Functional Magnetic Resonance Imaging

GitHub https://github.com/dcdace/fMRI_training

Dace [datza] Apšvalka

February 2026

Outline

- Introduction
- Experimental design
- **Data management**
- Pre-processing
- Statistical analysis
- Practical demo

Example Dataset

https://openneuro.org/datasets/ds000117/

www.nature.com/scientificdata

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES

» Electroencephalography -EEG

» Brain imaging

» Functional magnetic resonance imaging

» Cognitive neuroscience

A multi-subject, multi-modal human neuroimaging dataset

Daniel G. Wakeman[1,2] & Richard N. Henson[2]

We describe data acquired with multiple functional and structural neuroimaging modalities on the same nineteen healthy volunteers. The functional data include Electroencephalography (EEG), Magnetoencephalography (MEG) and functional Magnetic Resonance Imaging (fMRI) data, recorded while the volunteers performed multiple runs of hundreds of trials of a simple perceptual task on pictures of familiar, unfamiliar and scrambled faces during two visits to the laboratory. The structural data include T1-weighted MPRAGE, Multi-Echo FLASH and Diffusion-weighted MR sequences. Though only from a small sample of volunteers, these data can be used to develop methods for integrating multiple modalities from multiple runs on multiple participants, with the aim of increasing the spatial and temporal resolution above that of any one modality alone. They can also be used to integrate measures of functional and structural connectivity, and as a benchmark dataset to compare results across the many neuroimaging analysis packages. The data are freely available from https://openfmri.org/.

Received: 07 April 2014

Accepted: 05 January 2015

Published: 20 January 2015

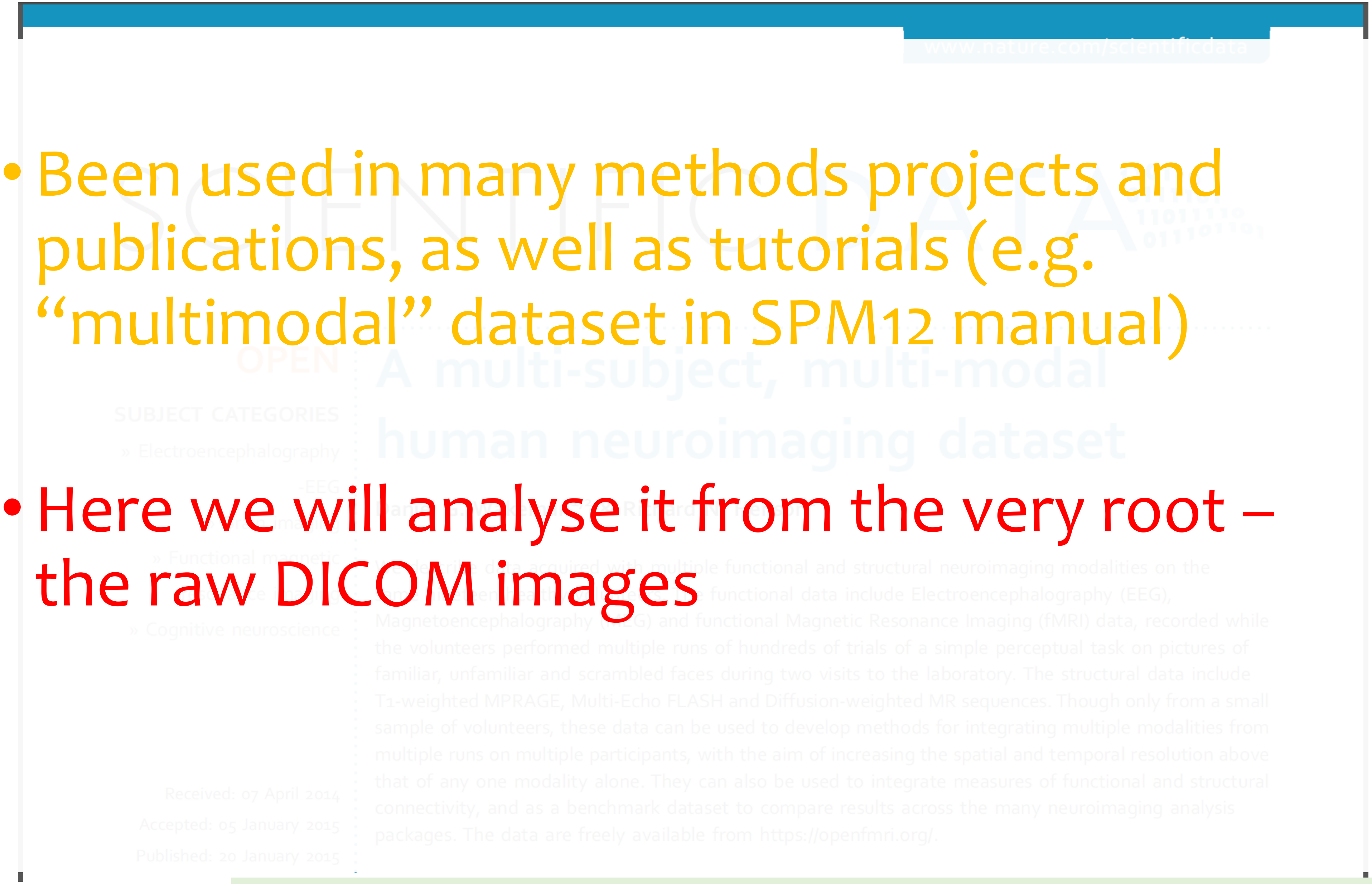
Wakeman & Henson (2015), *Scientific Data*, http://www.nature.com/articles/sdata20151

Example Dataset

https://openneuro.org/datasets/ds000117/

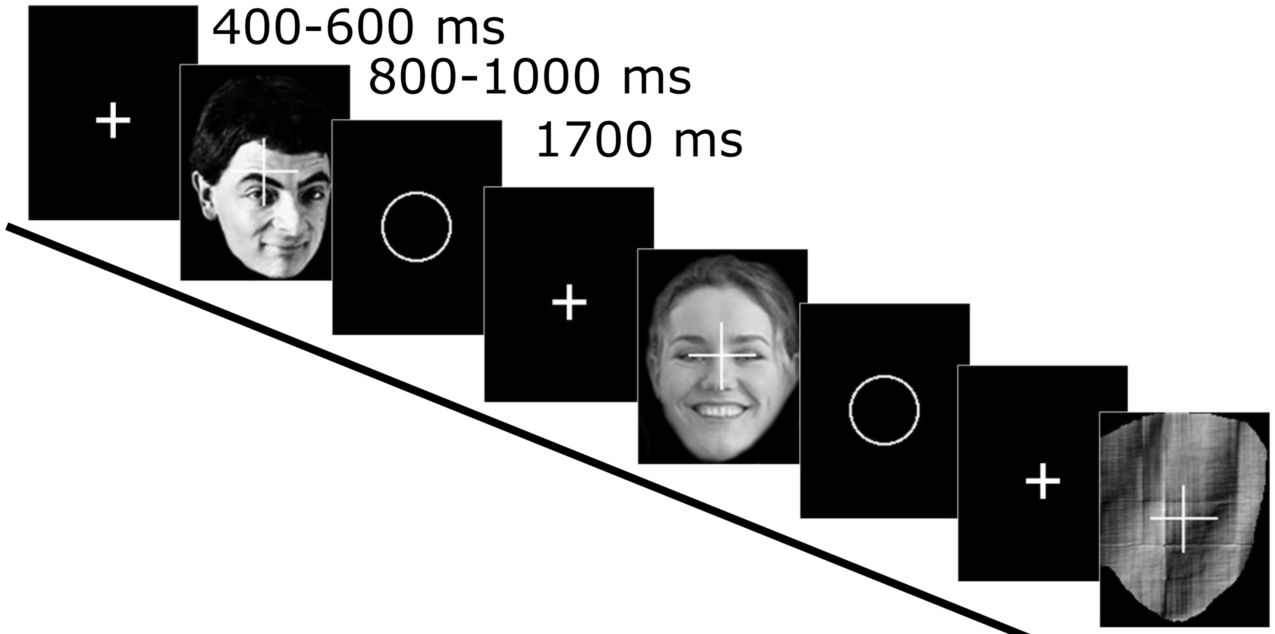
- Been used in many methods projects and publications, as well as tutorials (e.g. “multimodal” dataset in SPM12 manual)

- Here we will analyse it from the very root – the raw DICOM images



Wakeman & Henson (2015), *Scientific Data*, http://www.nature.com/articles/sdata20151

Example Experiment: Face Recognition



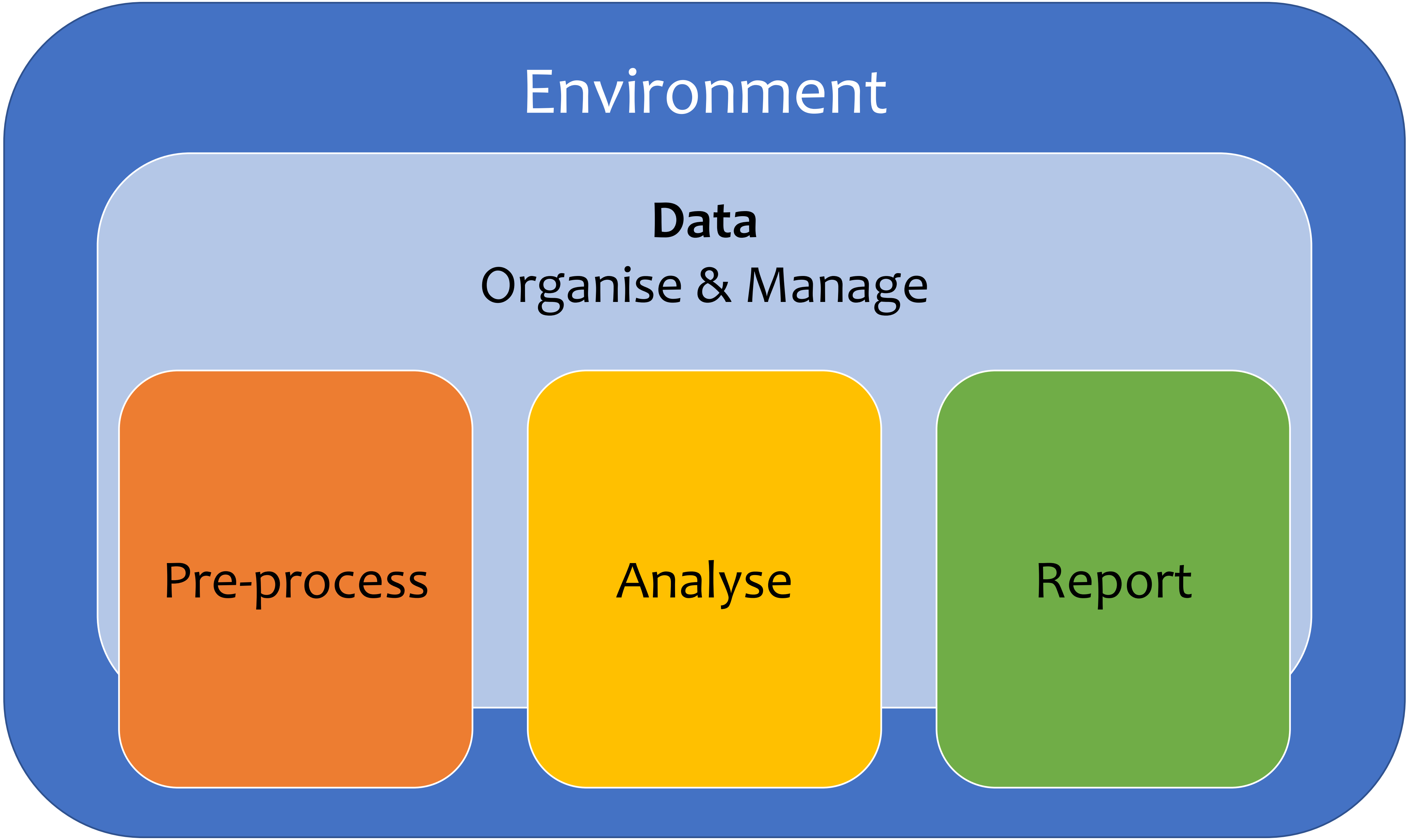
N = 16 subjects

Stimuli: 3 types of greyscale face images:

~150 x Familiar
~150 x Unfamiliar
~150 x Scrambled

Task: Judge face symmetry

Each image was presented twice, with the second presentation occurring either immediately after (**Immediate Repeats**), or after 5–15 intervening stimuli (**Delayed Repeats**), with 50% of each type of repeat.


Environment
Data
Organise & Manage
Pre-process
Analyse
Report

Environment

PROGRAMMING LANGUAGES

BASH & Shell Scripts

A low-level programming language providing a command line user interface for Unix-like operating systems (e.g., Linux, macOS).
Used to automate repetitive tasks and manage system processes and resources.

python

A high-level, general-purpose programming language.
License-free – good for reproducible & open code.

MATLAB

A high-level programming language designed for engineers and scientists.
Requires a license. Provides loads of useful resources for Neuroimaging analysis.

