

Strategies for Processing, Distributing, and Annotating Large Neuroimaging Datasets for Open Science

Brent McPherson, PhD
McGill University, ORIGAMI Lab
2023-08-26



Topics

- What I Work On
 - UK Biobank
 - Parkinson's Disease
- Processing
 - Apptainer (Singularity)
 - Writable Overlays (.ext3)
- Distributing
 - Locally - Read-only archives (.squashfs)
 - NiPoppy - Sharing the structure and analyses
- Annotating
 - Standard organization schemes - nipoppy
 - Neurobagel - Enable federated search



What I Work On

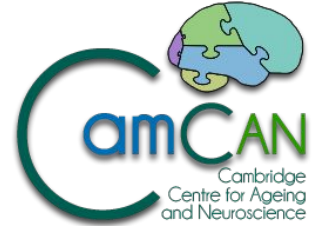
Analyzing Population Samples to Understand Aging

- My PhD was modeling Alzheimer's Disease with diffusion MRI (dMRI) and connectomics.



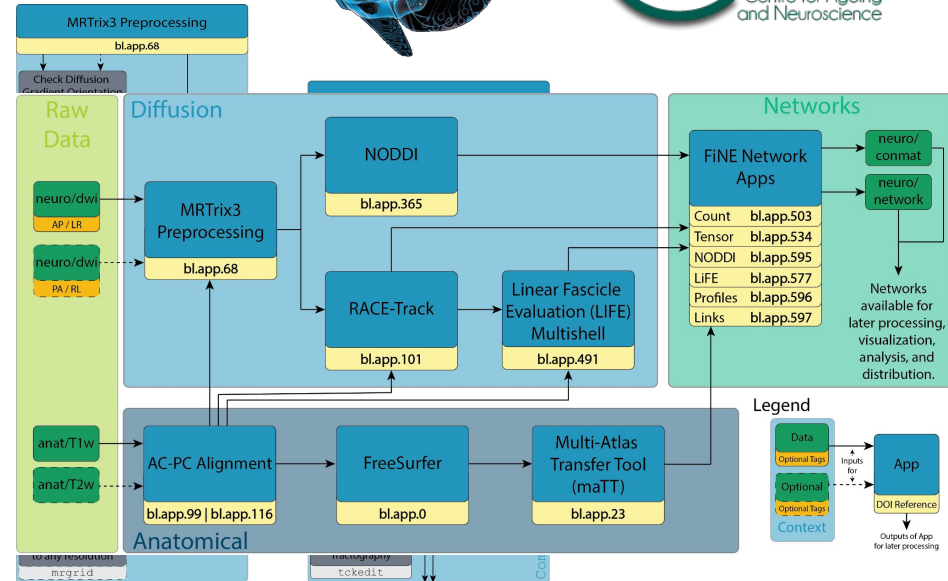
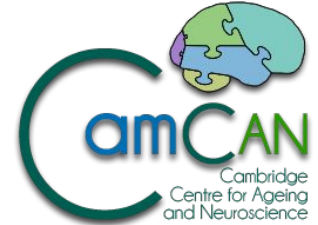
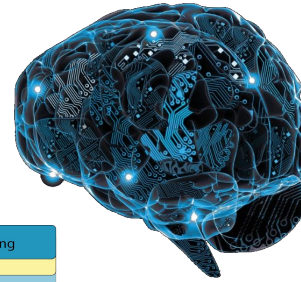
Analyzing Population Samples to Understand Aging

- My PhD was modeling Alzheimer's Disease with diffusion MRI (dMRI) and connectomics.
- However, the best sample I found to work with was the CamCAN dataset.



Analyzing Population Samples to Understand Aging

- My PhD was modeling Alzheimer's Disease with diffusion MRI (dMRI) and connectomics.
- However, the best sample I found to work with was the CamCAN dataset.
- I developed a pipeline on brainlife.io to go from converted .nii.gz data to connectomes and applied it to CamCAN.



