Reproducible, generalizable, and scalable analytic software for large neuroimaging datasets

Dissertation Proposal Defense

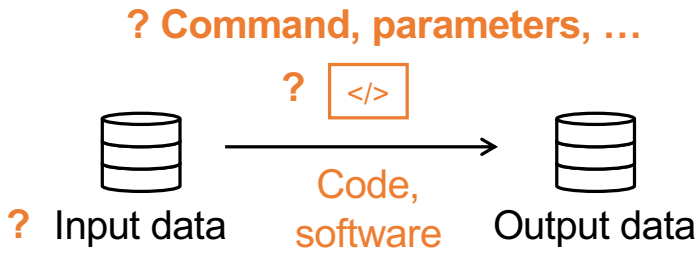
Chenying Zhao

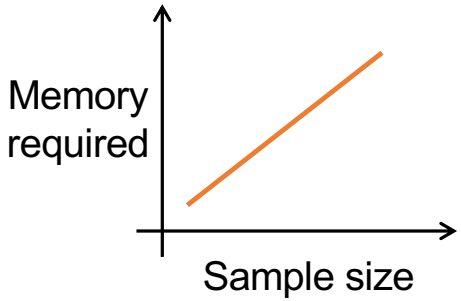
April 1st, 2022

Penn
University of Pennsylvania

Overview – The challenge: Reproducibility in large-scale neuroimaging research

- Reproducibility crisis in neuroscience and neuroimaging, esp. with large-scale datasets
  - Fully reproducible research requires a full audit trail of data processing; however existing tools are not user-friendly
  - Memory intensive for e.g., mass-univariate hypothesis testing
  - Analytic tools are often not generalizable for various image types, data types and processing methods.





- fMRI, dMRI, …
- Volume, surface data, …
- QSIPrep, ASLPrep, …

Reproducibility **Scalability** **Generalizability**

Overview – Proposed solutions

To develop reproducible, generalizable, and scalable analytic software for large neuroimaging datasets

- **Aim 1:** To develop and release an open-source software package for reproducible image processing at scale.
- **Aim 2:** To develop and release a generalizable, memory-efficient, open-source R package for mass-univariate hypothesis testing of large neuroimaging datasets.

Outline

- Aim 1: Software for reproducible image processing at scale
- Aim 2: Memory-efficient, generalizable software for statistical analysis
- Software development

Outline

- Aim 1: Software for reproducible image processing at scale
- Aim 2: Memory-efficient, generalizable software for statistical analysis
- Software development

Outline: Approach for Aim 1

To develop and release an open-source software package for reproducible image processing at scale

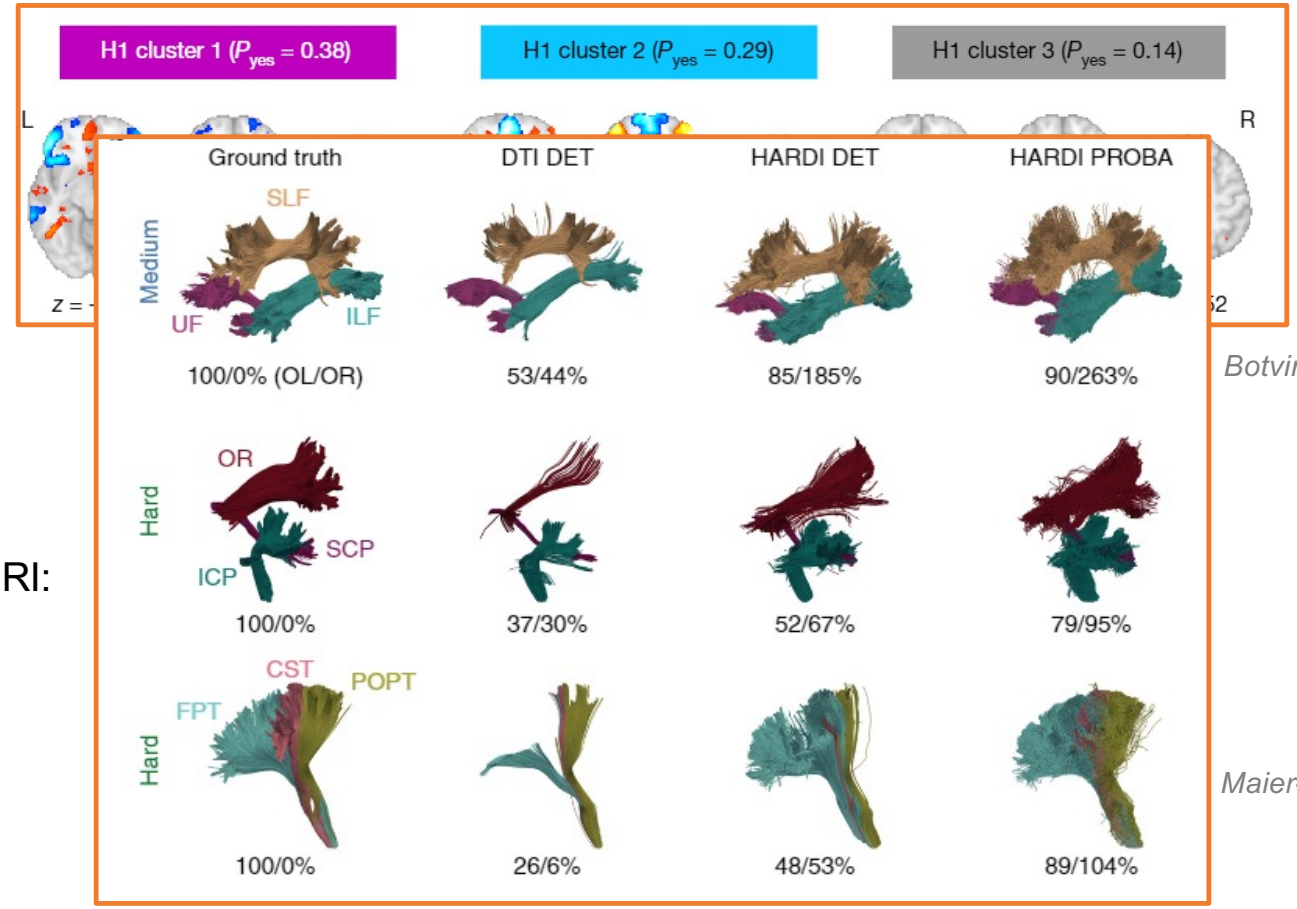
- Approach for **Aim 1**:
  - Background
  - BIDS and BIDS apps
  - Method for data provenance tracking using DataLad
  - Workflow and functionality of proposed software
  - Compatibility with computing cluster management systems
  - Evaluation data and details

Aim 1 – Background

Reproducibility crisis in neuroimaging studies

- Even with the same data, different analytical methods and tools can lead to discordant results:

Functional MRI:



Botvinik-Nezer et al., Nature 2020

Diffusion MRI:

Maier-Hein et al., Nat Comm 2017

Aim 1 – Background

Large-scale neuroimaging datasets can enhance statistical power

- Human Connectome Project
  - Young adult (n=1,200)
  - Development (n=1,300)
- Philadelphia Neurodevelopmental Cohort (PNC, n=1,601)
- Healthy Brain Network (n=5,000)
- UK Biobank (n=40,000+)
- …

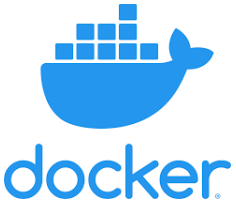
Aim 1 – Background

Challenges in reproducibly processing large-scale datasets and emerging solutions

- Large-scale datasets:
  - ☺ enhance statistical power
  - ☹ complexity, heterogeneity, and huge sample size

- Emerging solutions:

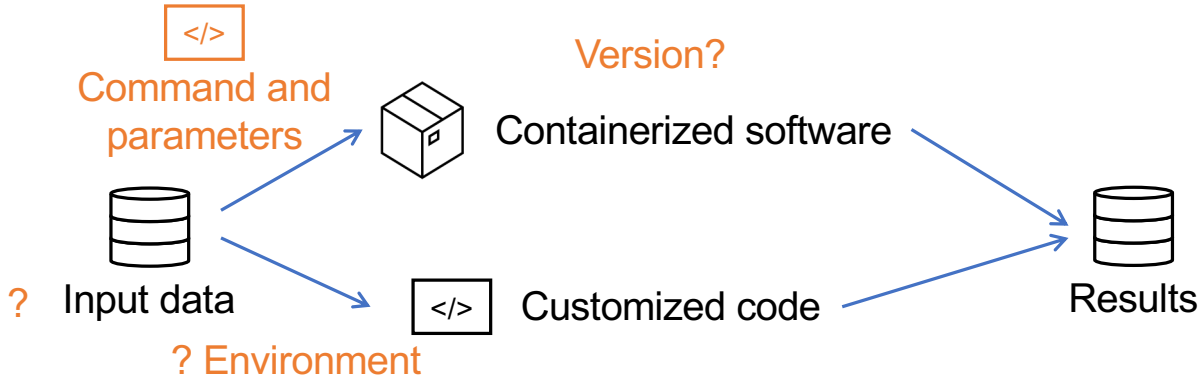
Code version control tools



Portable containers

High performance computing (HPC) clusters

Aim 1 – Still, there are obstacles…
Complete reproducibility with automatic data provenance tracking



✗ Manually track data provenance

✓ Automatically track data provenance

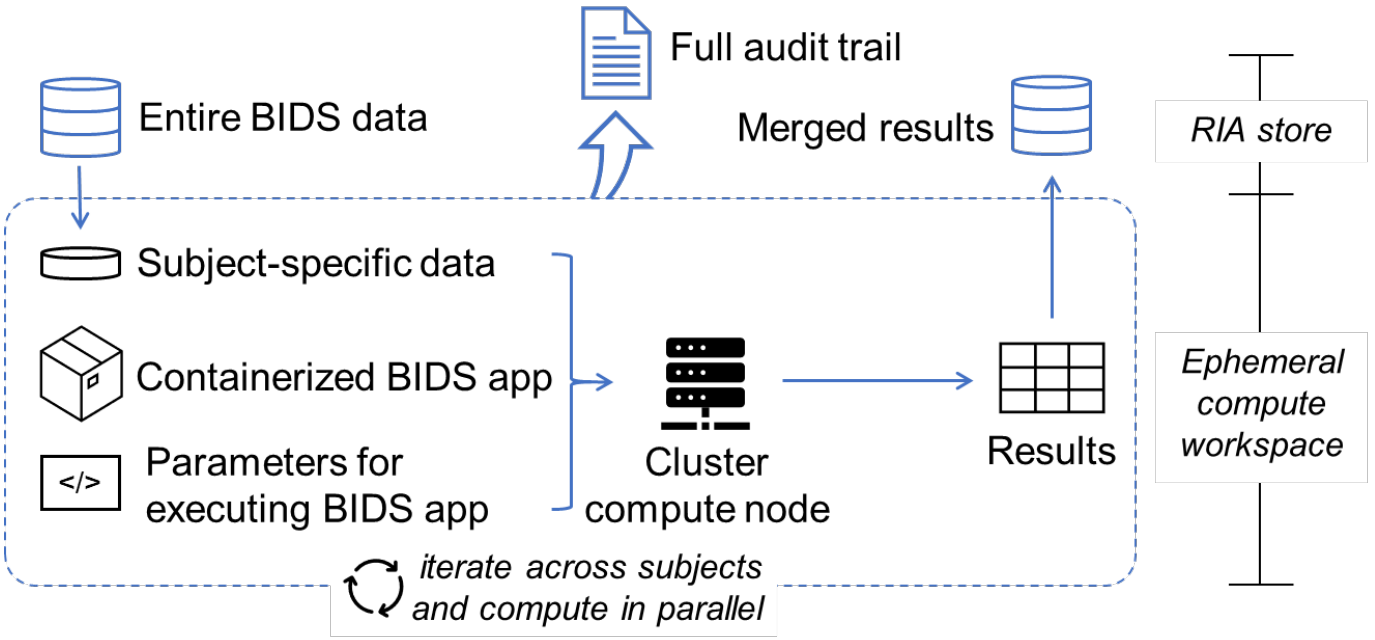
- DataLad[1]
- FAIRly big workflow[2]

- However, not practical for most academic investigators
  - Numerous steps and DataLad functions
  - Highly customized code

[1]Halchenko et al., J Open Source Softw 2021; [2]Wagner et al., bioRxiv 2021

Aim 1 – Proposed software: BABS

Proposed software package: BIDS-App Bootstrap (BABS)



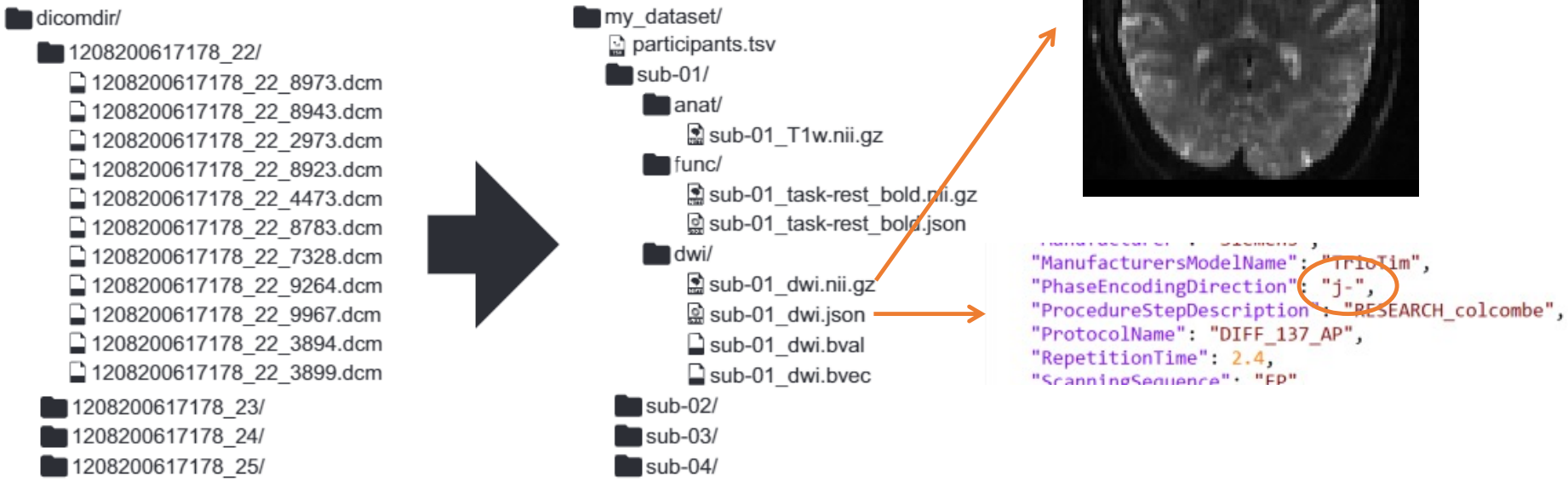
→ Automatically tracking all data provenance

→ User-friendly, generalizable software

Aim 1 – BIDS and BIDS apps

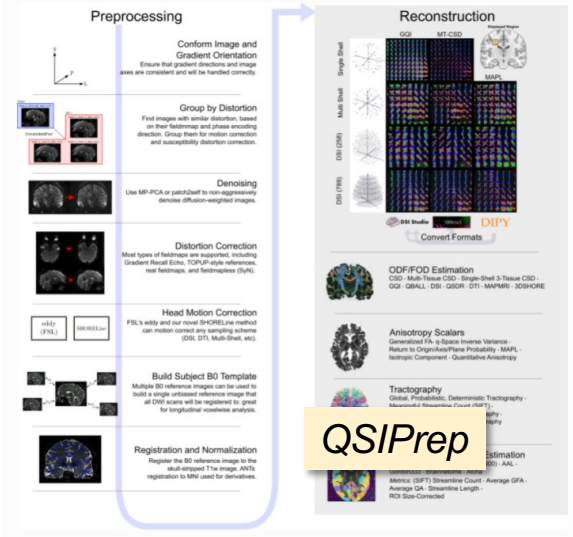
What is Brain Imaging Data Structure (BIDS) format?

