MRC Cognition and Brain Sciences Unit
UNIVERSITY OF CAMBRIDGE

Functional MRI
pre-processing and analysis

Dace Apšvalka
[Datza]

COGNESTIC, 2025

Functional MRI (fMRI)

Diagram adapted from Glover, 2011

Functional MRI (fMRI)

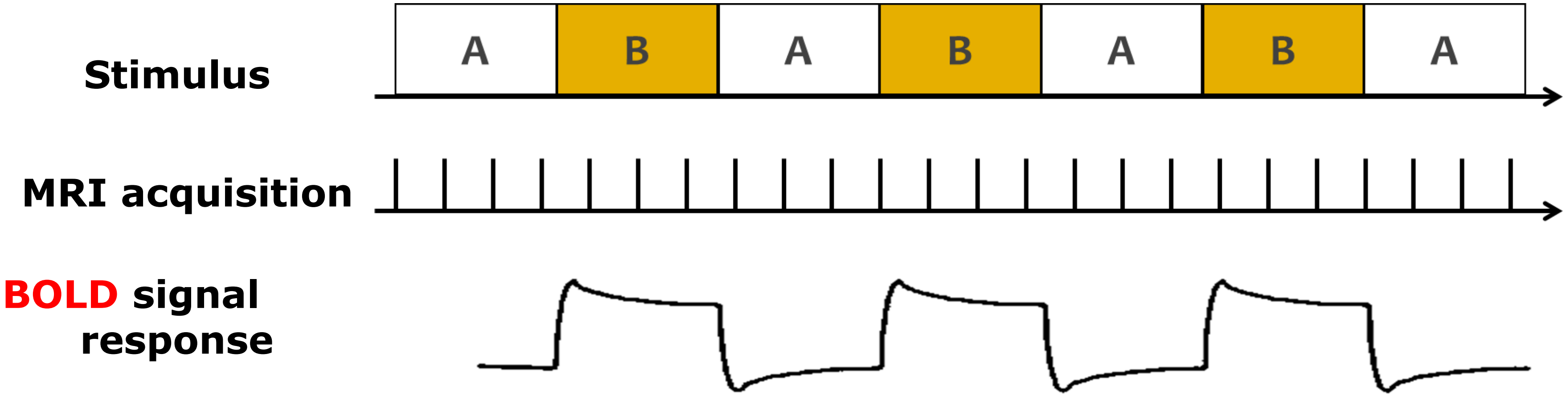
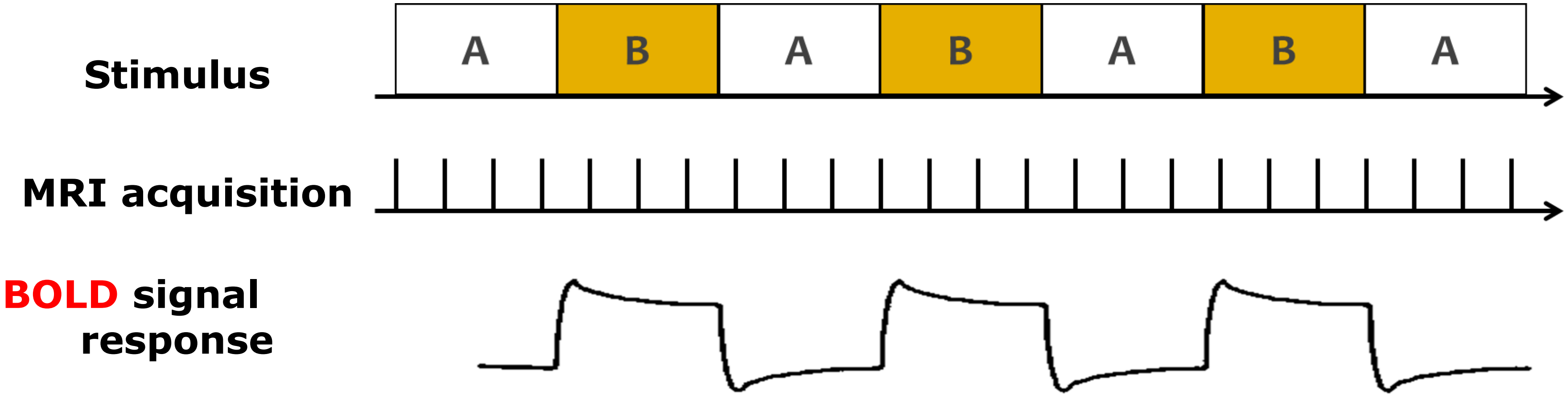
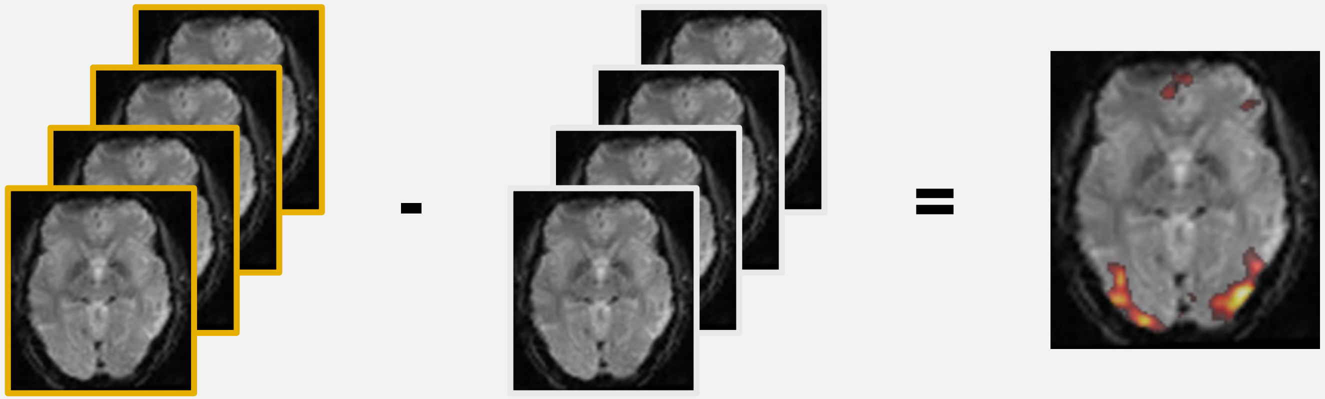


Diagram adapted from Glover, 2011

Functional MRI (fMRI)