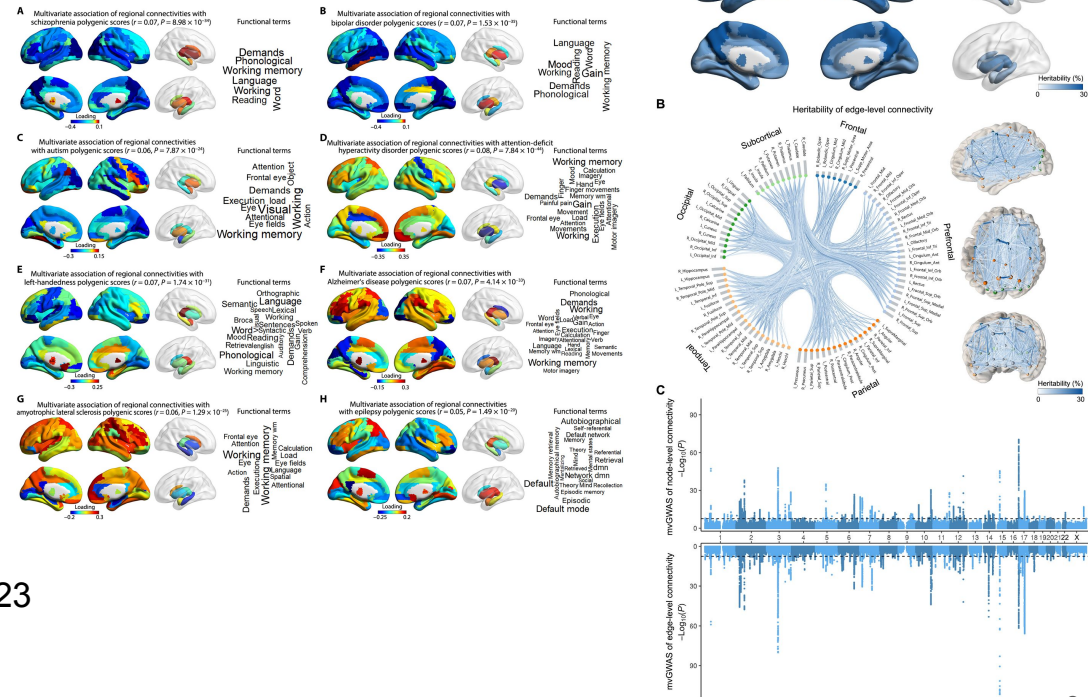
Analyzing Population Samples to Understand Aging

- At McGill I've continued working on high-throughput processing analyzing the UK Biobank.
- I am also applying the same pipelines to multiple datasets studying Parkinson's Disease (PD).
- I'm working on incorporating the longitudinal neuroimaging observations with behavior and genetics to identify biomarkers of PD and other dementias.

The UK Biobank

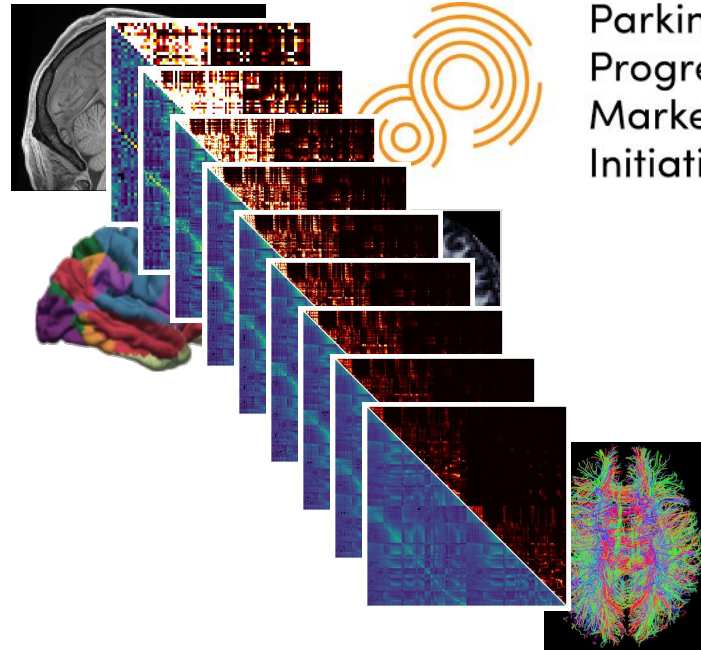


- Following 500,000 individuals
- 30 years of follow up
- Full health history
- Neuroimaging - 48k / 100k
 - ~4,500 have 2 time points
- Genetics - 488k
- Genetics of Imaging Derived Phenotypes
 - Gong et al. 2021; Zabad et al., 2023



Parkinson's Disease

- The Quebec Parkinson's Network (QPN)
- The Progressive Parkinson's Measurement Initiative (PPMI)
- NIMHANS (Bangalore)
- Producing common preprocessing and derivatives for T1w and dMRI.



Parkinson's
Progression
Markers
Initiative



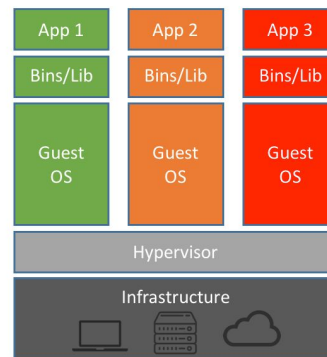
Common Challenges Across Datasets

- Prepare data for analysis (and re-analysis).
- Create useful and replicable derivatives for distribution.
- Reliably distribute and sharing both the tools and data.
- Safely sharing access to the data and derivatives.