Environment

PROGRAMMING LANGUAGES

BASH & Shell Scripts

A low-level programming language providing a command line user interface for Unix-like operating systems (e.g., Linux, macOS).
Used to automate repetitive tasks and manage system processes and resources.

python™

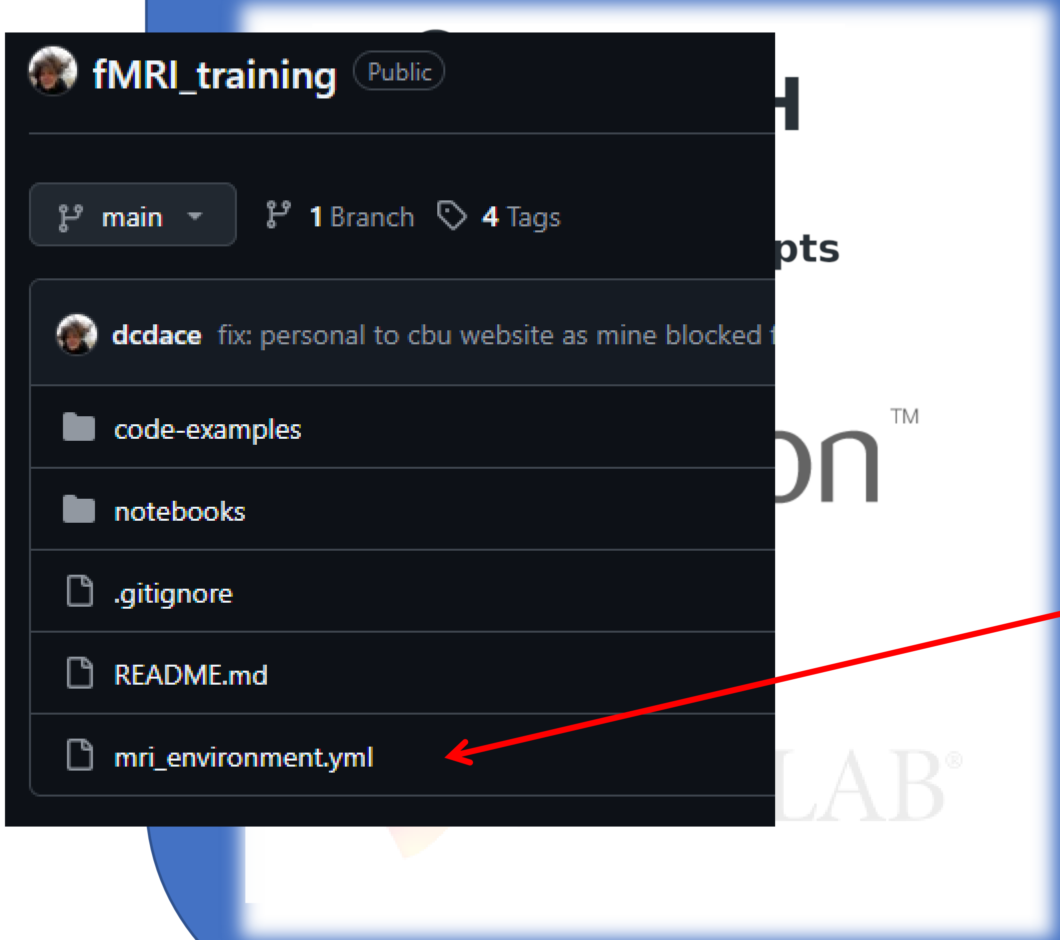
A high-level, general-purpose programming language.
License-free – good for reproducible & open code.

MATLAB®

A high-level programming language designed for engineers and scientists.
Requires a license. Provides loads of useful resources for Neuroimaging analysis.

Environment

PROGRAMMING LANGUAGES



PACKAGE MANAGER

Conda is an open-source, cross-platform, language-agnostic package manager and environment management system.

With conda, you can use environments that have different versions of Python and packages installed in them.

You can, for example, create your **MRI analysis environment** that includes packages needed for your analysis work.

```yaml
name: mri
channels:
  - conda-forge
  - defaults
dependencies:
  - dcm2niix=1.0.20240202
  - heudiconv=1.1.6
  - nipype=1.8.5
  - pip=24.2
  - pytest=8.3.2
  - python=3.11.10 # dipy v1.9.0 dependency
  - seaborn=0.13.2
  - traits=6.4.3
  - wheel=0.44.0
  - pip:
      - antspyx==0.5.3
      - atlasreader==0.3.2
      - dipy==1.9.0
      - fury==0.11.0 # dipy v1.9.0 viualisa
      - ipykernel==6.29.3
      - ipython==8.22.1
      - ipywidgets==8.1.5
      - jupyter==1.1.1
      - matplotlib==3.8.3
```

Environment

LANGUAGES

SH

cripts

hon™

PACKAGE MANAGER

CONDA

Conda is an open-source, cross-platform, language-agnostic package manager and environment management system.

With conda, you can use environments that have different versions of Python and packages installed in them.

You can, for example, create your **MRI analysis environment** that includes packages needed for your analysis work.

https://conda.io/projects/conda/en/latest/user-guide/tasks/manage-environments.html

Creating an environment from a .yml file

```
conda env create -f mri_environment.yml
```

Environment

PROGRAMMING LANGUAGES

BASH
&
Shell Scripts

python™

MATLAB®

PACKAGE MANAGER

CONDA

CODE EDITOR

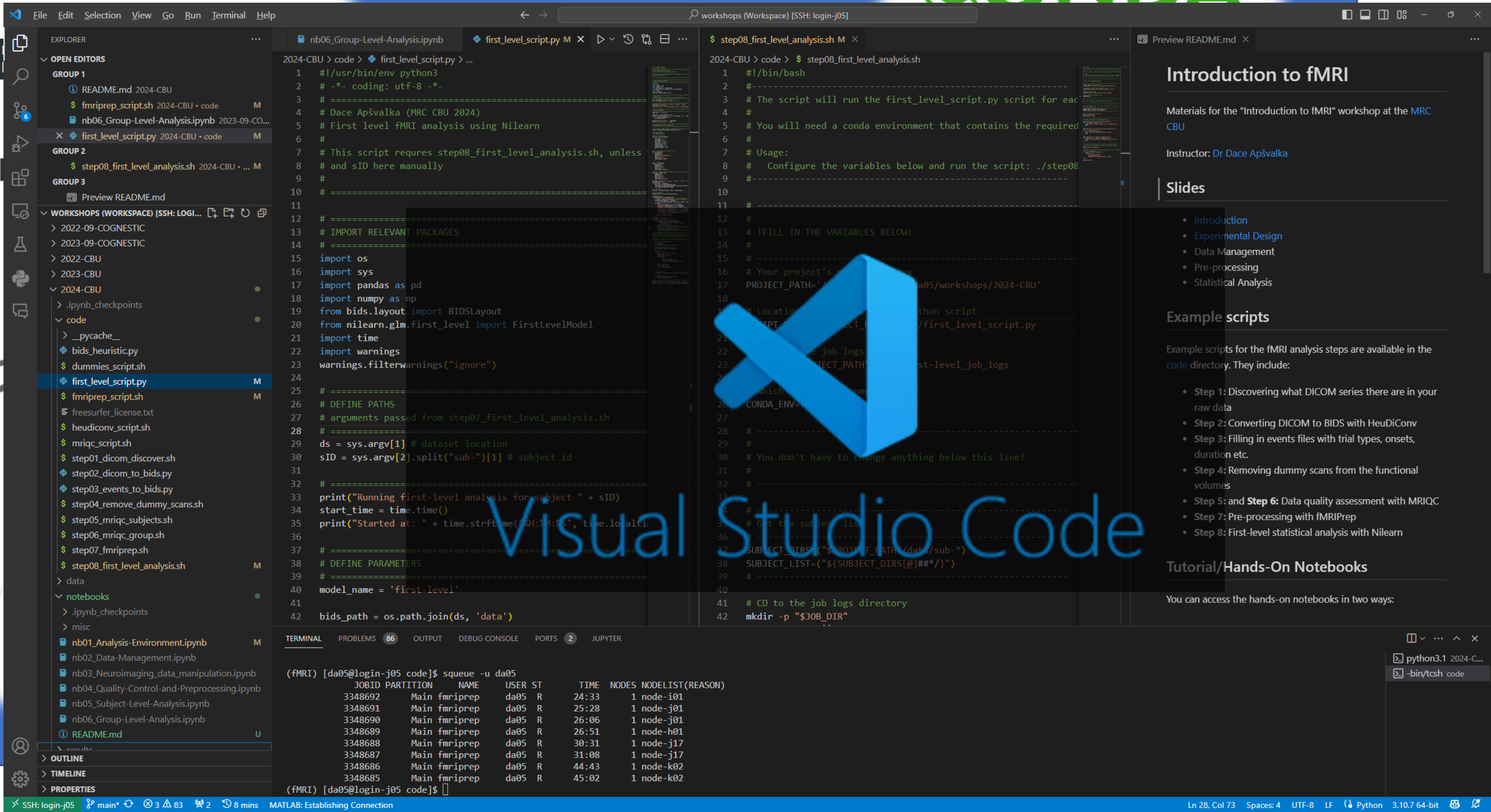
Code editors simplify and speed up typing of source code. They also support running and debugging the code.

The best editors are cross-language, cross-platform and support version control.

Environment

PROGRAMMING LANGUAGES

PACKAGE MANAGER



https://www.youtube.com/watch?v=h9cQE0WUrm4

Environment

PROGRAMMING LANGUAGES

BASH

&

Shell Scripts

python™

MATLAB®

PACKAGE MANAGER

CONDA

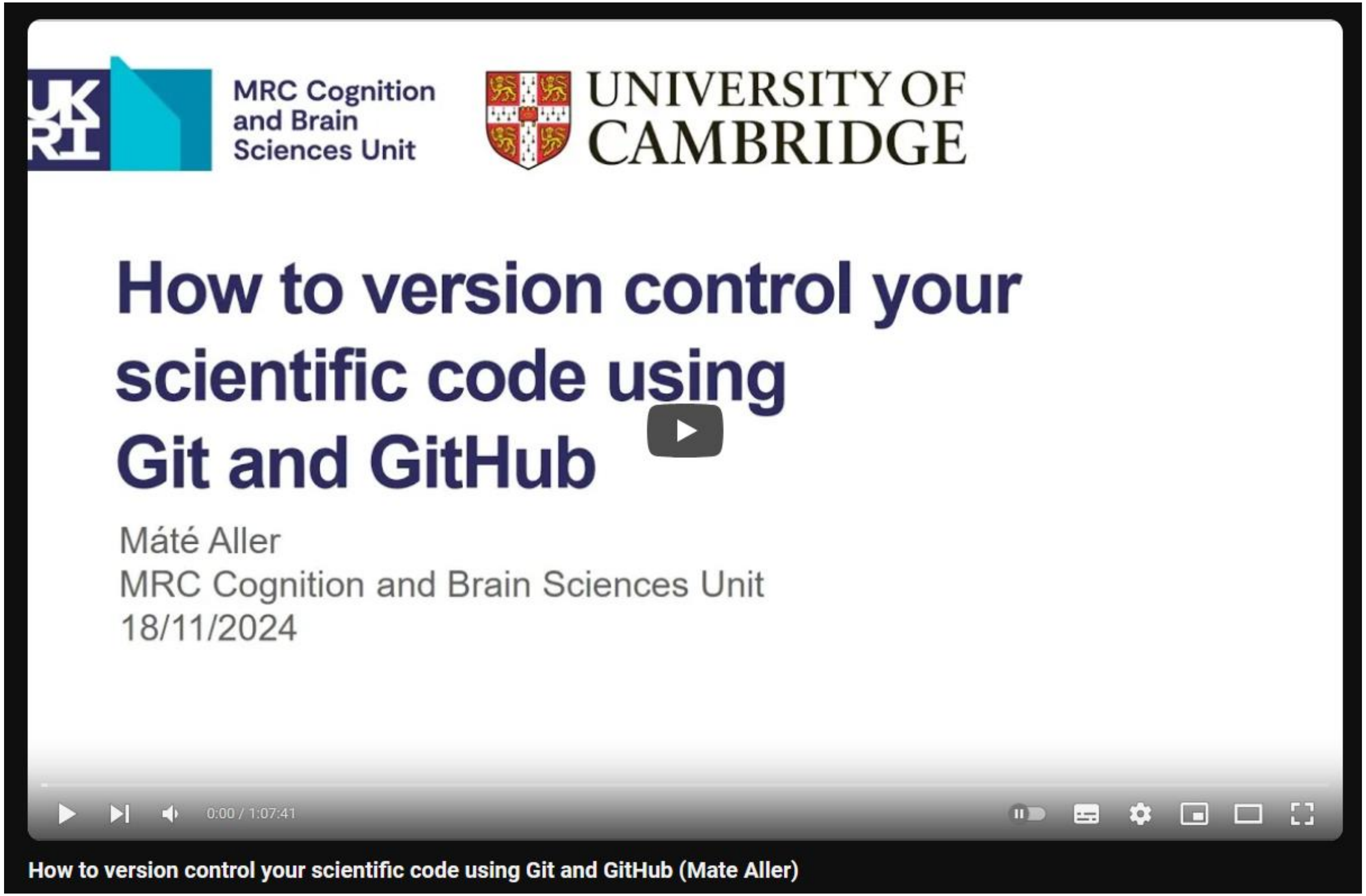
CODE EDITOR

Visual Studio Code

VERSION CONTROL

git

The practice of tracking and managing changes to software code - your analysis code.
It allows to revert selected files back to a previous state, revert the entire project back to a previous state, compare changes over time, do collaborative coding etc.