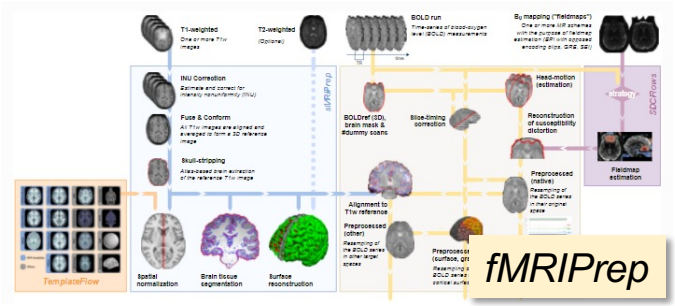
- BIDS includes images + their metadata in sidecar JSON files:



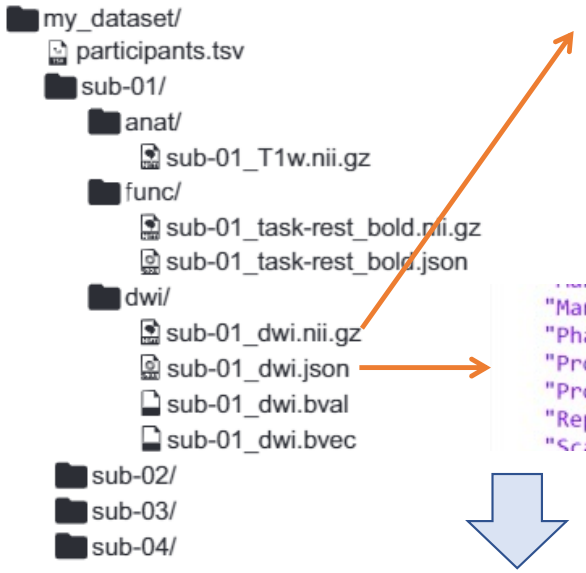
Gorgolewski et al., SciData 2016

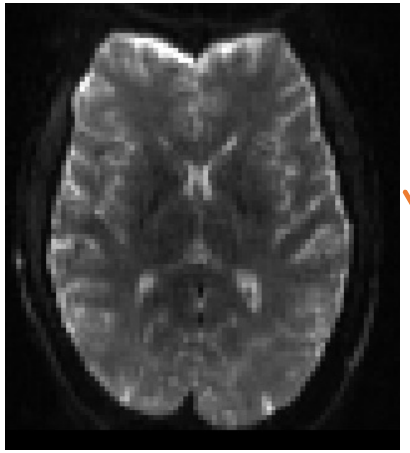
Aim 1 – BIDS and BIDS apps
What are BIDS apps?



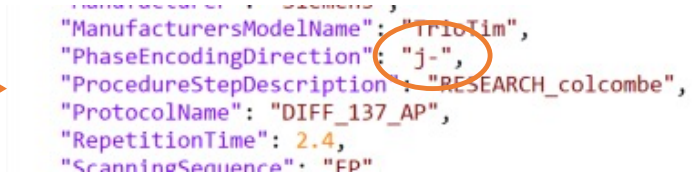


...





Phase encoding direction



BIDS Apps: ✓ Automatically configure

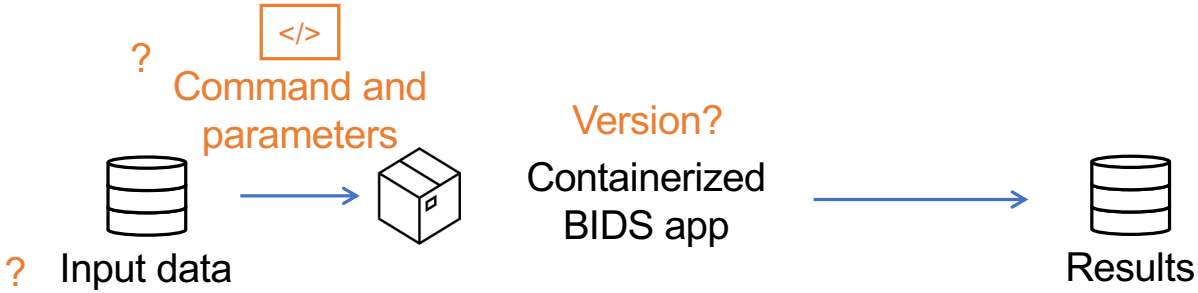
✓ Containerized

Cieslak et al., Nat Methods 2021; Esteban et al., Nat Methods 2019

Gorgolewski et al., Plos Compt Biol 2017

Aim 1 – BIDS and BIDS apps
What is lacked for complete reproducibility?

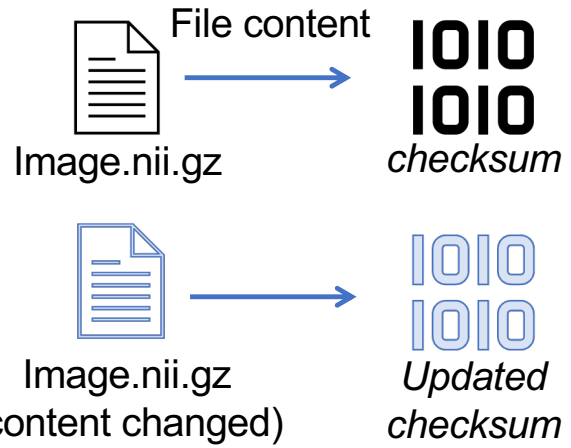
Applying containerized BIDS apps enhances reproducibility, however…



BABS will fill this gap and maximize reproducibility

Aim 1 – Method for data provenance tracking using DataLad
Data version control

For data version control



- No matter how large the file is
- (sensitive) contents in the file cannot be retrieved from the checksum

Aim 1 – Method for data provenance tracking using DataLad

Functions in DataLad

- DataLad provides machine-readable, re-executable provenance record

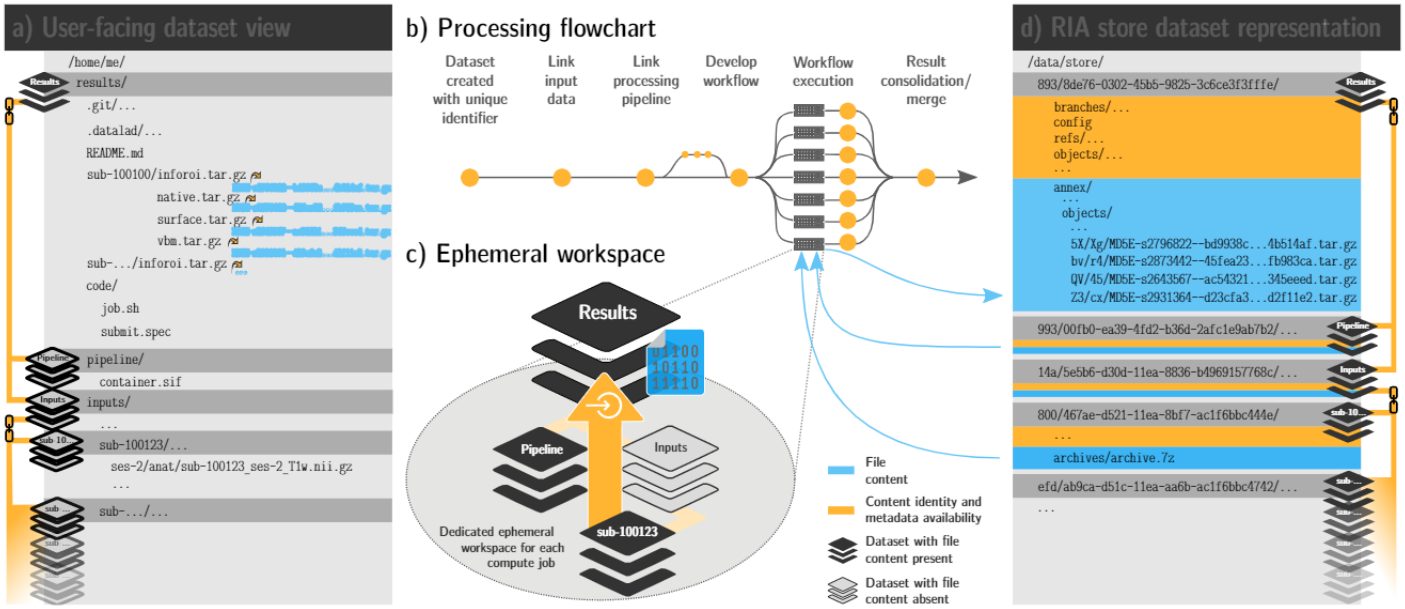
Example provenance record:

```
commit e035f896s45c9fac70cn7cc4dbd0dad43907755p
Author:     Jane Doe <j.doe@fz-juelich.de>
AuthorDate: Wed Feb 10 18:05:30 2021 +0100
Commit:     Jane Doe <j.doe@fz-juelich.de>
CommitDate: Wed Feb 10 18:05:30 2021 +0100

[DATALAD RUNCMD] Compute sub-6025043/ses-2

   === Do not change lines below ===
   {
    "chain": [],
    "cmd": "singularity exec -B {pwd} --cleanenv code/pipeline/.datalad/
            environments/cat/image sh -e -u -x -c  [...]'
    "dsid": "8938de76-0302-45b5-9825-3c6ce3f3fffe",
    "exit": 0,
    "extra_inputs": [
     "code/pipeline/.datalad/environments/cat/image"
    ],
    "inputs": [
     "inputs/ukb/sub-6025043/ses-2/anat/sub-6025043_ses-2_T1w.nii.gz",
     "code/cat_standalone_batch.txt",
     "code/finalize_job_outputs.sh"
    ],
    "outputs": [
     "sub-6025043/ses-2"
    ],
    "pwd": "."
   }
   ^^^ Do not change lines above ^^^
---
 sub-6025043/ses-2/inforoi.tar.gz | 1 +
 sub-6025043/ses-2/native.tar.gz  | 1 +
 sub-6025043/ses-2/surface.tar.gz | 1 +
 sub-6025043/ses-2/vbm.tar.gz     | 1 +
 4 files changed, 4 insertions(+)
```

Wagner et al., bioRxiv 2021

Aim 1 – Workflow of BABS

- BABS will follow the workflow of FAIRly big framework
  - FAIR = findability, accessibility, interoperability, and reusability
  - FAIRly big is a DataLad-based framework for reproducible processing of large-scale datasets.
  - This facilities a full audit trail for processing data at scale



Wagner et al., bioRxiv 2021

Background – Still, there are obstacles…
Impractical adoption of existing effort

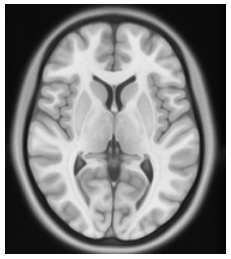
```
#!/bin/bash
# fail on any issue, show commands
set -e -u -x
# name arguments for readability
dssource="$1"
pushgitremote="$2"
subid="$3"

# obtain the analysis dataset, which
# also tracks the required inputs
datalad clone "${dssource}" ds
cd ds

# register location for result
# deposition, separate from the input
# source for performance reasons only
git remote add outputstore
  ↪ "$pushgitremote"

# all job results will be put into
# a job-specific, dedicated branch
git checkout -b "job-$JOBID"

# START OF APPLICATION-SPECIFIC CODE
# pull down input data manually,
# only needed for wildcard-based file
# selection in the next command
datalad get -n "inputs/ukb/${subid}"

# datalad containers-run executes
# the "cat" computational pipeline.
# specified inputs are auto-obtained,
# specified outputs are saved with
# provenance record
datalad containers-run \
  -m "Compute subject ${subid}" \
  -n cat \
  --explicit \
  --output "${subid}" \
  --input
  ↪ "inputs/ukb/${subid}/*T1w.nii.gz"
  "<container invokation arguments>"
# END OF APPLICATION-SPECIFIC CODE

# push result file content to the
# configured "storage-remote"
datalad push --to storage-remote

# push branch with provenance records
# needs a global lock to prevent
# write conflicts
flock "$DSLOCKFILE" git push
  ↪ outputstore

# log entry to mark non-error exit
echo SUCCESS
```

