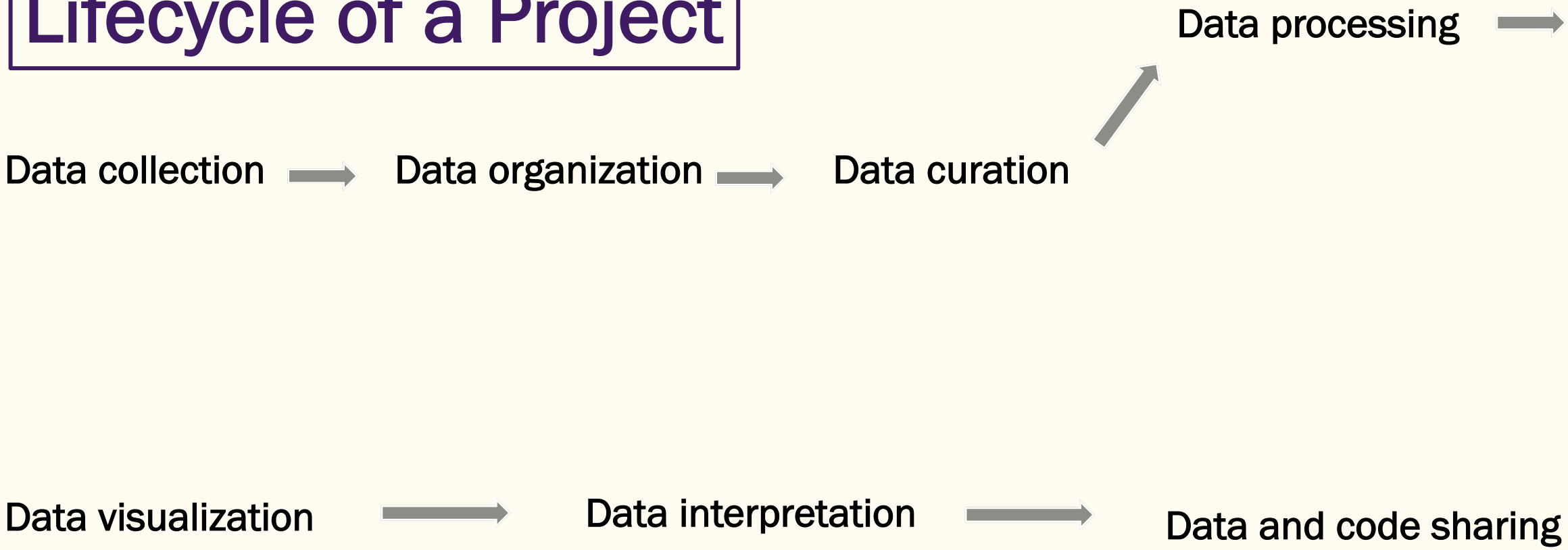
REPRODUCIBLE NEURO-INFORMATICS

Data organization and curation, software containers,
data annotation tools, github & gh-pages

There are many tools available for efficient, state-of-the-art, reproducible neuroimaging data analysis.

These tools will facilitate discovery and good science practices.

Lifecycle of a Project



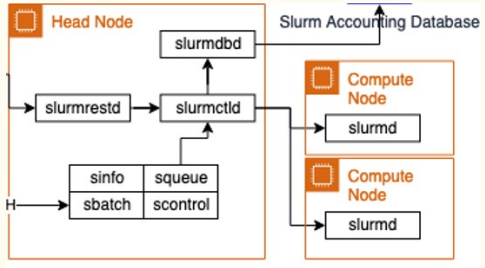
Lifecycle of a Project

Data collection → Data organization → Data curation → Data processing →

BIDS
BRAIN IMAGING DATA STRUCTURE

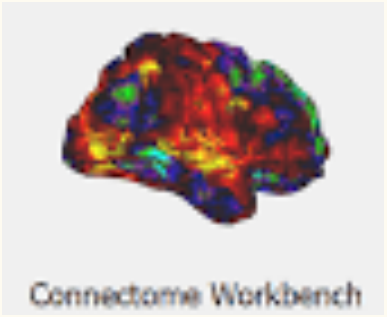
CuBIDS:
Curation of BIDS

docker



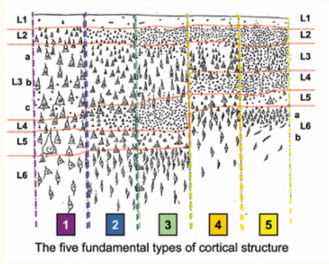
Data visualization → Data interpretation → Data and code sharing

ggseg



neurosynth.org

neuromaps



GitHub Pages

Data **collection** → **Data organization** →

BIDS
BRAIN IMAGING DATA STRUCTURE

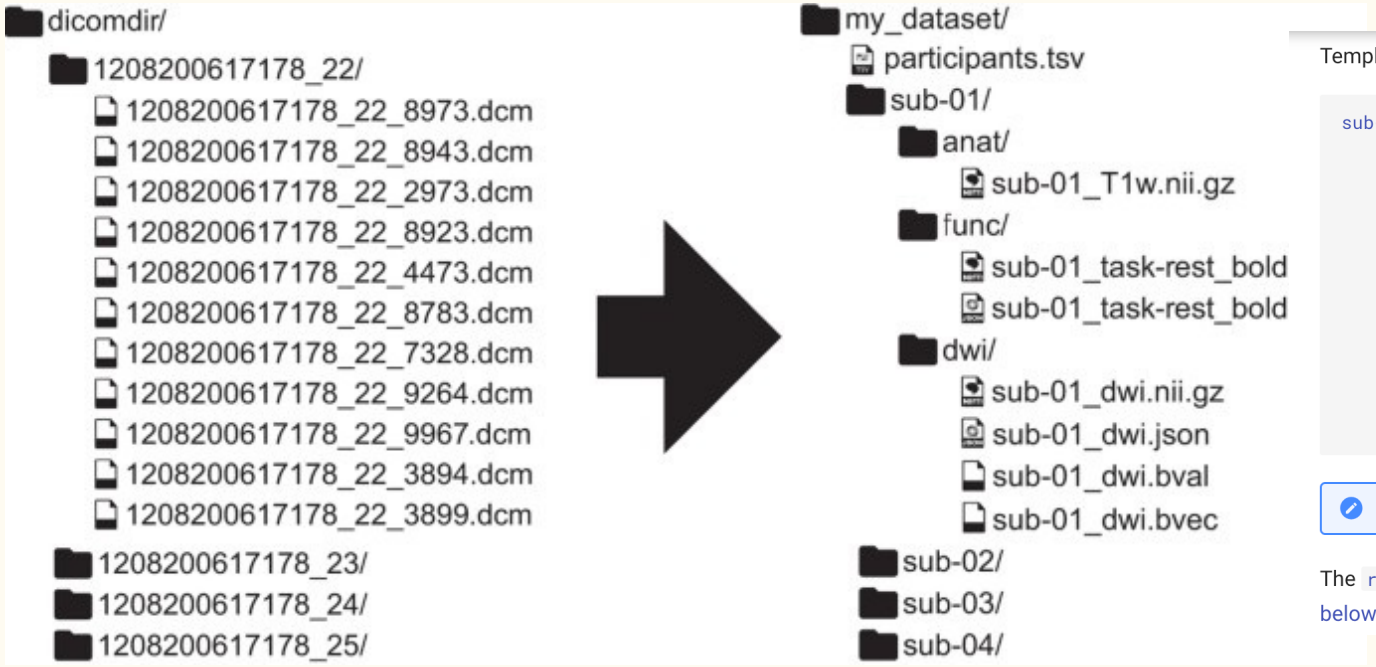

NOT SURE IF I AM CLEANING
OR ORGANIZING A MESS


I MEAN, I HAVE WON AWARDS FOR MY
ORGANIZATIONAL SKILLS.

Brain Imaging Data Structure (BIDS)

A standardized way to organize and name your neuroimaging data!

- Modality-specific naming conventions
- Scans identifiable via subject and session hierarchy
- Many helpful naming modifiers



Template:

```
sub-<label>/
    [ses-<label>/]
        dwi/
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>][_acq-<label>][_rec-<lab
            sub-<label>[_ses-<label>]_task-<label>[_acq-<labe
            sub-<label>[_ses-<label>]_task-<label>[_acq-<labe
            sub-<label>[_ses-<label>]_task-<label>[_acq-<labe
            sub-<label>[_ses-<label>]_task-<label>[_acq-<labe
```

Legend:

The `run-<index>` entity is RECOMMENDED to encode the splits of multi
below for more information on multipart DWI schemes).

Modality specific files

- Magnetic Resonance Imaging
- Magnetoencephalography
- Electroencephalography
- Intracranial Electroencephalography
- Task events
- Physiological and other continuous recordings
- Behavioral experiments (with no neural recordings)
- Genetic Descriptor
- Positron Emission Tomography

Brain Imaging Data Structure (BIDS)

A standardized way to organize and name your neuroimaging data!

- Modality-specific naming conventions
- Scans identifiable via subject and session hierarchy
- Many helpful naming modifiers
- **Allows you to use BIDS apps! that know, by default, the structure of your data**

Brain Imaging Data Structure (BIDS)

A standardized way to organize and name your neuroimaging data!

```
if "MP2RAGEPTX_TR6000_1mmiso_INV1_PHS_FILT" in s.series_description:
    info[inv1_phase].append(s.series_id)
if ("MP2RAGEPTX_TR6000_1mmiso_INV1" in s.series_description) and ("M" in s.image_type):
    info[inv1_mag].append(s.series_id)
if "MP2RAGEPTX_TR6000_1mmiso_INV2_PHS_FILT" in s.series_description:
    info[inv2_phase].append(s.series_id)
if ("MP2RAGEPTX_TR6000_1mmiso_INV2" in s.series_description) and ("M" in s.image_type):
    info[inv2_mag].append(s.series_id)
if "MP2RAGEPTX_TR6000_1mmiso_UNI_Images" in s.series_description:
    info[UNI].append(s.series_id)
if ("MP2RAGEPTX_TR6000_1mmiso_UNI-DEN" in s.series_description) and ("mlBra
    info[UNIDEN].append(s.series_id)
if "MP2RAGEPTX_TR6000_1mmiso_T1_Images" in s.series_description:
    info[T1].append(s.series_id)
#if "B1-abs-1slc-singleChannelMode-B1" in s.dcm_dir_name:
    #info[tfl_b1map].append(s.series_id)
```

```
## MP2RAGE
inv1_phase = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_inv-1_part-mag_MP2RAGE')
inv1_mag = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_inv-1_part-phase_MP2RAGE')
inv2_phase = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_inv-2_part-mag_MP2RAGE')
inv2_mag = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_inv-2_part-phase_MP2RAGE')
UNI = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_UNIT1')
UNIDEN = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_acq-UNIDENT1_T1w')
T1 = create_key(
    'sub-{subject}/{session}/anat/sub-{subject}_{session}_T1map')
```

Brain Imaging Data Structure (BIDS)