https://www.youtube.com/watch?v=ohTW4FJdmeQ

Environment

PROGRAMMING LANGUAGES

BASH & Shell Scripts

python™

MATLAB®

PACKAGE MANAGER

CONDA

CODE EDITOR

Visual Studio Code

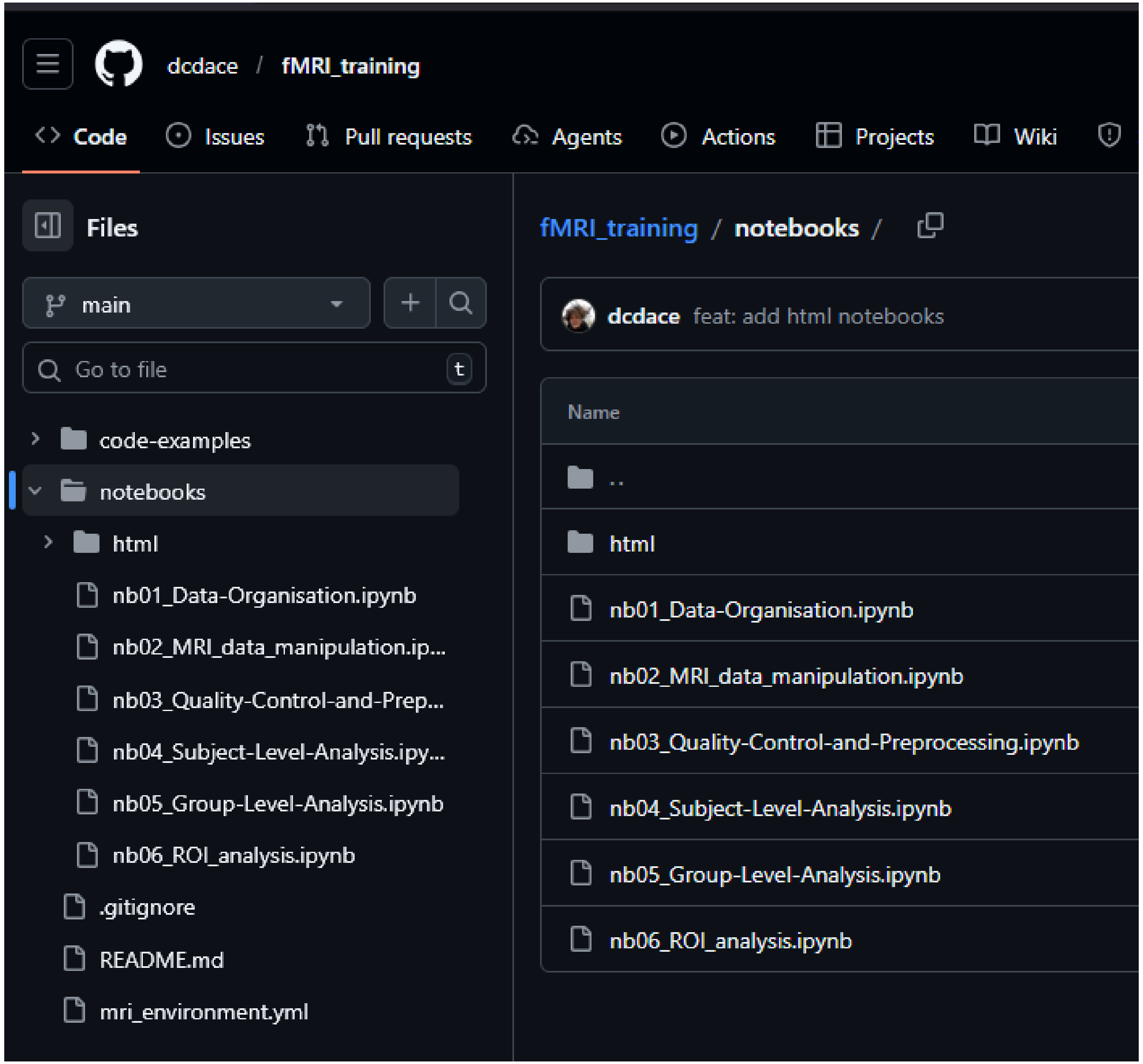
VERSION CONTROL

git

ANALYSIS NOTEBOOK

jupyter

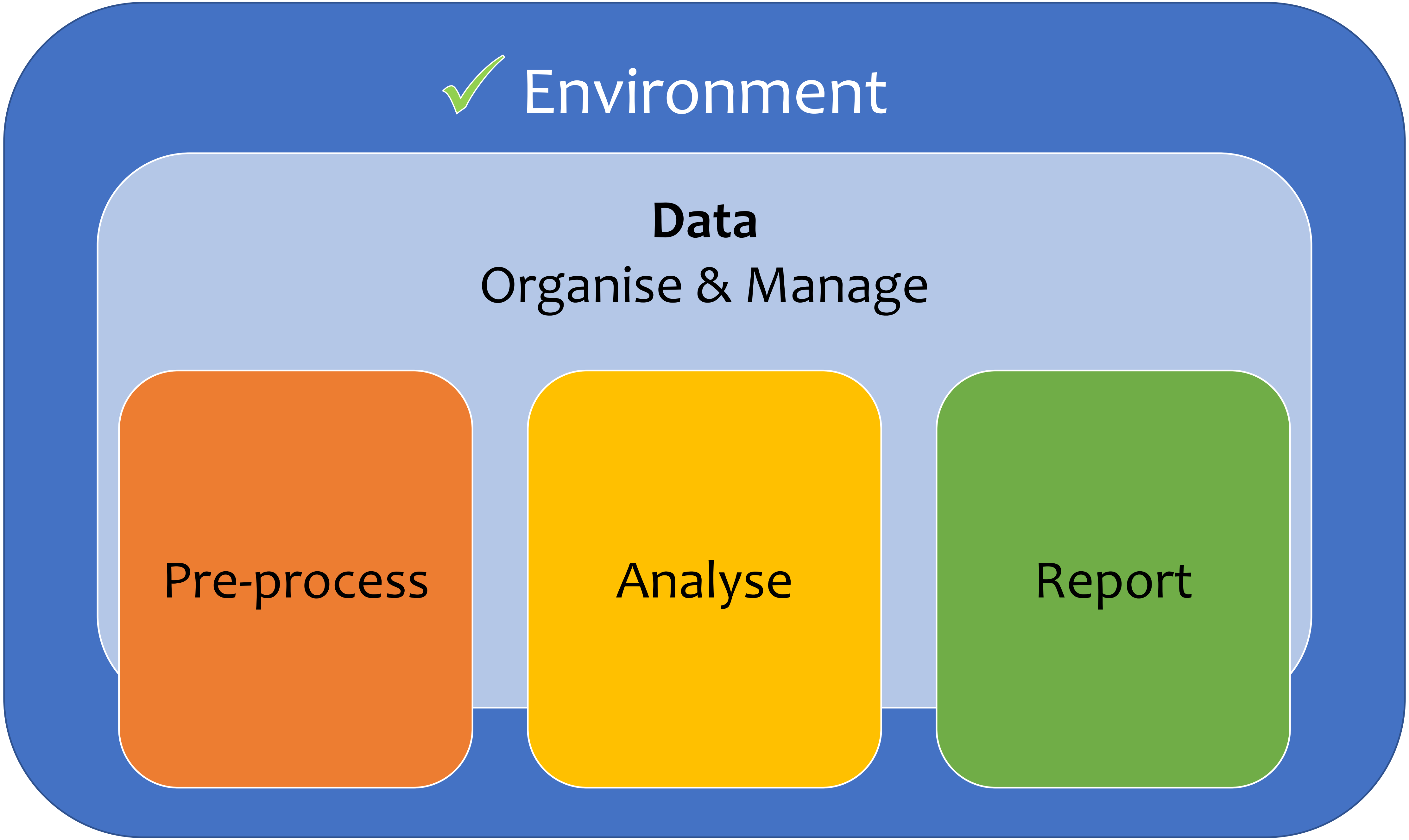
Analysis notebooks



GitHub

https://github.com/dcdace/fMRI_training

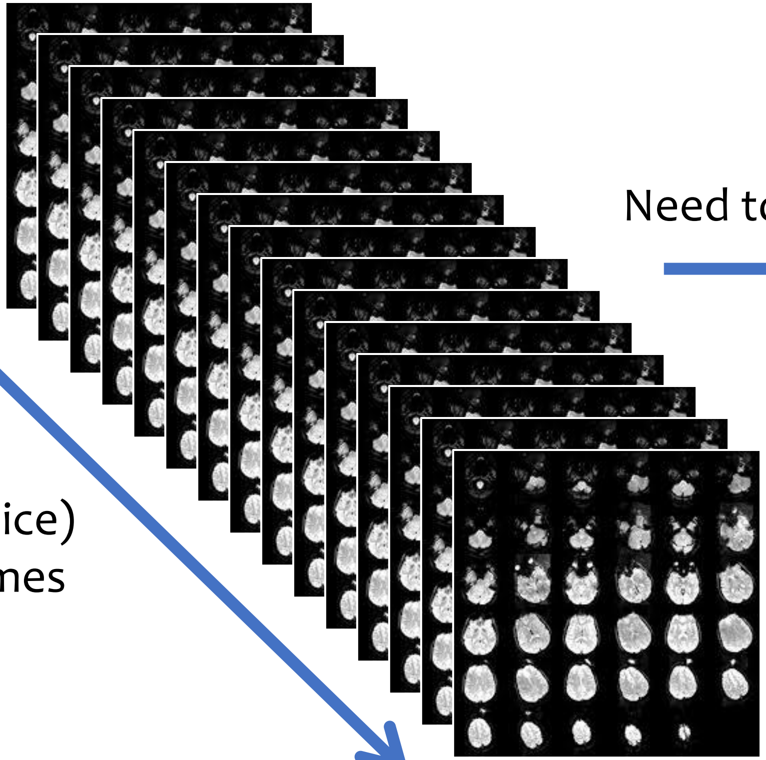
Example data (~3GB):
https://cloud.mrc-cbu.cam.ac.uk/index.php/s/rDzG98znXae9Zsy
(email me for password)