Part of the full script

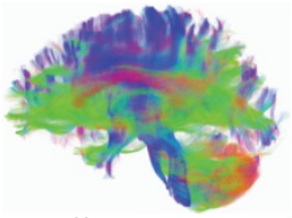
- FAIRly big framework is not user-friendly
- Long scripts
  - Numerous steps …
  - Numerous DataLad functions …

Background – Still, there are obstacles…
Poor generalizability of existing effort

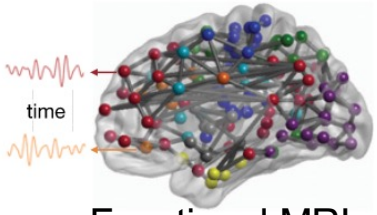
Image types



Structural MRI



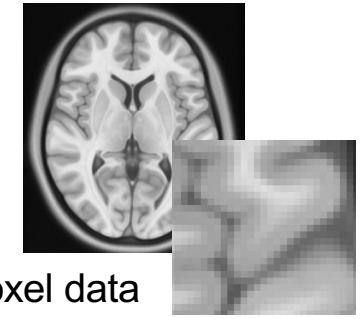
Diffusion MRI



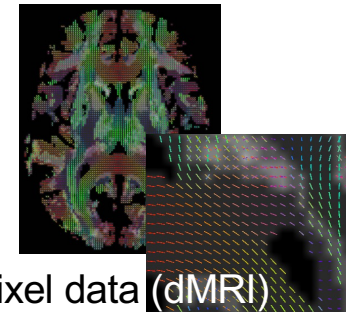
Functional MRI

…

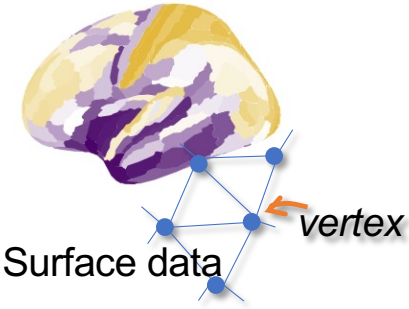
Data types



Voxel data



Fixel data (dMRI)



Surface data

…

Processing methods and software

ANTs

QSIPrep

MRtrix3

…

Figures adapted from: Baum et al., PNAS 2020; Sydnor et al., Neuron 2021

Aim 1 – Still, there are obstacles…
Poor generalizability of existing effort

Datasets

HCP-D
HBN
PNC
NKI-RS
…

×

Image types and Processing BIDS apps

sMRI: FreeSurfer, ANTs, …

fMRI: fMRIPrep, …

dMRI: QSIPrep, MRtrix, …

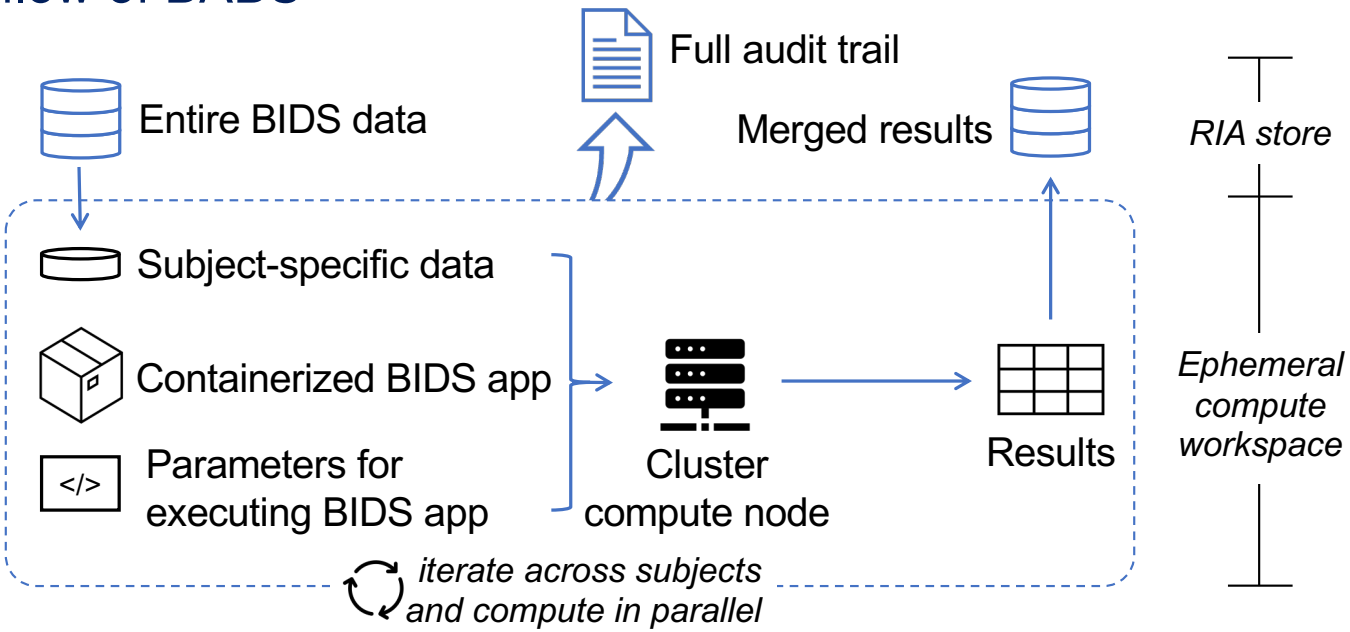
…

How FAIRly big workflow works currently in practice…

Numerous customized scripts for various BIDS apps and datasets

BABS will be user-friendly, generalizable software

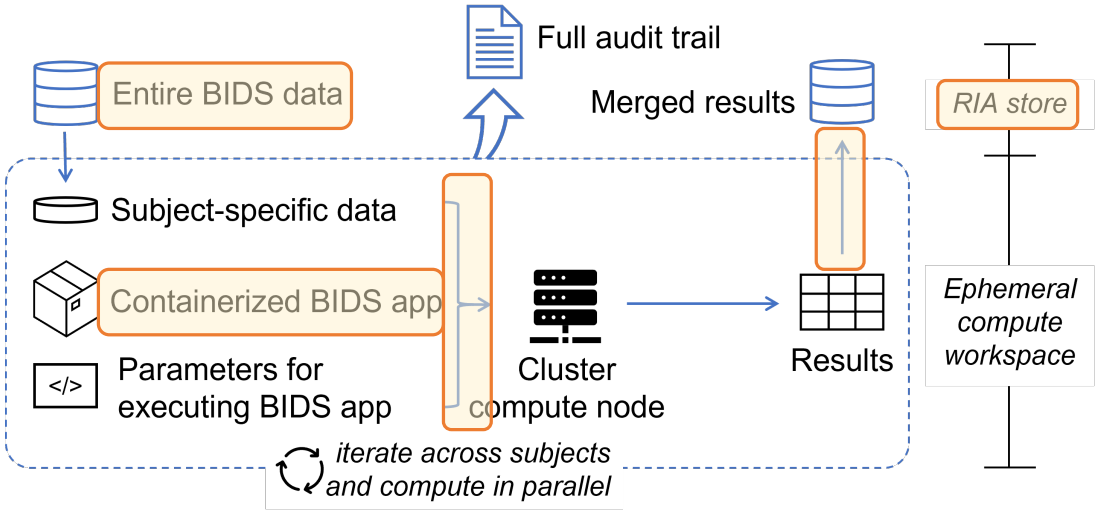
Aim 1 – Workflow of BABS



- ✓ All provenance tracked
- ✓ Fully reproducible rerun
- ✓ Generalizability: not depending on specific information of BIDS dataset and app

Aim 1 – Functionality of BABS

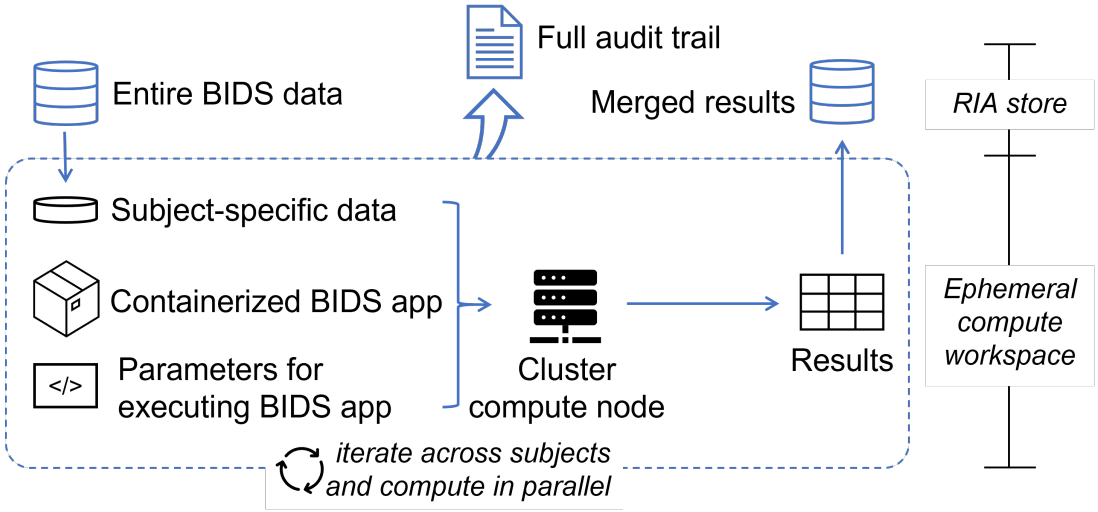
- `babs-init`: initialize necessities for each run
  - Create RIA store
  - Ask DataLad to start to track data provenance
  - Automatically generate the scripts that will later be used internally
- `babs-check-setup`
- `babs-submit`
- `babs-status`
- `babs-merge`
- `babs-unzip`



Proposed BABS workflow

Aim 1 – Functionality of BABS

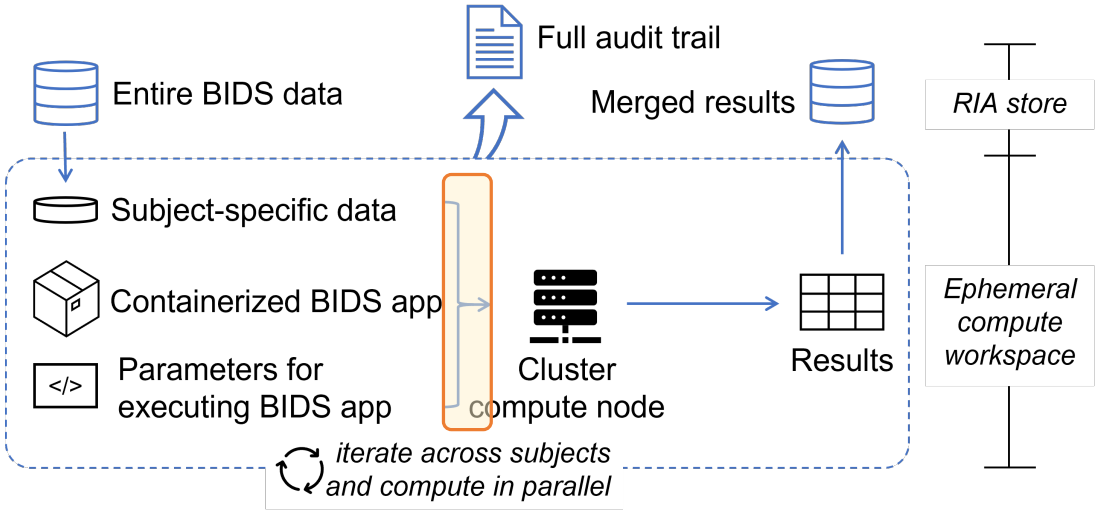
- `babs-init`: initialize necessities for each run
- `babs-check-setup`: check all necessaries are correctly set up by `babs-init`
- `babs-submit`
- `babs-status`
- `babs-merge`
- `babs-unzip`



Proposed BABS workflow

Aim 1 – Functionality of BABS

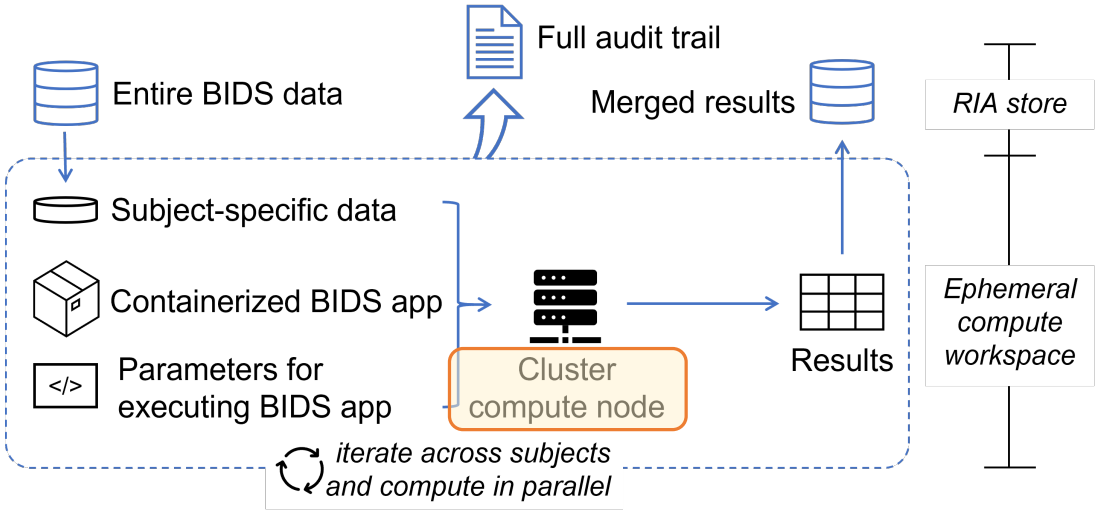
- `babs-init`: initialize necessities for each run
- `babs-check-setup`: verify that data and container configuration is appropriate
- `babs-submit`: submit the jobs to be executed in parallel on cluster compute nodes. Options include:
  - submitting only one subject for testing
  - submitting all subjects but excluding the subjects with successful outputs
- `babs-status`
- `babs-merge`
- `babs-unzip`