A standardized way to organize and name your neuroimaging data!

```
# Baseline session

    #if "Localizer" in s.protocol_name:
    if ("SpinEchoFieldMap_AP" in s.protocol_name) or ("fmap_acq-topup_dir-AP_epi" in s.protocol_name) or ("fmap_acq-spinecho_dir-AP_epi" in s.protocol_name):
        info[fmap_se_AP].append(s.series_id)
    if ("SpinEchoFieldMap_PA" in s.protocol_name) or ("fmap_acq-topup_dir-PA_epi" in s.protocol_name) or ("fmap_acq-spinecho_dir-PA_epi" in s.protocol_name):
        info[fmap_se_PA].append(s.series_id)
    if ("rfMRI_REST_AP" in s.protocol_name) or ("func_task-rest_acq-TR2s_dir-AP_bold" in s.protocol_name) or ("func_task-rest_dir-AP_bold" in s.protocol_name):
        info[rest_AP].append(s.series_id)
    if ("rfMRI_REST_PA" in s.protocol_name) or ("func_task-rest_acq-TR2s_dir-PA_bold" in s.protocol_name) or ("func_task-rest_dir-PA_bold" in s.protocol_name):
        info[rest_PA].append(s.series_id)
    if ("T1w_MPR" in s.protocol_name) or ("anat_acq-mpr_T1w" in s.protocol_name):
        info[t1w].append(s.series_id)
    if ("BOLD_02_nback_348" in s.protocol_name) or ("func_task-nback02_run-01_bold" in s.protocol_name):
        info[nback_02].append(s.series_id)
    if ("DTI_64dir_MB2" in s.protocol_name) or ("dwi_acq-multiband2dir64_dwi" in s.protocol_name):
        info[dti].append(s.series_id)
    #if s.series_description.endswith('_M0'):
        #info[m0].append(s.series_id) #Removed conversion of MZero to nifti. With current versions of
        #dcm2niix, the M0 will not convert correctly (one slice is deleted, resulting in errors). Error
        #is reproduced across different studies using this M0 protocol- not fixed for now. MZero not
        #used in processing so no conversion to nifti needed.
    if s.series_description.endswith('_ASL'):
        info[raw_asl].append(s.series_id)
    if s.series_description.endswith('_MeanPerf'):
        info[mean_perf].append(s.series_id)
    if ((s.protocol_name.startswith('B0map')) and ('IFG' not in s.dcm_dir_name) and ('sgACC' not in s.dcm_dir_name) and ('FP' not in s.d
        if "P" in s.image_type:
            info[fmap_phPA_baseline].append(s.series_id)
        elif "M" in s.image_type:
            info[fmap_magPA_baseline].append(s.series_id)
```

```
# TMS session

    if "IFG_Rest_PreIFG" in s.dcm_dir_name:
        info[preIFG_rest].append(s.series_id)
    if "IFG_sp80_PreIFG" in s.dcm_dir_name:
        info[preIFG_sp80].append(s.series_id)
    if "IFG_sp100_PreIFG" in s.dcm_dir_name:
        info[preIFG_sp100].append(s.series_id)
    if "IFG_sp120_PreIFG" in s.dcm_dir_name:
        info[preIFG_sp120].append(s.series_id)
    if "IFG_TBS_IFG" in s.dcm_dir_name:
        info[IFG_iTBS].append(s.series_id)
    if "IFG_sp120_PostIFG" in s.dcm_dir_name:
        info[postIFG_sp120].append(s.series_id)
    if "IFG_Rest_PostIFG" in s.dcm_dir_name:
        info[postIFG_rest].append(s.series_id)
    if "IFG_B0Maps" in s.dcm_dir_name:
        if "P" in s.image_type:
            info[fmap_phPA_IFG].append(s.series_id)
        elif "M" in s.image_type:
            info[fmap_magPA_IFG].append(s.series_id)
```

Brain Imaging Data Structure (BIDS)

A standardized way to organize and name your neuroimaging data!

```
cd $input_dicomdir

for sub in 1* ; do
	subject=${sub%_*}
	session=${sub#*_}

	if ! [ -d $bids_dir/sub-$subject/ses-$session ] ; then

	heudiconv -d $bidscompliant_dicomdir/{subject}/{session}/*/* -s $subject -ss $session -f
/Volumes/Hera/Projects/corticalmyelin_development/code/corticalmyelin_maturation/BIDS/BIDS_heudico
nv/7TBrainMech_MP2RAGE_heuristic.py -c dcm2niix -b -o $bids_dir --grouping custom &>
$logs_dir/${subject}_${session}-log.txt

	fi
done
```

Data collection → Data organization → Data curation

BIDS
BRAIN IMAGING DATA STRUCTURE

CuBIDS:
Curation of BIDS

When your armor doesn't match but it increases your stats

Curation of BIDS on Disk (CuBIDS)

A convenient way to check image acquisition parameters across all scans and modalities and identify variants!

KeyGroup	ParamGroup	Counts	Dim1Size	Dim2Size	Dim3Size	EchoTime	FlipAngle	HasFieldmap	ImageOrientation	KeyGroupCount	Modality
acquisition-UNIDENT1_datatype-anat_suffix-T1w	1	273	184	210	192	0.00287		False	RAS+	2126	anat
acquisition-UNIDENT1_datatype-anat_suffix-T1w	2	1	168	210	192	0.00287		False	RAS+	2126	anat
datatype-anat_suffix-T1map	1	269	184	210	192	0.00287		False	RAS+	2104	anat
datatype-anat_suffix-T1map	2	1	168	210	192	0.00287		False	RAS+	2104	anat

```
sub-NDARYD666FL0/ses-HBNsiteCBIC/dwi/sub-NDARYD666FL0_ses-HBNsiteCBIC_acq-64dirVARIANTObliquity_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARYH480GTD/ses-HBNsiteSI/dwi/sub-NDARYH480GTD_ses-HBNsiteSI_acq-64dirVARIANTNoFmap_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARYN595JMA/ses-HBNsiteRU/dwi/sub-NDARYN595JMA_ses-HBNsiteRU_acq-64dirVARIANTNoFmap_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARYW170CAA/ses-HBNsiteCUNY/dwi/sub-NDARYW170CAA_ses-HBNsiteCUNY_acq-64dirVARIANTObliquity_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARYW789GNP/ses-HBNsiteCBIC/dwi/sub-NDARYW789GNP_ses-HBNsiteCBIC_acq-64dirVARIANTObliquity_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARZE389XF0/ses-HBNsiteRU/dwi/sub-NDARZE389XF0_ses-HBNsiteRU_acq-64dirVARIANTObliquity_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARZF288FB7/ses-HBNsiteSI/dwi/sub-NDARZF288FB7_ses-HBNsiteSI_acq-64dirVARIANTObliquity_space-T1w_desc-preproc_gqi.mif.gz@
sub-NDARZH629NVF/ses-HBNsiteSI/dwi/sub-NDARZH629NVF_ses-HBNsiteSI_acq-64dirVARIANTEchoTimePhaseEncodingDirection_space-T1w_desc-preproc_gqi.mif.gz@
```

```
sub-3445089839/ses-PNC1/dwi/sub-3445089839_ses-PNC1_acq-VARIANTRepetitionTime_space-T1w_desc-preproc_gqi.mif.gz@
sub-3520758666/ses-PNC1/dwi/sub-3520758666_ses-PNC1_acq-VARIANTNoFmap_space-T1w_desc-preproc_gqi.mif.gz@
sub-3818082974/ses-PNC1/dwi/sub-3818082974_ses-PNC1_acq-VARIANTNoFmap_space-T1w_desc-preproc_gqi.mif.gz@
```

Curation of BIDS on Disk (CuBIDS)

A convenient way to check image acquisition parameters across all scans and modalities and identify variants!

```
#!/bin/bash

# Use CuBIDS to look at metadata and acquisition heterogeneity

## Install CuBIDS
conda activate cubids
pip install CuBIDS
conda install nodejs
npm install -g bids-validator@1.7.2

bids_dir=/Volumes/Hera/Projects/corticalmyelin_development/BIDS
cubids-print-metadata-fields $bids_dir #ensure metadata of interest is in jsons
cubids-group $bids_dir $bids_dir/CuBIDS/v0 #create spreadshseets with metadata information and key param groups
cubids-add-nifti-info $bids_dir
cubids-group $bids_dir $bids_dir/CuBIDS/v1 #create spreadshseets with metadata information and key param groups after running cubids-add-nifti-info
```

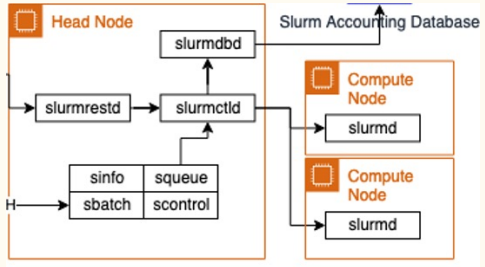
Lifecycle of a Project

Data collection → Data organization → Data curation → Data processing →

BIDS
BRAIN IMAGING DATA STRUCTURE

CuBIDS:
Curation of BIDS

docker




IF YOU'RE RUNNING DIFFERENT SOFTWARE VERSIONS
INSTRUCTOR
YOU'RE GONNA HAVE A


I can't remember everything.
I am not a magician.


PLEASE WAIT
WHILE THE WIZARD INSTALLS THE SOFTWARE

Containers

Image software packages that ~contain~ specific versions of many softwares plus pipeline-specific run scripts all wrapped up for you!

- Hosted on and downloadable from Docker
- Easy to build
- Portable between computers, networks, etc.
- Reproducible

A container is a standard unit of software that packages up code and all its dependencies so the application runs quickly and reliably from one computing environment to another. A Docker container image is a lightweight, standalone, executable package of software that includes everything needed to run an application: code, runtime, system tools, system libraries and settings.

BIDS apps

Software applications, built by experts, that help you process your BIDS-formatted data

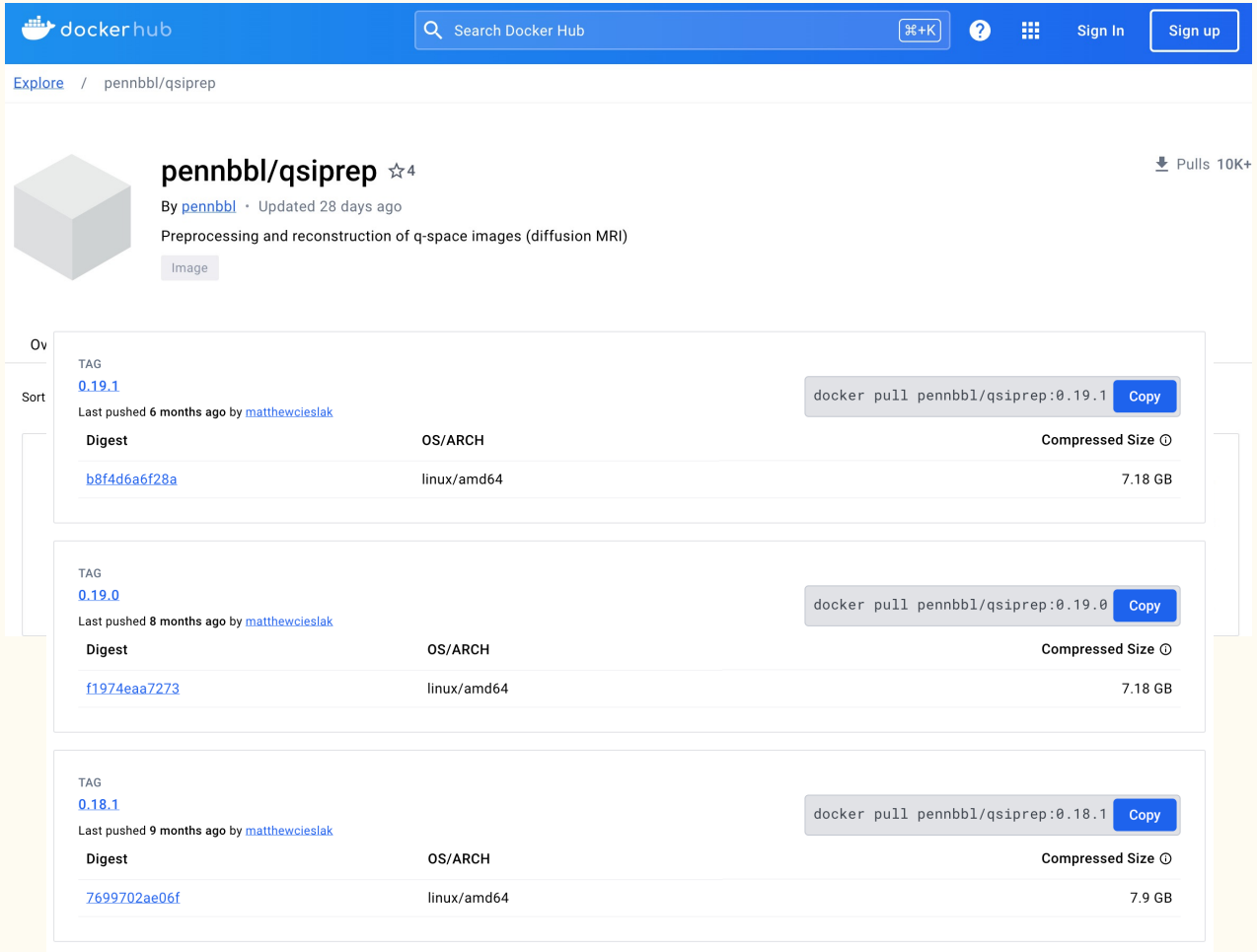
- Work on BIDS valid datasets
- Immediately recognize what modalities/types of data you do have, how many runs, what the acquisition parameters are and can configure the right pipeline for your data
- Often combine best-practices in the field and execute multi-steps pipelines with one command with many user-specified options

BIDS apps

Software applications, built by experts, that help you process your data

- antsCorticalThickness
- baracus
- BrainSuite
- HCPPipelines
- MAGeTbrain
- mindboggle
- ndmg
- QAP
- bidspm
- fmriprep
- dmriprep
- mriqc
- nibabies
- nirodents
- smriprep
- BIDSonym
- aslprep
- xcp_d
- qsiprep

BIDS apps 🤝 Containers



Positional Arguments

`bids_dir`

the root folder of a BIDS valid dataset (sub-XXXXX folders should be found at the top level in this folder).