B state images

A state images

Contrast/**B activation** map

Diagram adapted from Glover, 2011

Blood **O**xygen **L**evel-**D**ependent (BOLD) signal

dampens the MRI signal

Oxygenated blood cells
$HbO2$

De-oxygenated blood cells
Hb

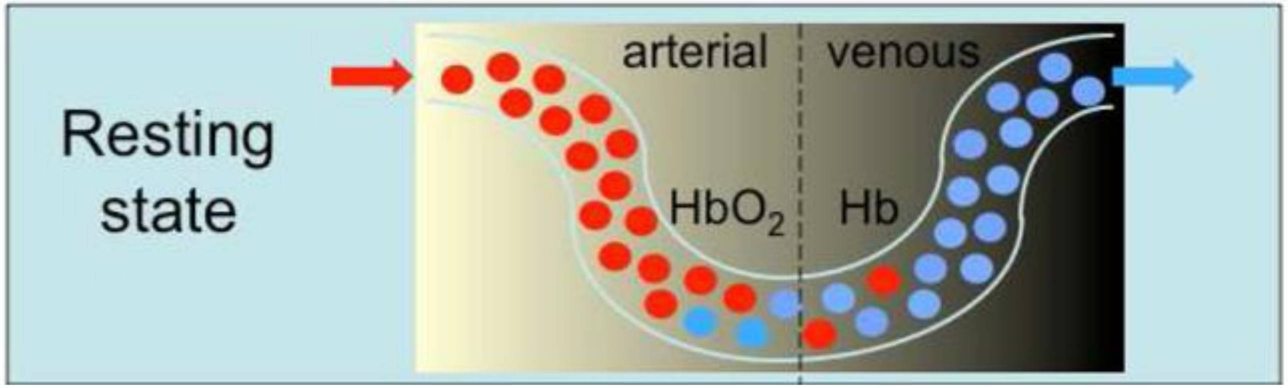


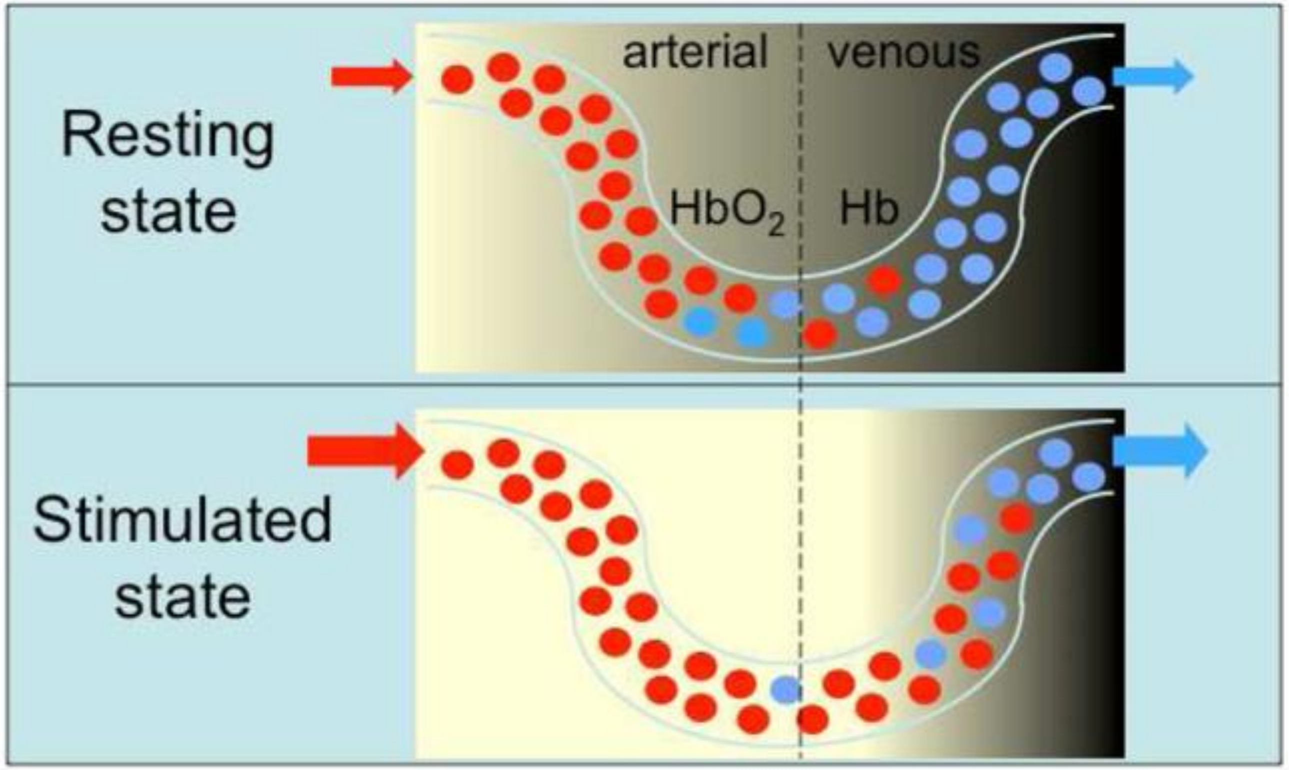
Diagram adapted from Glover, 2011

Blood **O**xygen **L**evel-**D**ependent (BOLD) signal

dampens the MRI signal

Oxygenated blood cells
HbO2

De-oxygenated blood cells
Hb



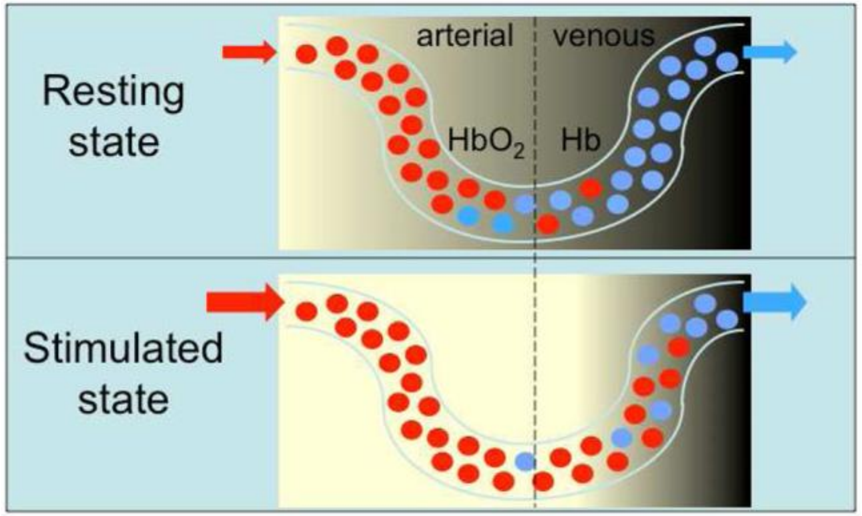
Neural activity-induced increase in blood flow **sweeps the "de-ox" away**, causing an MRI **signal increase**

Diagram adapted from Glover, 2011

Blood **O**xygen **L**evel-**D**ependent (BOLD) signal

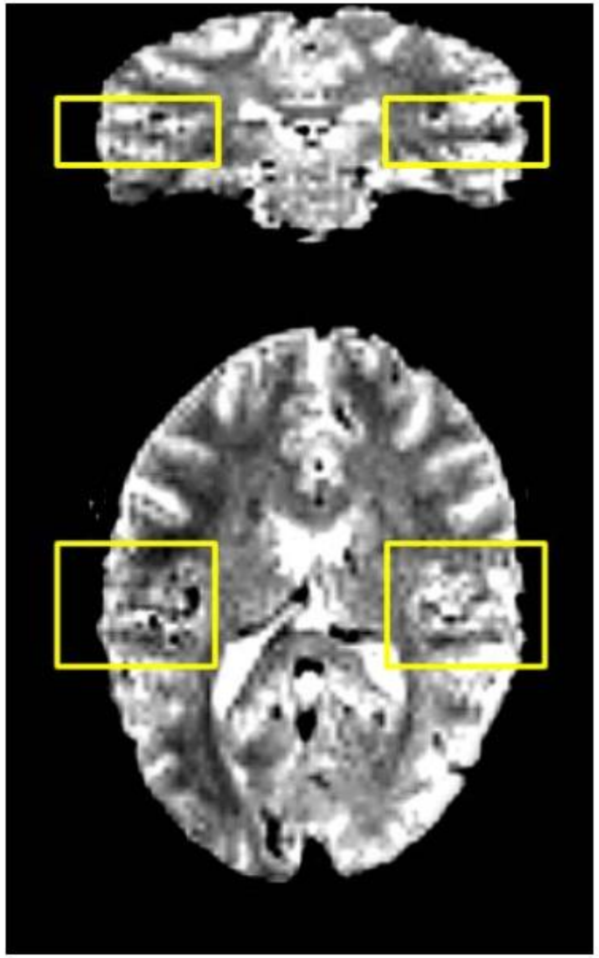
- This difference in the magnetic properties of de-oxygenated and oxygenated Hb is used in BOLD fMRI to create BOLD contrast in images – reflecting activity in different brain regions