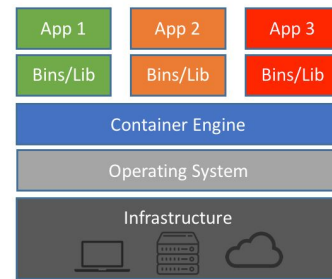
Processing

Containerization - `apptainer` / `singularity`

- The key to modern scientific computing.
- Deploy stable and versioned analyses across any HPC system.
- This is a valuable way to distribute and replicate compute environments.
- But it can also help curate your data handling.



Machine Virtualization



Containers

Binding Data to the Environment

- `apptainer` will bind or mount data within the container.
 - The mount option is more configurable bind
- Binds (`-B` or `--bind`)
 - Let you assign a **path on your system** to a **path within the container**.
 - Simple **source:destination** syntax
 - `-B /path/to/data : /data`
 - The commands invoked in the will follow `/data` to `/path/to/data` on the host system disk.

Working with Modern Pipelines

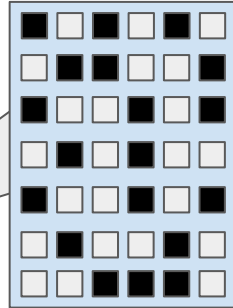
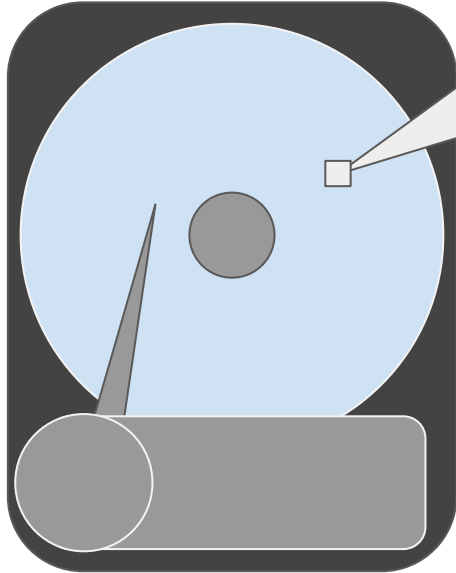
- Increasingly they are fully containerized.
 - This greatly facilitates replication.
- However, many modern pipelines (i.e. [fMRIPrep](#)) create a LOT of additional logs and working files in a dense directory structure.
 - Not always optimized for use on a HPC.
- This utilizes a lot of **inodes** per participant, and can rapidly reduce the number of subjects that can be processed or stored on a standard allocation.

What are **inodes**?

- **inodes** (Index Nodes) are the available pointers within a file system that can identify different files.
 - Inodes are defined when the file system is initialized and cannot be adjusted.
 - Each file on disk needs 1 inode.
- The default ratio of inodes to disk space per file does not work well when many small files are needed.
- When processing a few input files into many small output files, you are likely to run out of inodes before you run out of disk space.
 - This is a common issue on HPC compute disk.

What are inodes?

Hard Disk



Operating System

File System

Inodes

Other Metadata

Inode 0

Inode 1

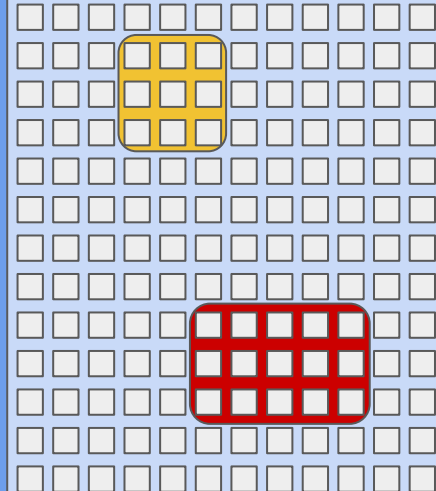
.

.

.

.

Inode N



How can `appthainer` help with this?

Writable Overlays (`.ext3`)

Writable Overlays (`.ext3`)

- `apptainer` can create a **writable overlay** (or a **persistent overlay**).
- This is a file system image that apptainer can utilize similar to a bind.
 - They use a different argument: `--overlay /path/to/overlay.ext3`
- However, data can be **written** to the paths on this image.
- This image will be seen as a single inode on the host system, but uses its own internal inodes for managing files within it.

Writable Overlay Usage

- I've used writable overlays for dealing with pipelines ([fMRIPrep](#), [TractoFlow](#)) that create a large number of intermediary files.
- I will create a writable overlay for each subject to store the working output of the pipeline.
- I can then archive the results and discard the compartmentalized working directory.
- This greatly increased the concurrent throughput of analyses.