`output_dir`

the output path for the outcomes of preprocessing and visual reports

`analysis_level`

Possible choices: participant

processing stage to be run, only “participant” in the case of qsiprep (see BIDS-Apps specification).

```
usage: qsiprep [-h] [--version] [--skip_bids_validation]
               [--participant_label PARTICIPANT_LABEL [PARTICIPANT_LABEL ...]]
               [--bids-database-dir BIDS_DATABASE_DIR]
               [--bids-filter-file FILE] [--interactive-reports-only]
               [--recon-only] [--recon-spec RECON_SPEC]
               [--recon-input RECON_INPUT]
               [--recon-input-pipeline {qsiprep,ukb,hcpya}]
               [--freesurfer-input FREESURFER_INPUT] [--skip-odf-reports]
               [--nthreads NTHREADS] [--omp-nthreads OMP_NTHREADS]
               [--mem_mb MEM_MB] [--low-mem] [--use-plugin USE_PLUGIN]
               [--anat-only] [--dwi-only] [--infant] [--boilerplate] [-v]
               [--anat-modality {T1w,T2w,none}]
               [--ignore {fieldmaps,phase} [{fieldmaps,phase} ...]]
               [--longitudinal] [--b0-threshold B0_THRESHOLD]
               [--dwi_denoise_window DWI_DENOISE_WINDOW]
               [--denoise-method {dwidenoise,patch2self,none}]
               [--unringing-method {none,mrdegibbs,rpg}] [--dwi-no-biascorr]
               [--b1-biascorrect-stage {final,none,legacy}]
               [--no-b0-harmonization] [--denoise-after-combining]
               [--separate_all_dwis]
               [--distortion-group-merge {concat,average,none}]
               [--write-local-bvecs]
               [--anatomical-template {MNI152NLin2009cAsym}]
               --output-resolution OUTPUT_RESOLUTION
               [--b0-to-t1w-transform {Rigid,Affine}]
               [--intramodal-template-iters INTRAMODAL_TEMPLATE_ITERS]
               [--intramodal-template-transform {Rigid,Affine,BSplineSyN,SyN}]
               [--b0-motion-corr-to {iterative,first}]
               [--hmc-transform {Affine,Rigid}]
               [--hmc_model {none,3dSHORE,eddy,tensor}]
               [--eddy-config EDDY_CONFIG] [--shoreline_iters SHORELINE_ITERS]
               [--impute-slice-threshold IMPUTE_SLICE_THRESHOLD]
               [--skull-strip-template {OASIS,NKI}] [--skull-strip-fixed-seed]
               [--skip-anat-based-spatial-normalization]
               [--fs-license-file PATH] [--do-reconall]
               [--pepolar-method {TOPUP,DRBUDDI,TOPUP+DRBUDDI}]
               [--denoised_image_sdc] [--prefer_dedicated_fmaps]
               [--fmap-bspline] [--fmap-no-demean] [--use-syn-sdc]
               [--force-syn] [-w WORK_DIR] [--resource-monitor]
               [--reports-only] [--run-uuid RUN_UUID] [--write-graph]
               [--stop-on-first-crash] [--notrack] [--sloppy]
               bids_dir output_dir {participant}
```

BIDS apps 🤝 Containers

singularity build /Volumes/Hera/Projects/corticalmyelin_developent/software/freesurfer-7.4.1/sif docker://bids/freesurfer:7.4.1-202309

```
# A script to run longitudinal freesurfer with the freesurfer bids-app singularity container on individual participants.

usage() {
        cat << EOF >&2
Usage: $script_name [-s] [-a] [-b] [-f] [-c] [-l] [-r] [-n]

-s <subject_id>: The subject identifier. This is the top-level BIDS sub-id
-a <acquisition_label>: If the input BIDS dataset contains multiple _T1w images from different acquisitions, the BIDS -a
-b <bids_directory>: Full path to the BIDS directory containing the <subject_id> folder
-f <freesurfer_directory>: The freesurfer directory where output files should be stored
-c <freesurfer_sif>: Full path to the singularity SIF container with the freesurfer bids-app
-l <freesurfer_license>: Full path to a freesurfer license.txt
-r <rerun>: If freesurfer output already exists for this <subject_id>, should it be rerun? Setting rerun to TRUE will de
-n <cores>: The number of cores/CPUs available for running freesurfer-bids. Defaults to 1
EOF
```

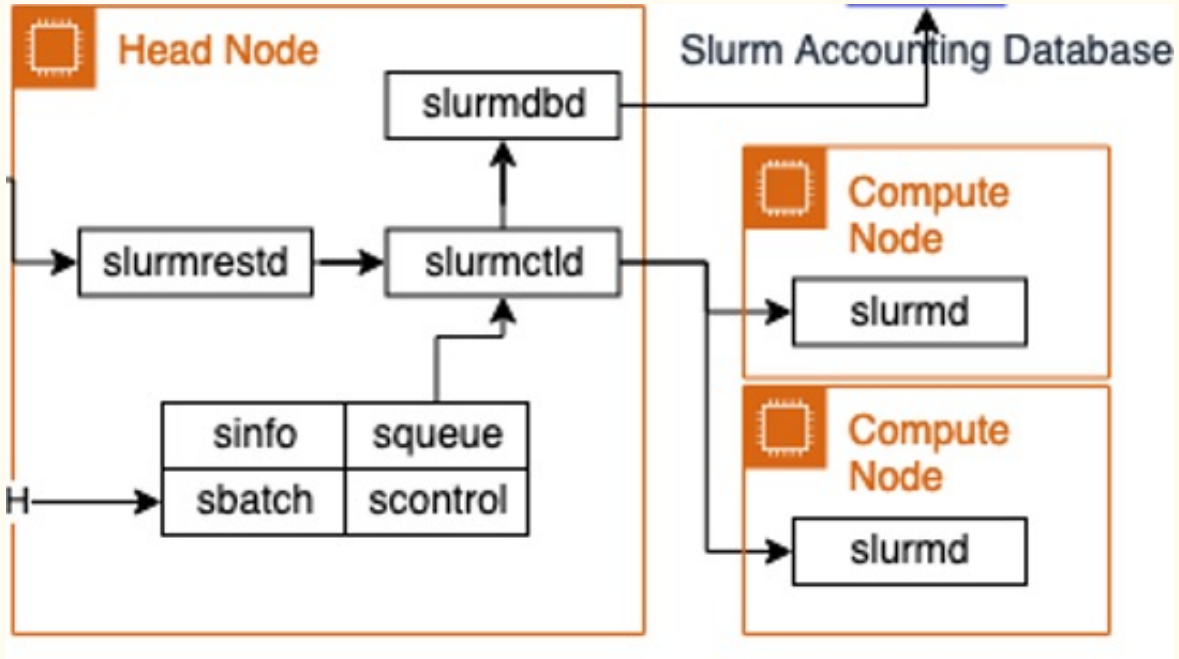
singularity run -B ${bids_dir}:/BIDS -B ${freesurfer_dir}:/Freesurfer_output -B ${freesurfer_license}:/license.txt $freesurfer_sif /BIDS /Freesurfer_output participant --participant_label ${subject_id} --acquisition_label ${acquisition_label} --license_file /license.txt --stages all --steps {cross-sectional,template,longitudinal} --multiple_sessions longitudinal --skip_bids_validator --n_cpus ${cores}

Super computing clusters

Submit computationally expensive scripts as parallel running jobs with high performance computing, lots of memory

```
#Launch job via sbatch
     sbatch \
   -J "${subses}-R1-vol2surf" \
   --time 00:30:00 \
   -p RM-shared \
   --nodes 1 \
   --ntasks-per-node 1  \
   -o "${subses}-R1-vol2surf.o" \
   -e "${subses}-R1-vol2surf.e" \
   --export="ALL,SUBJECT_ID=$subject_id,MOVING_IMAGE=$movin
   "$script_dir/voltosurf_projection_nativefsaverage.sh"
```