- By controlling for all other factors, any observed **differences in the BOLD signal** are inferred to be due to **differences in neuronal activity**

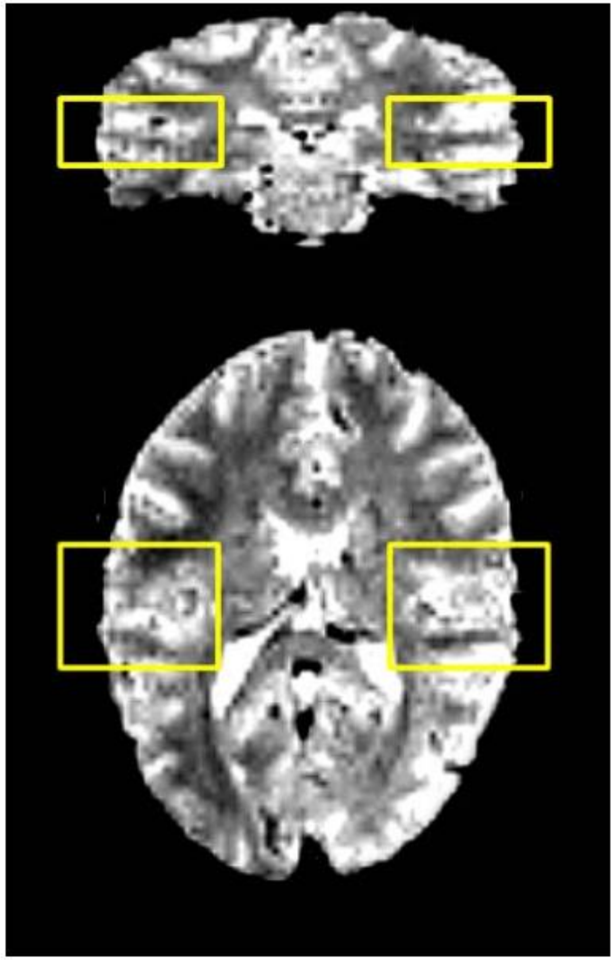


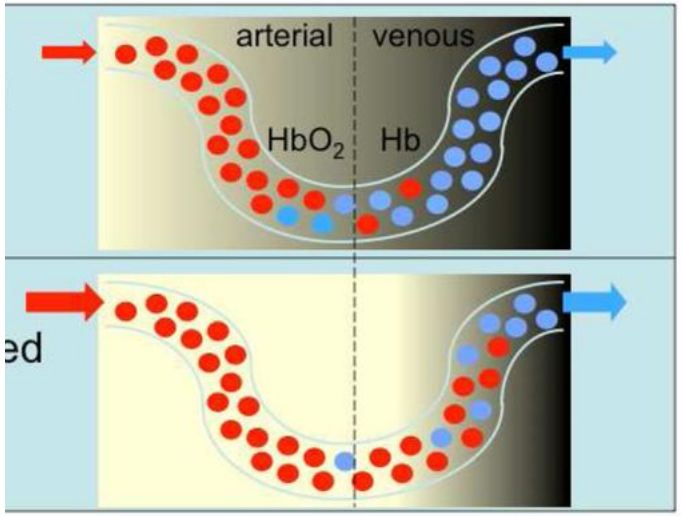
Blood **O**xygen **L**evel-**D**ependent (BOLD) signal

Baseline



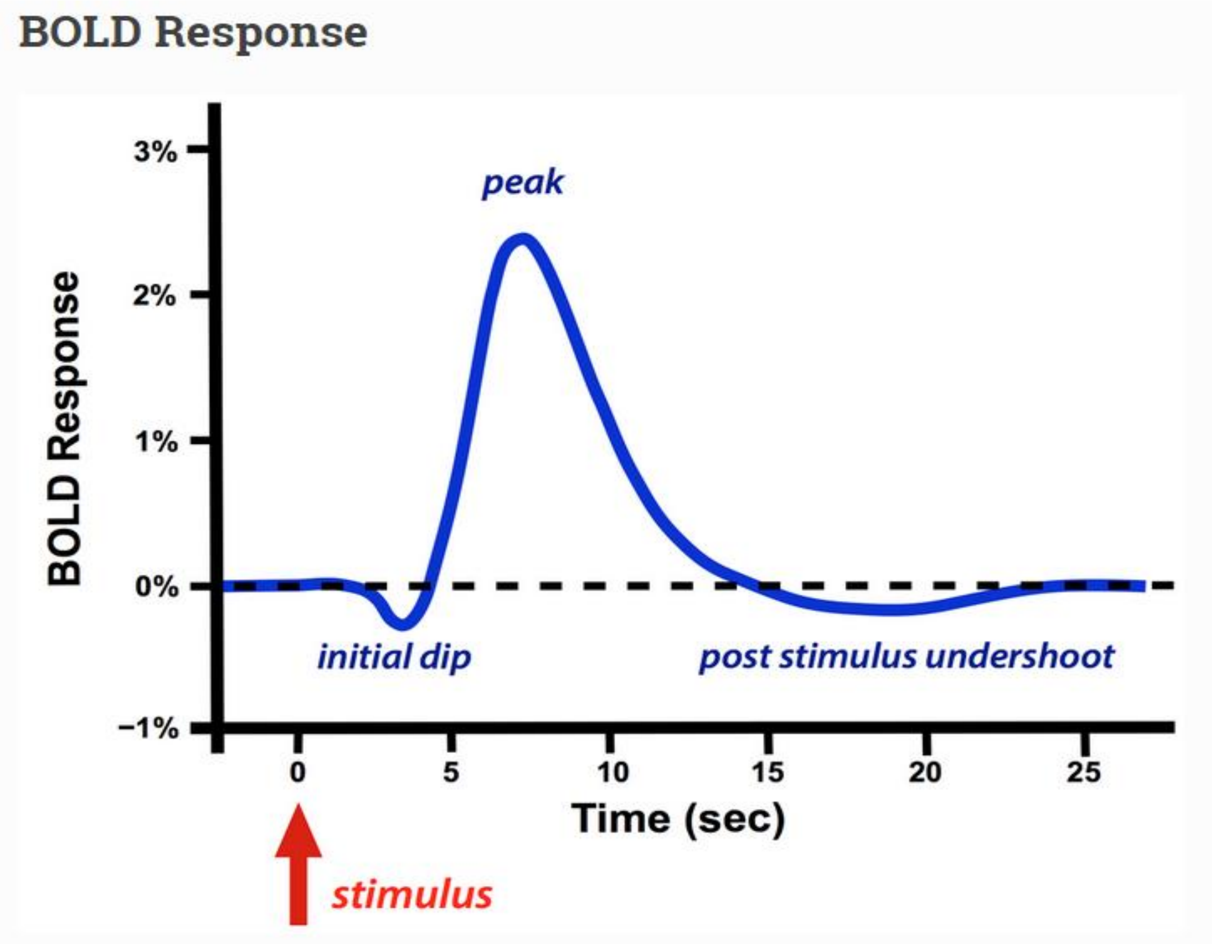
Neural Activity





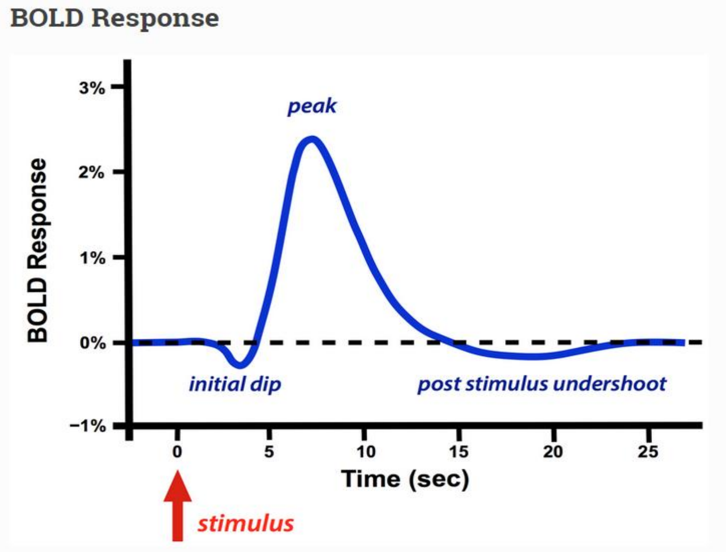
BOLD response

Hemodynamic response function (HRF)