Proposed BABS workflow

Aim 1 – Functionality of BABS

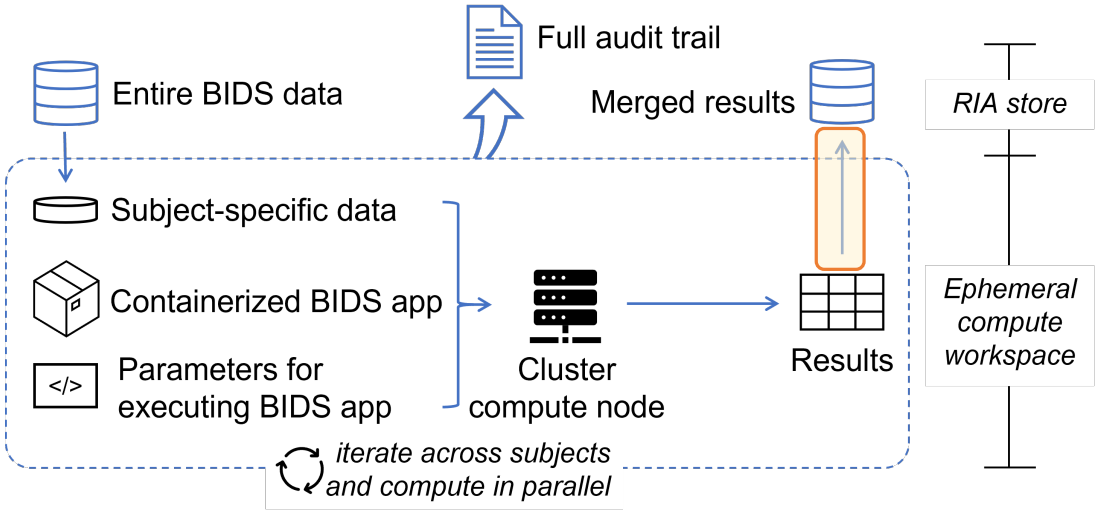
- `babs-init`: initialize necessities for each run
- `babs-check-setup`: verify that data and container configuration is appropriate
- `babs-submit`: submit the jobs to be executed in parallel on cluster compute nodes
- `babs-status`: easily check all jobs' status, and it allows users to automatically resubmit the jobs if failed or stalled
- `babs-merge`
- `babs-unzip`



Proposed BABS workflow

Aim 1 – Functionality of BABS

- `babs-init`: initialize necessities for each run
- `babs-check-setup`: verify that data and container configuration is appropriate
- `babs-submit`: submit the jobs to be executed in parallel on cluster compute nodes
- `babs-status`: easily check all jobs' status and automatically resubmit if needed
- `babs-merge`: merge results and provenance from all jobs. Results are now compressed at the RIA store.
- `babs-unzip`

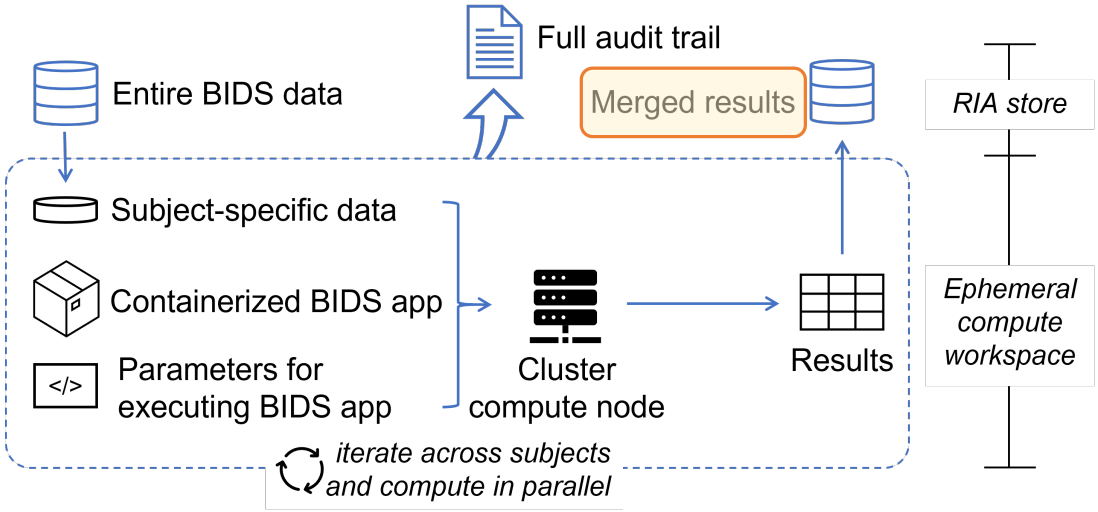


Proposed BABS workflow

Aim 1 – Functionality of BABS

- `babs-init`: initialize necessities for each run
- `babs-check-setup`: verify that data and container configuration is appropriate
- `babs-submit`: submit the jobs to be executed in parallel on cluster compute nodes
- `babs-status`: easily check all jobs' status and automatically resubmit if needed
- `babs-merge`: merge results and provenance from all jobs
- `babs-unzip`: allow users to decompress the zipped data and only get the requested files

- ✓ Parsimonious set of commands
- ✓ Generalizable functionality of BABS
  - any BIDS dataset
  - any BIDS-app
- ✓ Fully reproducible way with complete provenance tracking



Proposed BABS workflow

Aim 1 – Compatibility with computing cluster management systems

- There are various management systems for clusters, and the commands for job scheduling can vary across systems:


Cluster

User Commands	SGE	Slurm
Job submission	qsub [script_file]	sbatch [script_file]
Job deletion	qdel [job_id]	scancel [job_id]
Job status by job	qstat -u * [-j job_id]	squeue [job_id]
…	…	…

- Design of BABS: will be compatible with Slurm and SGE
- Test of BABS: on Slurm and SGE systems, using
  - Slurm cluster through collaborations with University of Minnesota
  - SGE-based CUBIC system at Penn Medicine

Table adapted from: https://srcc.stanford.edu/sge-slurm-conversion

Aim 1 – Evaluation data and details
BIDS data and app

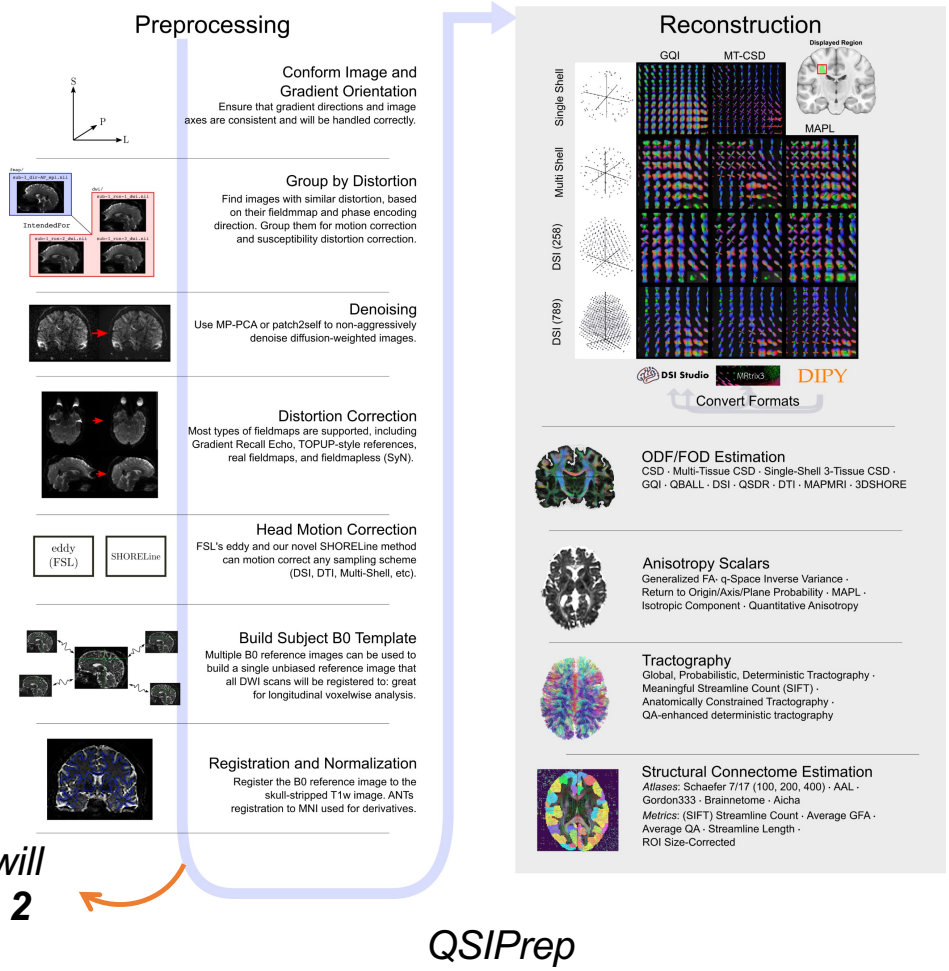
- Input BIDS data: large-scale diffusion MRI (dMRI) from Reproducible Brain Charts (RBC) effort
  - Four datasets with different dMRI acquisition protocols
  - In total n=6,372 dMRI

Datasets	dMRI			
	Sample size (n)	Resolution	b-values (s/mm^2)	Number of directions
Philadelphia Neurodevelopmental Cohort (PNC)[1]	1409	1.875x1.875x2 mm	1000	64
Healthy Brain Network (HBN)[2]	2151	1.8mm isotropic	1000, 2000	64
NKI Rockland Sample (NKI-RS)[3]	2169	2mm isotropic	1500	137
Human Connectome Project - Development (HCP-D)[4]	643	1.5mm isotropic	1500, 3000	185
Total	**6372**			

[1]Satterthwaite et al., NeuroImage 2014;
[2]Alexander et al., Sci Data 2017;
[3]Nooner et al., Front Neurosci 2012;
[4]Somerville et al., NeuroImage 2018

Aim 1 – Evaluation data and details
BIDS data and app

- Containerized BIDS app: QSIPrep
  - For dMRI preprocessing + reconstruction
  - Can process the dMRI with nearly all sampling schemes



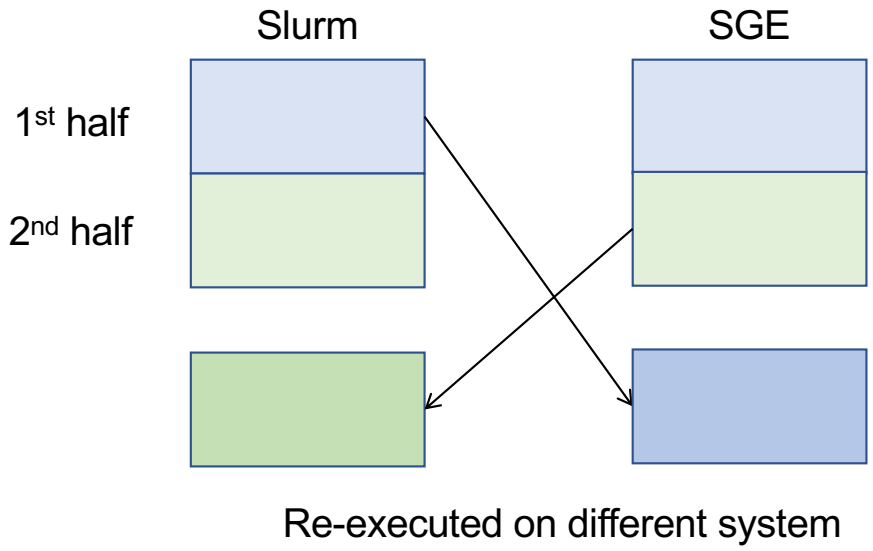
*Preprocessed data will also be used in **Aim 2***

QSIPrep

Figure adapted from: Cieslak et al., Nat Methods 2021

Aim 1 – Evaluation data and details
Evaluation details

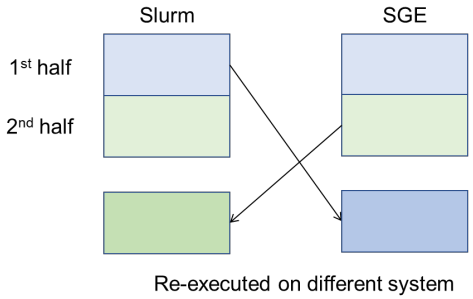
1. **Compatibility** with both cluster systems: BABS will be applied to all datasets on both Slurm and SGE systems
2. **Reproducibility and capability** of capturing full process provenance even for dataset at scale:
   - To perform a split-half cross-validation:

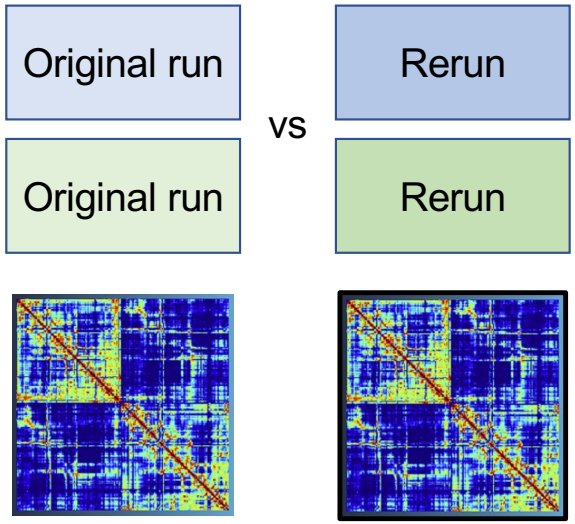


Aim 1 – Evaluation data and details

Evaluation details: split-half cross-validation

2. To demonstrate the **reproducibility and capability** of capturing full process provenance even for dataset at scale: To perform a split-half cross-validation:
   - To compare the differences between the output connectivity matrices from original run and rerun
   - Expect that there is minimal difference





Connectivity matrix

Figure adapted from:
Baum et al., NeuroImage 2018

Outline

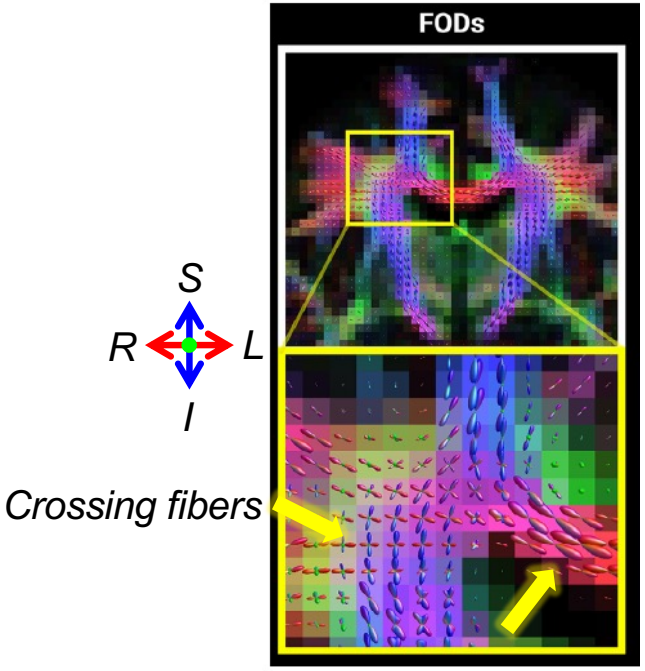
- Aim 1: Software for reproducible image processing at scale
- Aim 2: Memory-efficient, generalizable software for statistical analysis
- Software development

Outline: Approach for Aim 2

To develop and release a generalizable, memory-efficient, open-source R package for mass-univariate hypothesis testing of large neuroimaging datasets

- Approach for **Aim 2:**
  - Background
  - Method for efficient memory
  - Workflow and functionality of proposed software
  - Evaluation data and details
  - Application of GAMs to developmental datasets
  - Generalizability and extensibility

Aim 2 – Background: What is fixel data?