Pittsburgh super computer uses slurm

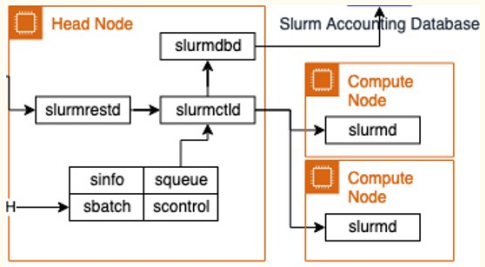


Data collection → Data organization → Data curation → Data processing →

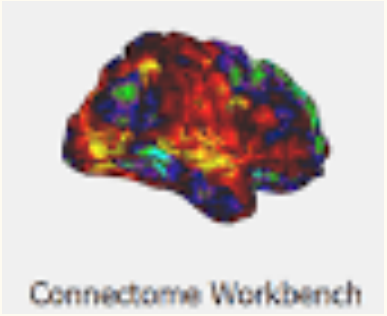
BIDS
BRAIN IMAGING DATA STRUCTURE

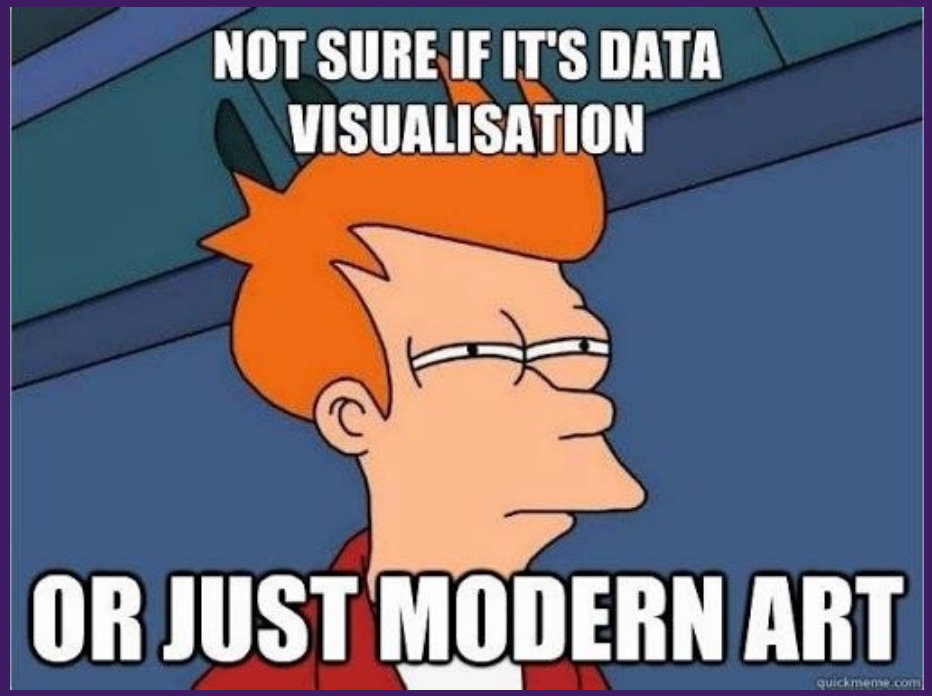
CuBIDS:
Curation of BIDS

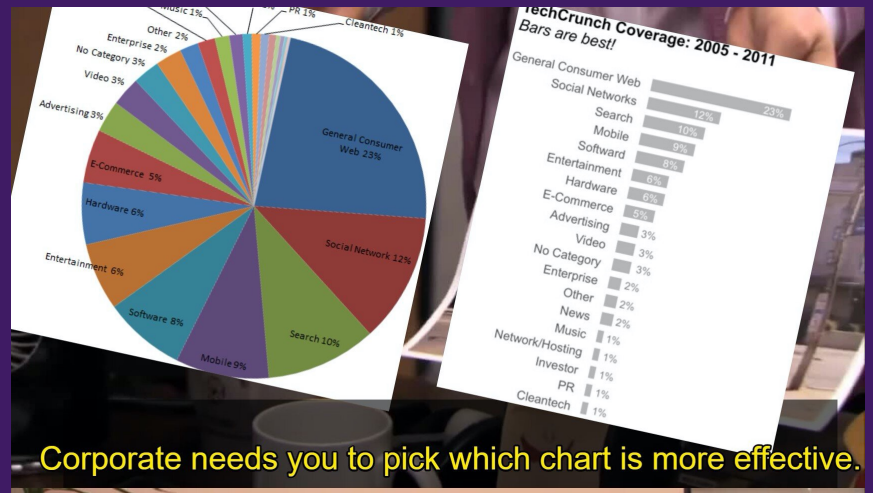
docker



Data visualization →




NOT SURE IF IT'S DATA VISUALISATION
OR JUST MODERN ART
quickmeme.com

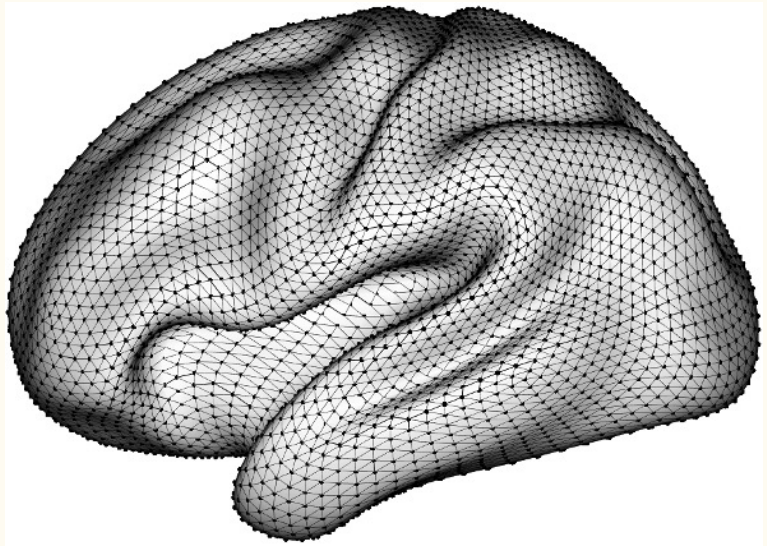

General Consumer Web 23%
Social Network 12%
Search 10%
Mobile 9%
Software 8%
Entertainment 6%
Hardware 6%
E-Commerce 5%
Advertising 3%
Video 3%
No Category 3%
Enterprise 2%
Other 2%
Cleantech 1%
TechCrunch Coverage: 2005 - 2011
Bars are best!
Corporate needs you to pick which chart is more effective.


IN
KAPWING
They're the same. Both are effective.

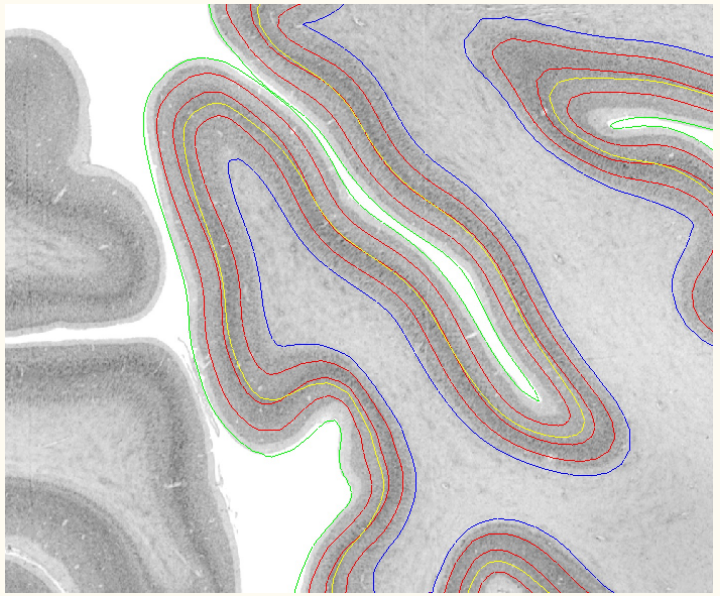
Surface-based Analyses

Project your data to a (usually cortical) surface mesh

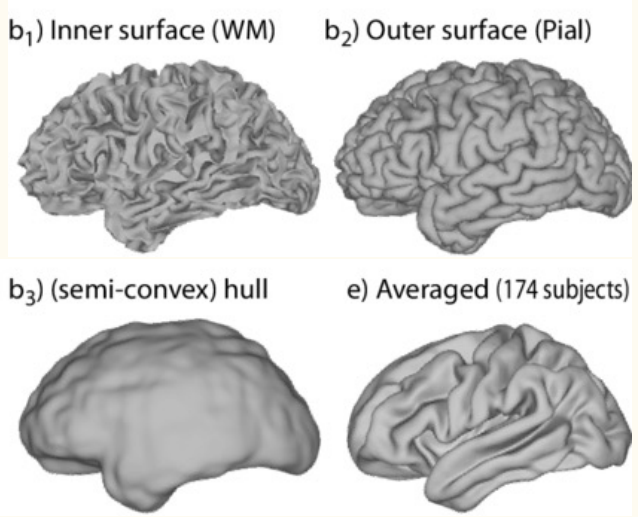
Surface positions are represented by vertices



Volumetric data can be mapped to the surface at varying depths



There are multiple surface geometries



There are native and template surfaces

Surface-based Analyses

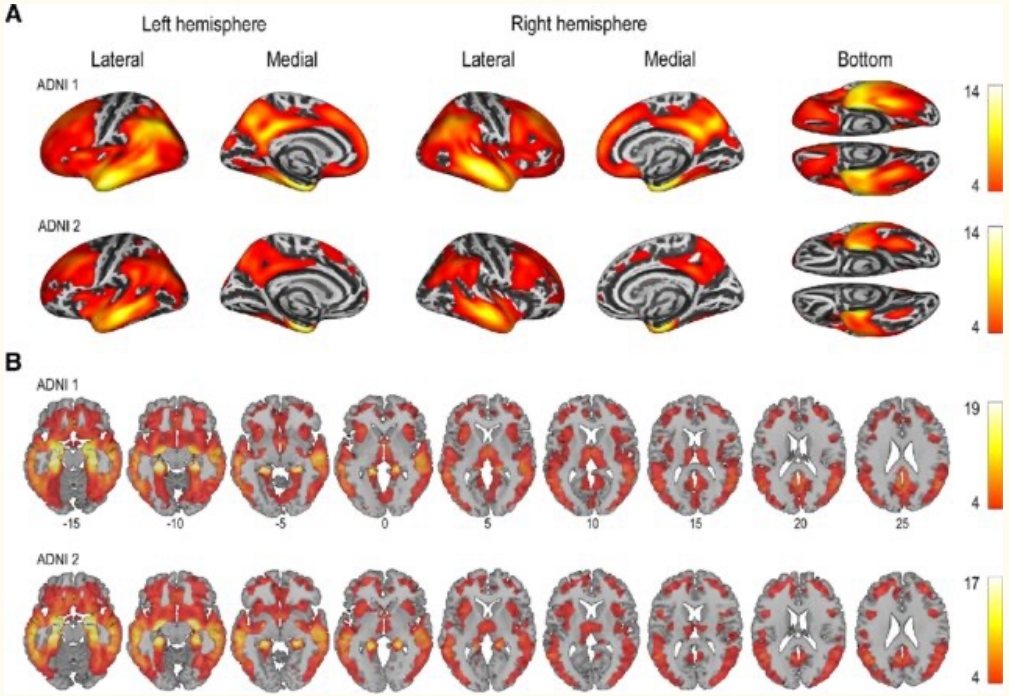
Project your data to a (usually cortical) surface mesh

- Easier data visualization for cortical localization
- Preserves true surface anatomy (e.g., folds and distance)

- Better smoothing, dilation, cluster correction (?) as it can be informed by surface geometry

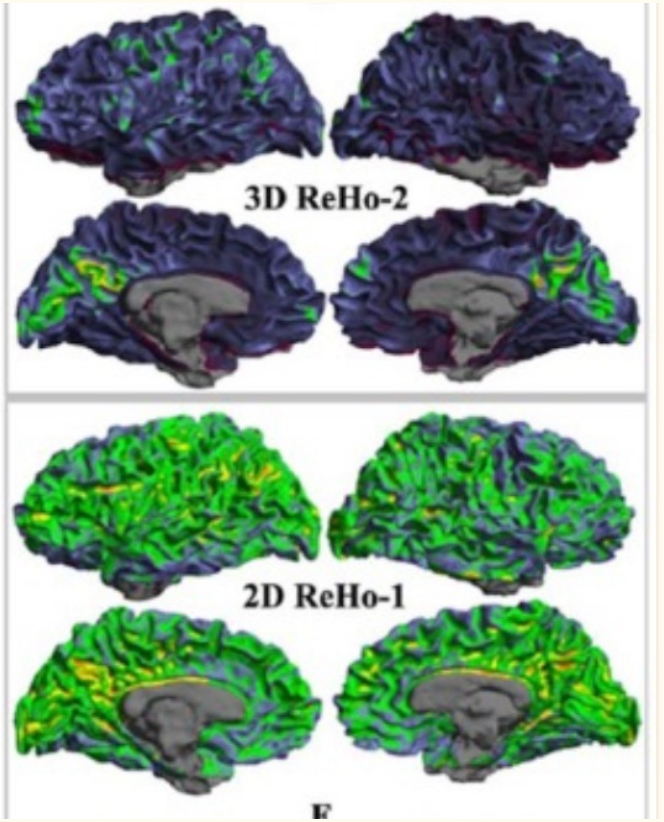
- Better spatial normalization (for cortex) and alignment of cortical landmarks
- Higher SNR due to better across-subject alignment in measures

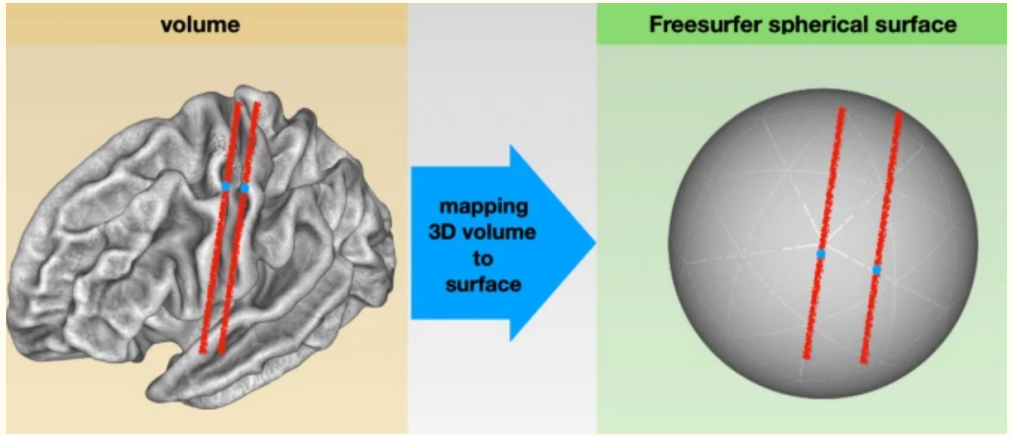
Surface-based Analyses



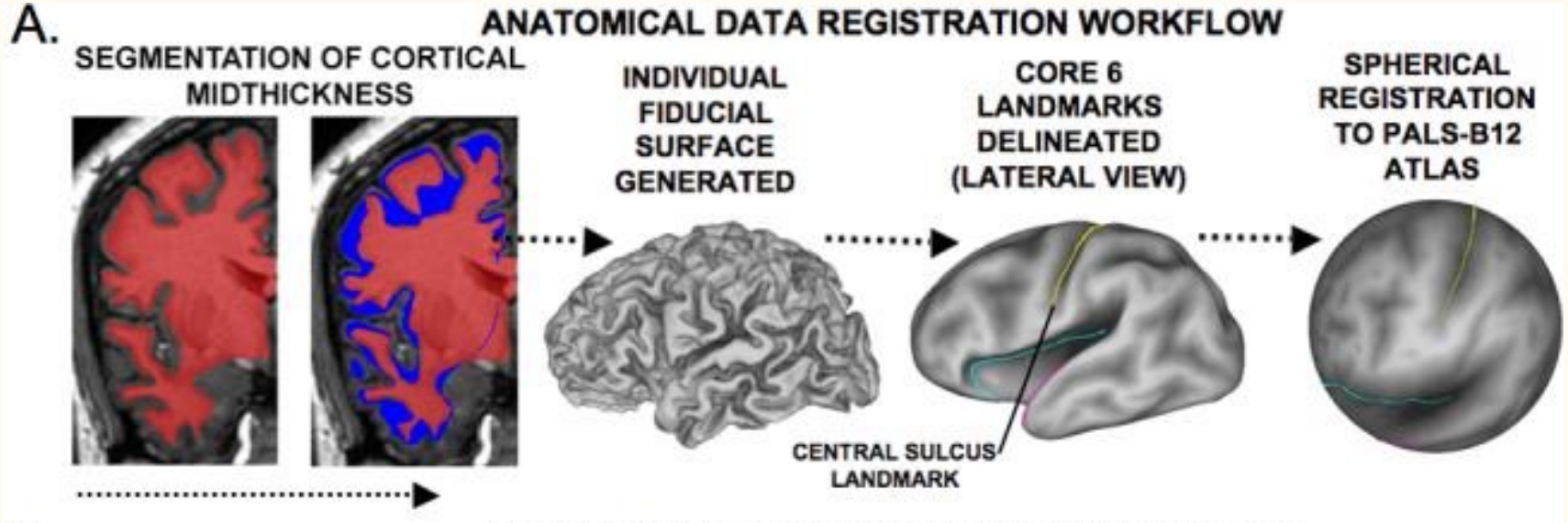
Article | Open access | Published: 31 March 2020