✓ Environment
Data
Organise & Manage
Pre-process
Analyse
Report

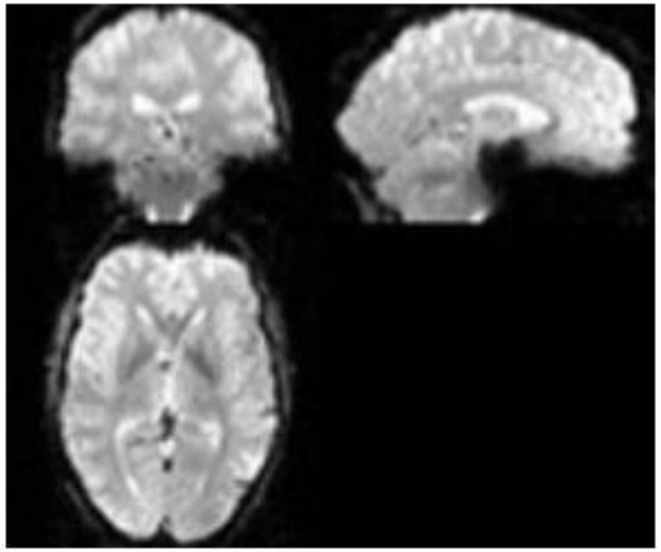
fMRI file formats

Collect the data

DICOM format



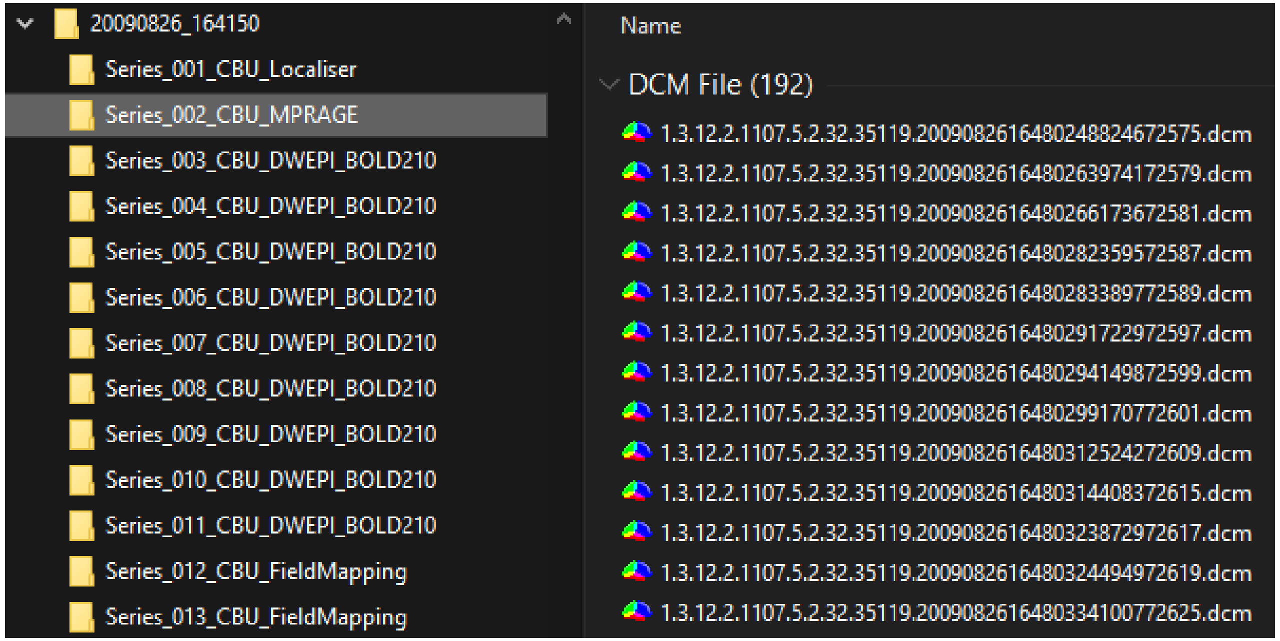
Need to convert to **NIfTI**



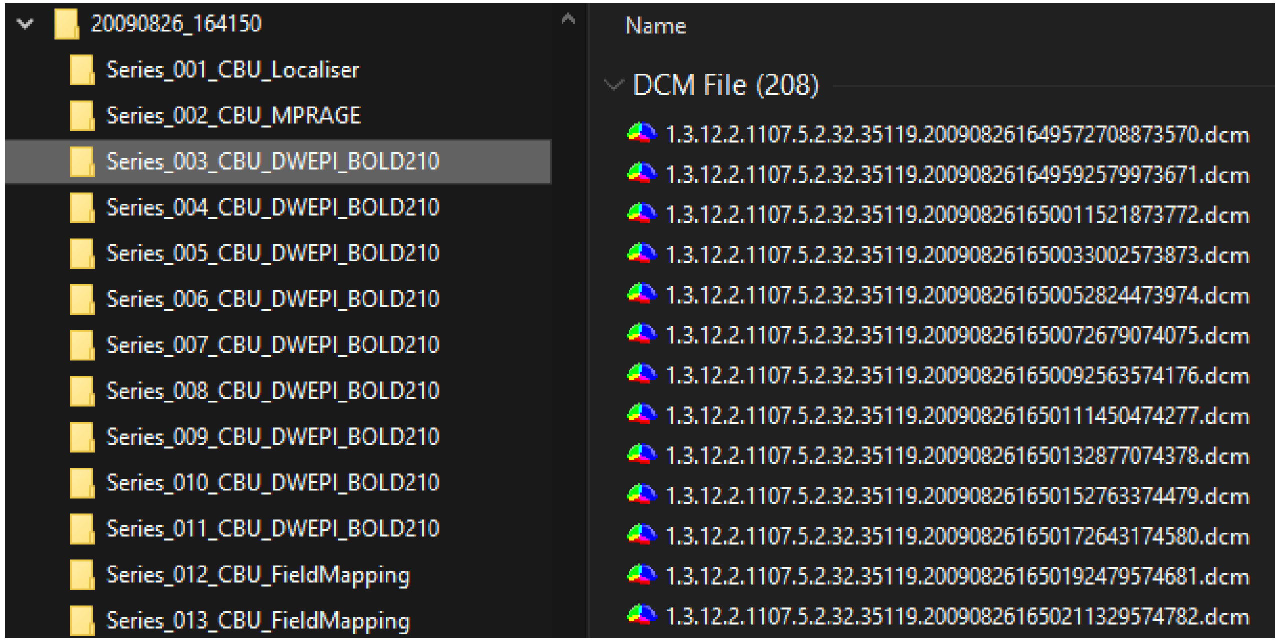
functional scan
A brain volume (slice-by-slice)
scanned every 2s > 100 times

DICOM

Anatomical (T1w) scans



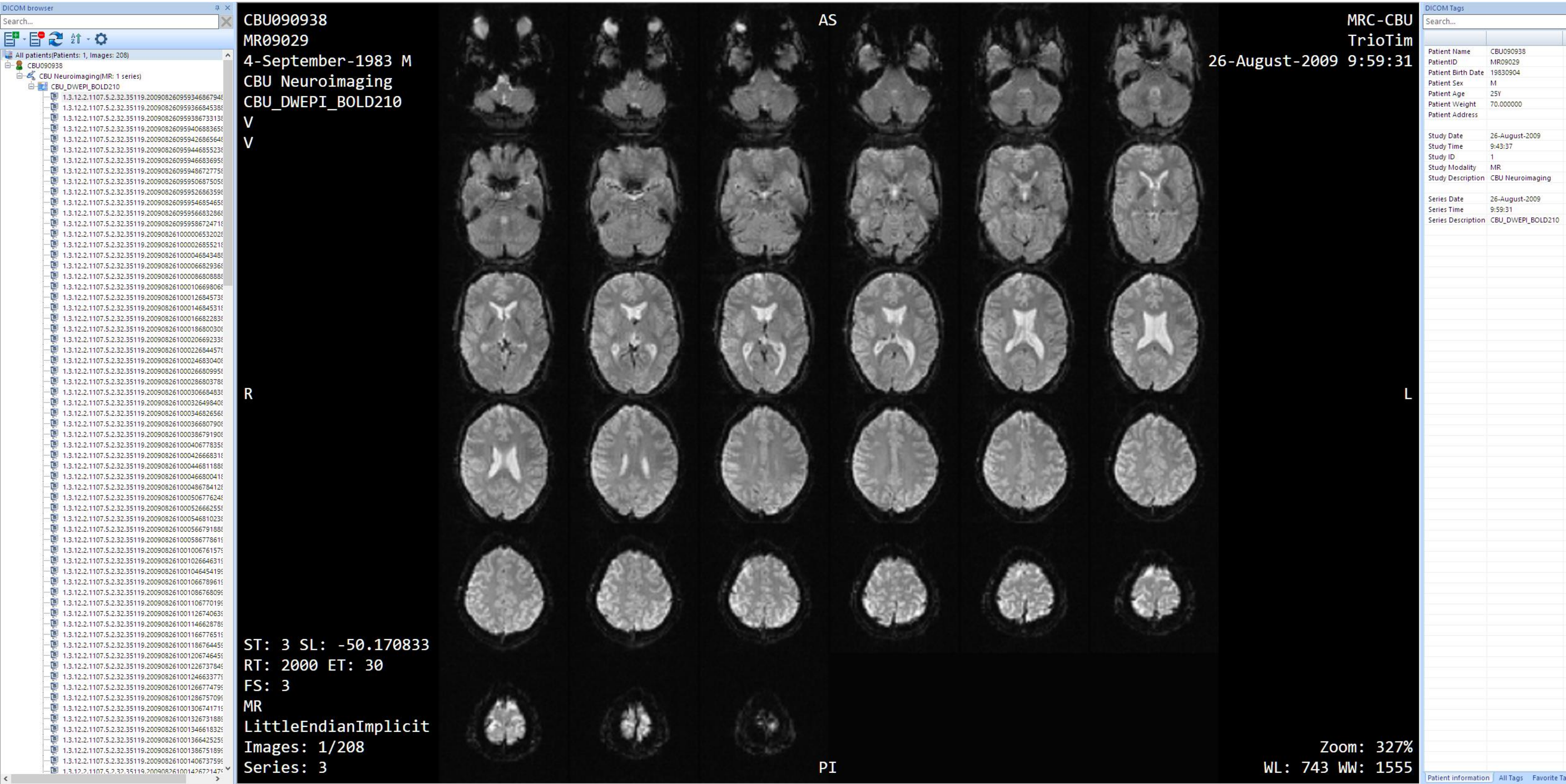
Functional (T2*/BOLD) scans



- **DICOM** - **Di**gital **I**maging and **Co**mmunications in **M**edicine (.dcm)
  - Raw data standard for storing and communicating **medical images**
  - Contains a **header** (meta data) and the actual **image** itself
  - Images are stored in **2D layers**
  - A separate file for each **slice** (T1w) or **volume** (BOLD)

DICOM: Anatomical

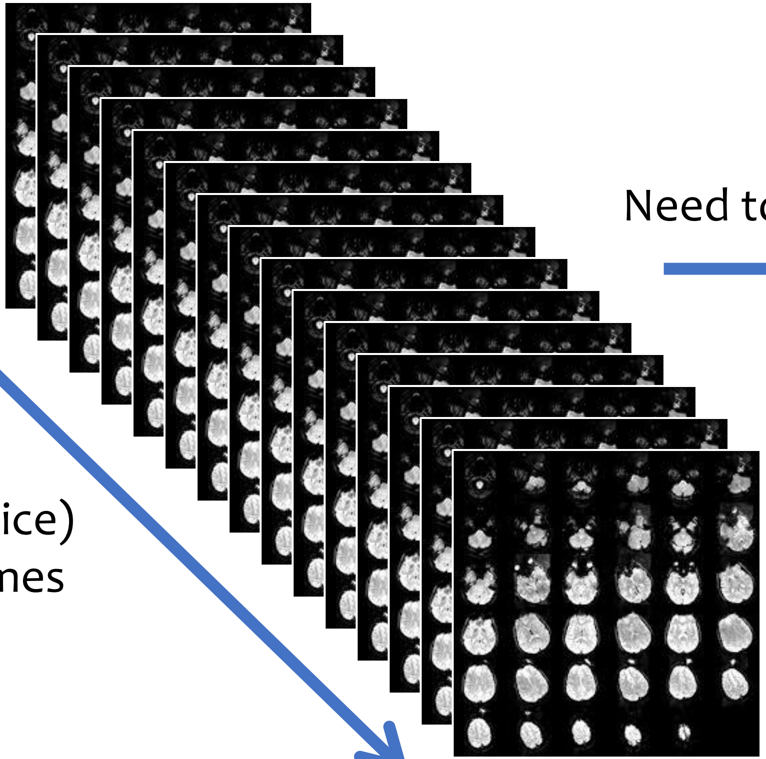
DICOM: Functional



Anatomical (T1w) image & **Functional** (T2*/BOLD) image

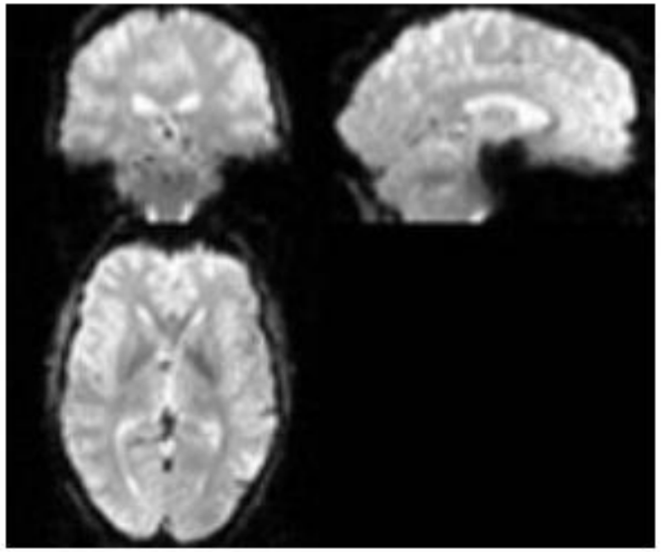
Collect the data

DICOM format



functional scan
A brain volume (slice-by-slice)
scanned every 2s > 100 times

Need to convert to **NIfTI**

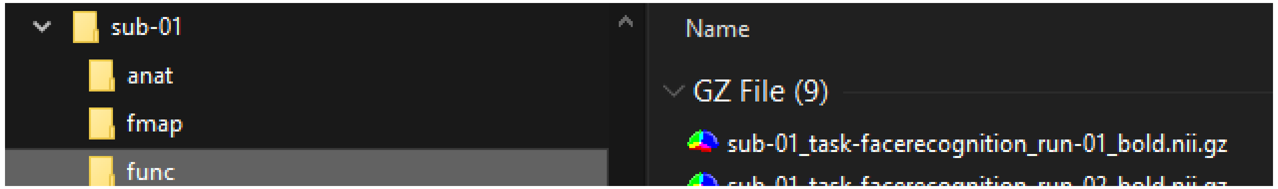


NIfTI

Anatomical (T1w) scans

3D file

Functional (T2*/BOLD) scans



4D file

- **NIfTI** – **N**euroimaging **I**n**f**ormatics **T**echnology **I**nitiative (.nii, .nii.gz)
  - Standardised representation of **brain images**, cross-platform, cross-software
  - Contains **header** and **image**
  - 3D or 4D files (all slices/volumes in a single file)

- **DICOM vs NIFTI**
  - All medical vs brain images
  - 2D layer files vs 3D/4D files
  - More vs less metadata
    - When converting DICOM to NIfTI need to be aware of that and save all metadata that might be needed

- **DICOM to NIfTI**
  - Several tools are available
    - For example, ***dcm2niix*** Python package

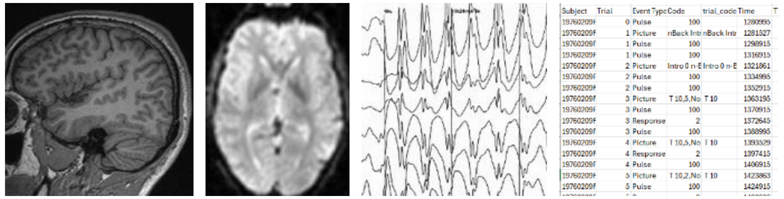
File organisation

Complexity of Brain Imaging Data

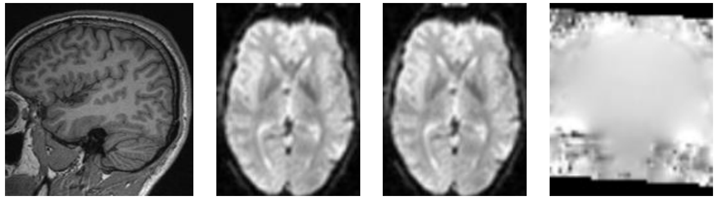
- A single study often involves:
  - Multiple imaging protocols
  - Many participants, sometimes across sessions

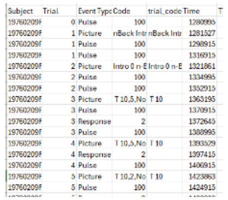
- Produces numerous files in diverse formats:
  - From simple text files (logs, metadata)
  - To large multidimensional images