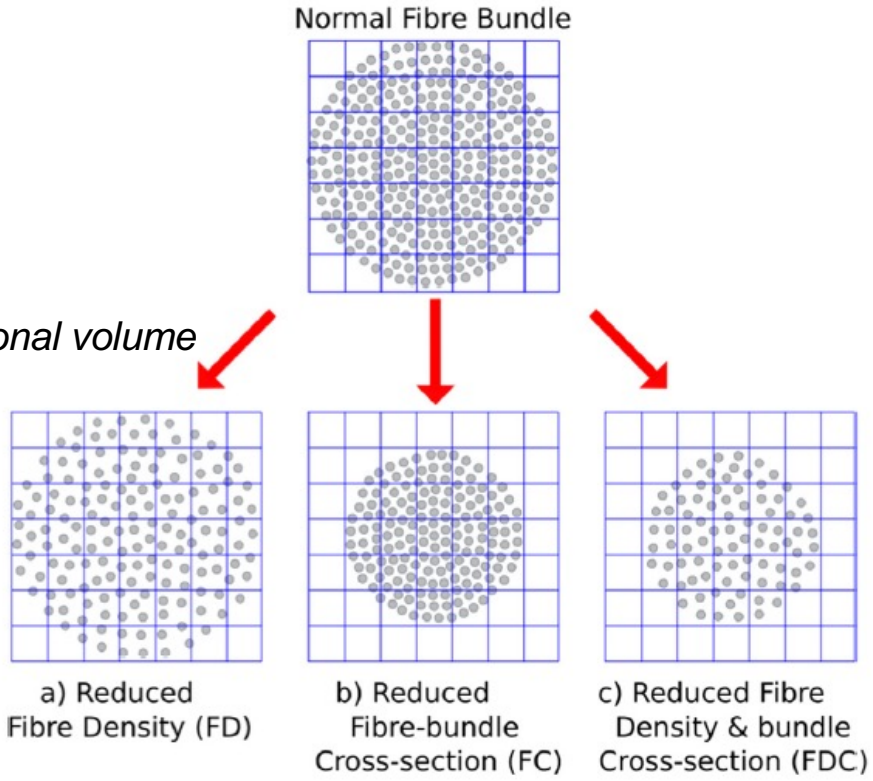
FOD, fiber orientation distribution
FD, fiber density

- "Fixel": an individual **fi**ber population within a vo**xel**

Dhollander et al., NeuroImage 2021
Raffelt et al., NeuroImage 2015

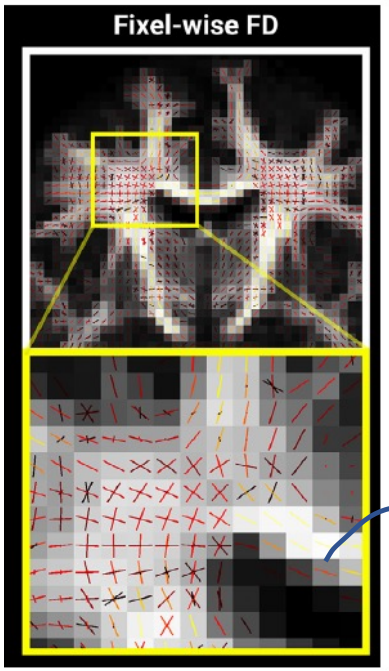
Aim 2 – Background: What is fixel data?
Fixel-wise metrics

Change to intra-axonal volume



Raffelt et al., NeuroImage 2017

Aim 2 – Background: Mass-univariate hypothesis testing



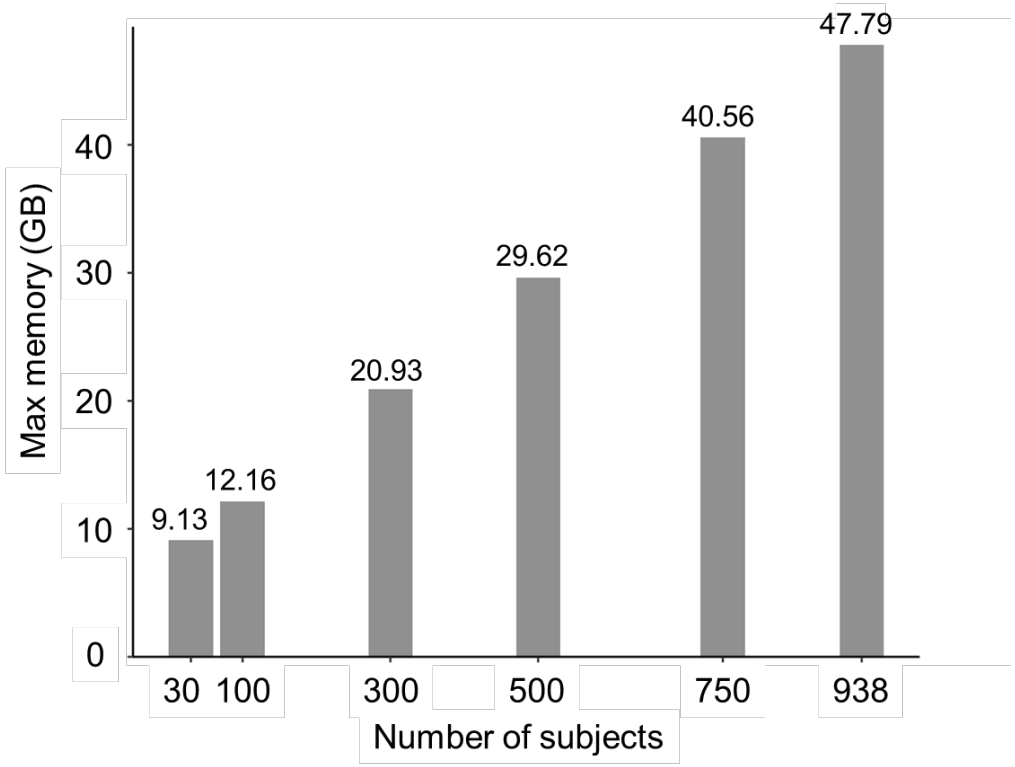
Does fixel-wise FD change over development?

→ To perform a mass-univariate hypothesis testing:
@ each fixel, test FD ~ age

Figure adapted from: Dhollander et al., NeuroImage 2021

Aim 2 – Background

Challenge #1: Intensive memory requirement for large-scale dataset



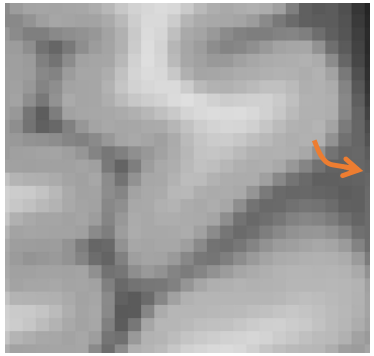
Software MRtrix, for fixel-wise linear model: FDC ~ age

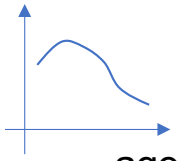
Using HCP cluster?

Aim 2 – Background

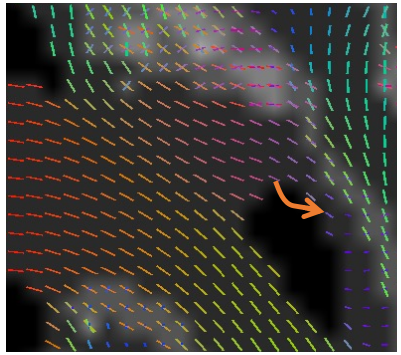
Challenge #2: No consistent statistical analysis tool generalized across image types

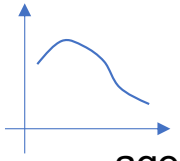
Voxel data



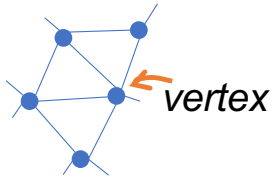


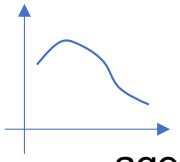
Fixel data





Surface vertex data





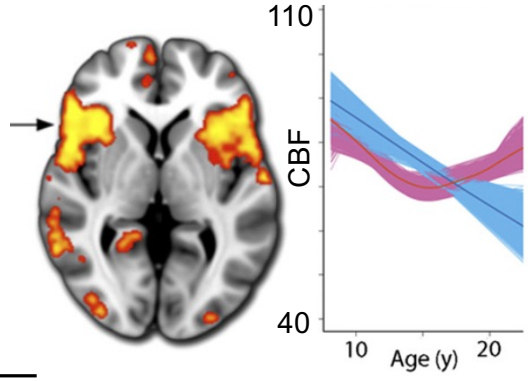
Consistent statistical analysis tool?

Sydnor et al., Neuron 2021

Aim 2 – Background

Challenge #3: Users cannot easily model nonlinear age effects with current tools

- Lifespan age effects are often nonlinear
- Current tools often only support generalized linear model (GLM):
  - Limited flexibility: not supporting GAM with penalized splines
  - Generating design matrix and contrast matrix for GLM is not as easy as writing a formula in R



Using Generalized Additive Model (GAM)

Satterthwaite et al., PNAS 2014

	GLM		Formula in R
	Design matrix	*Contrast matrix*	
	intercept sex age		
Linear regression	1 1 21.75 1 1 20.50 1 0 20.50 1 1 20.92 1 0 20.92 1 0 21.50 …	0 0 1	lm(): FD ~ age + sex
GAM	???		gam(): FD ~ s(age) + sex

Aim 2 – Proposed software: ModelArray

Current challenges in mass-univariate hypothesis testing for large-scale datasets:

1. Scalability: Massive memory resources needed by large datasets
2. Generalizability: No consistent statistical analysis tool generalized across data types
3. User-friendly: Users cannot easily model nonlinear age effects with current tools

To address these challenges, I propose to develop **ModelArray**, a generalizable, memory-efficient, open-source R package for mass-univariate hypothesis testing of large neuroimaging datasets.

Aim 2 – Method for efficient memory

- HDF5 file format:
  - An HDF5 file (.h5) stores large dataset hierarchically and provides fast data access.
- R package DelayedArray:
  - Accessing and analyzing data ***without*** loading the entire data into memory

File size on the disk:

ltn_FDC_n938.h5 2.5 GB

```
# in-memory size in R:
> object.size(fixelarray)
> 15584 bytes
```

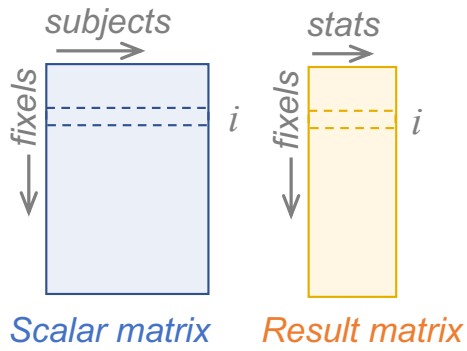
An example .h5 file for fixel data:

Information on fixel data

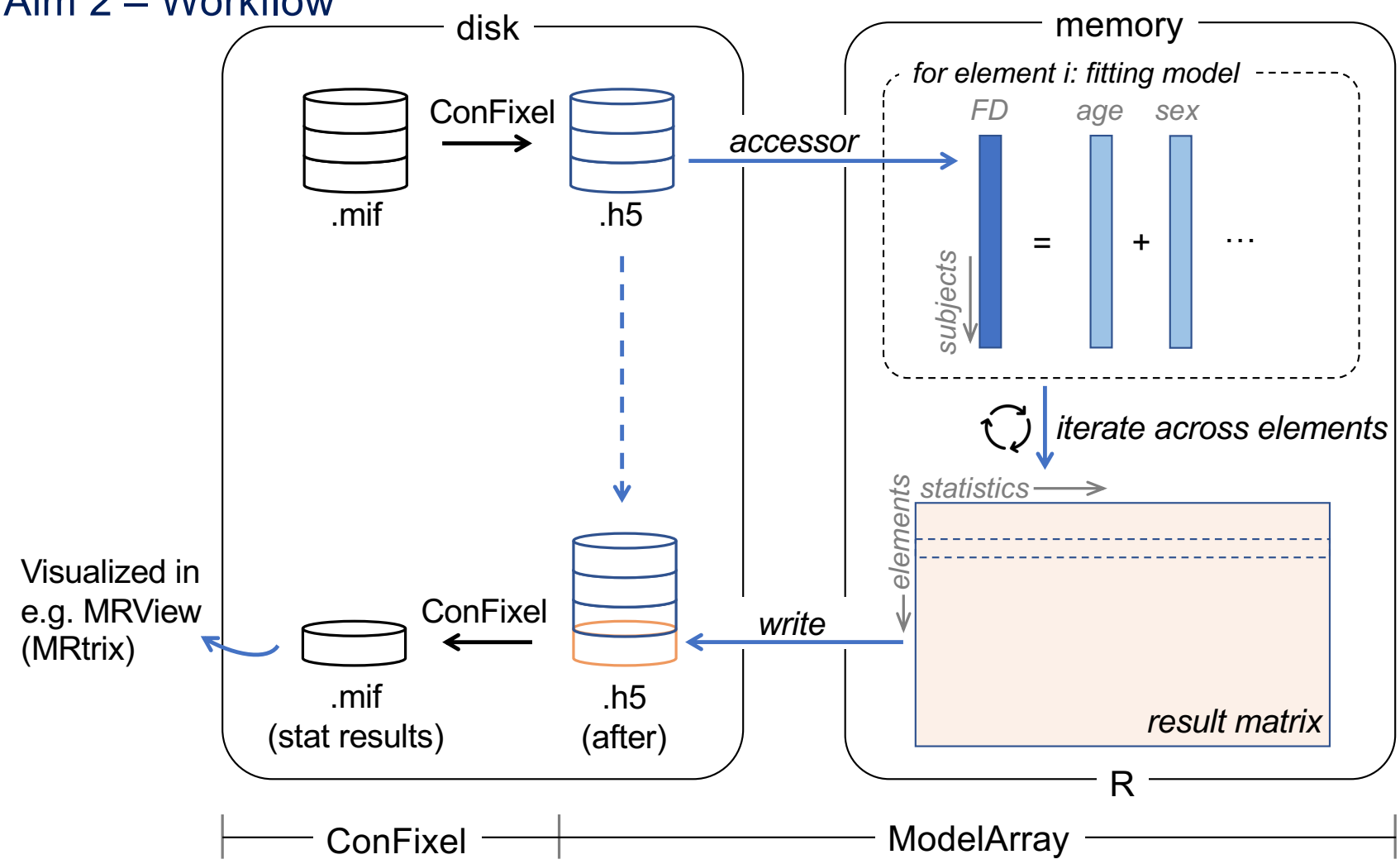
Scalar matrix

Result matrix

```
/fixels                    602229 x 5
/voxels                    433909 x 4
/scalars
  /FDC
     /ids                  1 x 938
     /values               602229 x 938
/results
  /lm
    /results_matrix        602229 x 13
```



Aim 2 – Workflow

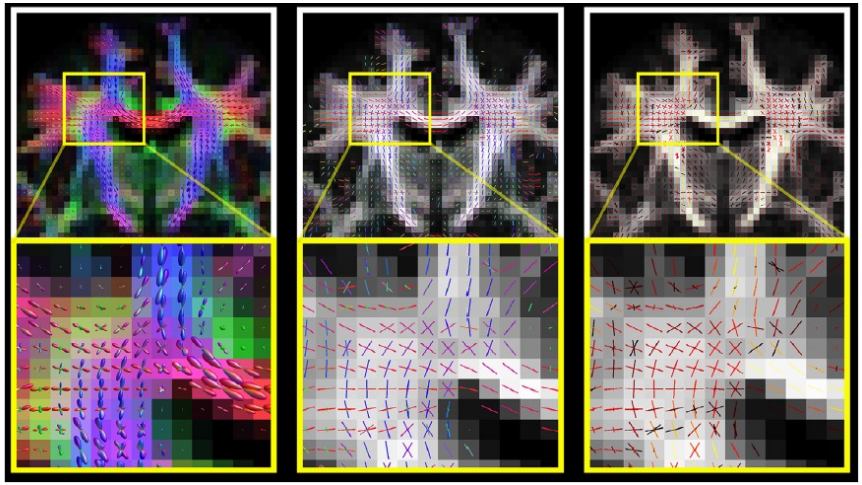


Aim 2 – Functionality of ModelArray

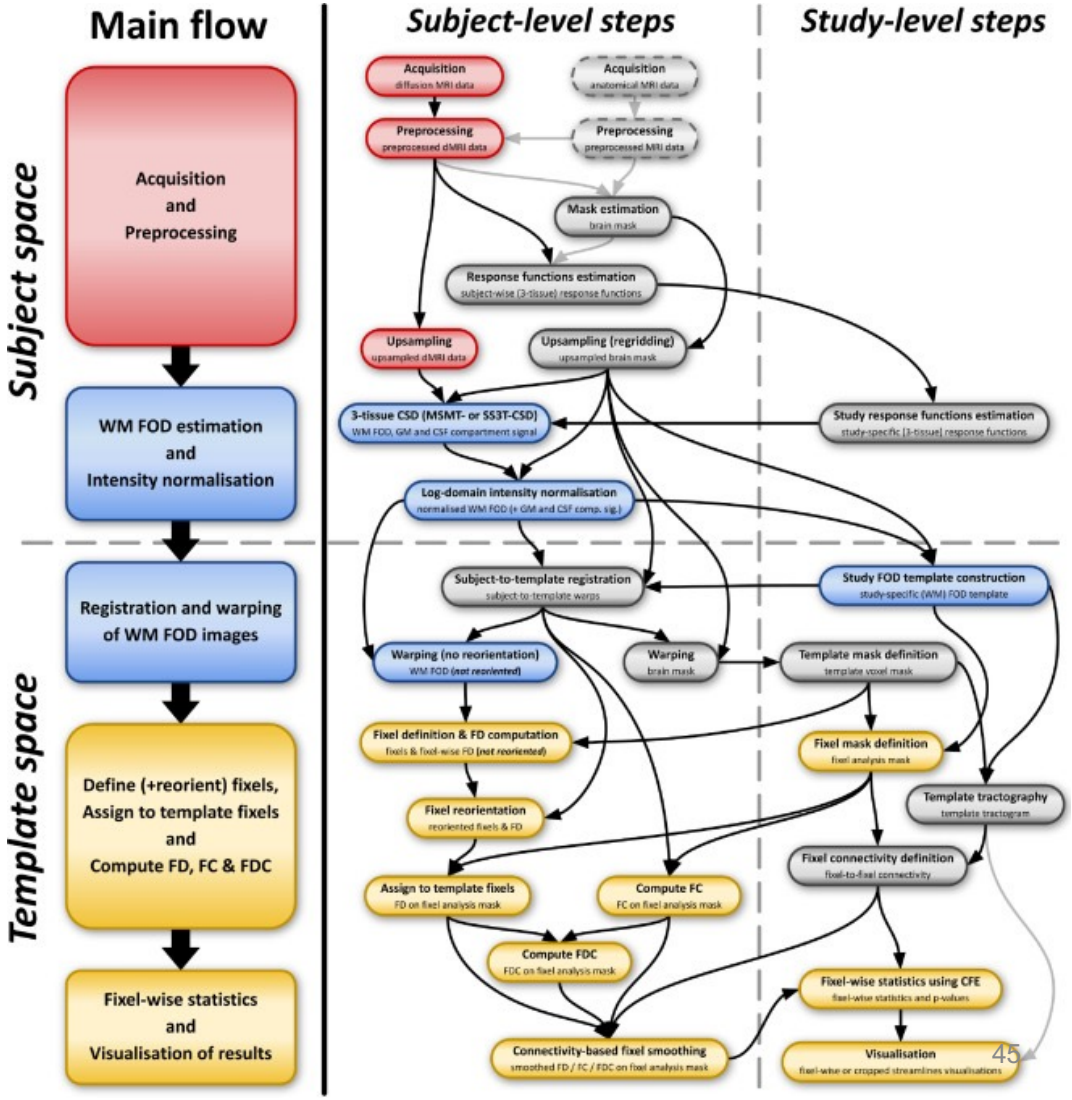
Example script of the whole procedure of running ModelArray in R

```
> library(ModelArray)

> filename <- "example.h5"
> modelarray <- ModelArray(filename, scalar_types = "FDC")
> phenotypes <- read.csv("example.csv")

# Linear model:
> lm.outputs <- ModelArray.lm(FDC~age, modelarray, phenotypes, "FDC")
# Generalized additive model (GAM):
> gam.outputs <- ModelArray.gam(FDC~s(age)+sex+motion, modelarray, phenotypes, "FDC")

> writeResults(filename, df.output=lm.outputs, analysis_name="lm")
> writeResults(filename, df.output=gam.outputs, analysis_name="gam")
```

Aim 2 – Evaluation of ModelArray
Data: dMRI fixel data from RBC



Fixel-wise metric: FDC
(combination of FD and FC, more sensitive)



Figures adapted from Dhollander et al., NeuroImage 2021

Aim 2 – Evaluation of ModelArray

- Evaluations:
  - Memory profiling
  - Runtime profiling
- Evaluation details:
  - Will compare to primary existing tool for fixel-wise statistical analysis: the function "fixelcfestats" in MRtrix[1].
  - Will evaluate on a standalone computer with Linux system to avoid interference from other users.
  - Will use simple linear model: FDC ~ age

[1]*Raffelt et al., NeuroImage 2015*

Aim 2 – Evaluation of ModelArray

Memory profiling

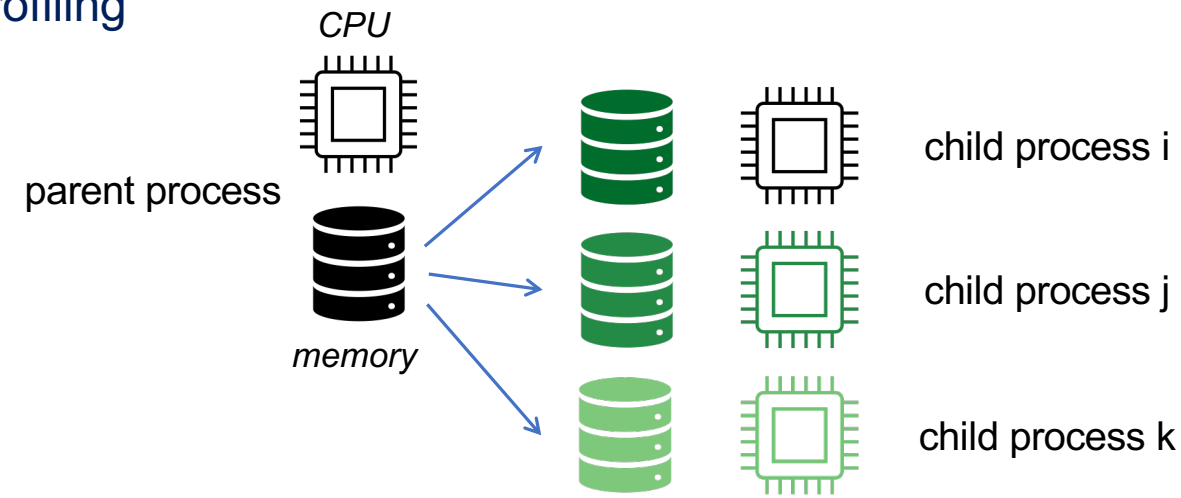


Figure adapted from : https://www.blasbenito.com/post/02_parallelizing_loops_with_r/

```
$ htop
```

child processes

parent process →

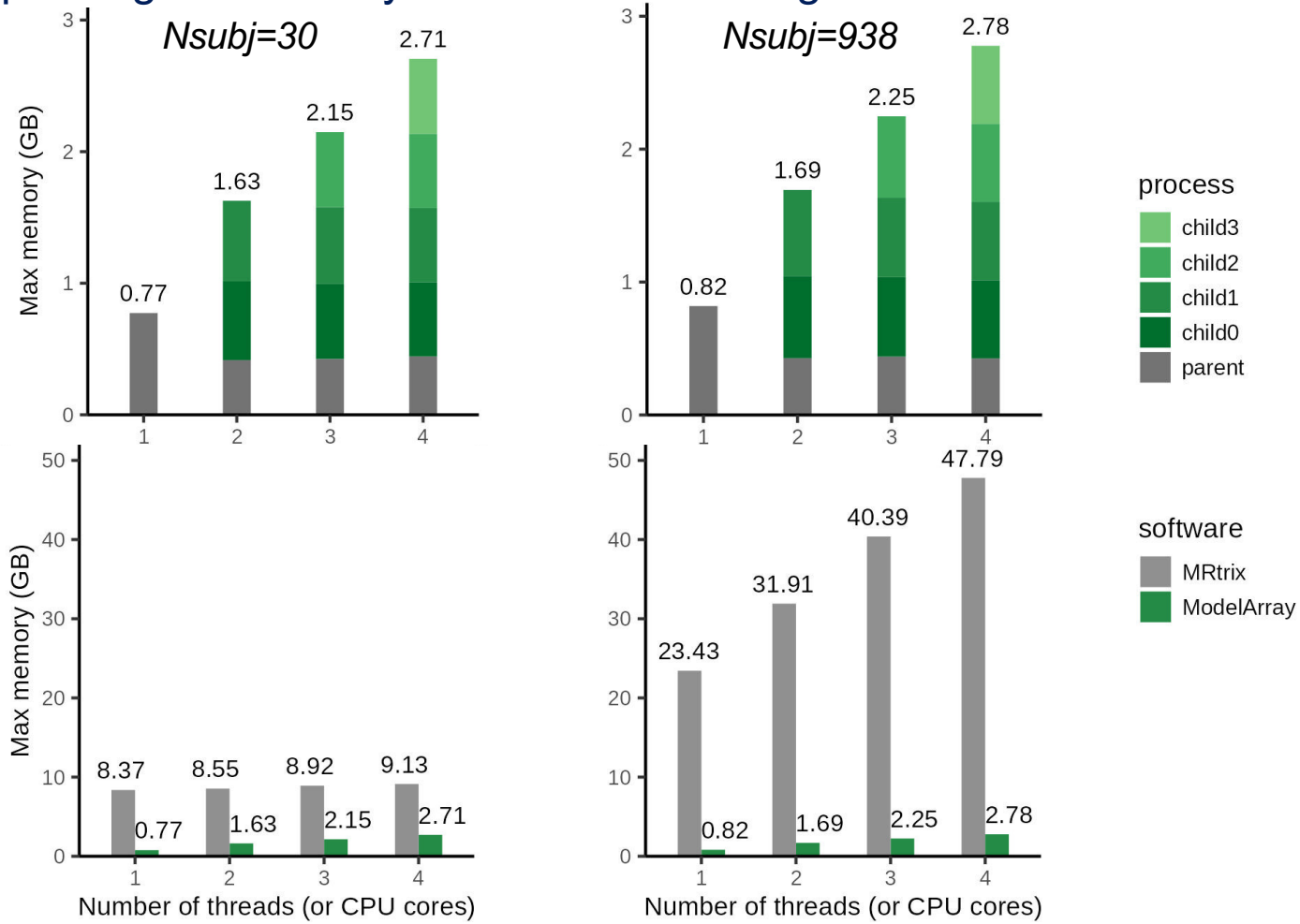
PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
10652	chenying	20	0	1135M	471M	9568	R	99.8	0.9	1:25.39	R --no-echo --no-r
10658	chenying	20	0	1135M	471M	9560	R	99.8	0.9	1:25.40	R --no-echo --no-r
10656	chenying	20	0	1135M	471M	9560	R	99.2	0.9	1:25.43	R --no-echo --no-r
3566	chenying	20	0	1081M	439M	82148	S	0.0	0.8	0:09.83	R --no-echo --no-r
2445	chenying	20	0	4148M	300M	82168	S	0.0	0.5	0:14.12	gnome-shell