Surface-based analysis increases the specificity of cortical activation patterns and connectivity results





Surface-based Analyses



Surface-based Analyses

Surface data comes in different flavors

- Ciftis and giftis / mgh
- Scalar and label data

Surface-based Analyses

In freesurfer:

singularity exec --writable-tmpfs -B $input_image_dir:/input -B $freesurfer_dir:/opt/freesurfer/subjects -B $freesurfer_license:/opt/freesurfer/license.txt $freesurfer_sif mri_vol2surf --src /input/$input_image_name --hemi $hemi --projfrac $depth --srcreg /opt/freesurfer/subjects/$subject_id/mri/${moving_image_basename}_coreg_T1.lta --out /opt/freesurfer/subjects/$subject_id/surf/${hemi}.${input_image_type}.${depth}%.mgh --interp trilinear

In neuromaps, for MNI data:

```
import nibabel as nib
import pandas as pd
from neuromaps import transforms

#Read in the MNI-space EEG atlas and map it to the fsaverage surface
EEG_volume = nib.load("/Volumes/Hera/Projects/corticalmyelin_development/Maps/EEG_electrode_atlas/electrodeLo
EEGatlas_fsaverage = transforms.mni152_to_fsaverage(EEG_volume, '164k', method='nearest') #mni to fsaverage
```

Surface-based Analyses

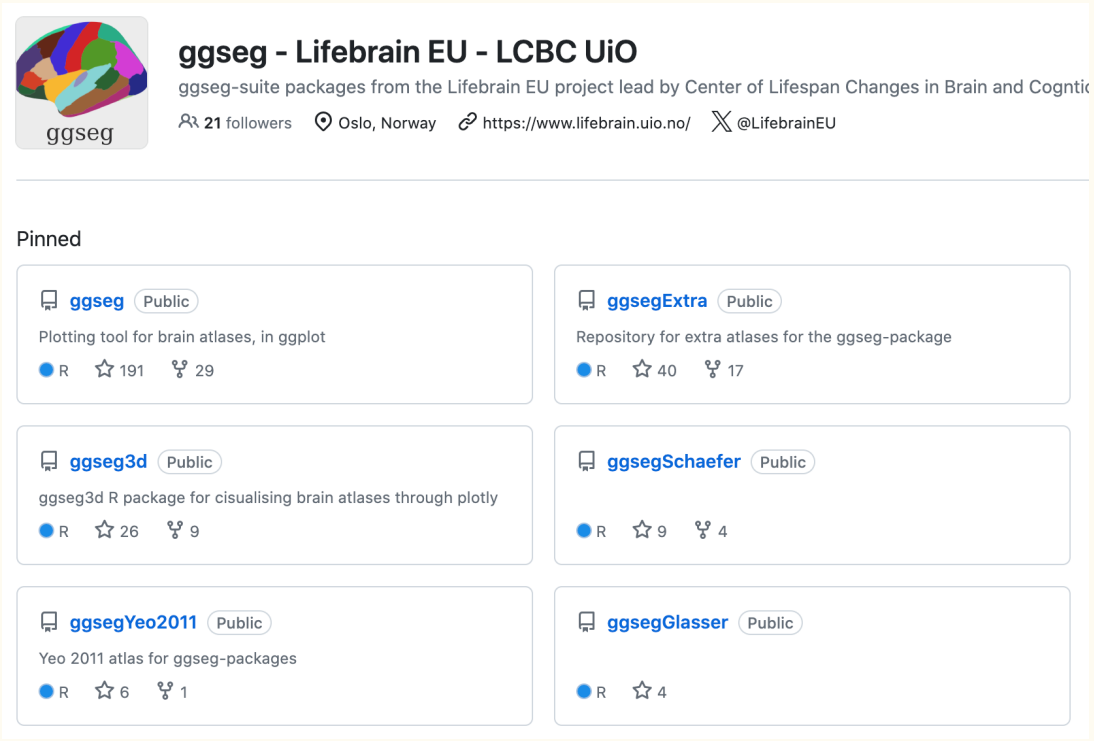
In connectome workbench:

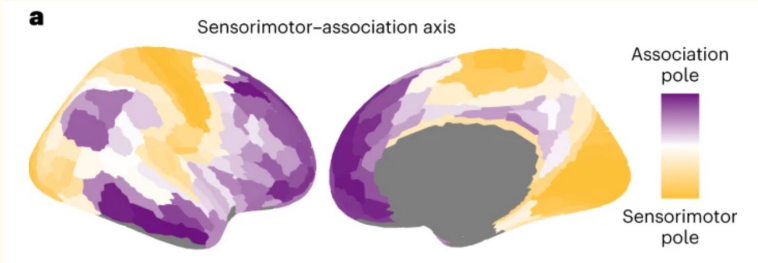
-volume-to-surface-mapping

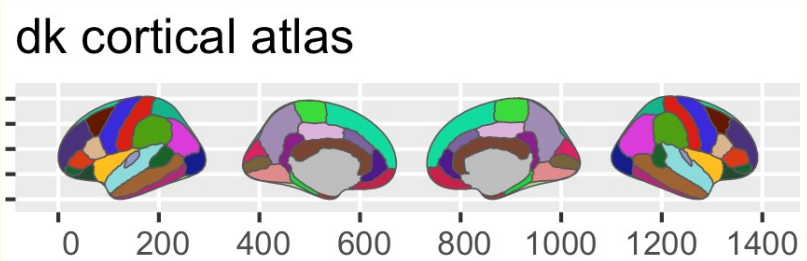
```
MAP VOLUME TO SURFACE
   wb_command -volume-to-surface-mapping
      <volume> - the volume to map data from
      <surface> - the surface to map the data onto
      <metric-out> - output - the output metric file

      [-trilinear] - use trilinear volume interpolation

      [-enclosing] - use value of the enclosing voxel

      [-cubic] - use cubic splines

      [-ribbon-constrained] - use ribbon constrained mapping algorithm
         <inner-surf> - the inner surface of the ribbon
         <outer-surf> - the outer surface of the ribbon

         [-volume-roi] - use a volume roi
            <roi-volume> - the roi volume file

            [-weighted] - treat the roi values as weightings rather than binary

         [-voxel-subdiv] - voxel divisions while estimating voxel weights
            <subdiv-num> - number of subdivisions, default 3

         [-thin-columns] - use non-overlapping polyhedra

         [-gaussian] - reduce weight to voxels that aren't near <surface>
```