- Datasets become large, diverse, and complex



Complexity of Brain Imaging Data





Subject	Trial	Event Typ	Code	trial_code	Time	T
19760209F	0	Pulse	100		1280995	
19760209F	1	Picture	nBack Intr	nBack Intr	1281527	
19760209F	1	Pulse	100		1298915	
19760209F	1	Pulse	100		1316915	
19760209F	2	Picture	Intro 0 n-E	Intro 0 n-E	1321861	
19760209F	2	Pulse	100		1334995	
19760209F	2	Pulse	100		1352915	
19760209F	3	Picture	T 10,5,No	T 10	1363195	
19760209F	3	Pulse	100		1370915	
19760209F	3	Response	2		1372645	
19760209F	3	Pulse	100		1388995	
19760209F	4	Picture	T 10,5,No	T 10	1393529	
19760209F	4	Response	2		1397415	
19760209F	4	Pulse	100		1406915	
19760209F	5	Picture	T 10,2,No	T 10	1423863	
19760209F	5	Pulse	100		1424915	

```
scans/
├── subj01/
│   ├── s01_t1w.nii.gz
│   ├── s01_bold1_run1.nii.gz
│   ├── s01_bold1_run2.nii.gz
│   └── s01_fieldmap.nii.gz
├── subj02/
│   ├── s02_t1w.nii.gz
│   ├── s02_bold1_run1.nii.gz
│   ├── s02_bold1_run2.nii.gz
│   └── s02_fieldmap.nii.gz
...
├── subj100/
│   ├── s100_t1w.nii.gz
│   ...
logs/
├── subj01/
│   ├── events_taskA_run1.tsv
│   ├── events_taskA_run2.tsv
├── subj02/
│   ├── events_taskA_run1.tsv
│   ├── events_taskA_run2.tsv
...
├── subj100/
│   ...
protocols/
...
```

and even more files
(more modalities, sessions, tasks)

Complexity of Brain Imaging Data


```
scans/
├── subj01/
│   ├── s01_t1w.nii.gz
│   ├── s01_bold1_run1.nii.gz
│   ├── s01_bold1_run2.nii.gz
│   └── s01_fieldmap.nii.gz
├── subj02/
│   ├── s02_t1w.nii.gz
│   ├── s02_bold1_run1.nii.gz
│   ├── s02_bold1_run2.nii.gz
│   └── s02_fieldmap.nii.gz
│   ├── s100_t1w.nii.gz
├── subj01/
│   ├── events_taskA_run1.tsv
│   ├── events_taskA_run2.tsv
├── subj02/
│   ├── events_taskA_run1.tsv
│   ├── events_taskA_run2.tsv
...
├── subj100/
│   ...
protocols/
...
```

(more modalities, sessions, tasks)

Many possible ways to name and organise the data

Inconsistent Data Organisation

- Difficult for others (and you!) to understand data and track changes
- Scripts must be adapted, can't be easily reused
- Huge effort to automate workflows, no way to automatically validate datasets
- Increased risk of errors (wrong files, outdated versions)
- Time wasted searching and reorganising
- Harder to reproduce results and collaborate

Inconsistent Data Organisation

- Difficult for others (and you!) to understand data and track changes
- Scripts must be adapted, can't be easily reused
- Huge effort to automate workflows, no way to automatically validate datasets
- Increased risk of errors (wrong files, outdated versions)
- Time wasted searching and reorganising
- Harder to reproduce results and collaborate

Wouldn't it be much easier if everybody organised the files in the same way?

Brain **I**maging **D**ata **S**tructure (BIDS)

- A standard for organising data and metadata across various neuroscience modalities (MRI, MEG, EEG, PET)
- Enables easier sharing, reuse, and application of automated pipelines and quality assurance protocols

Brain **I**maging **D**ata **S**tructure (BIDS)

- A standard for organising data and metadata across various neuroscience modalities (MRI, MEG, EEG, PET)
- Enables easier sharing, reuse, and application of automated pipelines and quality assurance protocols

BIDS
BRAIN IMAGING DATA STRUCTURE

SCIENTIFIC DATA

OPEN

SUBJECT CATEGORIES
» Data publication and archiving
» Research data

The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments

Krzysztof J. Gorgolewski[1], Tibor Auer[2], Vince D. Calhoun[3,4], R. Cameron Craddock[5,6], Samir Das[7], Eugene P. Duff[8], Guillaume Flandin[9], Satrajit S. Ghosh[10,11], Tristan Glatard[7,12], Yaroslav O. Halchenko[13], Daniel A. Handwerker[14], Michael Hanke[15,16], David Keator[17], Xiangrui Li[18], Zachary Michael[19], Camille Maumet[20], B. Nolan Nichols[21,22], Thomas E. Nichols[20,23], John Pellman[6], Jean-Baptiste Poline[24], Ariel Rokem[25], Gunnar Schaefer[1,26], Vanessa Sochat[27], William Triplett[1], Jessica A. Turner[3,28], Gaël Varoquaux[29] & Russell A. Poldrack[1]

Received: 18 December 2015
Accepted: 19 May 2016
Published: 21 June 2016

PLOS | COMPUTATIONAL BIOLOGY

RESEARCH ARTICLE

BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods

Krzysztof J. Gorgolewski[1]*, Fidel Alfaro-Almagro[2], Tibor Auer[3], Pierre Bellec[4,5], Mihai Capotă[6], M. Mallar Chakravarty[7,8], Nathan W. Churchill[9], Alexander Li Cohen[10], R. Cameron Craddock[11,12], Gabriel A. Devenyi[7,8], Anders Eklund[13,14,15], Oscar Esteban[1], Guillaume Flandin[16], Satrajit S. Ghosh[17,18], J. Swaroop Guntupalli[19], Mark Jenkinson[2], Anisha Keshavan[20], Gregory Kiar[21,22], Franziskus Liem[23], Pradeep Reddy Raamana[24,25], David Raffelt[26], Christopher J. Steele[7,8], Pierre-Olivier Quirion[15], Robert E. Smith[26], Stephen C. Strother[24,25], Gaël Varoquaux[27], Yida Wang[6], Tal Yarkoni[28], Russell A. Poldrack[1]

BIDS
BRAIN IMAGING DATA STRUCTURE

Raw DICOM format

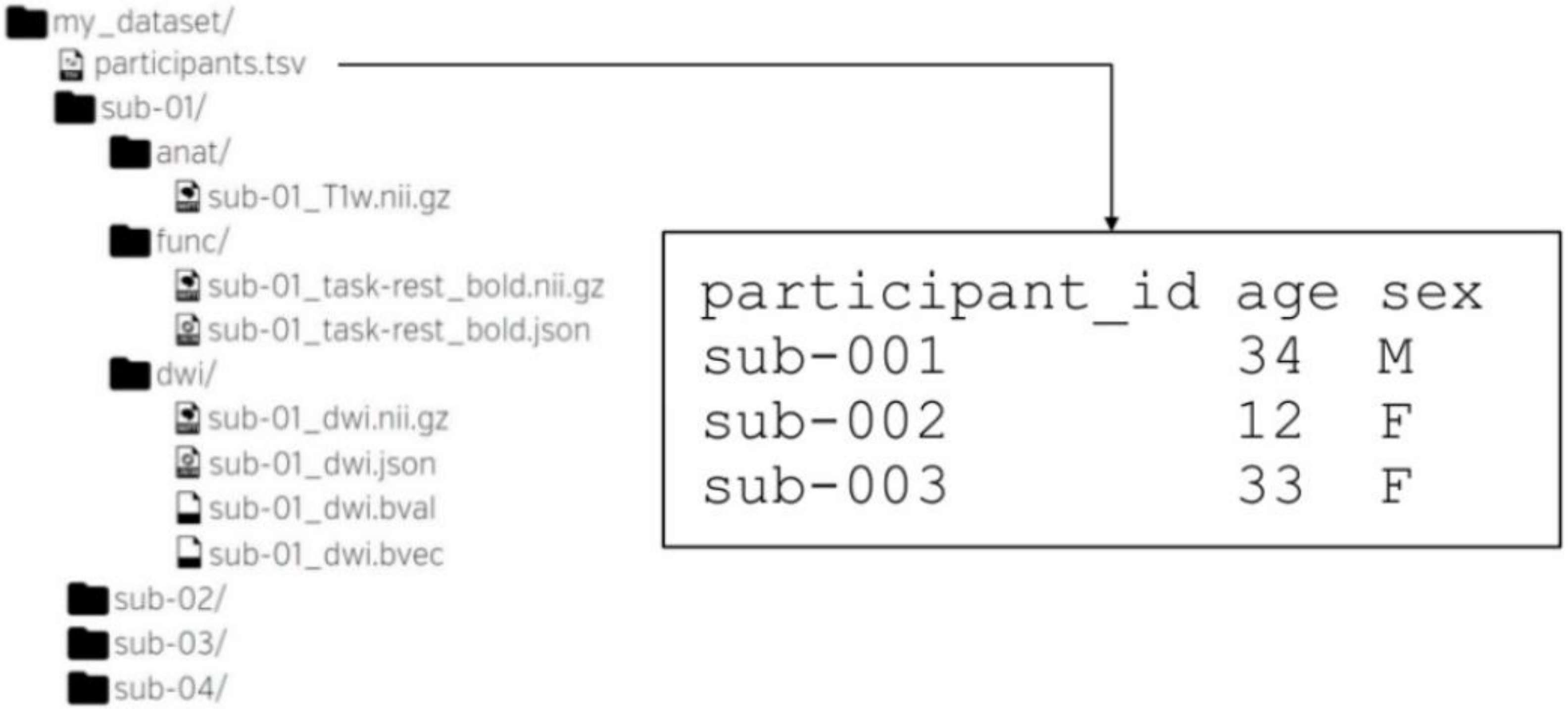
```
mridata/
└── CBU090962_MR09029
    └── 20090902_100102
        ├── Series_001_CBU_Localiser
        ├── Series_002_CBU_MPRAGE
        ├── Series_003_CBU_DWEPI_BOLD210
        ├── Series_004_CBU_DWEPI_BOLD210
        ├── Series_005_CBU_DWEPI_BOLD210
        ├── Series_006_CBU_DWEPI_BOLD210
        ├── Series_007_CBU_DWEPI_BOLD210
        ├── Series_008_CBU_DWEPI_BOLD210
        ├── Series_009_CBU_DWEPI_BOLD210
        ├── Series_010_CBU_DWEPI_BOLD210
        ├── Series_011_CBU_DWEPI_BOLD210
        ├── Series_012_CBU_FieldMapping
        └── Series_013_CBU_FieldMapping
```

BIDS NIfTI format

```
├── sub-15
│   └── ses-mri
│       ├── anat
│       │   ├── sub-15_ses-mri_T1w.json
│       │   └── sub-15_ses-mri_T1w.nii.gz
│       ├── fmap
│       │   ├── sub-15_ses-mri_acq-func_magnitude1.json
│       │   ├── sub-15_ses-mri_acq-func_magnitude1.nii.gz
│       │   ├── sub-15_ses-mri_acq-func_magnitude2.json
│       │   ├── sub-15_ses-mri_acq-func_magnitude2.nii.gz
│       │   ├── sub-15_ses-mri_acq-func_phasediff.json
│       │   └── sub-15_ses-mri_acq-func_phasediff.nii.gz
│       ├── func
│       │   ├── sub-15_ses-mri_task-facerecognition_run-01_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-01_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-01_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-02_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-02_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-02_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-03_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-03_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-03_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-04_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-04_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-04_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-05_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-05_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-05_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-06_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-06_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-06_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-07_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-07_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-07_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-08_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-08_bold.nii.gz
│       │   ├── sub-15_ses-mri_task-facerecognition_run-08_events.tsv
│       │   ├── sub-15_ses-mri_task-facerecognition_run-09_bold.json
│       │   ├── sub-15_ses-mri_task-facerecognition_run-09_bold.nii.gz
│       │   └── sub-15_ses-mri_task-facerecognition_run-09_events.tsv
│       └── sub-15_ses-mri_scans.tsv
```