Memory usage will be captured by Working Set Size (WSS) Tools for Linux

https://www.brendangregg.com/wss.html

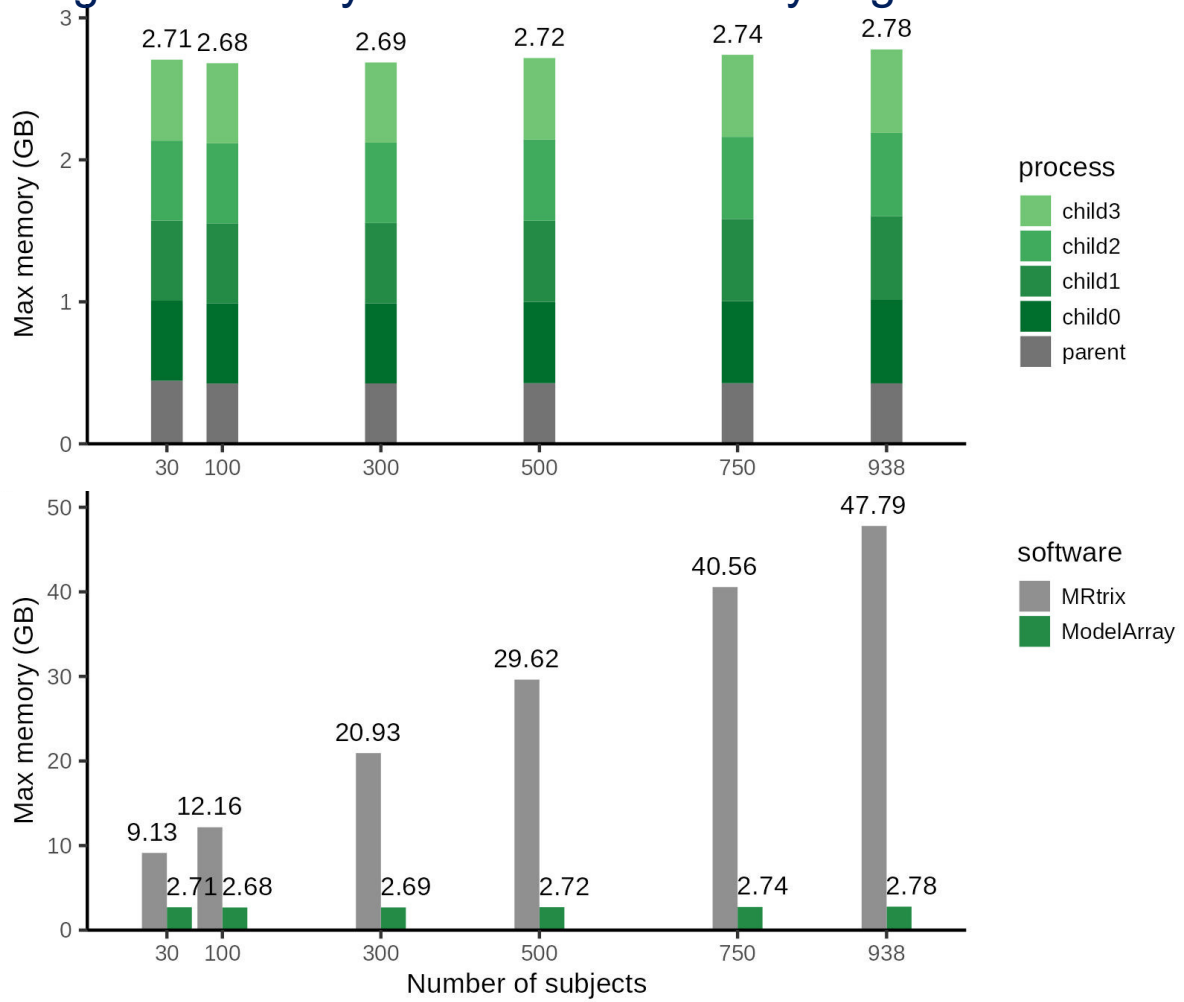
Aim 2 – Evaluation of ModelArray

Memory profiling: Preliminary results – when using different number of threads (or CPU cores)



Aim 2 – Evaluation of ModelArray

Memory profiling: Preliminary results – when analyzing different number of subjects



parallelization factor = 4

- *4 CPU cores requested for ModelArray*
- *4 threads requested for MRtrix*

Aim 2 – Application of GAMs to developmental datasets
Methods

- I will also demonstrate the use of GAMs in ModelArray for modelling nonlinear developmental effects with penalized splines:

$$FDC \sim s(Age) + sex + motion$$

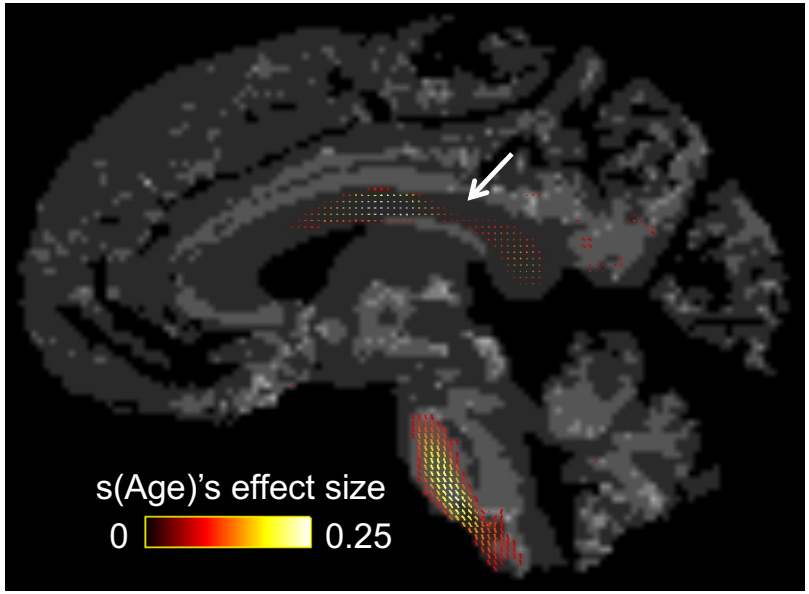
- $motion$: mean relative volume-to-volume displacement of 7 b=0 images
- $Effect\ size\ = R^2_{full}\ - R^2_{reduced}$

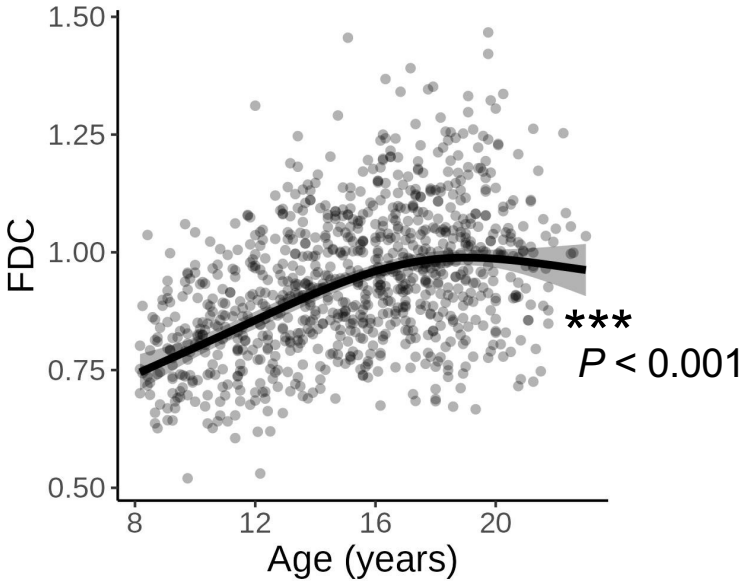
$$FDC \sim sex + motion$$

Roalf et al., NeuroImage 2016

Aim 2 – Application of GAMs to developmental datasets

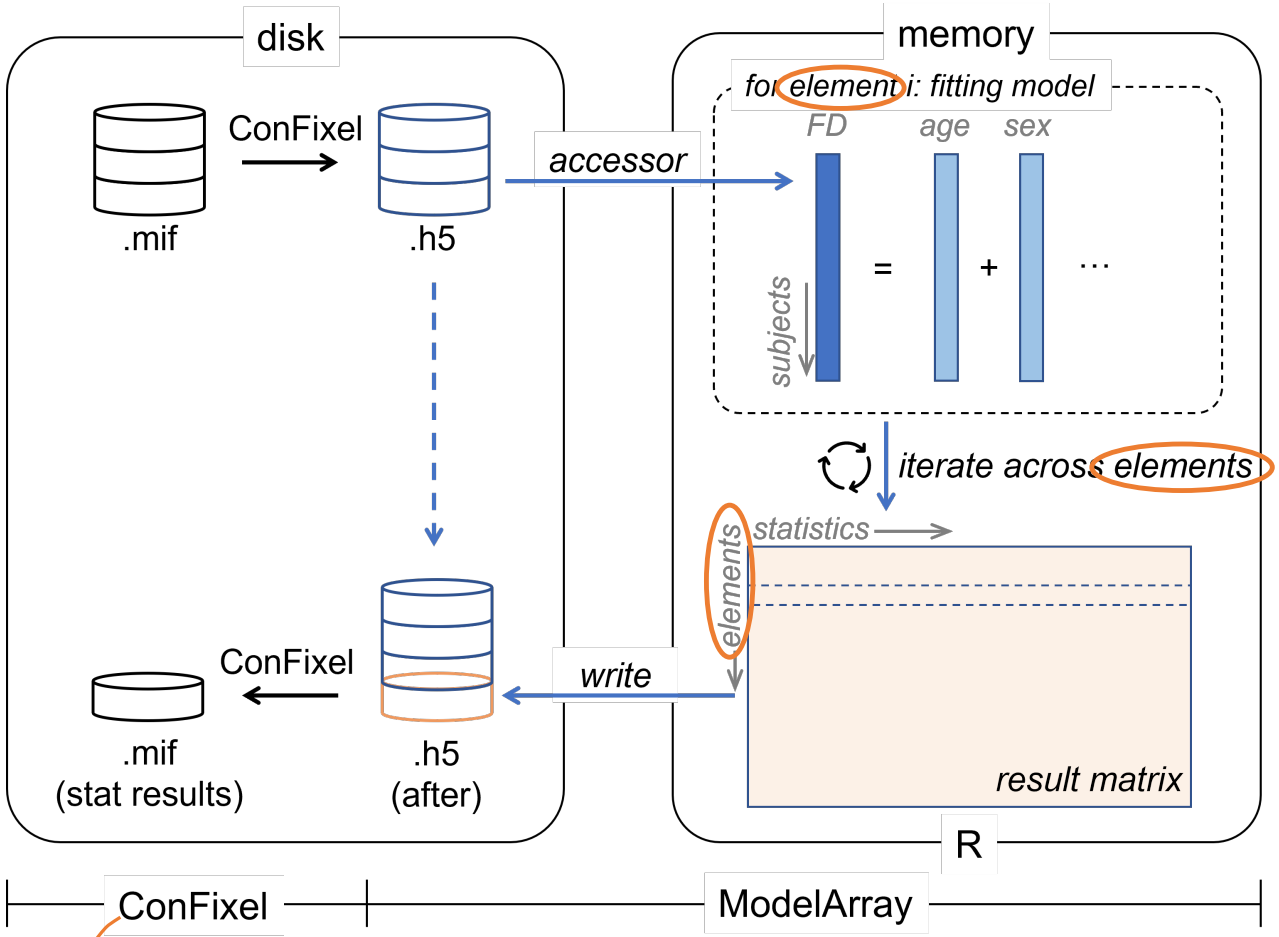
Preliminary data: nonlinear age effect in adolescent development





- Data: PNC dataset (n=938), aged 8-22 years
- Threshold: *P* value of s(Age) < $1x10^{-15}$
- GAM Formula: $FDC \sim s(Age) + sex + motion$

Aim 2 – Generalizability and extensibility of ModelArray



ConVoxel for voxel data (NIfTI file format)
ConCifti for surface data (CIFTI grayordinate file format)

Extensibility:

Model fitting functions in ModelArray:

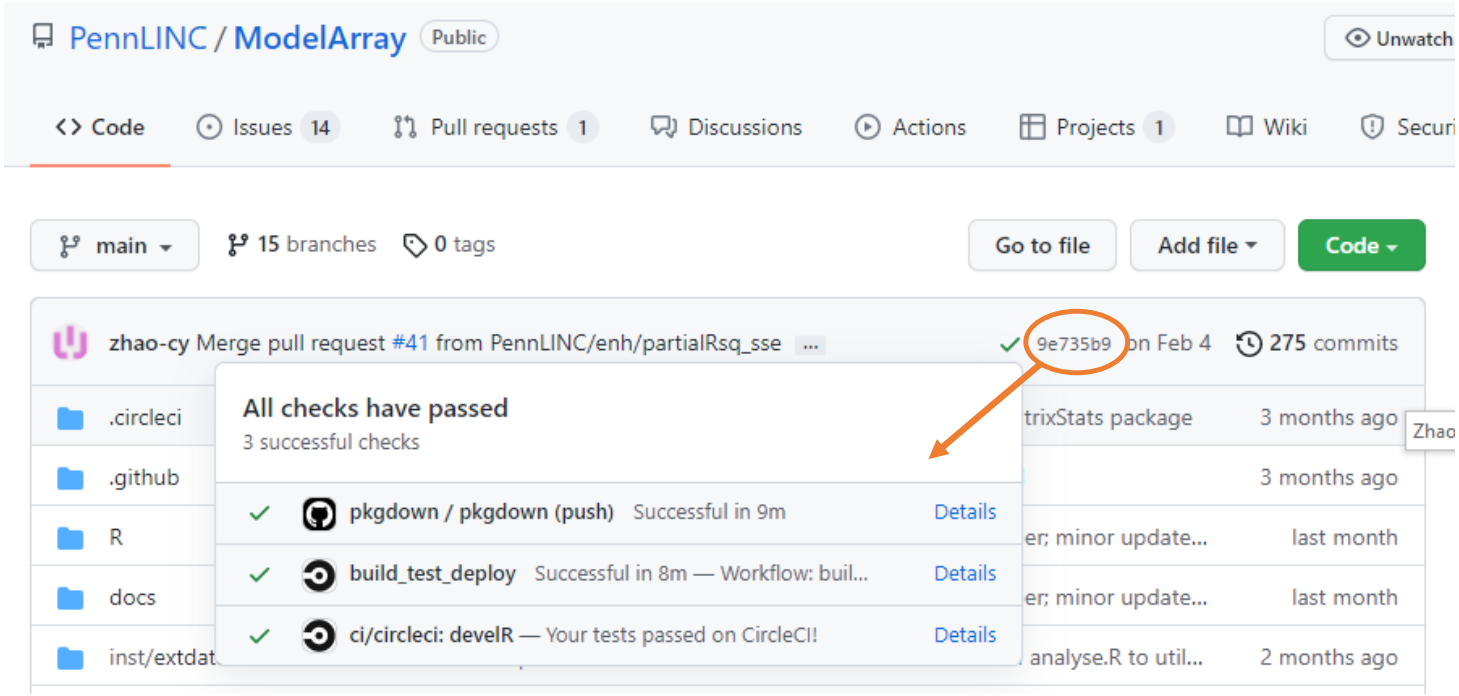
- ModelArray.lm() for linear model
- ModelArray.gam() for GAM
- ModelArray.new() for ?

Generalizability
- I/O interface
- workflow

Outline

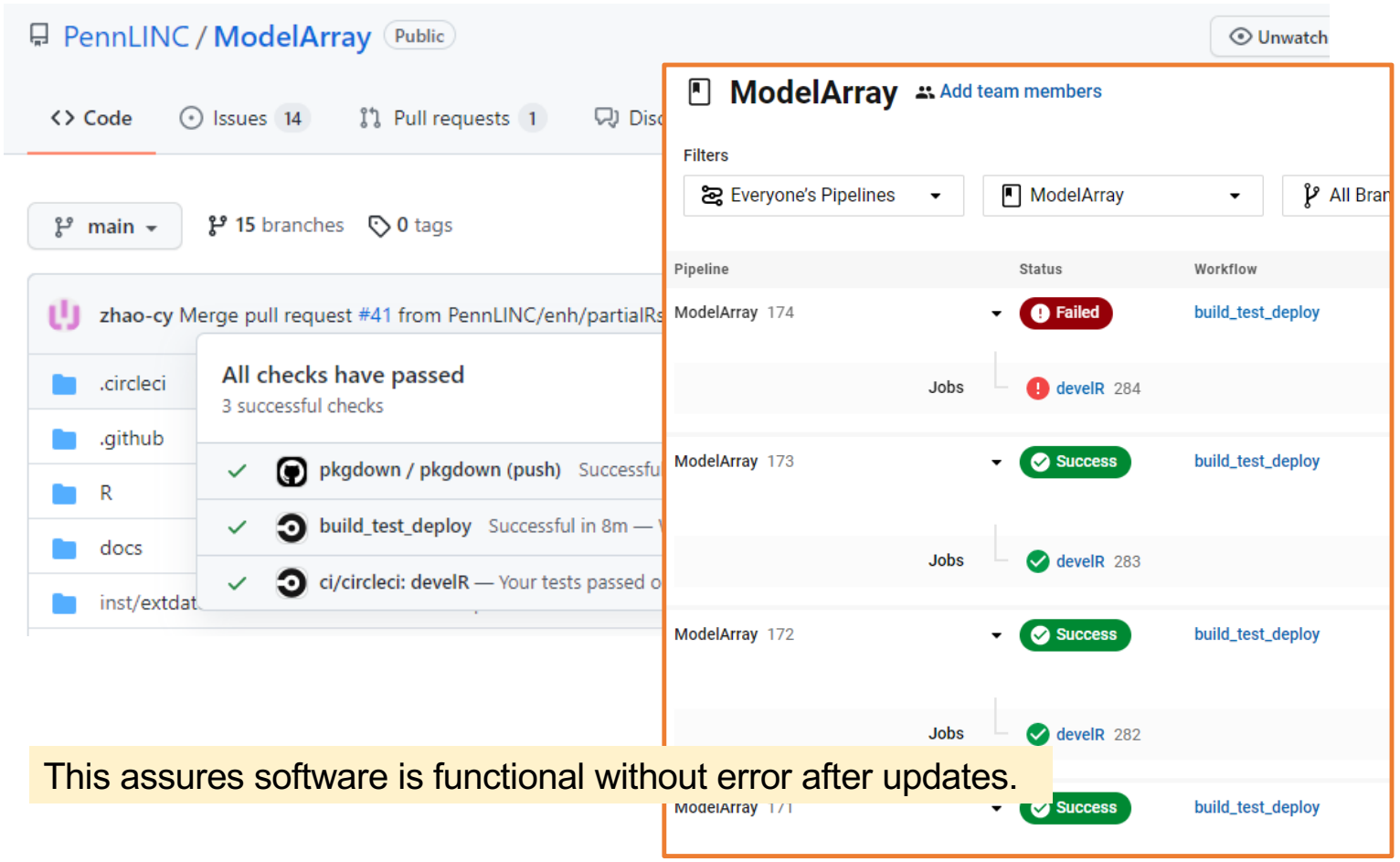
- Aim 1: Software for reproducible image processing at scale
- Aim 2: Memory-efficient, generalizable software for statistical analysis
- Software development

Software development
Version control and CI testing



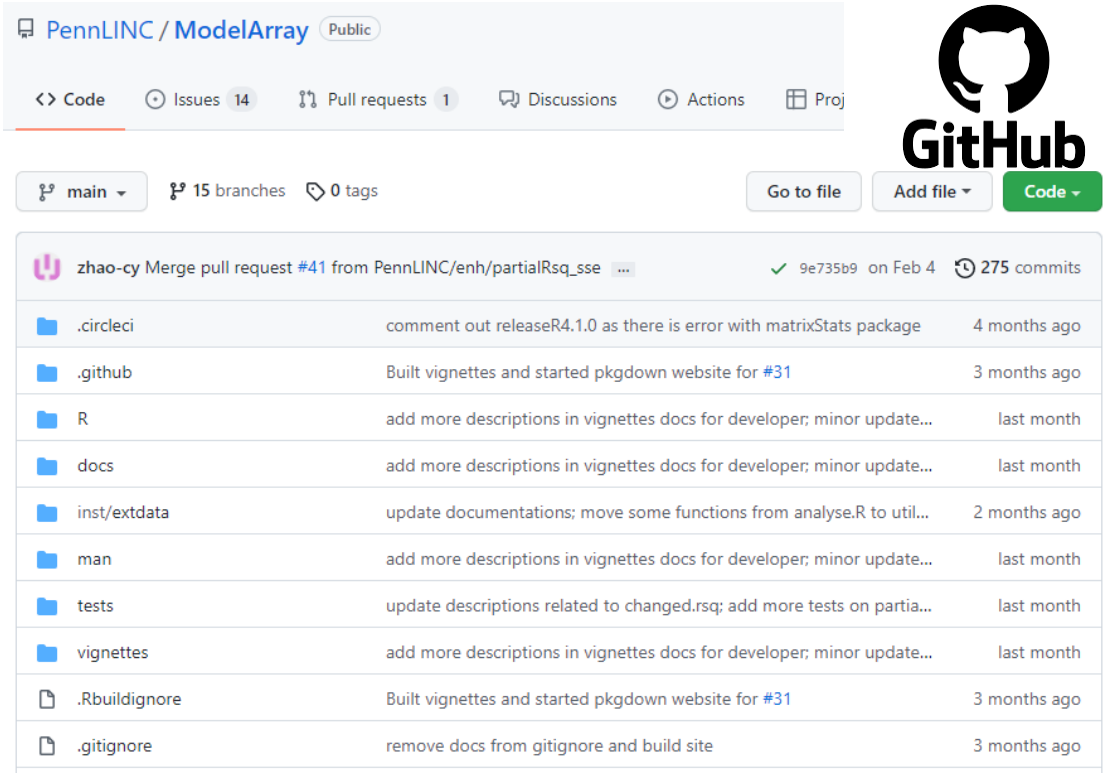
CI = Continuous Integration

Software development
Version control and CI testing



This assures software is functional without error after updates.

Software development
Containerization and release

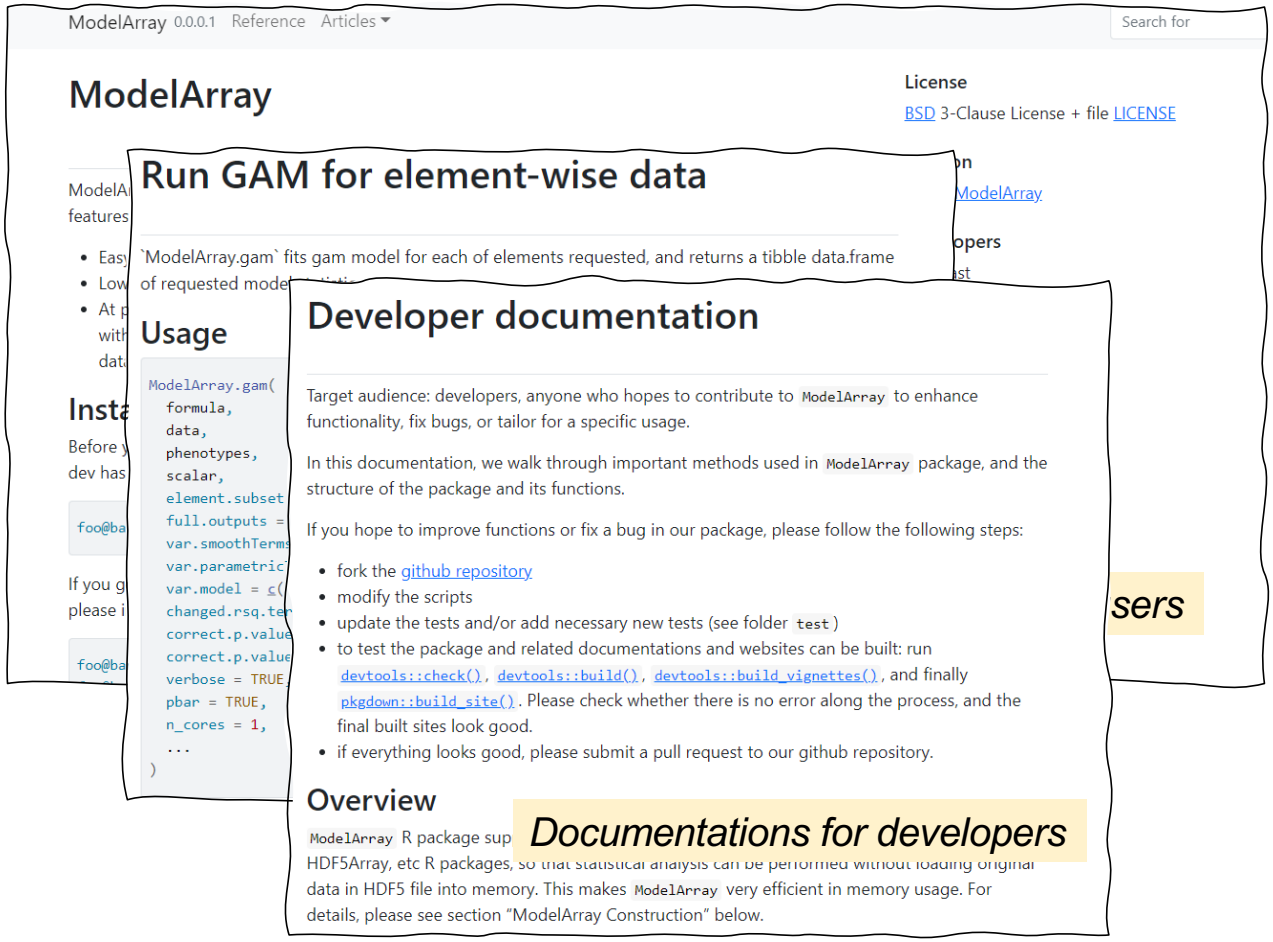


Source code openly available on GitHub

- Will build a container to enhance portability
- Will release on DockerHub

This workflow ensures both version control and frictionless portability.

Software development
Comprehensive documentations



Thank you !!!