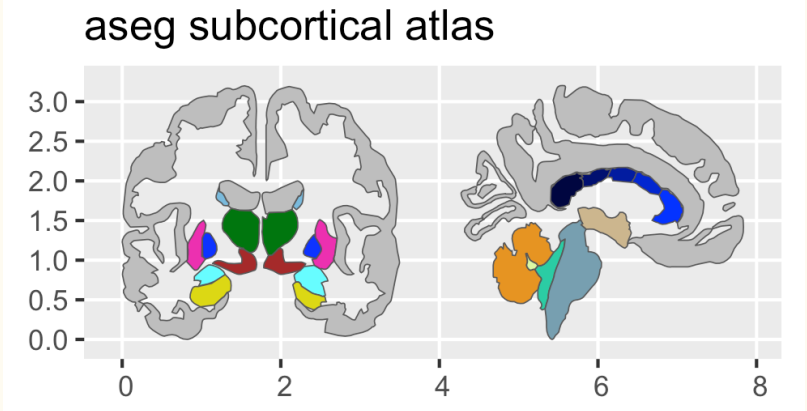
Visualization Tools

ggseg: parcellated cortical and subcortical data





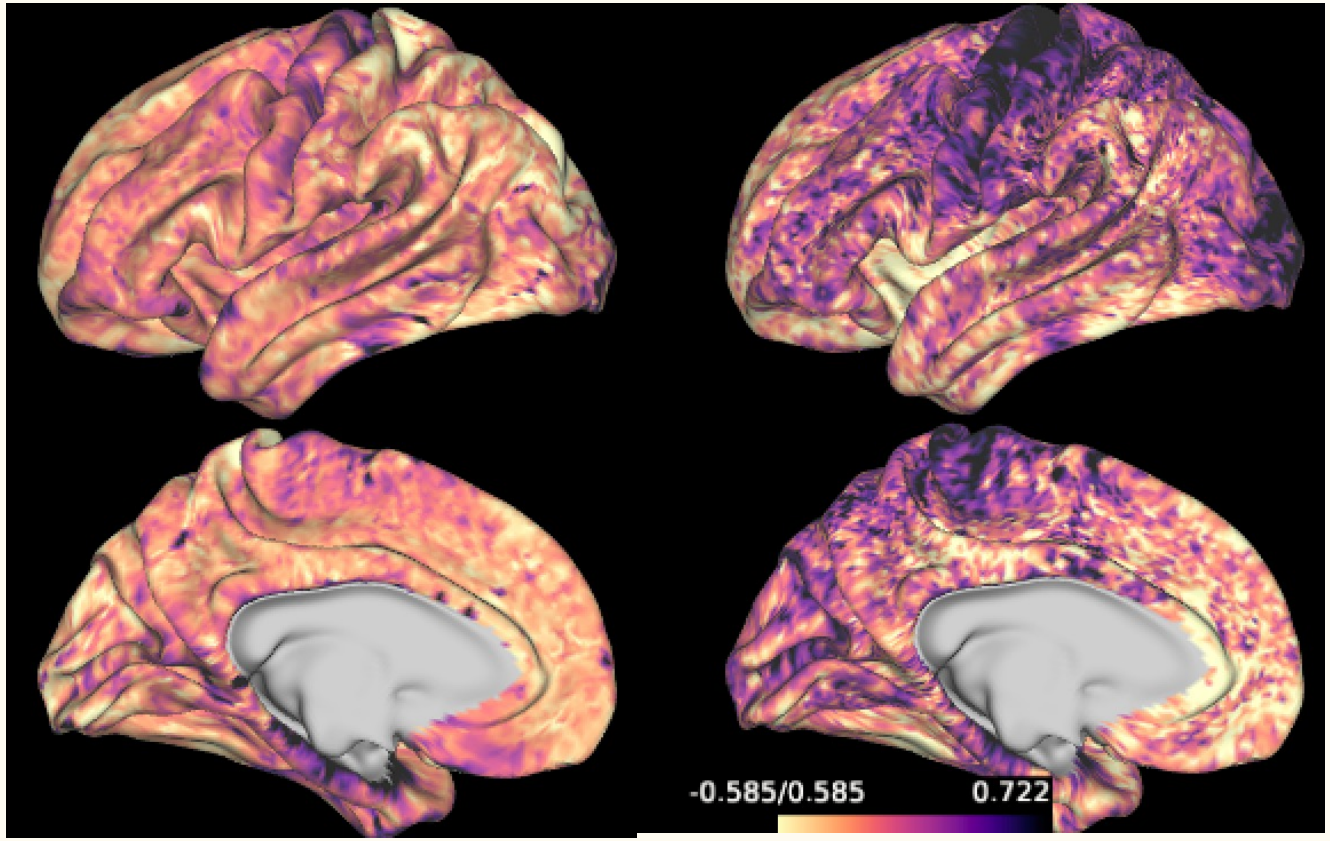


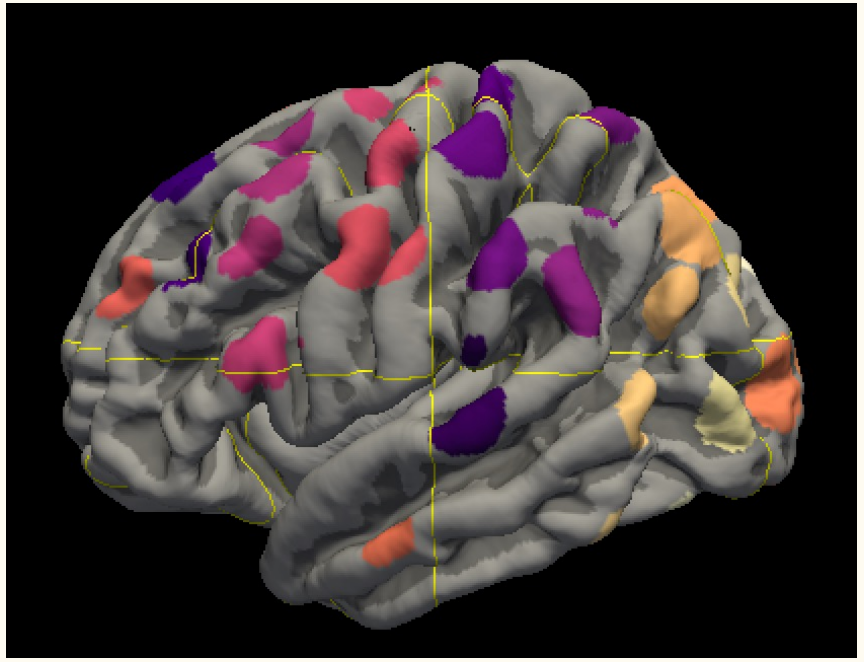


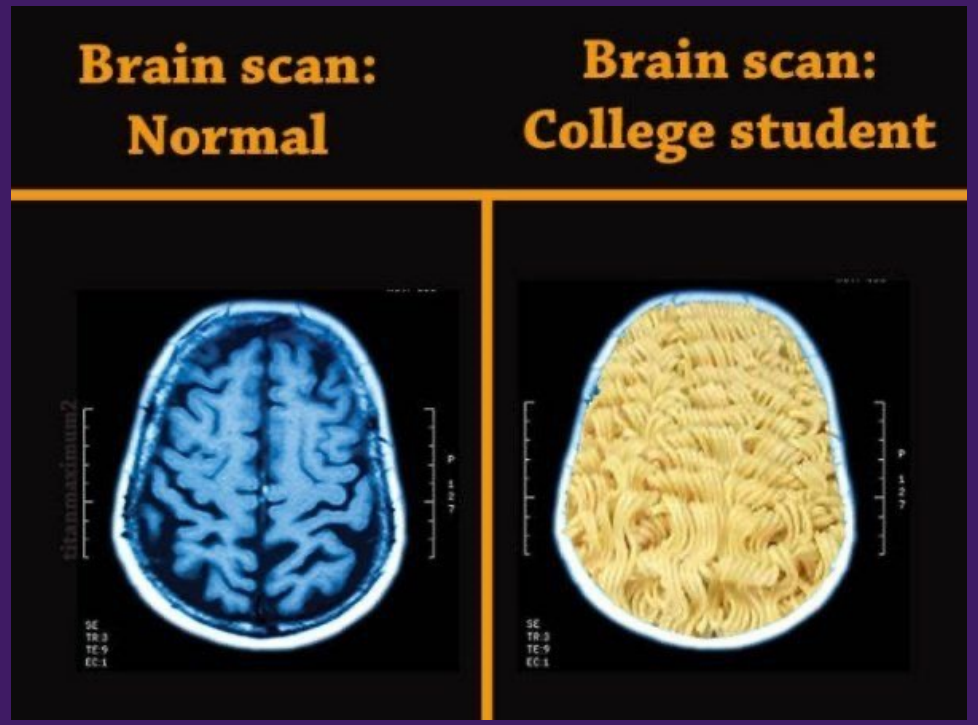

```
ggseg(.data = ALFF.sub.df, atlas = "glasser", mapping=aes(fill = mean.ALFF), position = c("stacked")) + paletteer::scale_fill_paletteer_c("pals::ocean.matter", direction = -1)
```

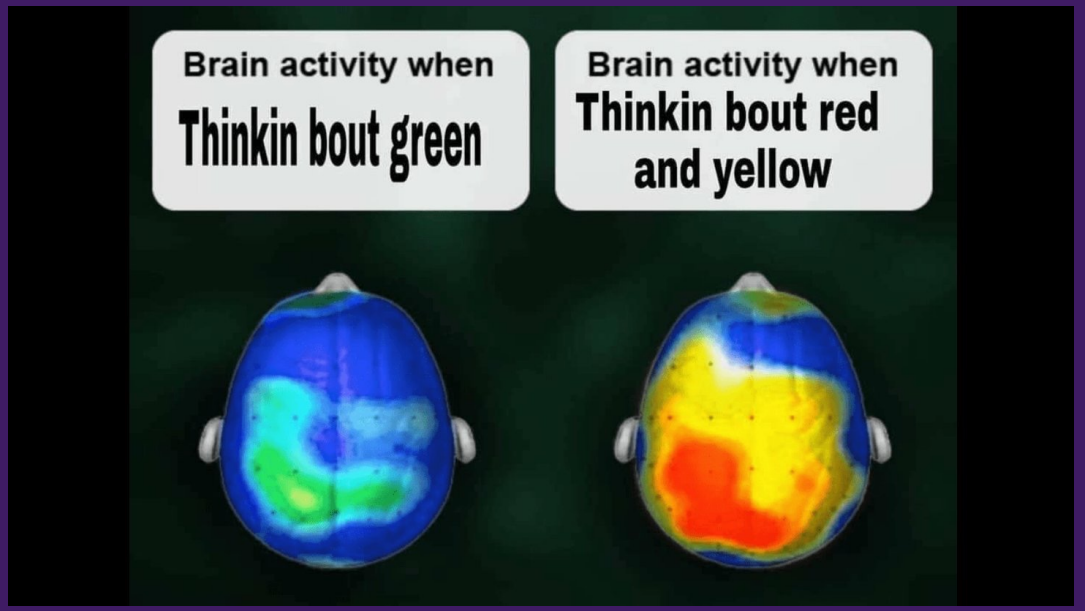
Visualization Tools

Connectome workbench and freeview: vertex surface and voxel volume data






Brain scan:
Normal
Brain scan:
College student

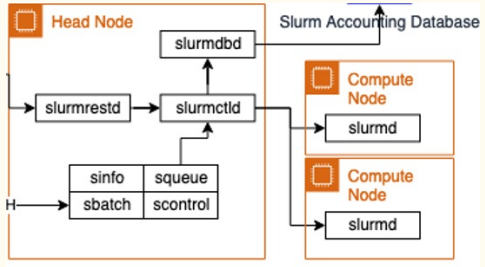

Brain activity when
Thinkin bout green
Brain activity when
Thinkin bout red
and yellow

Data collection → Data organization → Data curation → Data processing →

BIDS
BRAIN IMAGING DATA STRUCTURE

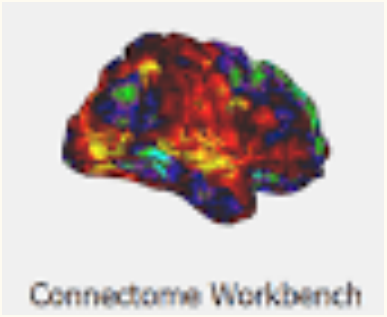
CuBIDS:
Curation of BIDS

docker



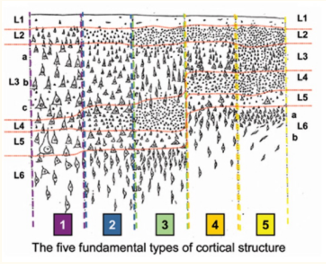
Data visualization → Data interpretation

ggseg

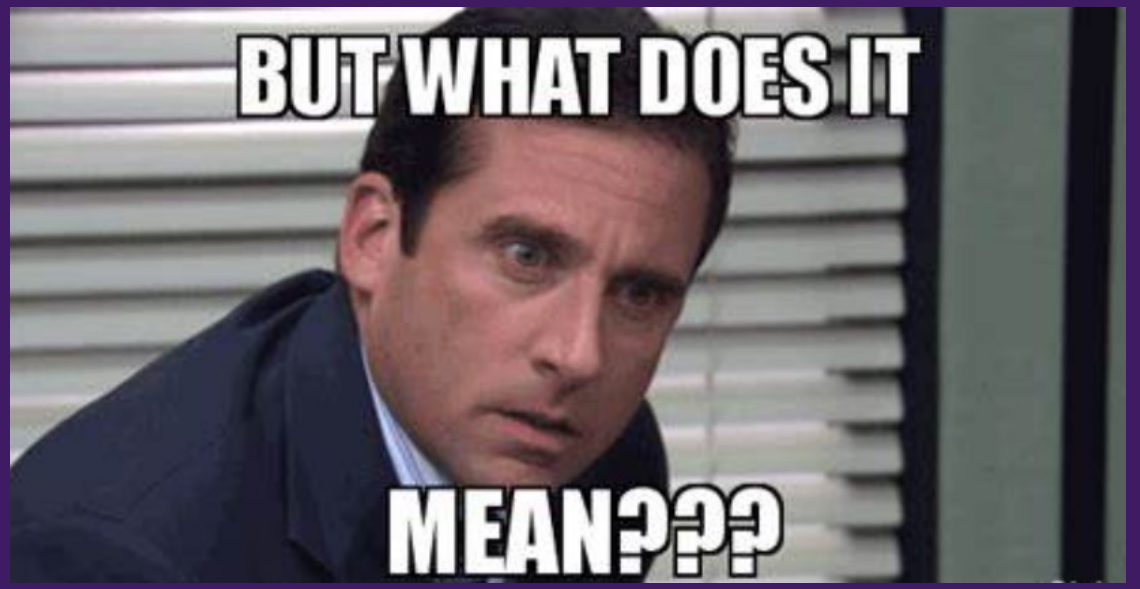


neurosynth.org

neuromaps




DATA
DATA EVERYWHERE

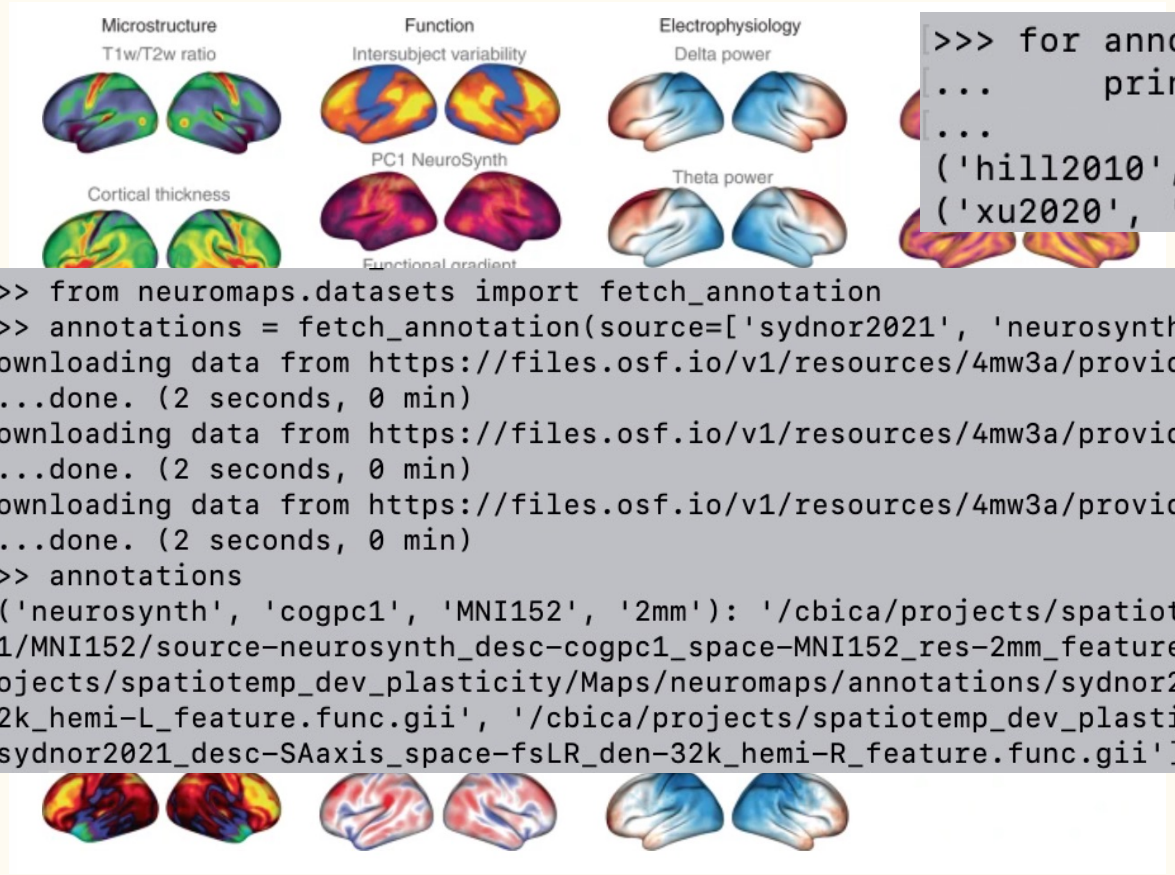

BUT WHAT DOES IT
MEAN???