©Andy Jahn

Hemodynamic response function (HRF)



- Depends on stimulus intensity and duration
- Varies across individuals
- Varies with healthy ageing and development
- Varies with common stimulants such as caffeine
- Varies across the brain, both at a distant and local scale

- The most common solution to HRF variability is to pretend it doesn't exist and use a generic model for all participants

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

Received: 07 April 2014

Accepted: 05 January 2015

Published: 20 January 2015

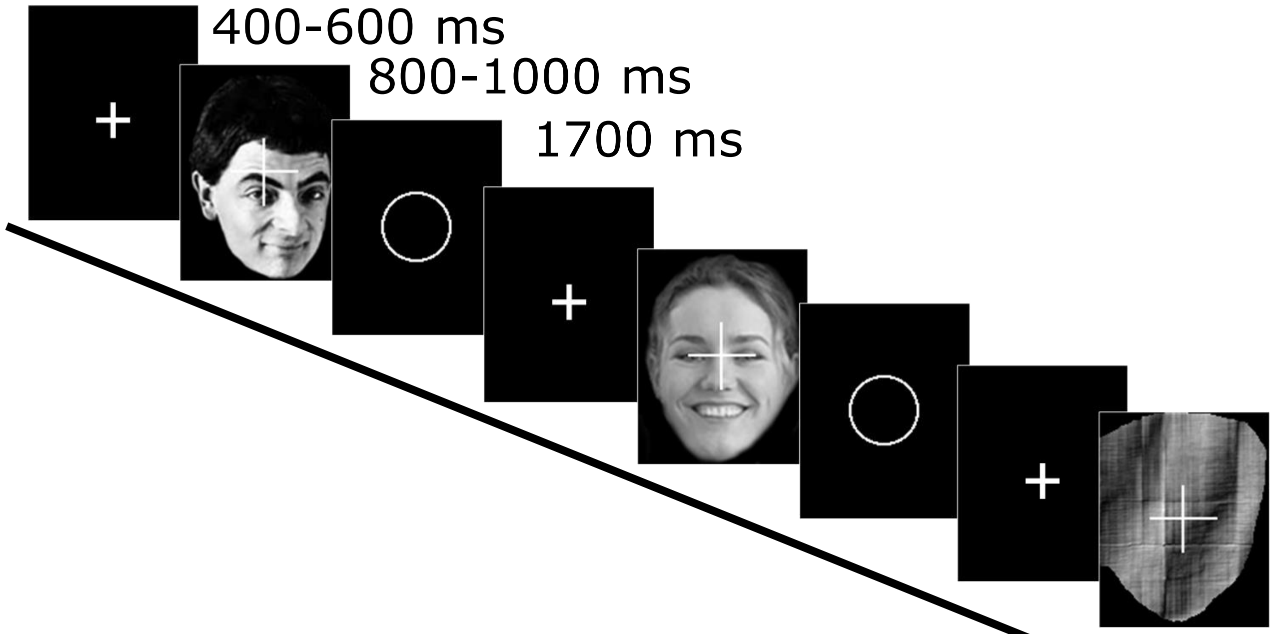
A multi-subject, multi-modal human neuroimaging dataset

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/.

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:

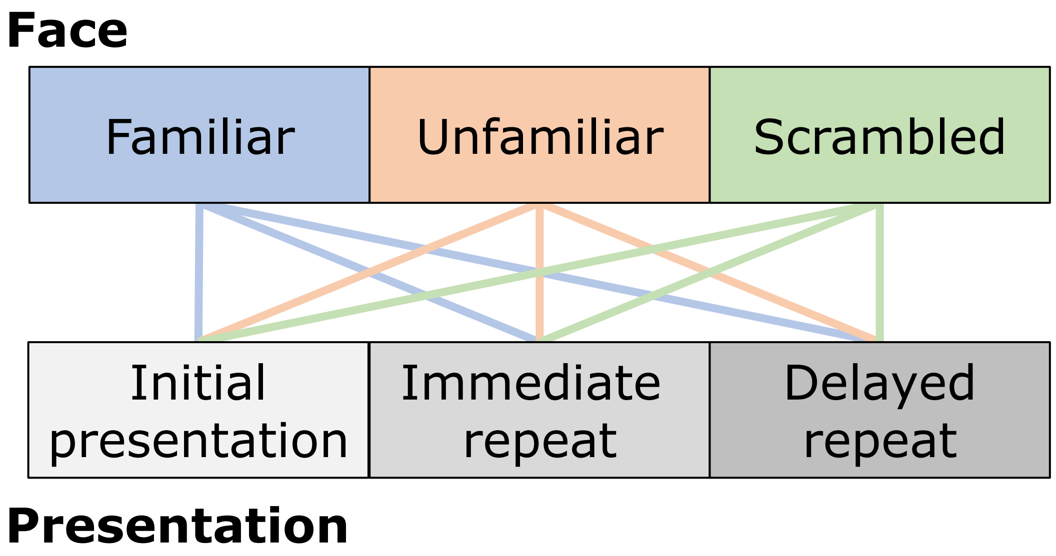
- ~150 x Familiar
- ~150 x Unfamiliar
- ~150 x Scrambled

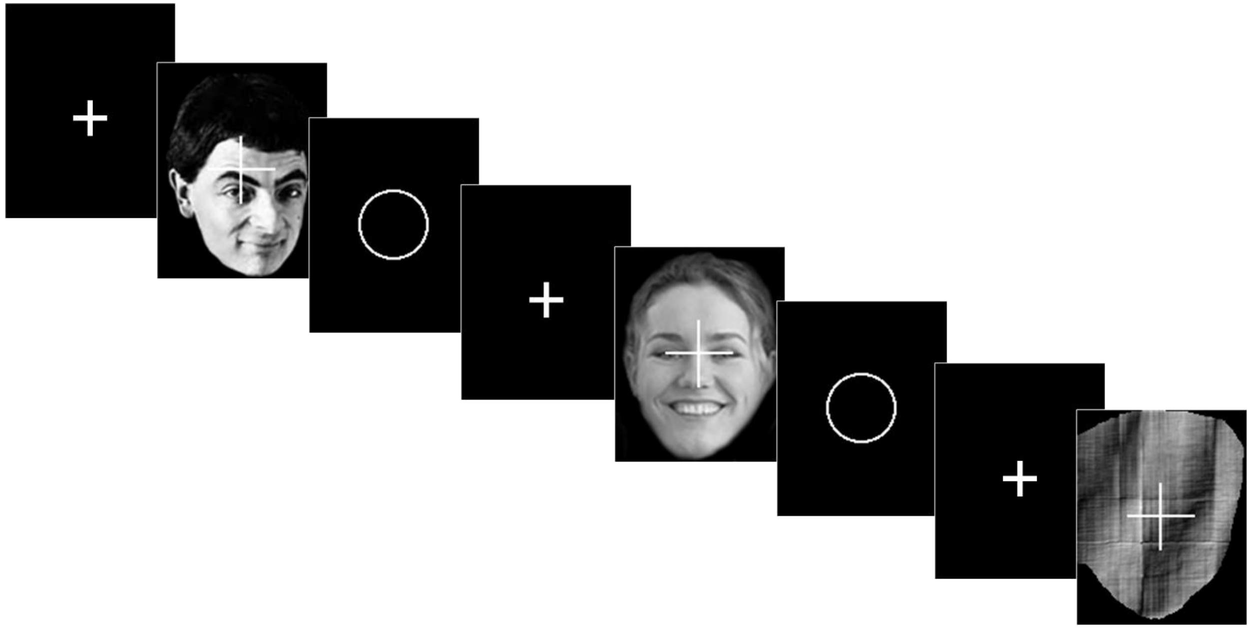
Task: Judge face symmetry

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.