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

1. Create the overlay with the desired output folders:

```
apptainer overlay create --size 5120 --sparse \  
--create-dir /work --create-dir /results \  
/path/to/image.ext3
```

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

1. Create the overlay with the desired output folders:

```
apptainer overlay create --size 5120 --sparse \  
--create-dir /work --create-dir /results \  
/path/to/image.ext3
```

size (MB) of image - do not zero image

create the directory path - writable directories

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

2. Use the overlay to catch all the working files:

```
apptainer run -B /path/to/data/:/data --overlay  
/path/to/image.ext3 /path/to/container.sif --input /data  
--output /results
```

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

2. Use the overlay to catch all the working files:

```
apptainer run -B /path/to/data/:/data --overlay  
/path/to/image.ext3 /path/to/container.sif --input /data  
--output /results
```

path to data on host - path to data within container - writable directories

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

3. Verify and archive the output:

```
apptainer exec -B /path/to/archive:/archive --overlay  
/path/to/image.ext3 tar cvzfh /archive/${SUBJ}-output.tar.gz  
/results
```

Working with Writable Overlays

The general way to work with these overlays is in 3 stages:

3. Verify and archive the output:

```
apptainer exec -B /path/to/archive:/archive --overlay  
/path/to/image.ext3 tar cvzfh /archive/${SUBJ}-output.tar.gz  
/results
```

path to archive on host - path to data within overlay

writable directories - path to archive in container

Distributing Locally

Archive Overlays (`.squashfs`)

Squashing a File System

- A `.squashfs` filesystem is a read-only, compressible filesystem.
 - It was originally used for creating boot images on floppy drives and CDs.
- However, it can be passed as an overlay to apptainer.
 - This will add the paths and data in the `.squashfs` image to the paths in the container.
- Because the filesystem cannot be modified, there is no risk of accidentally modifying the data within it.
- This is how the raw data and finalized derivatives are distributed for the UKBB on Compute Canada.
 - It's also being developed for archiving DICOM files with `nipoppy`.

Deploying large fixed file datasets with SquashFS and Singularity

Pierre Rioux *

Montreal Neurological Institute,
McGill University, Montreal, QC,
Canada

pierre.rioux@mcgill.ca

Gregory Kiar

Montreal Neurological Institute,
McGill University, Montreal, QC,
Canada

gkiar@mcin.ca

Alexandre Hutton

Montreal Neurological Institute,
McGill University, Montreal, QC,
Canada

ahutton@cim.mcgill.ca

Alan C. Evans

Montreal Neurological Institute,
McGill University, Montreal, QC,
Canada

alan@bic.mni.mcgill.ca

Shawn T. Brown

Pittsburgh Supercomputing Center,
Pittsburgh, PA, USA
stbrown@psc.edu

Creating a `.squashfs` Filesystem

- `mksquashfs` can be used similar to `tar`
 - `mksquashfs /path/to/data /path/to/archive.squashfs -keep-as-directory`
 - They are compressible.
- Because the systems are read only, modifications are a challenge.
 - `unsquashfs`, make the changes, and `re-mksquashfs` it.
 - You can append to them (to a limit).

Limitations of `.squashfs`

- There is a hard limit of 2.2 TB of space an image can take up.
- They can be frustrating to make or correct.
- 1 large compressed archive file can be riskier to store long term.

Distributing Globally

What Exists to Share Data

- Standards

- BIDS



- Platforms

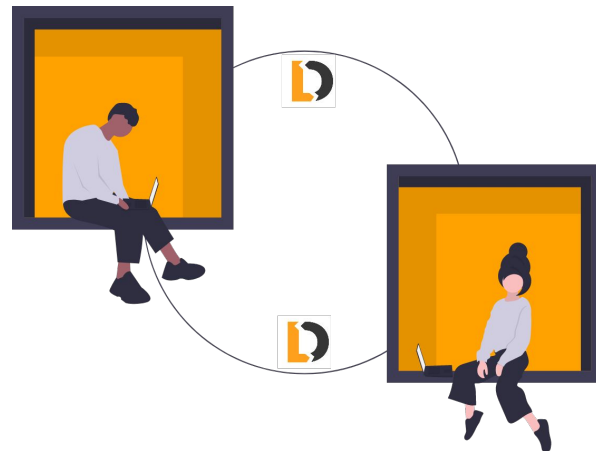
- brainlife.io
- Loris / CBRAIN



- Standards aren't enough and Platforms may not interact cleanly.
- Sharing data is only part of the problem - how can I share what I've **done** to the data?

