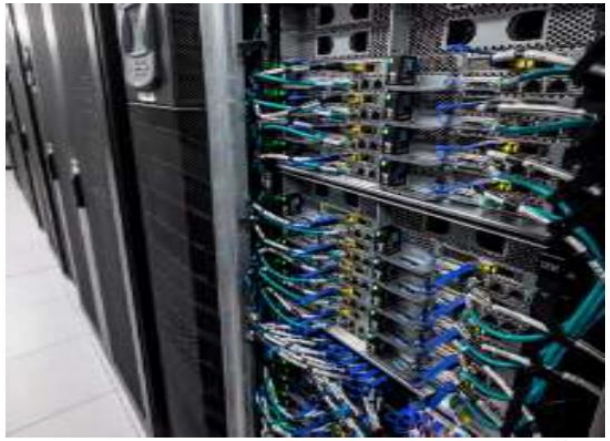
NIH FUNDED NODES

BODE2 Partition:

- $2M S10 BODE2 awarded by NIH (Kovatch PI)
- 78 compute nodes - Intel 8268, 2.9 GHz -48 cores per node - **192 GB**/node

CATS Partition:

- $2M CATS awarded by NIH (Kovatch PI)
- 55 compute nodes - Intel 8358, 2.6 GHz- 64 cores per node -**1.5 TB** / node



Storage: **32**PB of high-speed online storage as an IBM General Parallel File System (**GPFS**)

- **Path /sc/arion** : Use the system path environment variable in scripts **$GPFS**

Before you start…

- Request user account: https://acctreq.hpc.mssm.edu/
- Request project allocation ($100/TB per year): https://labs.icahn.mssm.edu/minervalab/minerva-project-allocation/
- Sign HIPAA agreement form

HIPAA

- Minerva is HIPAA compliant as of October 1st, 2020, i.e., Protected Health Information (PHI) data is allowed to be stored and processed on Minerva.
- All users have to read the HIPAA policy and complete Minerva HIPAA Agreement Form **annually** (every Dec.) at https://labs.icahn.mssm.edu/minervalab/hipaa/
- Users who have not signed the agreement will have their accounts locked until the agreement is signed.

Logging in

Minerva is a Linux machine with Centos 7.6

- Linux is command line based, not GUI
- Logging in requires campus network, **SSH client** installed on your machine, **username, memorized password,** and **one-time code** obtained from a Symantec VIP token

Detailed procedures:

- Campus network (School VPN needed if off-campus)
- Apply for an account at https://acctreq.hpc.mssm.edu/
  - Apply account for external users following here
- Complete HIPAA form at https://labs.icahn.mssm.edu/minervalab/hipaa/ to activate your account
- Register your token at the Self Service Portal **school site** (https://register4vip.**mssm**.edu/vipssp/)
- SSH client: terminal (Mac), MobaXterm/Putty (Windows)
- Logging info at https://labs.icahn.mssm.edu/minervalab/logging-in/

*Note: Minerva is school resource, so use your **school password and school portal** for register*

Logging in - Linux / Mac

Connect to Minerva via ssh

- Open a terminal window on your workstation
- ssh *your_userID*@minerva.hpc.mssm.edu
- To display graphics remotely on your screen, pass the "-X" or "-Y" flag:
  - ssh -X *your_userID*@minerva.hpc.mssm.edu
  - Mac: Install XQuartz on your mac first
  - Test by running the command: xclock
    - Should see a clock
- Landed on one of the login nodes, and at your home directory
  - Never run jobs on login nodes
  - For file management, coding, compilation, check/manage jobs etc., purposes only
  - Basic linux command: **cd, ls and** more

```
imac:~ gail01$ ssh -X gail01@minerva.hpc.mssm.edu
Please input your password and two factor token:
Password:
Last login: Mon Sep 13 16:24:06 2021 from 10.254.167.11
==========================================================
====
Run "Minerva_help" for useful Minerva commands and websites

   === Upcoming Minerva Training Sessions ===
Session 1: 15 Sep 2021, 11:00AM-12:00PM – Introduction to Minerva
Session 2: 22 Sep 2021, 11:00AM-12:00PM – LSF Job Scheduler
Session 3: 29 Sep 2021, 11:00AM-12:00PM – Globus: Data Transfer
Zoom link for all sessions:
https://mssm.zoom.us/j/5420563013


   === Send ticket to hpchelp@hpc.mssm.edu ===
WE DO NOT BACKUP USER FILES
        PLEASE ARCHIVE/BACKUP YOUR IMPORTANT FILES
   === Send ticket to hpchelp@hpc.mssm.edu ===
==================================================
gail01@li03c04: ~ $ pwd
/hpc/users/gail01
gail01@li03c04: ~ $ xclock
```