Example Experiment: Face Recognition

- Nine Conditions (3 x 3)



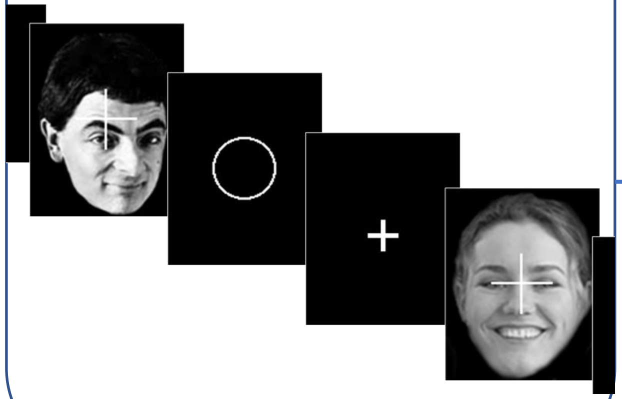


Possible questions to investigate

- Brain areas for Faces
- Brain areas for Face Familiarity
- Response to Initial vs Repeated presentations
- Response to the Repetition of Familiar vs Repetition of Unfamiliar
- ...

An fMRI study

Design an experiment

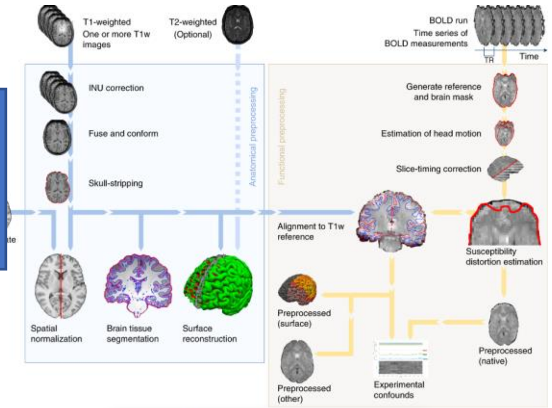


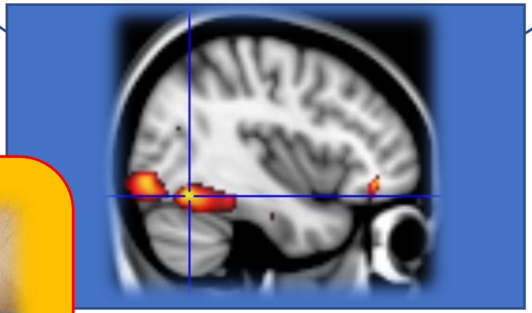
Collect the MRI data

Data

Anatomical image
Functional images
Event details

Pre-process & Analyse





An fMRI study

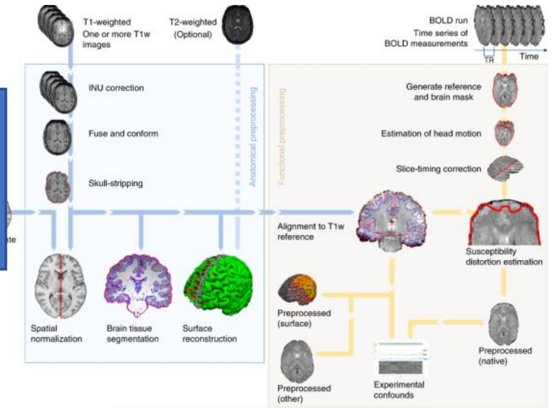
Design an experiment

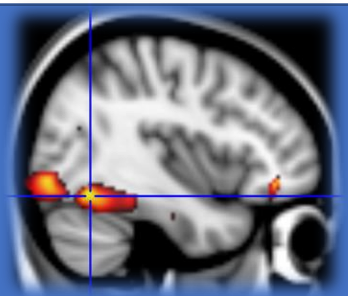
Collect the MRI data

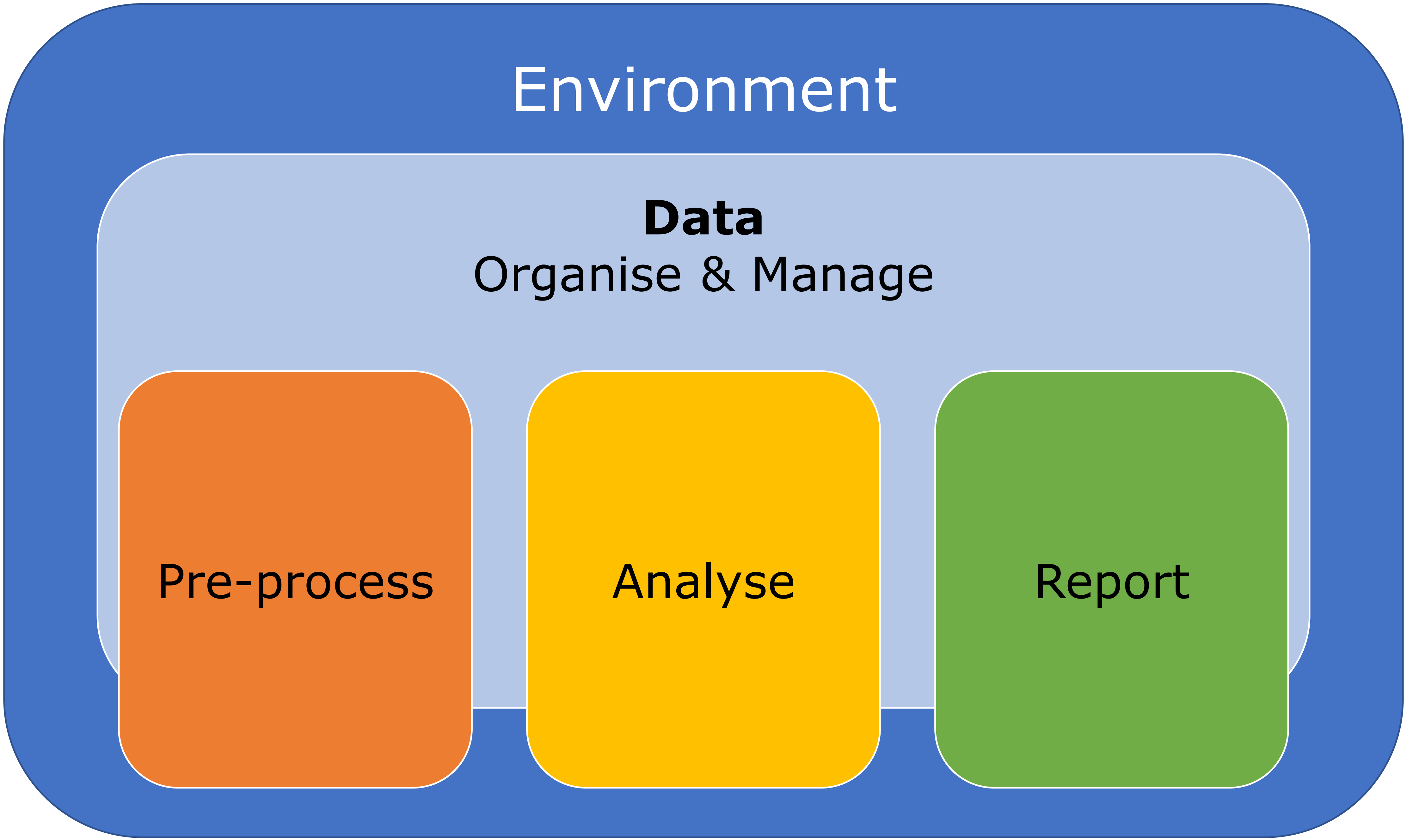
Data

Anatomical image
Functional images
Event details

Pre-process & Analyse






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

MING LANGUAGES

BASH
&
Shell Scripts

ython™

MATLAB®

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.

```yaml
name: mri
channels:
  - conda-forge
  - defaults
dependencies:
  - dcm2niix=1.0.20250506
  - heudiconv=1.3.3
  - pip=24.2
  - pytest=8.3.2