What Exists to Share Data

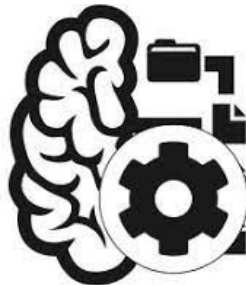
- Datalad
 - Interfaces with git-annex to store large files in a git-like way.
- It has many large open datasets indexed already.
 - OpenNeuro, HCP, etc.
- Can version runs of various functions for re-execution or re-application to a new subject.
 - Store command provenance for replicable reuse.



* Similar ability to compress data with 7zip like overlays, but (currently) is meant for archiving not computational speedups.

What Exists to Share Pipelines

- BIDS Apps
- brainlife.io Apps



- Nipype / pydra
- Nextflow
- Many pipeline engines not specific to neuroimaging

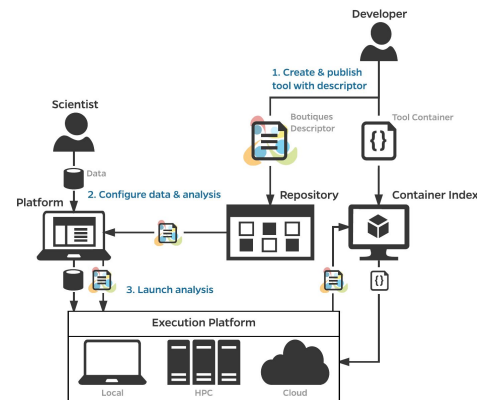


Nipype:
Neuroimaging in Python
Pipelines and Interfaces



What Exists to Share Pipelines

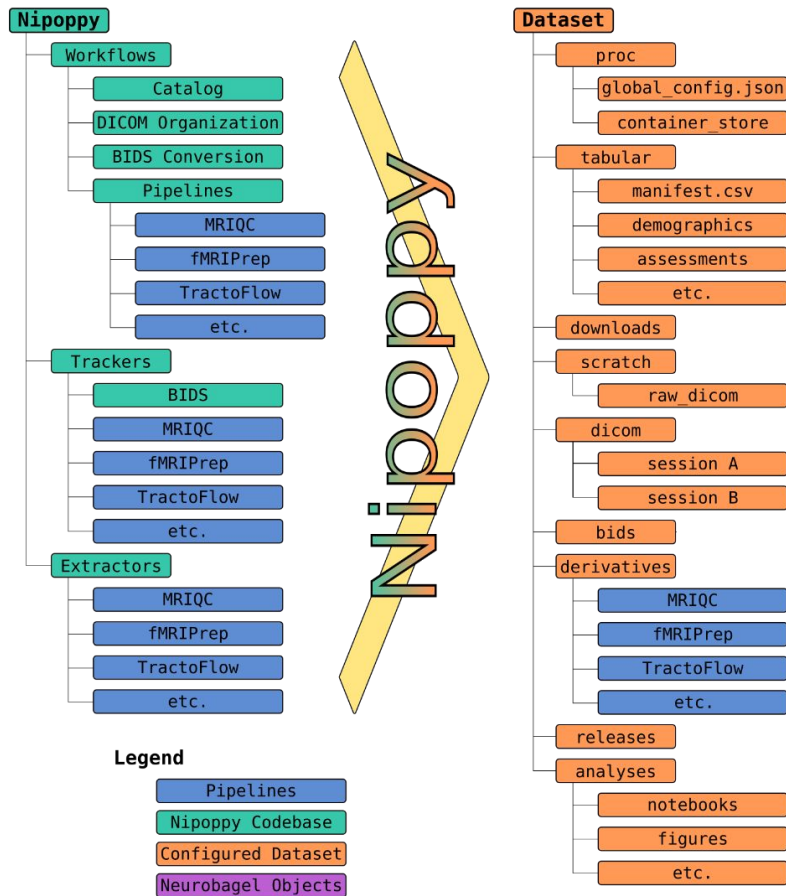
- Boutiques
 - Used for multiple projects within our group.
- Creates a .json “sidecar” that describes an analysis script.
 - Inputs parameters
 - Possible outputs
 - Container
- This descriptor is easily shared to deploy the containerized call on any system.



What are we working on?