All right, then. Keep your secrets.
made with mematic

Data annotation tools: brain maps and cognitive terms

neuromaps

```
>>> from neuromaps.datasets import available_annotations
>>> for annotation in available_annotations():
...     print(annotation)
...
('neurosynth', 'cogpc1', 'MNI152', '2mm')
('norgaard2021', 'flumazenil', 'MNI152', '1mm')
('norgaard2021', 'flumazenil', 'fsaverage', '164k')
```



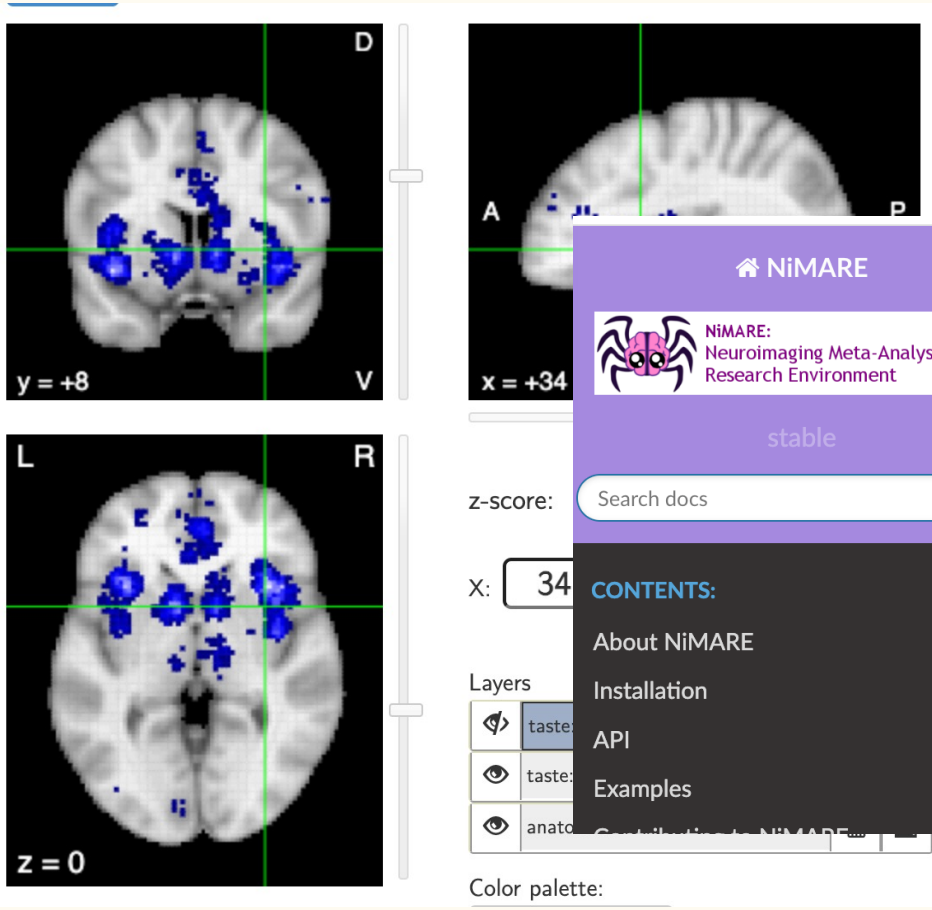

```
>>> for annotation in available_annotations(desc=['evo']):
...     print(annotation)
...
('hill2010', 'evoexp', 'fsLR', '164k')
('xu2020', 'evoexp', 'fsLR', '32k')
```

```
>>> from neuromaps.datasets import fetch_annotation
>>> annotations = fetch_annotation(source=['sydnor2021', 'neurosynth'])
Downloading data from https://files.osf.io/v1/resources/4mw3a/providers/osfstorage/60c22953f3ce9401fa24e651 ...
 ...done. (2 seconds, 0 min)
Downloading data from https://files.osf.io/v1/resources/4mw3a/providers/osfstorage/61e06d4ac99ebd0254018176 ...
 ...done. (2 seconds, 0 min)
Downloading data from https://files.osf.io/v1/resources/4mw3a/providers/osfstorage/61e06d3c466d150209d58122 ...
 ...done. (2 seconds, 0 min)
>>> annotations
{('neurosynth', 'cogpc1', 'MNI152', '2mm'): '/cbica/projects/spatiotemp_dev_plasticity/Maps/neuromaps/annotations/neurosynth/cogpc1/MNI152/source-neurosynth_desc-cogpc1_space-MNI152_res-2mm_feature.nii.gz', ('sydnor2021', 'SAaxis', 'fsLR', '32k'): ['/cbica/projects/spatiotemp_dev_plasticity/Maps/neuromaps/annotations/sydnor2021/SAaxis/fsLR/source-sydnor2021_desc-SAaxis_space-fsLR_den-32k_hemi-L_feature.func.gii', '/cbica/projects/spatiotemp_dev_plasticity/Maps/neuromaps/annotations/sydnor2021/SAaxis/fsLR/source-sydnor2021_desc-SAaxis_space-fsLR_den-32k_hemi-R_feature.func.gii']}
```

```
('tuominen', 'feobv', 'MNI152', '2mm')
('turtonen2020', 'carfentanil', 'MNI152', '1mm')
('xu2020', 'FChomology', 'fsLR', '32k')
('xu2020', 'evoexp', 'fsLR', '32k')
```

Data annotation tools: brain maps and cognitive terms

neurosynth.org



NiMARE

NiMARE: Neuroimaging Meta-Analysis Research Environment

stable

Search docs

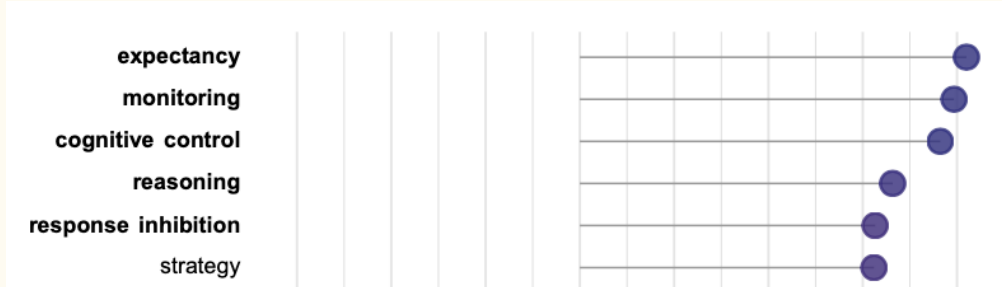
CONTENTS:

About NiMARE

Installation

API

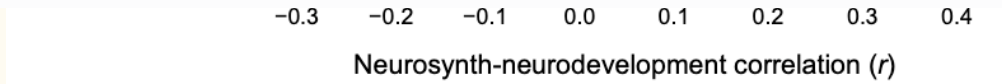
Examples



Download Neurosynth

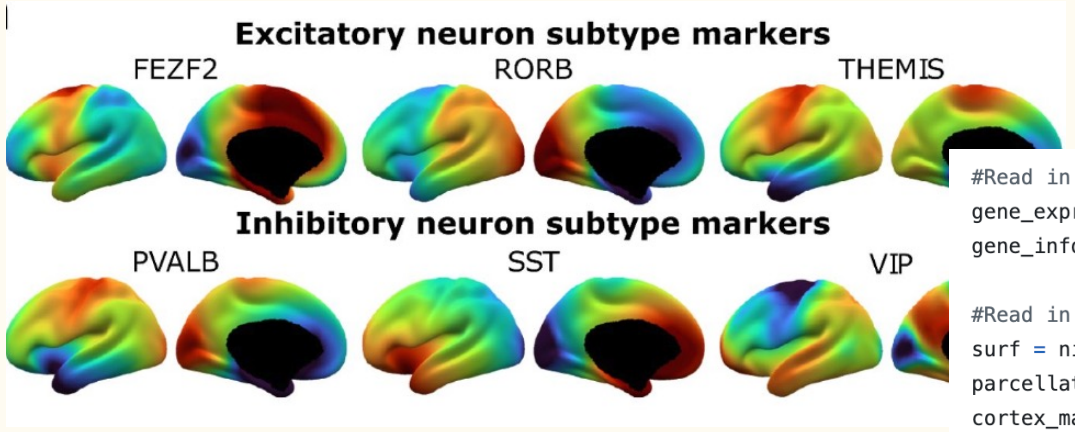
Neurosynth's data files are stored at https://github.com/neurosynth/neurosynth-data.

```
out_dir = os.path.abspath("../example_data/")
os.makedirs(out_dir, exist_ok=True)

files = fetch_neurosynth(
    data_dir=out_dir,
    version="7",
    overwrite=False,
    source="abstract",
    vocab="terms",
)
# Note that the files are saved to a new folder within "out_dir" named "neurosynth".
pprint(files)
neurosynth_db = files[0]
```



Data annotation tools: gene expression (AHBA)

Multiscale Atlas of Gene Expression for Integrative Cortical Cartography

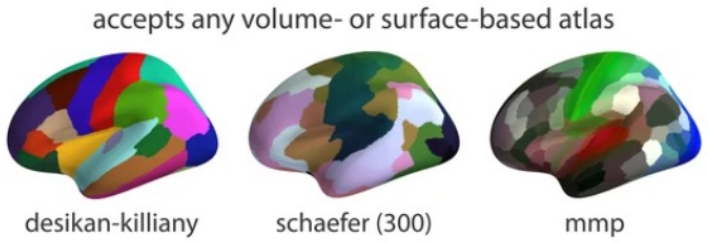



```
#Read in MAGICC gene expression and gene info data
gene_expression = np.load('/Volumes/Hera/Projects/corticalmyelin_development/Maps/AHBA_magicc/magicc_e
gene_info = pd.read_csv('/Volumes/Hera/Projects/corticalmyelin_development/Maps/AHBA_magicc/magicc_exp

#Read in relevant gifti files and create a cortex mask
surf = nib.load('/Volumes/Hera/Projects/corticalmyelin_development/Maps/AHBA_magicc/magicc_expression_
parcellation = nib.load('/Volumes/Hera/Projects/corticalmyelin_development/Maps/AHBA_magicc/magicc_exp
cortex_mask = parcellation.darrays[0].data>0 #cortex versus medial wall mask

#Get across-donor normalized gene expression for user-input gene of interest
mygene_name = sys.argv[1]
mygene_index = np.where(gene_info['gene.symbol']==mygene_name)[0][0]
mygene_expression = gene_expression[mygene_index]
mygene_expression_masked = mygene_expression*cortex_mask
mygene_expression_masked = mygene_expression_masked.astype(np.float32)

#Create lh gifti image and save
data_lh = nib.gifti.gifti.GiftiImage()
data_lh.add_gifti_data_array(nib.gifti.gifti.GiftiDataArray(data = mygene_expression_masked))
data_lh.meta['AnatomicalStructurePrimary'] = 'CortexLeft'
outputpath = sys.argv[2]
fname_lh = 'source-magicc_desc-{0}expression_space-fsLR_den-32k_hemi-L.func.gii'.format(mygene_name)
nib.save(data_lh, outputpath + "/" + fname_lh)
```

Data annotation tools: gene expression (AHBA)