/hpc/users/seeles01/

- land here when I log in
- store my .lsf files (scripts to run jobs)
- error and output log files written here

/sc/arion/scratch/seeles01

- not a place to store anything you want to keep
- write fmriprep work directories here since they're huge (10TB limit)

/sc/arion/projects/psychres/WTC_resilience_imaging

- project directory
- BIDS dataset and Singularity images live here
- fmriprep outputs written to subdirectory
- need to manually back up

Singularity

- A container system for secure high performance computing

- Containerization is the packaging of software code with just the operating system (OS) libraries and dependencies required to run the code to create a single lightweight executable (a container) that runs consistently on any platform/cloud/machine.

  - Containerization is ideal for applications with a large number of dependencies, like fMRIPrep, which draws on modules from many software libraries (FreeSurfer, FSL, AFNI, etc.)
  - Reproducibility: preserves exact versions of software used.
  - “Singularity image” (used to be .simg, now .sif)

User Software - Singularity Container Platform

Singularity tool is supported, instead of docker (Security concern)

- Docker gives superuser privilege, thus is better at applications on VM or cloud infrastructure
- It allows you to create and run containers that package up pieces of software in a way that is portable and reproducible. Your container is a single file and can be ran on different systems

To load singularity module: *$ module load singularity/3.6.4*

To pull a singularity image: *$ singularity pull --name hello.simg shub://vsoch/hello-world*

To create a container within a writable directory (called a sandbox):

$singularity build --sandbox lolcow/ shub://GodloveD/lolcow (create container within a writable directory)

To pull a docker image: *$singularity pull **docker://**ubuntu:latest*

To shell into a singularity image: *$ singularity shell hello.simg*

To run a singularity image: *$ singularity run hello.simg*

To get a shell with a specified dir mounted in the image

$ singularity run -B /user/specified/dir hello.simg

Note: /tmp, user home directory, and /sc/arion/is automatically mounted into the singularity image.

- fMRIPrep already published as a Docker image

- After logging into Minerva:

```
$ ml singularity
$ singularity pull docker://nipreps/fmriprep:latest
```

*or specific version

- Make sure you have Templateflow templates installed: https://www.nipreps.org/apps/singularity/#templateflow-and-singularity
  - Issue for HPC: fmriprep tries to download templates from server but no internet access on Minerva

Minerva resources requested

[Templateflow](#) = templates for spatial alignment/norm

binding directories to container (-B <dir>)

[fmriprep usage arguments](#)

```bash
#!/bin/bash
#BSUB -J sub-015                    #job name
#BSUB -P acc_federa03a              #project allocation number
#BSUB -q express                    #queue to run on
#BSUB -n 1                          #1 job slot
#BSUB -R affinity[core(16)]         #with 16 cores
#BSUB -R rusage[mem=120000]         #with 120GB memory dynamically allocated across cores
#BSUB -a openmp                     #use openmp multithreading
#BSUB -W 12:00                      #wall time
#BSUB -o %J.out                     #output file to t.out
#BSUB -e %J.err                     #error file to t.err

module purge
module load singularity

export SINGULARITYENV_TEMPLATEFLOW_HOME=/hpc/users/seeles01/.cache/templateflow
export OMP_NUM_THREADS=2

singularity run --cleanenv \
            -B /sc/arion/projects/psychres/WTC_resilience_imaging/data:/data \
            -B /hpc/users/seeles01/.cache/templateflow:/opt/templateflow \
            /sc/arion/projects/psychres/my_images/fmriprep-21.0.1.sif \
            /data/BIDS_data /data/BIDS_data/derivatives/fmriprep \
            --work /sc/arion/scratch/seeles01 \
            participant --participant-label 015 \
            --fs-license-file /hpc/users/seeles01/minerva/license.txt \
            --skip_bids_validation \
            --n_cpus 16 --omp-nthreads 2 --mem-mb 120000 \
            -vv --notrack
```