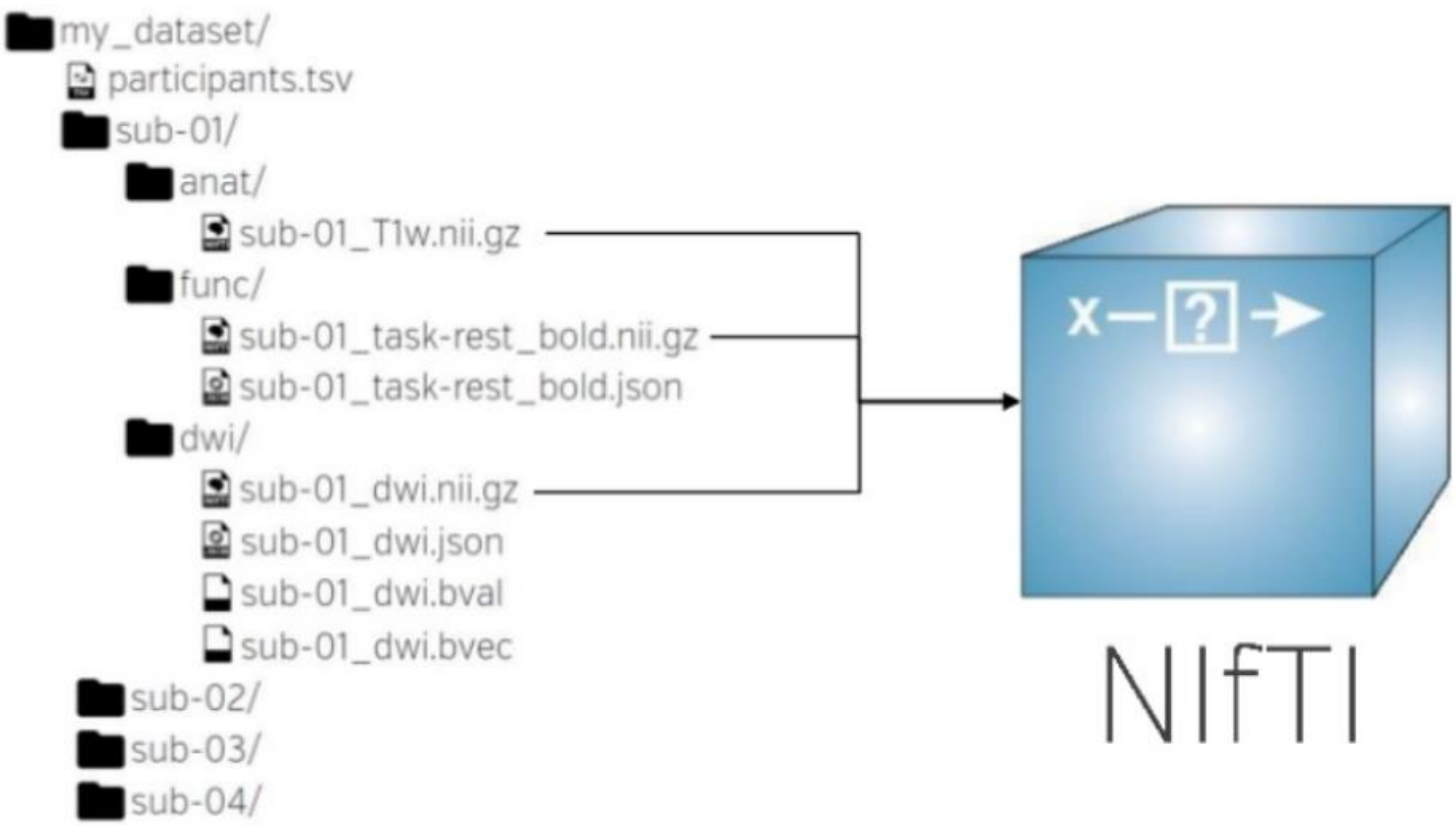
BIDS

- Contains participant information



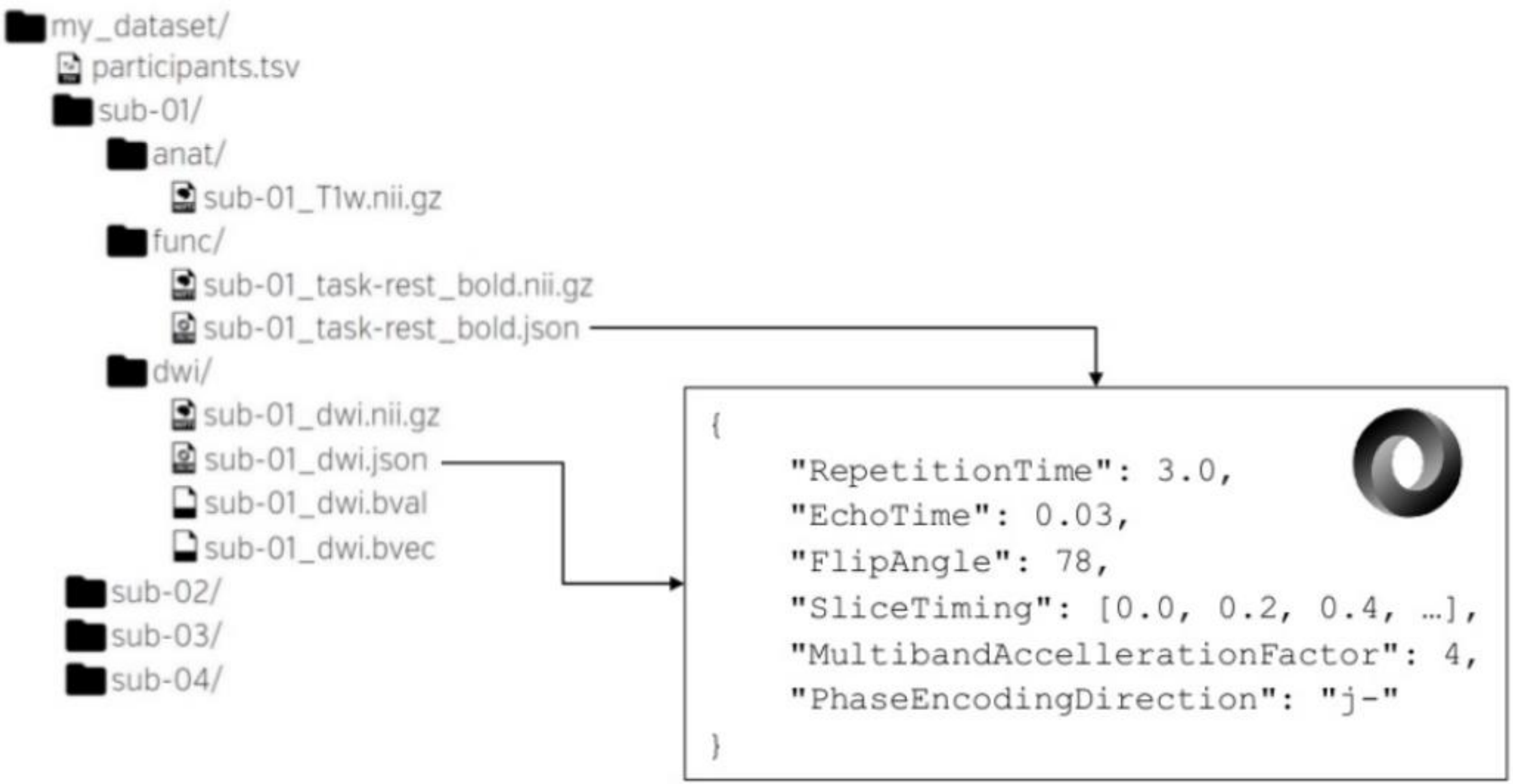
BIDS

- Contains data files



BIDS

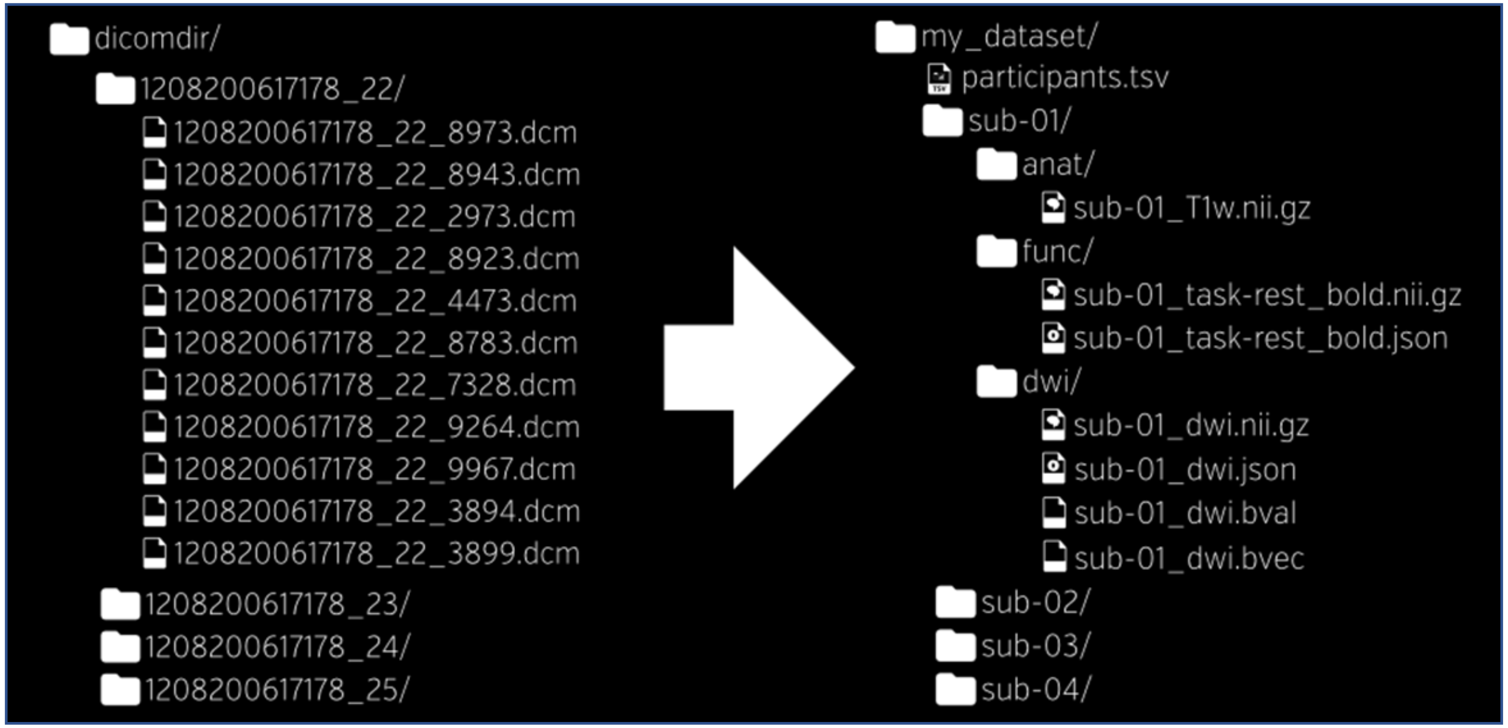
- Contains study specific JSON (metadata) files

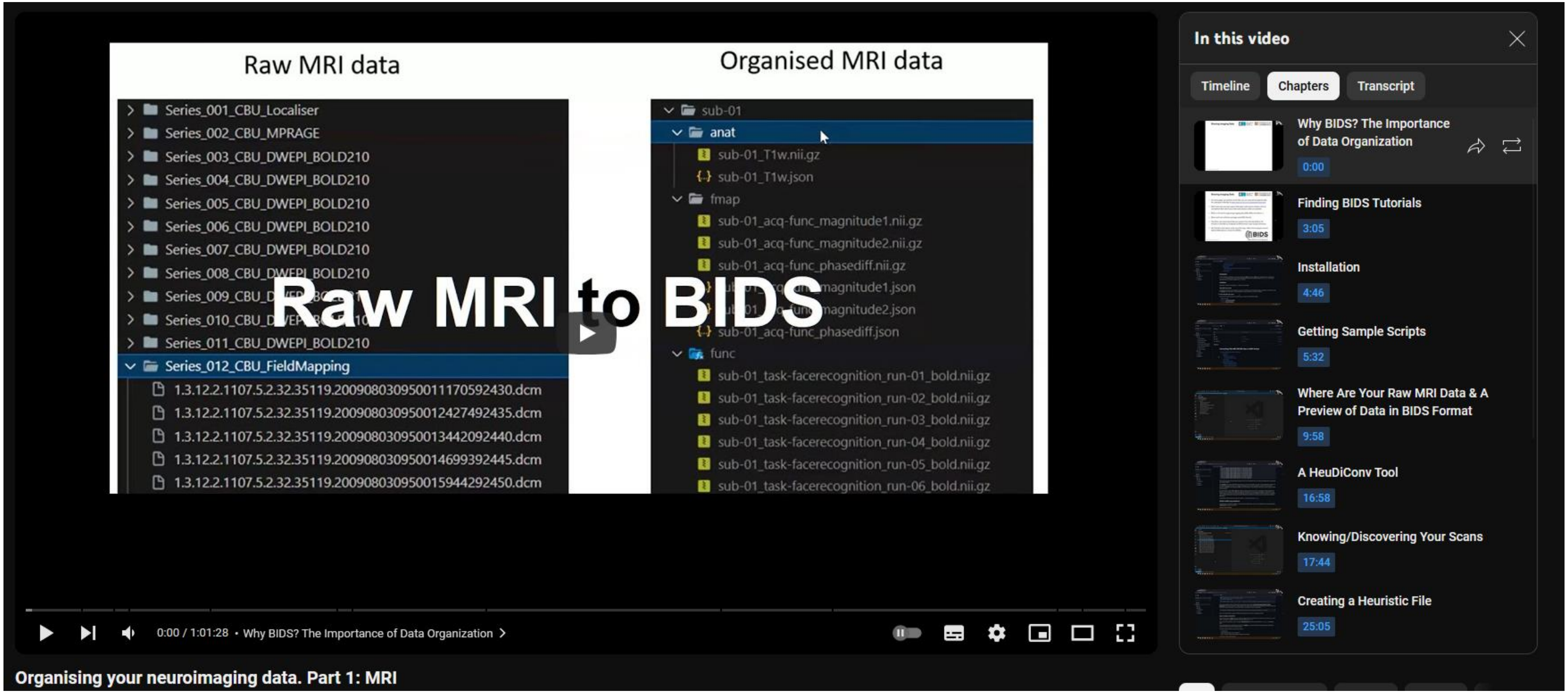


DICOM → BIDS

Many BIDS converters available

HeuDiConv (**Heu**ristic **Di**com **Conv**ersion)
provides sophisticated and flexible creation of BIDS datasets.