  - python=3.11.10 # dipy v1.9.0 dependency >=3.9, <3.12
  - 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
      - dcmstack==0.9
      - fury==0.11.0 # dipy v1.9.0 visualisation didn't work with fury 0.10.
      - ipykernel==6.29.3
      - ipython==8.22.1
      - jupyter==1.0.0
      - matplotlib==3.8.3
      - nibabel==5.3.1
      - nilearn==0.12.0
      - nipype==1.10.0
      - nipy==0.6.0
      - numpy==1.26.4 # dipy v1.9.0 dependency >=1.21.6, <1.27.0
      - nxviz==0.7.4
      - pandas==2.2.2
      - plotly==5.23.0
      - pybids==0.19.0
      - python-louvain==0.16
      - requests==2.31.0
      - rsatoolbox==0.1.5
      - scikit-image==0.24.0
```

ironment

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.

Environment

PROGRAMMING LANGUAGES

BASH
&
Shell Scripts

python™

PACKAGE MANAGER

CONDA

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

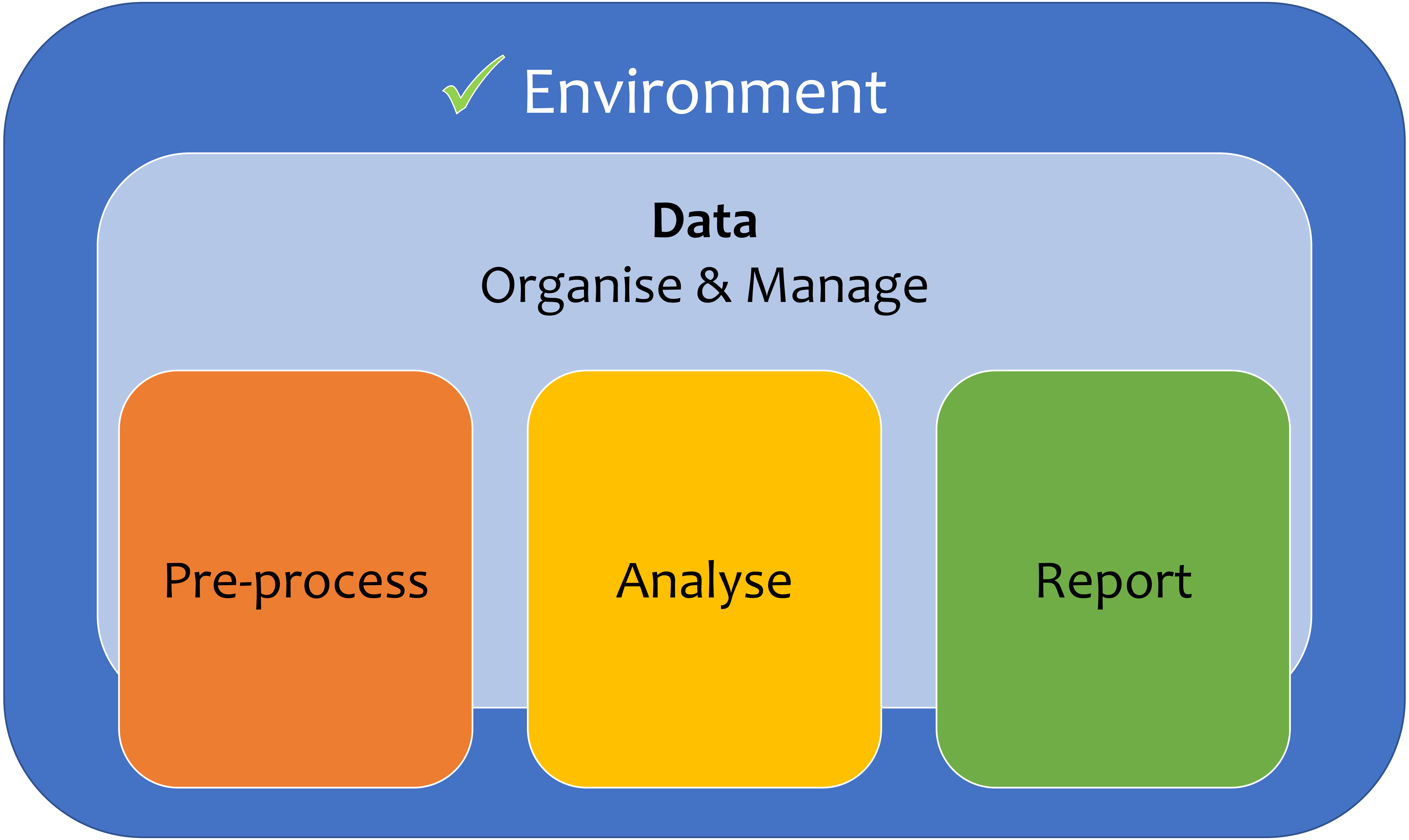
With conda, you can use environments that have different versions of Python and packages installed in them.

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
Data
Organise & Manage
Pre-process
Analyse
Report