collect_fmriprep_WTC.py | tedana_singularity.lsf | batch_process_tedana.sh

```
#       mask=compute_epi_mask(sub+'_task-'+task+'_dir-AP_run-0'+run+'_echo-1_space-native_desc-partialPre
#       mask.to_filename(sub+'_task-'+task+'_dir-AP_run-0'+run+'_echo-1_space-native_desc-mask.nii.gz');"


#process partially preproc (fmriprep) data using tedana ME-ICA:
#(no_gscontrol, low kdaw, and removal of a few initial volumes should help wih occasional convergence iss
tedana -d \
$sub"_task-"$task"_dir-AP_run-0"$r"_echo-1_space-native_desc-partialPreproc_bold.nii.gz" \
$sub"_task-"$task"_dir-AP_run-0"$r"_echo-2_space-native_desc-partialPreproc_bold.nii.gz" \
$sub"_task-"$task"_dir-AP_run-0"$r"_echo-3_space-native_desc-partialPreproc_bold.nii.gz" \
$sub"_task-"$task"_dir-AP_run-0"$r"_echo-4_space-native_desc-partialPreproc_bold.nii.gz" \
-e 11.0 29.7 48.4 67.1 --tedpca "kundu-stabilize" \
--mask epi1_mask.nii.gz \
--out-dir $sub_outdir --png \
--debug

#rename relevant output files so they're not overwritten:
#log file
mv $sub_outdir"/"report.txt $sub_outdir"/"$sub"_task-"$task"_run-0"$r"_report.txt"
#denoised optimally combined time series (for future analysis)
mv $sub_outdir"/"dn_ts_OC.nii $sub_outdir"/"$sub"_task-"$task"_run-0"$r"_space-native_desc-tedanaDenoised
```

Can also call your own scripts to run other processes

_WTC.py | tedana_singularity.lsf

```
Cresil_tedana          #job name
cc_federa03a     #project allocatio
xpress           #queue to run on
                 #1 job slot:
ffinity[core(16)] #with 16 cores
#BSUB -R rusage[mem=120000]     #with 120GB
#BSUB -W 06:00       #wall time
#BSUB -o %J.out                 #output file
#BSUB -e %J.err                 #error file

module load python/3.7.3

bash batch_process_tedana.sh sub-051
```

TemplateFlow issue fix:

From https://github.com/nipreps/fmriprep/issues/2717#issuecomment-1047238017

1. Log into Minerva
2. Run the following:

$ **wget https://raw.githubusercontent.com/nipreps/fmriprep/master/scripts/fetch_templates.py**

$ **python fetch_templates.py –h** *(to see script usage instructions)*

$ **python fetch_templates.py** *(to pull the templates)*

Then make sure the lines circled in green are in your LSF file:

1. changes the definition of the environment variable SINGULARITYENV_TEMPLATEFLOW_HOME to the path where you downloaded the templates

2. binds that location in your fMRIPrep container (so fMRIPrep can access it)

```
#!/bin/bash
#BSUB -J sub-015                    #job name
#BSUB -P acc_federa03a              #project allocation number
#BSUB -q express                    #queue to run on
#BSUB -n 1                          #1 job slot
#BSUB -R affinity[core(16)]         #with 16 cores
#BSUB -R rusage[mem=120000]         #with 120GB memory dynamically allocated across cores
#BSUB -a openmp                     #use openmp multithreading
#BSUB -W 12:00                      #wall time
#BSUB -o %J.out                     #output file to t.out
#BSUB -e %J.err                     #error file to t.err

module purge
module load singularity

export SINGULARITYENV_TEMPLATEFLOW_HOME=/hpc/users/seeles01/.cache/templateflow
export OMP_NUM_THREADS=2

singularity run --cleanenv \
            -B /sc/arion/projects/psychres/WTC_resilience_imaging/data:/data \
            -B /hpc/users/seeles01/.cache/templateflow:/opt/templateflow \
            /sc/arion/projects/psychres/my_images/fmriprep-21.0.1.sif \
            /data/BIDS_data /data/BIDS_data/derivatives/fmriprep \
            --work /sc/arion/scratch/seeles01 \
            participant --participant-label 015 \
            --fs-license-file /hpc/users/seeles01/minerva/license.txt \
            --skip_bids_validation \
            --n_cpus 16 --omp-nthreads 2 --mem-mb 120000 \
            -vv --notrack
```

- Submit a job:

  $ bsub < run_fmriprep_21.0.1_048.lsf

  - *NB: There is a better way to submit multiple jobs as a batch, but I haven't spent time to figure that out yet, see https://labs.icahn.mssm.edu/minervalab/resources/the-minerva-user-group-and-training-classes/*