https://www.youtube.com/watch?v=yQBGsZMttCc

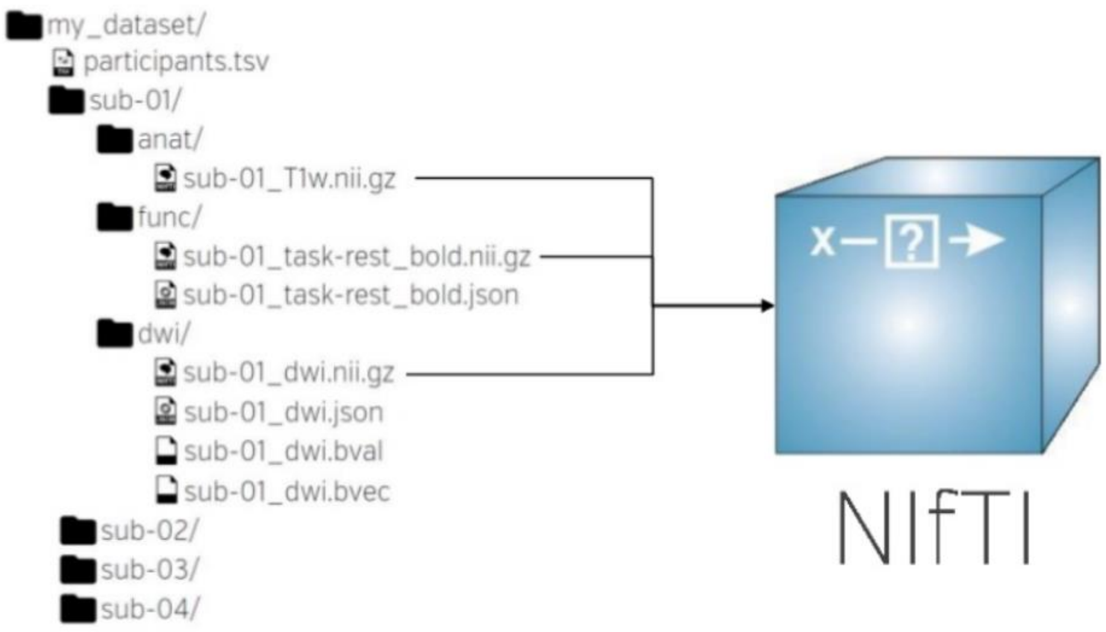
https://github.com/MRC-CBU/BIDS_conversion/tree/main/MRI

Environment

Data

Organise & Manage

Notebook: nb01_Data-Organisation.ipynb



Imaging data content

MRI data structure

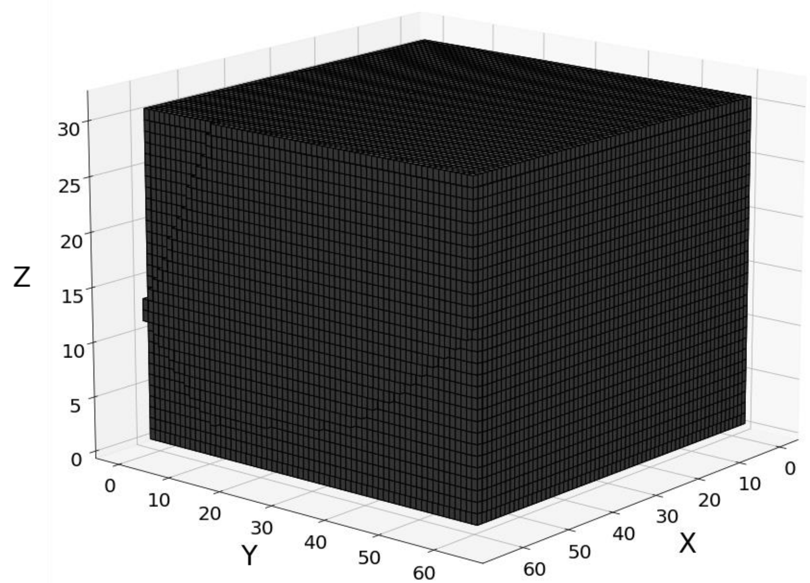
A 3D or 4D arrays of numbers

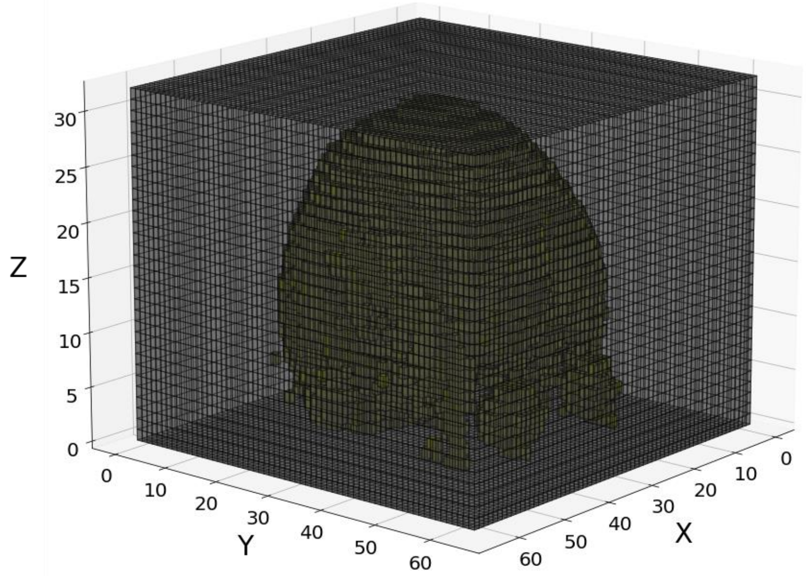
```
([[[ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.],
   ...,
   [ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.]],

  [[ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0., 25., 23., ..., 23., 32.,  0.],
   [ 0., 28., 21., ..., 25., 25.,  0.],
   ...,
   [ 0., 26., 24., ..., 40., 20.,  0.],
   [ 0., 44., 28., ..., 30., 21.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.]],

  [[ 0.,  0.,  0., ...,  0.,  0.,  0.],
   [ 0., 28., 26., ..., 31., 29.,  0.],
   [ 0., 32., 30., ..., 22., 21.,  0.],
   ...,
   [ 0., 27., 24., ..., 31., 30.,  0.],
   [ 0., 30., 23., ..., 37., 22.,  0.],
   [ 0.,  0.,  0., ...,  0.,  0.,  0.]],

  ...,
```





MRI data structure

A 3D or 4D arrays of numbers – intensity values

```
([[[ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.],
   ...,
   [ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.]],

  [[ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,  25.,  23., ...,  23.,  32.,   0.],
   [ 0.,  28.,  21., ...,  25.,  25.,   0.],
   ...,
   [ 0.,  26.,  24., ...,  40.,  20.,   0.],
   [ 0.,  44.,  28., ...,  30.,  21.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.]],

  [[ 0.,   0.,   0., ...,   0.,   0.,   0.],
   [ 0.,  28.,  26., ...,  31.,  29.,   0.],
   [ 0.,  32.,  30., ...,  22.,  21.,   0.],
   ...,
   [ 0.,  27.,  24., ...,  31.,  30.,   0.],
   [ 0.,  30.,  23., ...,  37.,  22.,   0.],
   [ 0.,   0.,   0., ...,   0.,   0.,   0.]],

  ...,
```

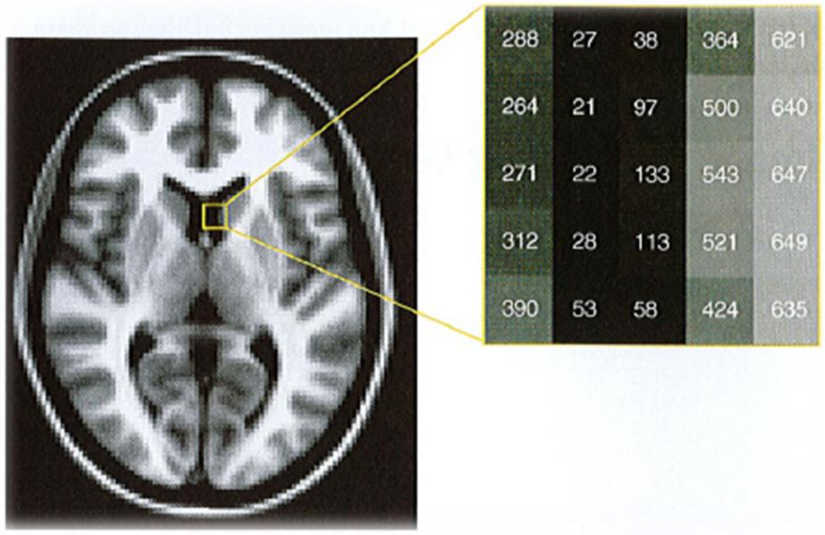
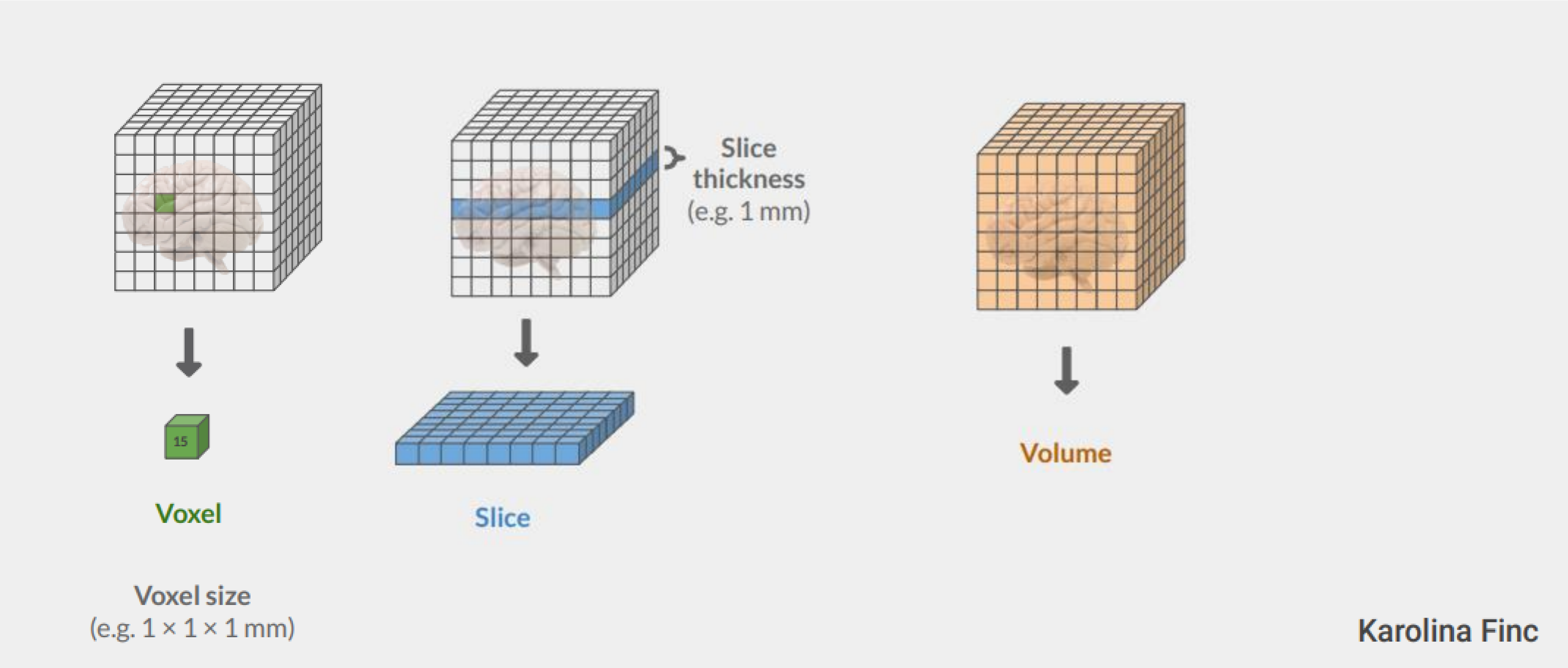