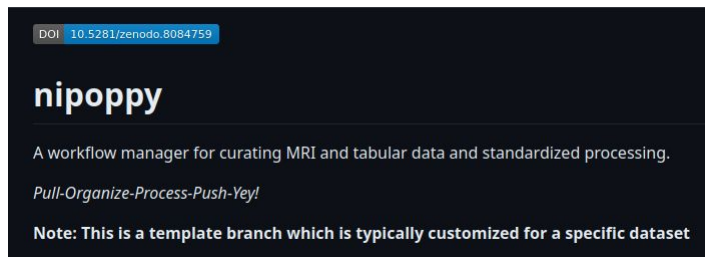
Nipoppy: ***P**ull - **O**rganize - **P**rocess - **P**ush - **Y**ey!*

- A standard organization structure for curating imaging data
 - Sort DICOMs for `heudiconv` conversion with `dcm2nii`
 - Convert DICOMs to BIDS structured `.nii.gz` files
 - **Track participants** to determine completed pipelines and available data.
 - Apply standard pipelines to the BIDS data:
 - [MRIQC](#)
 - [fMRIPrep](#) ([FreeSurfer](#))
 - [TractoFlow](#)
 - Extract features from pipeline results for analysis



Nipoppy Roadmap

- Documentation and improving usability *while* maintaining current projects.
- Better extractors for easier, automated processing.
- **Datalad** integration.
- Standard pipeline annotation with **Boutiques**.



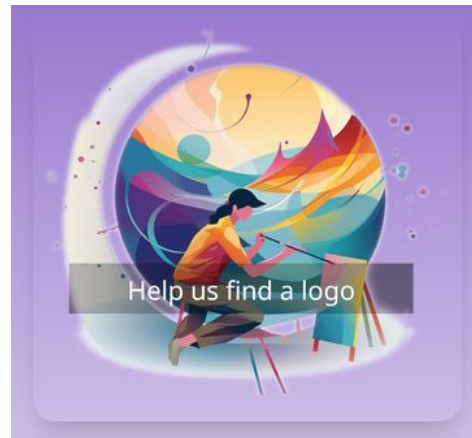
Annotating

Nipoppy facilitates **annotating** a dataset

- With data and analyses in a standard structure, it's easier to share them.
- But you still have to protect private information.
 - International regulations can provide an even bigger administrative block.
- How do you deal with different data standards or variable encodings?
 - Interoperability - The “I” in FAIR
 - These are the same issues that can plague platforms
- A standardized **annotation** of the data would make it findable without leaking sensitive information.

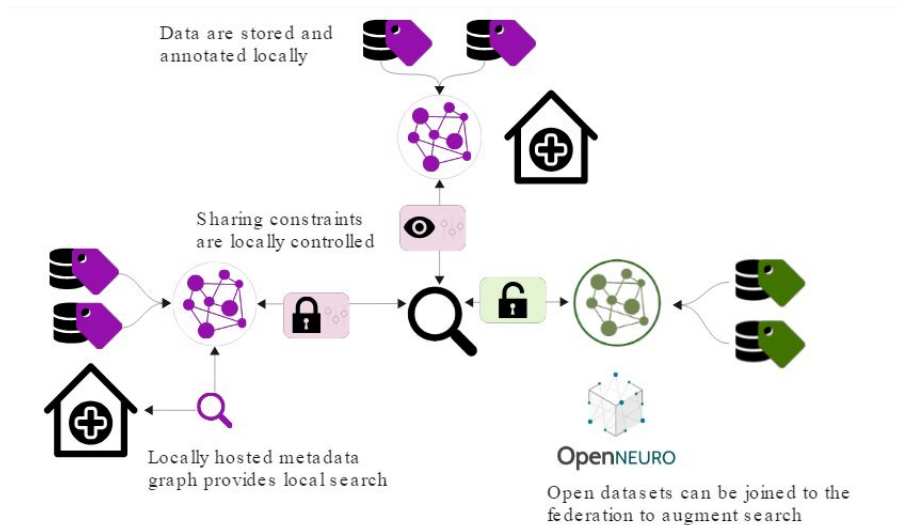
Neurobagel

- Is a suite of tools to create federated dataset harmonization and search.
- Annotate the modalities, derivatives, demographic, and clinical features of the dataset make it discoverable without uploading the data.
- A new way to collaboratively curate and find new, relevant data for replicating findings and generalizing models.
 - It also bolsters the impact of your own contributions.



Neurobagel

- Share a standardized annotation of your dataset so it can be discoverable while keeping it secure.
- Increase the impact of your research by being available for collaborations based on common data and analysis goals.