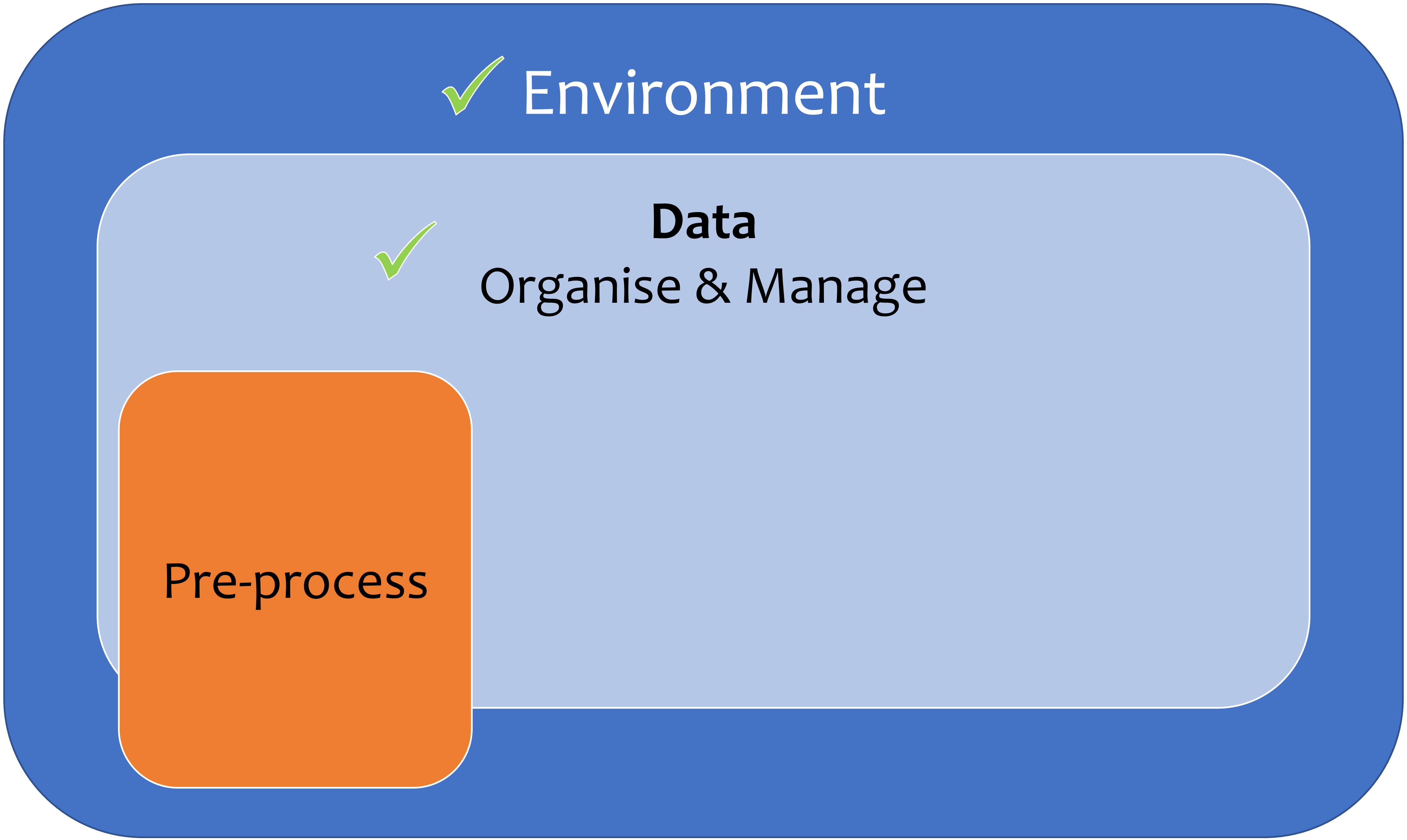
BIDS
BRAIN IMAGING DATA STRUCTURE

Raw DICOM format

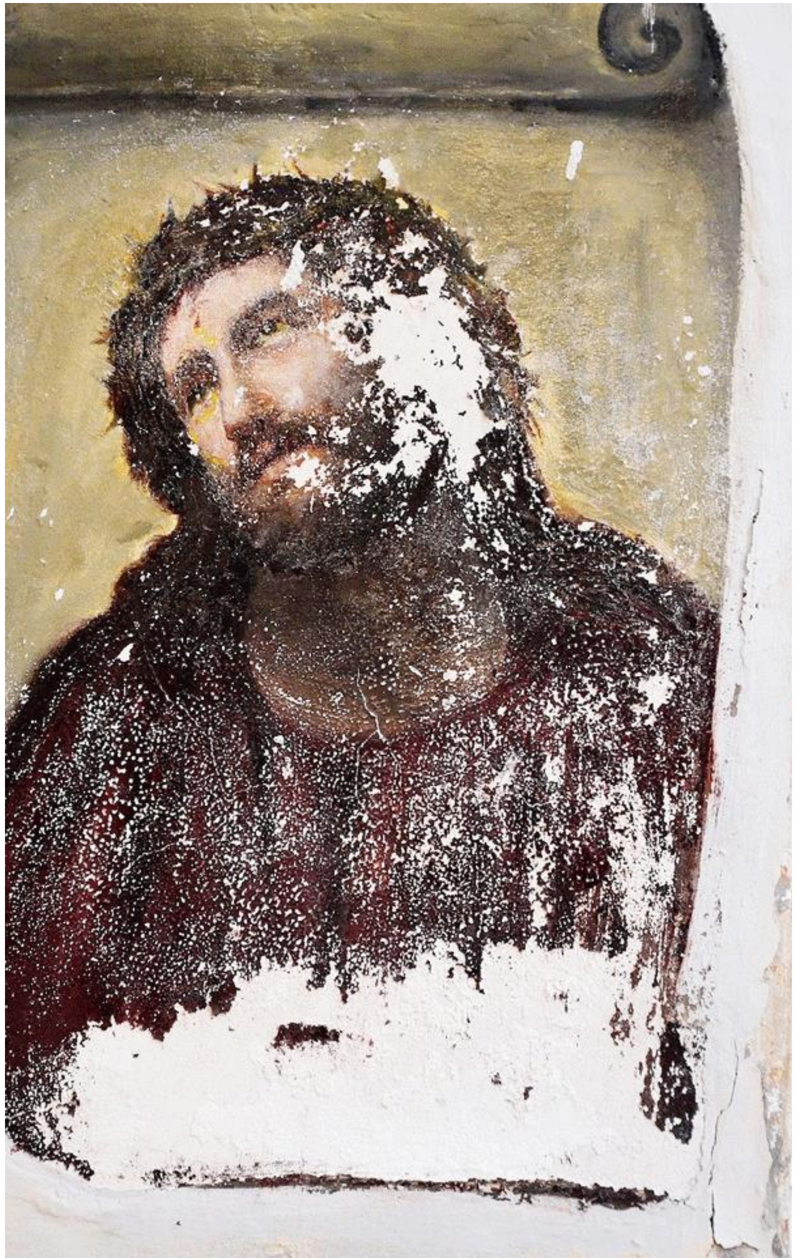
```
mridata/
└── CBU090962_MR09029
    └── 20090902_100102
        ├── Series_001_CBU_Localiser
        ├── Series_002_CBU_MPRAGE
        ├── Series_003_CBU_DWEPI_BOLD210
        ├── Series_004_CBU_DWEPI_BOLD210
        ├── Series_005_CBU_DWEPI_BOLD210
        ├── Series_006_CBU_DWEPI_BOLD210
        ├── 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

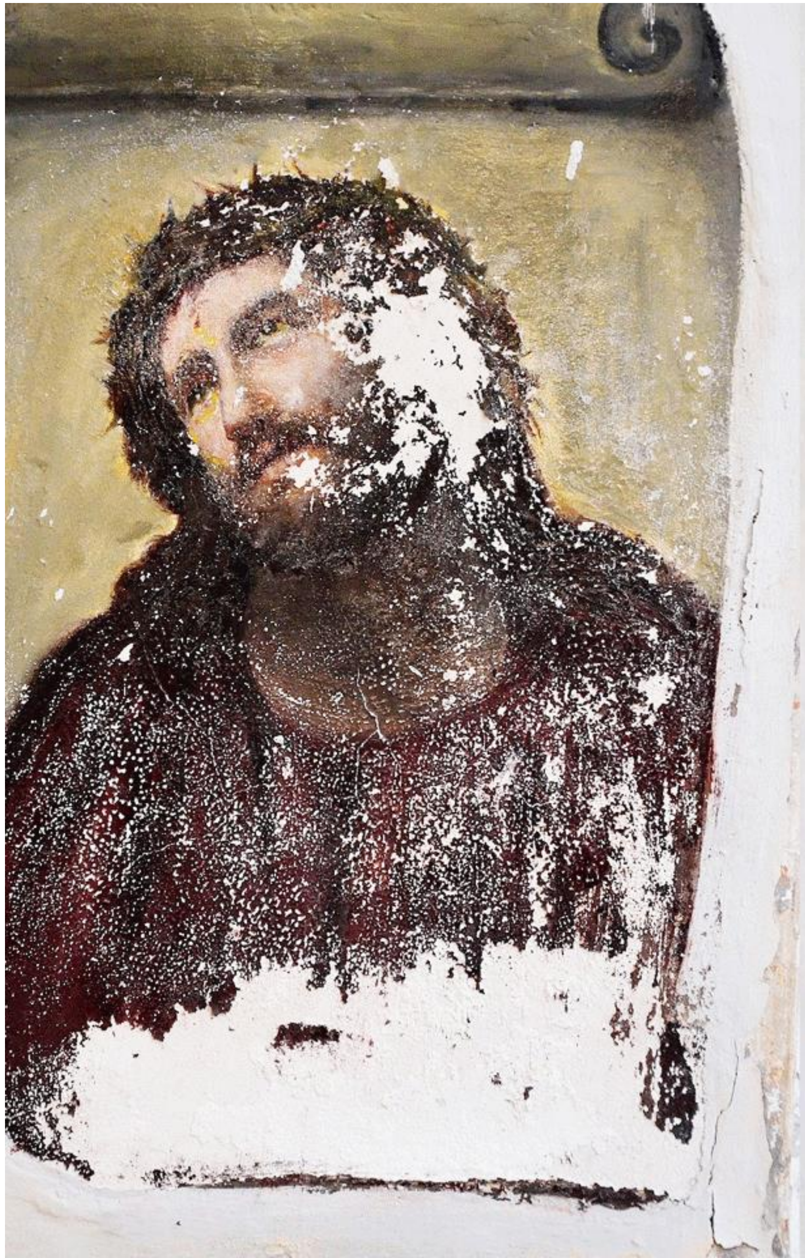
```
├── sub-15
│   └── ses-mri
│       ├── anat
│       │   ├── sub-15_ses-mri_T1w.json
│       │   └── sub-15_ses-mri_T1w.nii.gz
│       ├── fmap
│       │   ├── sub-15_ses-mri_acq-func_magnitude1.json