Image from Poldrack et al., 2011

MRI data structure

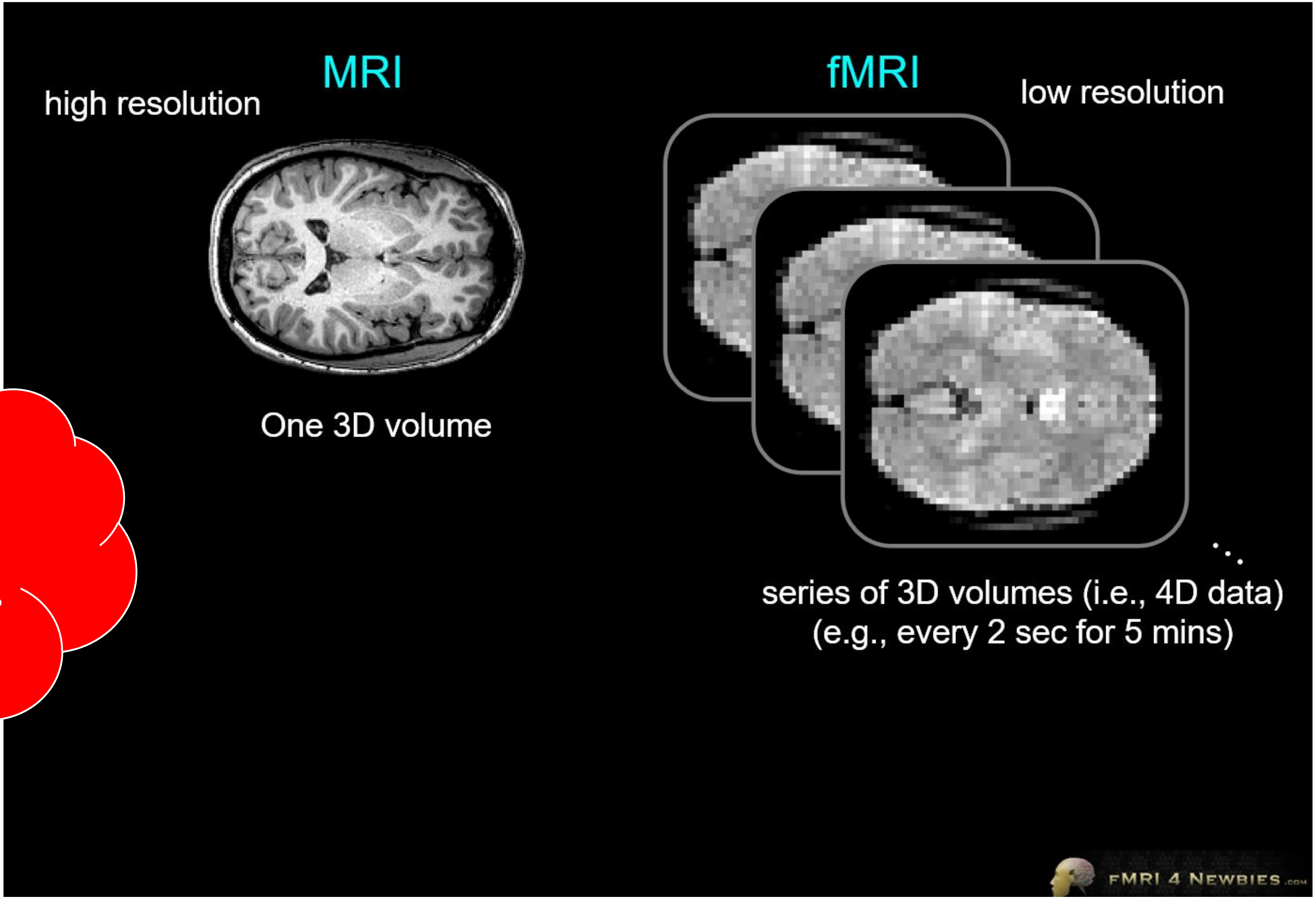


MRI data

MRI data

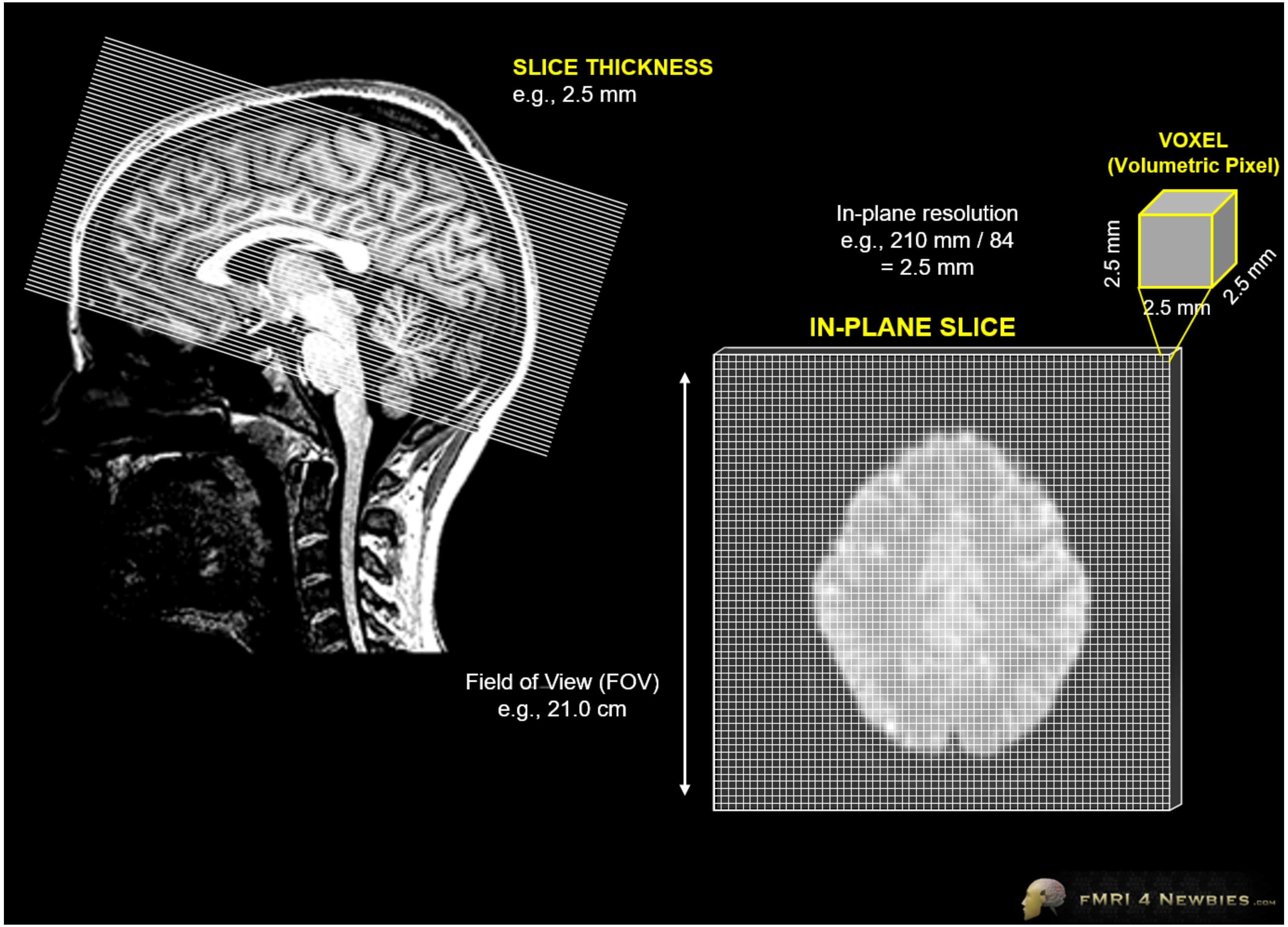
What determines the resolution?

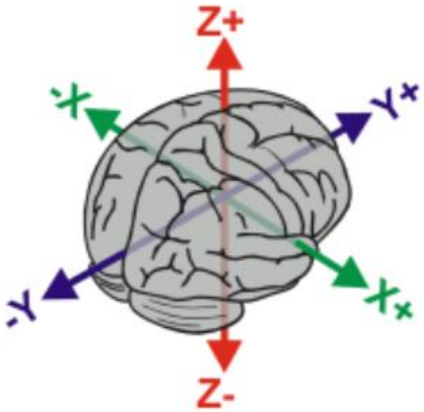
Why can't we acquire the functional images with higher resolution?

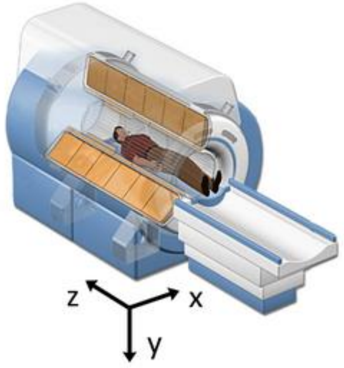


fMRI data

- Acquired in slices (usually axial; z-axis)

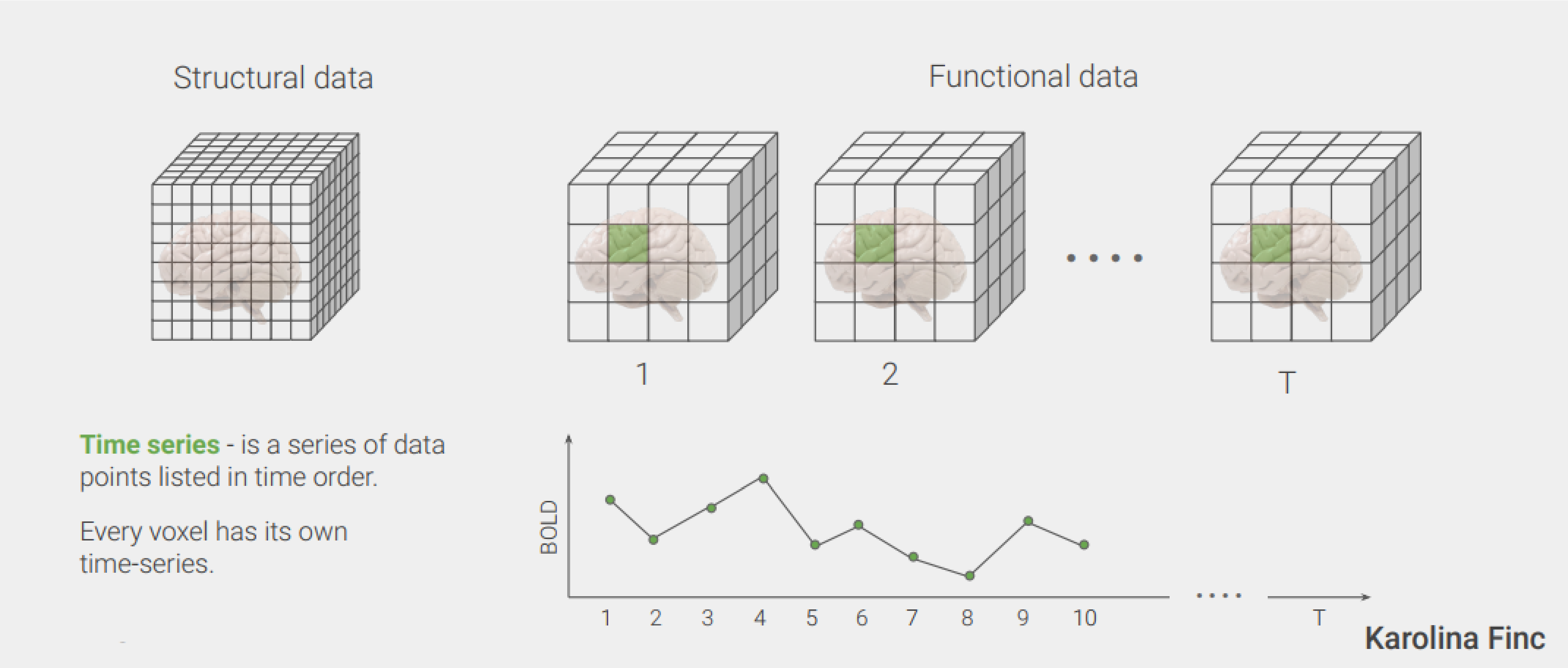






- Temporal resolution (TR), usually 1.5-3s
- Modern sequences allow acquiring multiple slices at the same time (multi-band)
- Typically, 30-50 slices acquired
- More slices = longer TR

MRI data structure

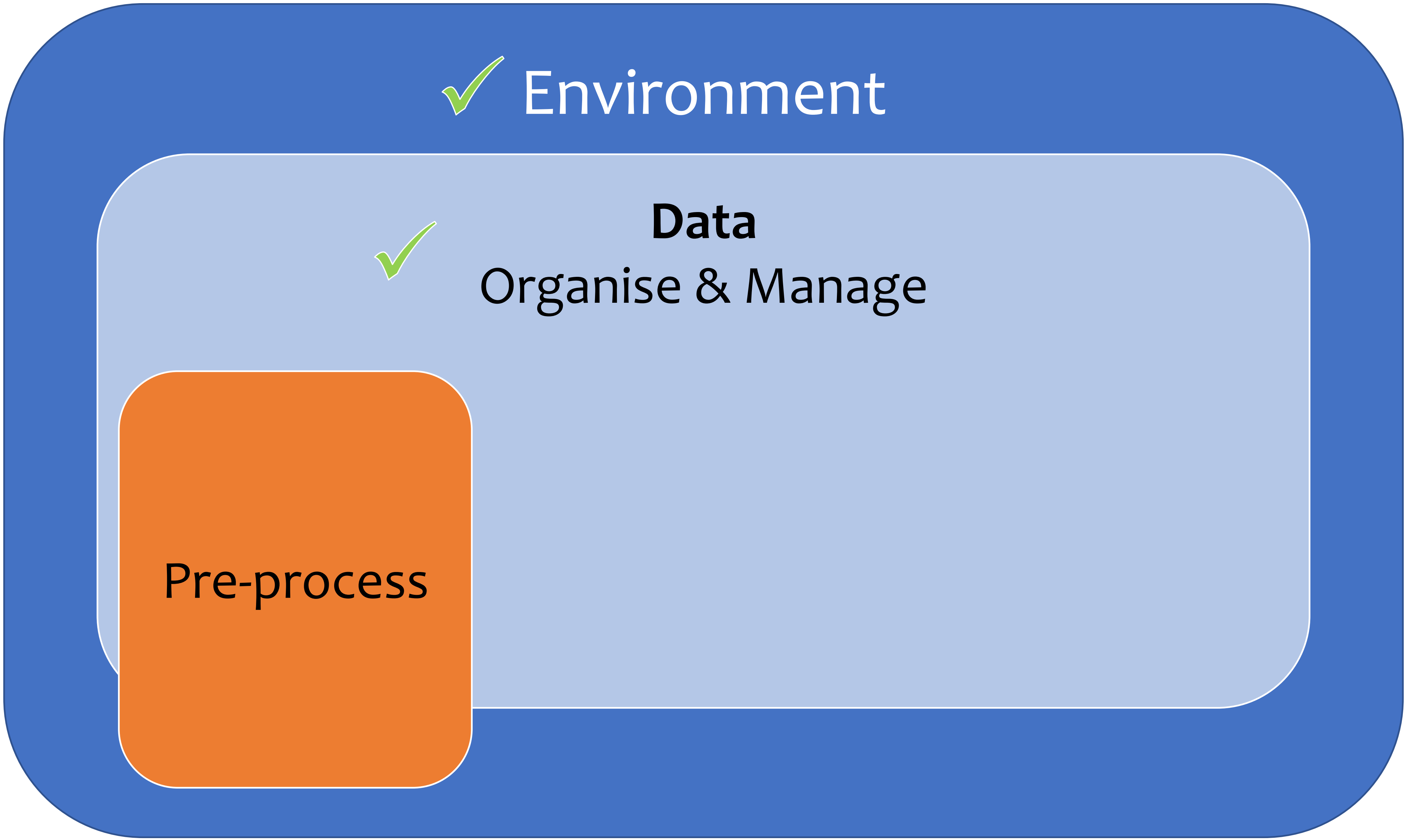


Environment

Data

Organise & Manage

Notebook: [nb02_MRI_data_manipulation.ipynb]


✓ Environment
✓ Data
Organise & Manage
Pre-process

Outline

- Introduction
- Experimental design
- Data management
- **Pre-processing**
- Statistical analysis
- Practical demo