Recommended Categories

Click category and then
corresponding column from tsv file

Subject ID

Age

Sex

Diagnosis

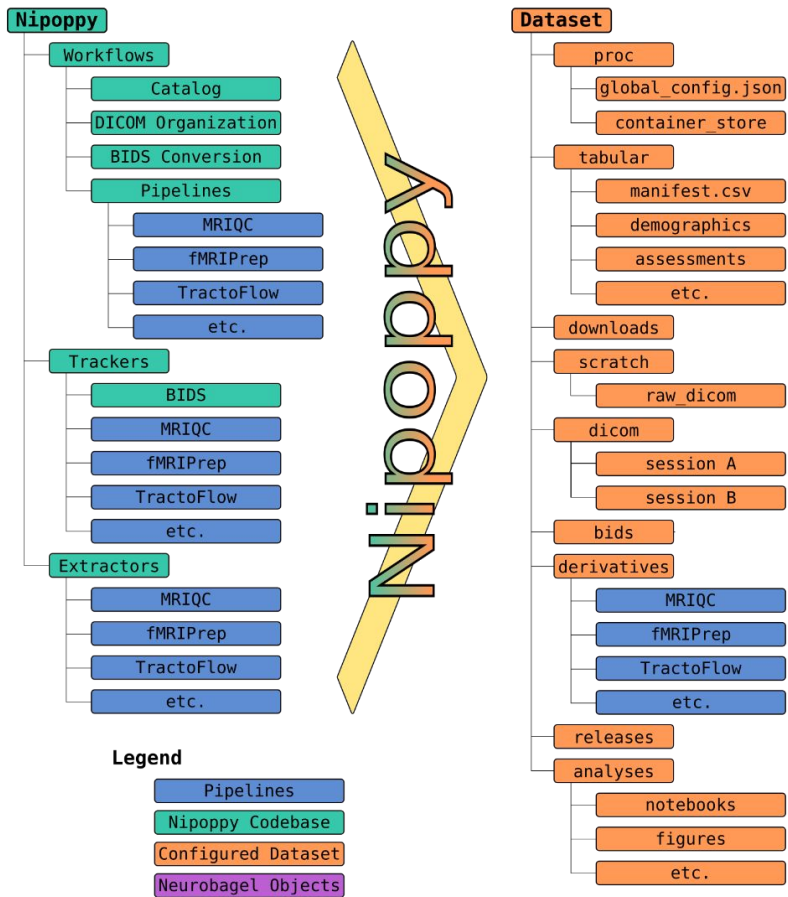
Assessment Tool

NEW CATEGORY

X

+

Column	Description
participant_id	
age	age of the participant
sex	sex of the participant as reported by the participant
group	diagnostic status determined by the study clinician at baseline
group_dx	specific diagnosis determined by the study clinician at baseline
number_comorbid_dx	a number of diagnoses comorbid with UD (e.g., GAD, PTSD)
medload	reflects the number of dosage of psychotropic medications taken by participants. Higher numbers correspond to more medications and/or higher medication dosage
iq	IQ derived based on the NART assessment.
session	scanning session



A Hackathon Idea

- The NA216 dataset could be annotated for Neurobagel
 - Convert to BIDS standard in a Datalad object.
 - Use BIDS extractor to create an initial Neurobagel .jsonld
 - Standardize the metadata (gender, age, weight, etc.)
- It won't cover facets that are not in the standard yet,
 - Non-human
 - Ex-vivo scans (?)
- But it would be a rapid way to see how these datasets might work encoded in a larger, federated database of clinical information.
 - Help with harmonizing ongoing collections.

Thanks and Acknowledgements



Collaborators

- Alain Dagher
- Yue Li
- Celia Greenwood



Nipoppy



Michelle Wang



Nikhil Bhagwat

Neurobagel



Sebastian
Urchs



Jonathan
Armoza



Arman
Jahanpour



Alyssa Dai



Remi Gau

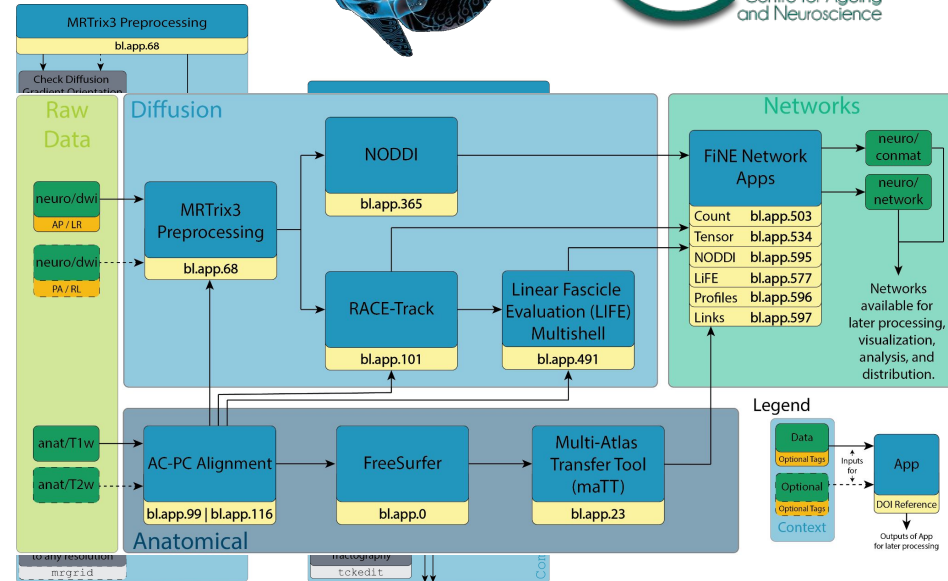
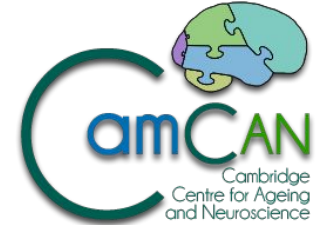
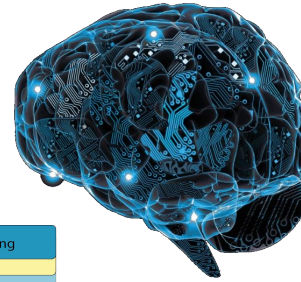