- Get list of jobs you have running for a specific project account:

  $ bjobs -P acc_federa03a

- Kill a job you started:

  $ bkill <jobid>

LSF: Queue structure (bqueues)

Queue structure in Minerva		
Queue	**Wall time limit**	**available resources**
interactive (Dedicated to interactive jobs)	12 hours	4 nodes+2 V100 GPU nodes
premium	6 days	275 nodes + 37 himem nodes+BODE2+CATS
express	12 hours	275 nodes + 4 dedicated nodes (may change)+BODE2+CATS
long	2 weeks	6 dedicated (288 cores) + 12 BODE2
gpu	6 days	40 V100 32 A100 8 A100-80GB
private	unlimited	private nodes

***default memory : 3,000MB / per core**

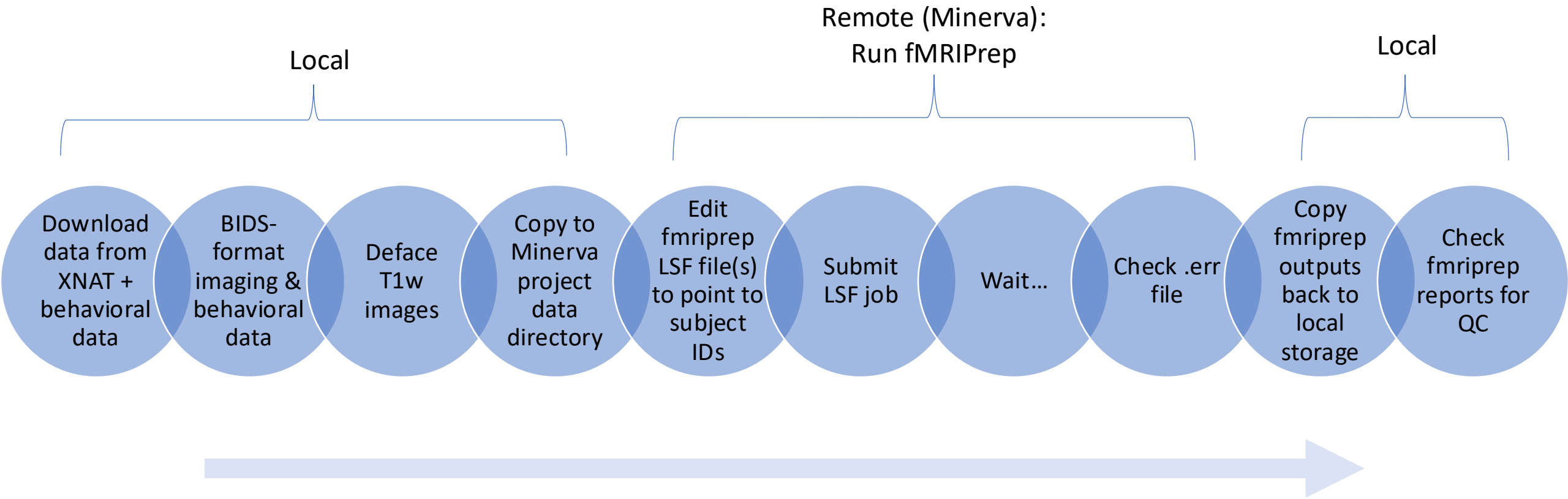
Transferring data to/from HPC

- Globus recommended: https://labs.icahn.mssm.edu/minervalab/documentation/services/globus-high-assurance-hipaa-file-manager/

- rsync also works (defaced images):

```
$ rsync -rv --<args> <from> <to>
```

```
rsync -rv --ignore-existing
seeles01+vkrb@chimera.hpc.mssm.edu:/sc/arion/projects/psychres/WTC_resilience_imaging/data
/BIDS_data/derivatives/fmriprep/
/Volumes/External/WTC_resilience_imaging/data/BIDS_data/derivatives/fmriprep
```



Next steps:

- *Finish report QC & flag subs/runs to drop*
- *TEDANA denoising (vs. regressors?)*
- *Fix subjects w/fmriprep errors & re-run*

Keeping track of fMRIPrep reports QC after inspecting them – example:

E35 | *fx* mask issue z=-6, y=16

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1			fmriprep	anat	IFT1	IFT2	IFT3	IFT4	RMET1	RMET2	FS1	FS2	rest	top confounds >.7 w/GS	
28	NW	sub-027	done												
29	NW	sub-028	done												
30	NW	sub-029	done												
31	NW	sub-030	done												
32	NW	sub-031	done												
33	NW	sub-032	done												
34	NW	sub-033	done												
35	NW	sub-034	done	ok	mask issue z	mask issue z	mask issue z	mask issue z	mask issue z	ok	ok	mask issue z	mask issue z=-7	tcompcor, csf_wm, white_matter	
36	NW	sub-035	done	ok	2mm max FI	3mm max FI	2.6 max FD,	ok	ok	2.2 max FD,	2.1mm max	2.3max FD, r	FD shows physi	tcompcor, csf, csf_wm, white_matter	
37	NW	sub-036	done	ok	2.5mm max	2.2mm max	ok	ok	ok	ok	ok	ok	>3.5mm FD spik	tcompcor, csf, csf_wm, white_matter	
38	NW	sub-037	done	ok	ok	mask issue y	ok	ok	ok	ok	<2mm FD bu	ok	ok	tcompcor, csf, csf_wm, white_matter	
39	NW	sub-038	done	ok	>3mm max F	ok	ok	ok	ok	ok	6mm max FI	5mm max FI	4mm max FD, l	tcompcor, csf, csf_wm, white_matter	
40	NW	sub-039	done	ok	ok	ok	ok	ok	ok	mask issue y	mask issue y	mask issue y	ok	tcompcor, csf, csf_wm, white_matter	
41	NW	sub-040	done	ok	>3mm max F	ok	ok	ok	>2mm max F	ok	bad mask, SI	ok	bad mask, SDC	tcompcor, csf, c_comp_cor_01	

```
[seeles01@li03c04 minerva]$ cat 60451070.err
Traceback (most recent call last):
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/cli/run.py", line 114, in main
    fmriprep_wf.run(**config.nipype.get_plugin())
  File "/opt/conda/lib/python3.8/site-packages/nipype/pipeline/engine/workflows.py", line 638, in run
    runner.run(execgraph, updatehash=updatehash, config=self.config)
  File "/opt/conda/lib/python3.8/site-packages/nipype/pipeline/plugins/base.py", line 192, in run
    report_nodes_not_run(notrun)
  File "/opt/conda/lib/python3.8/site-packages/nipype/pipeline/plugins/tools.py", line 96, in report_nodes_not_run
    raise RuntimeError(
RuntimeError: Workflow did not execute cleanly. Check log for details

During handling of the above exception, another exception occurred:

Traceback (most recent call last):
  File "/opt/conda/bin/fmriprep", line 8, in <module>
    sys.exit(main())
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/cli/run.py", line 175, in main
    failed_reports = generate_reports(
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/reports/core.py", line 105, in generate_reports
    report_errors = [
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/reports/core.py", line 106, in <listcomp>
    run_reports(
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/reports/core.py", line 88, in run_reports
    return Report(
  File "/opt/conda/lib/python3.8/site-packages/niworkflows/reports/core.py", line 291, in __init__
    self._load_config(Path(config or pkgrf("niworkflows", "reports/default.yml")))
  File "/opt/conda/lib/python3.8/site-packages/fmriprep/reports/core.py", line 47, in _load_config
    self.index(settings["sections"])
  File "/opt/conda/lib/python3.8/site-packages/niworkflows/reports/core.py", line 378, in index
    self.errors = [read_crashfile(str(f)) for f in error_dir.glob("crash*.*")]
  File "/opt/conda/lib/python3.8/site-packages/niworkflows/reports/core.py", line 378, in <listcomp>
    self.errors = [read_crashfile(str(f)) for f in error_dir.glob("crash*.*")]
  File "/opt/conda/lib/python3.8/site-packages/niworkflows/utils/misc.py", line 138, in read_crashfile
    return _read_txt(path)
  File "/opt/conda/lib/python3.8/site-packages/niworkflows/utils/misc.py", line 195, in _read_txt
    cur_key, cur_val = tuple(line.split(" = ", 1))
ValueError: not enough values to unpack (expected 2, got 1)
[seeles01@li03c04 minerva]$
```

← Example of .err file contents

Troubleshooting tips:

- Try googling “fmriprep <error message, e.g. *not enough values to unpack*>”
- Search posts on Neurostars forum (neurostars.com) and fMRIPrep Github issues forum
- Look at the last line, but also see if you can find exactly where fMRIPrep crashed (scroll up, look earlier in log)