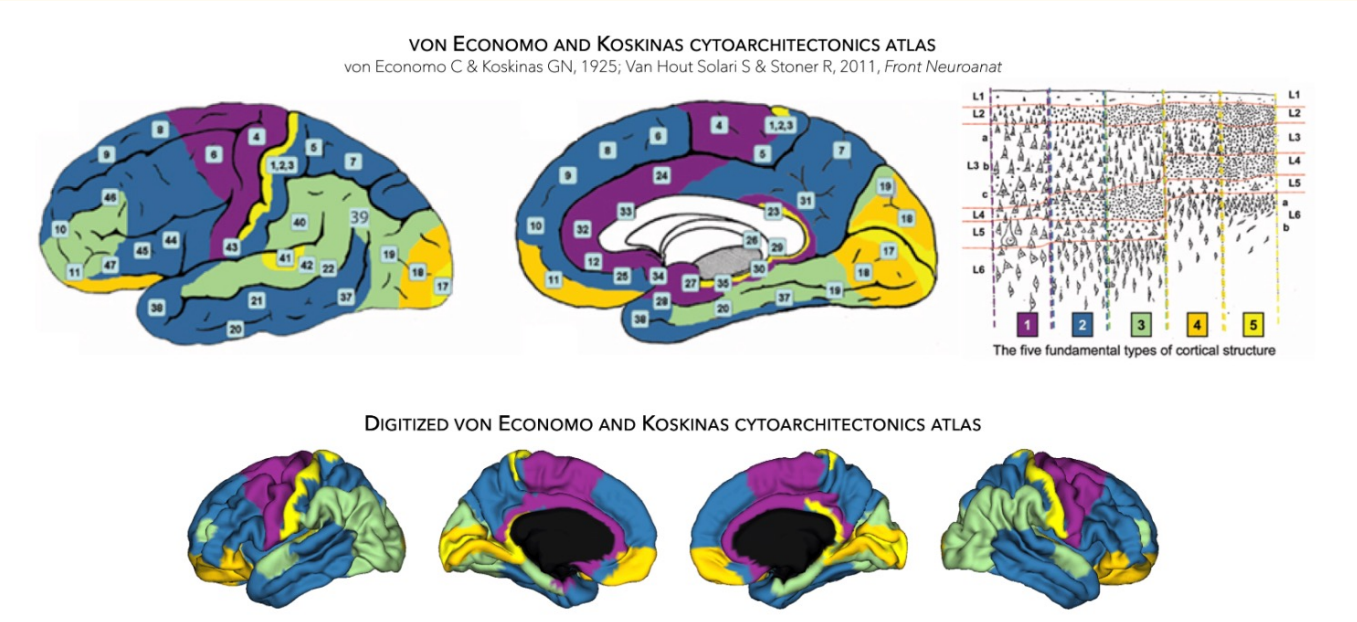
abagen: A toolbox for the Allen Brain Atlas genetics data

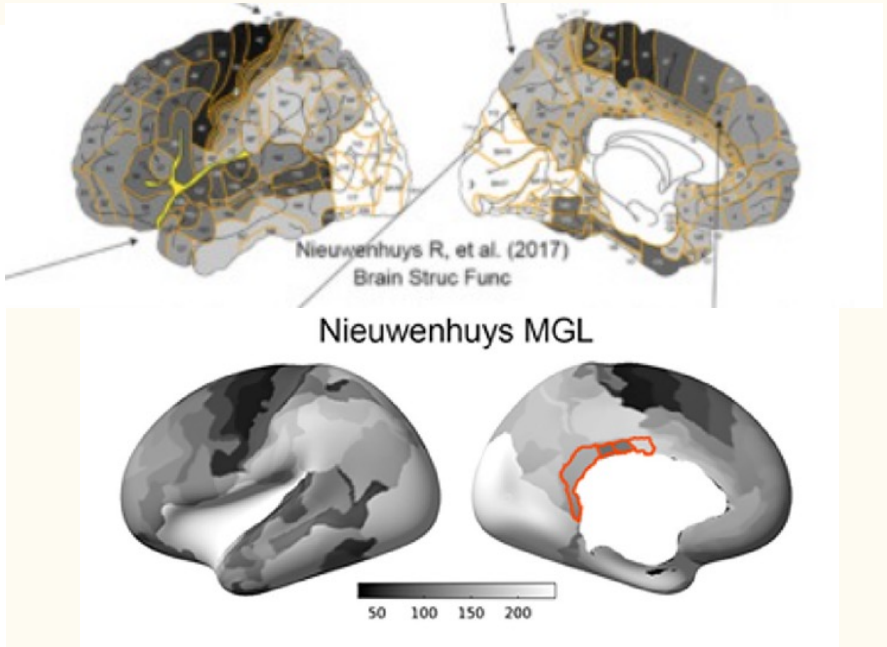
c | abagen features




```
# Set up abagen analysis parameters
runargs=(
        --verbose #verbose, turn on python parseltongue
        --data-dir ${project_dir}/software/abagen/microarray #download donor expression data to h
        --n-proc 6 #use one processor per donor to download the AHBA data
        --probe-selection rnaseq #select the probe with the highest correlation to rnaseq data
        --lr_mirror bidirectional #mirror microarray expression samples across hemispheres to inc
        --missing interpolate #assign nodes in missing regions the nearest tissue sample and crea
        --norm-all #normalizing across matched samples only should be set to false (i.e., the nor
        --tolerance 2 #use the default tolerance of 2 standard deviations for matching tissue sam
        --sample-norm scaled_robust_sigmoid #method to use for within-sample, across-gene normali
        --gene-norm scaled_robust_sigmoid #method to use to normalize gene-specific expression va
        --norm_structures #perform gene-norm only within structural classes (here, cortex by hemi
        --region-agg donors #use default (donors, not samples) option for averaging (agg-metric =
        --agg-metric mean
        --output-file ${project_dir}/Maps/AHBA/AHBA_geneexpression_glasser.csv #path where the re
)

# Run command line abagen to process AHBA data and generate gene x region matrices
abagen "${runargs[@]}" ${atlas_dir}/glasser_space-fsaverage5_den-10k_desc-atlas_hemi-R.label.gii
```

Data annotation tools: cortical cyto- and myelo-architecture



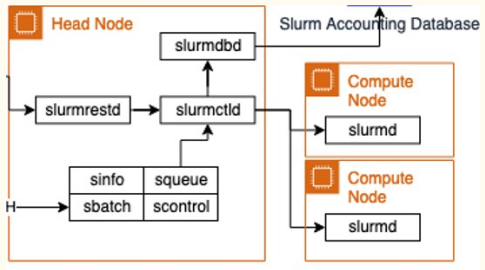


Data collection → Data organization → Data curation → Data processing →

BIDS
BRAIN IMAGING DATA STRUCTURE

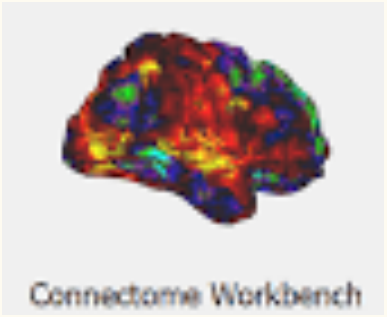
CuBIDS:
Curation of BIDS

docker



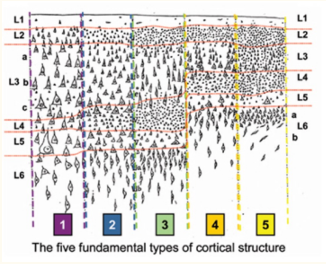
Data visualization → Data interpretation → Data and code sharing

ggseg



neurosynth.org

neuromaps



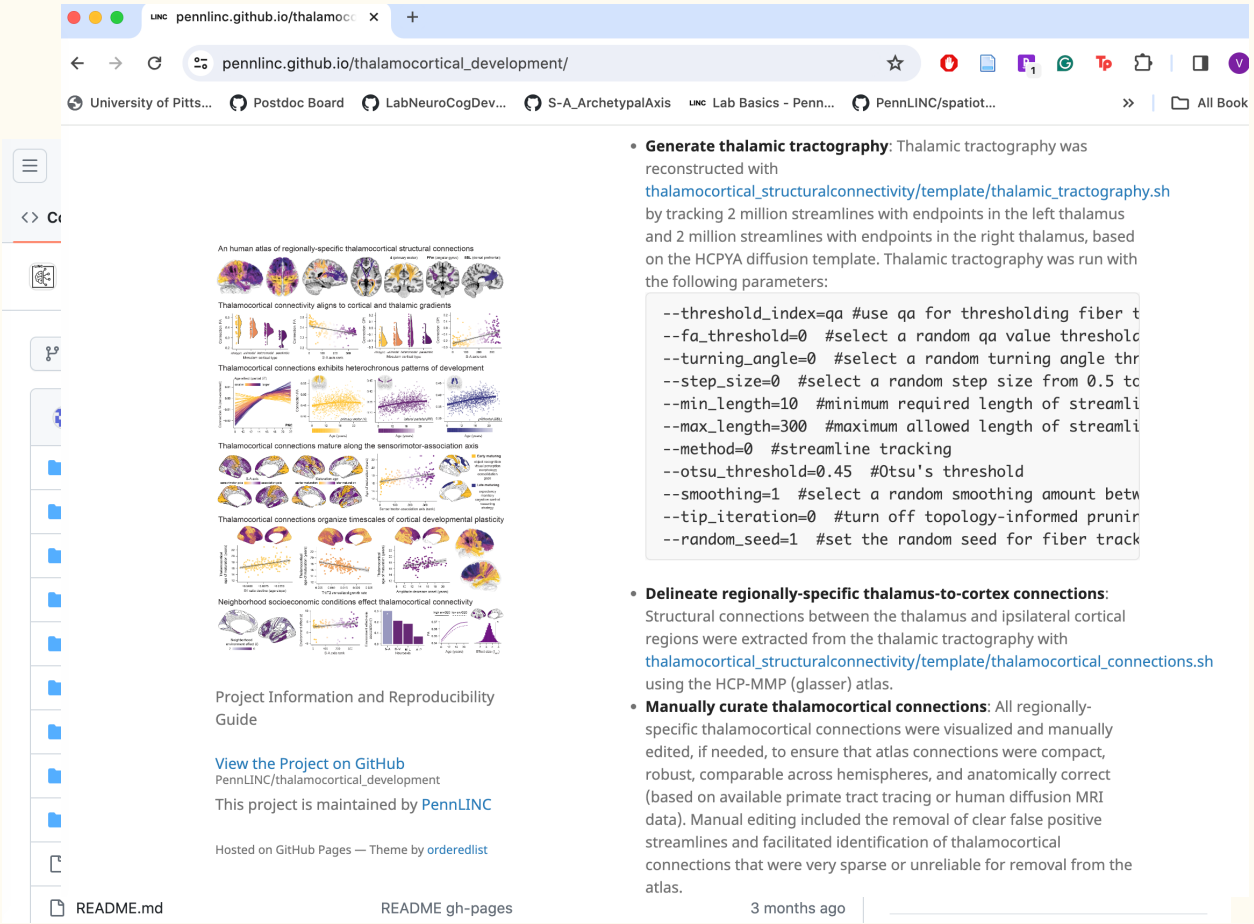
GitHub Pages

The project will have data documentation

Github and Github pages

Code storage, version history, collaboration (git)

Code descriptions (gh-pages)



Github and Github pages

```
title: <br>
logo: ./ProjectFigure.png
description: <br>Project Information and Reproducibility Guide
theme: jekyll-theme-minimal
```

```
<br>
<br>
# CODE DOCUMENTATION

The analytic and statistical workflow implemented in this research is described below and links to all corresponding code
<br>

### Creation of an Atlas of Human Thalamocortical Connections (HCP-Young Adult)
A novel thalamocortical structural connectivity tractography atlas was generated using a high quality diffusion template c

The thalamocortical structural connectivity tractography atlas was generated in the following steps:

* **Generate thalamic tractography**: Thalamic tractography was reconstructed with [thalamocortical_structuralconnectivity
```
--threshold_index=qa #use qa for thresholding fiber tracking
--fa_threshold=0 #select a random qa value threshold as termination criterion for each streamline
--turning_angle=0 #select a random turning angle threshold between 15-90 degrees as termination criterion for each stream
--step_size=0 #select a random step size from 0.5 to 1.5 voxels for each streamline
--min_length=10 #minimum required length of streamlines
--max_length=300 #maximum allowed length of streamlines
--method=0 #streamline tracking
--otsu_threshold=0.45 #Otsu's threshold
--smoothing=1 #select a random smoothing amount between 0% to 95% for each streamline; smoothing uses previous propagatic
--tip_iteration=0 #turn off topology-informed pruning
--random_seed=1 #set the random seed for fiber tracking
```
```