Jean-Baptiste
Poline

Fin
Merci Beaucoup!

Analyzing Population Samples to Understand Aging

- My PhD was modeling Alzheimer's Disease with diffusion MRI (dMRI) and connectomics.
- However, the best sample I found to work with was the CamCAN dataset.
- I developed a pipeline on brainlife.io to go from converted .nii.gz data to connectomes and applied it to CamCAN.



Picture of Paths “Overlaid” on the Filesystem?

```

1 #!/bin/bash
2
3 #SBATCH -J exmaple-job
4 #SBATCH --time=20:00:00
5 #SBATCH --account=def-jbpoline
6 #SBATCH --cpus-per-task=4
7 #SBATCH --mem-per-cpu=4G
8 #SBATCH --error=/scratch/bcmcpheer/project/logs/example-job-%A_%a.err
9 #SBATCH --output=/scratch/bcmcpheer/projectlogs/example-job-%A_%a.out
10 #SBATCH --array=4000-4499
11
12 ## get subj from array index
13 SAMPLE_LIST=(($(</scratch/bcmcpheer/project/subject_list.txt))
14 SUBJ=${SAMPLE_LIST[${SLURM_ARRAY_TASK_ID}]}
15
16 ## working directory
17 WORKDIR=/scratch/bcmcpheer/project
18 DATADIR=/lustre03/project/6008063/bcmcpheer/data
19
20 ## path to container
21 APP_IMAGE=${WORKDIR}/container/my_bids_app.sif
22
23 ## path to persistent disk
24 ARCHIVE=/lustre03/project/6008063/bcmcpheer/results
25
26 ## load apptainer
27 module load apptainer
28
29 ## create name for the tmp images
30 TMPIMAGE=${SUBJ}_workdir.ext3
31
32 ## create the writable overlay to use for the job
33 apptainer overlay create --size 5120 --sparse \
34   --create-dir /results --create-dir /work --create-dir /transfer \
35   ${WORKDIR}/work/${TMPIMAGE}
36
37 ## call a processing command
38 apptainer exec -B ${WORKDIR} -B ${DATADIR}:/input --overlay ${WORKDIR}/work/${TMPIMAGE} ${APP_IMAGE} \
39   my_bids_app --subj ${SUBJ} --input-dir /input --work-dir /work --results-dir /results
40
41 ## tar the results out of the scratch image
42 apptainer exec -B ${WORKDIR} -B ${ARCHIVE}:/transfer --overlay ${WORKDIR}/work/${TMPIMAGE} ${APP_IMAGE} \
43   tar cvzf /transfer/${SUBJ}_results.tar.gz /results
44

```

Archiving data

- How should data be stored after processing?
- Do you want to manage permissions across many different disk images?
 - `--fakeroot` makes this a real challenge.
- How can we monitor if files changed?
 - We are hoping to reuse and share pipeline outputs.

Sharing Locally

- If you're sharing locally, `.squashfs` isn't a bad solution
 - Keeping in mind its limitations.
- Standards to share pipelines and data organization makes sharing progress easier.
- There are proposed ways to share data, but with different trade offs.

Common Challenges

- Prepare data for analysis (and re-analysis).
 - BIDS, Boutiques, Platforms
- Create useful derivatives for distribution and replication efforts.
 - nipoppy, boutiques, apptainer
- Reliably distributing and sharing both the tools and data.
 - apptainer, .squashfs, datalad
- Safely sharing access to the data and derivatives.
 - Datalad, nipoppy, apptainer, .squashfs, Neurobagel