│       │   ├── sub-15_ses-mri_acq-func_magnitude1.nii.gz
│       │   ├── sub-15_ses-mri_acq-func_magnitude2.json
│       │   ├── 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
```


✓ Environment
✓ Data
Organise & Manage
Pre-process

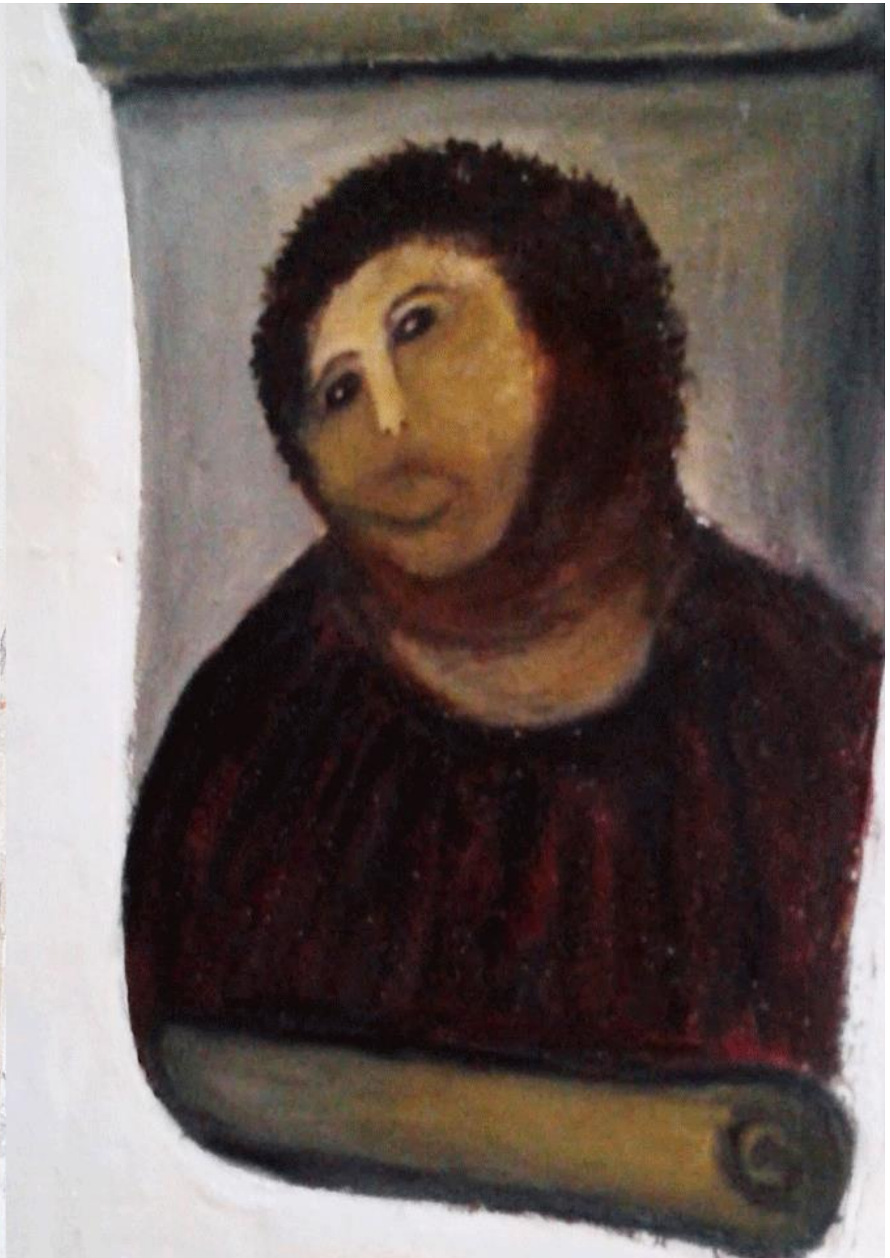
raw



raw

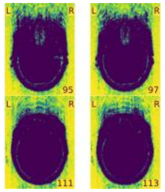


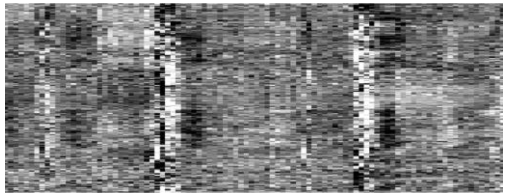
pre-processed



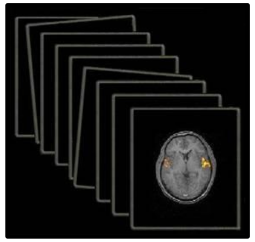
Typical fMRI pre-processing pipeline

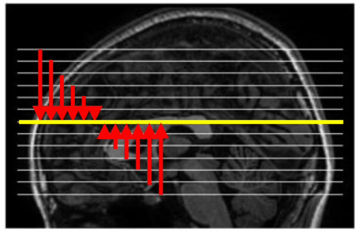
Data quality assessment

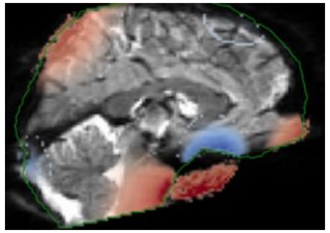


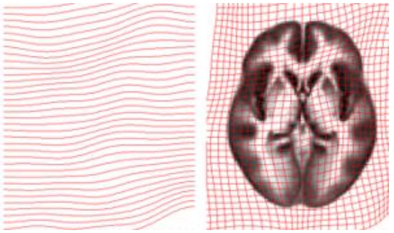


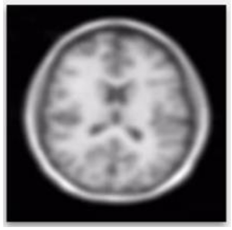
Motion-correction → Slice-time correction → Distortion correction → Normalisation to MNI space → Spatial smoothing



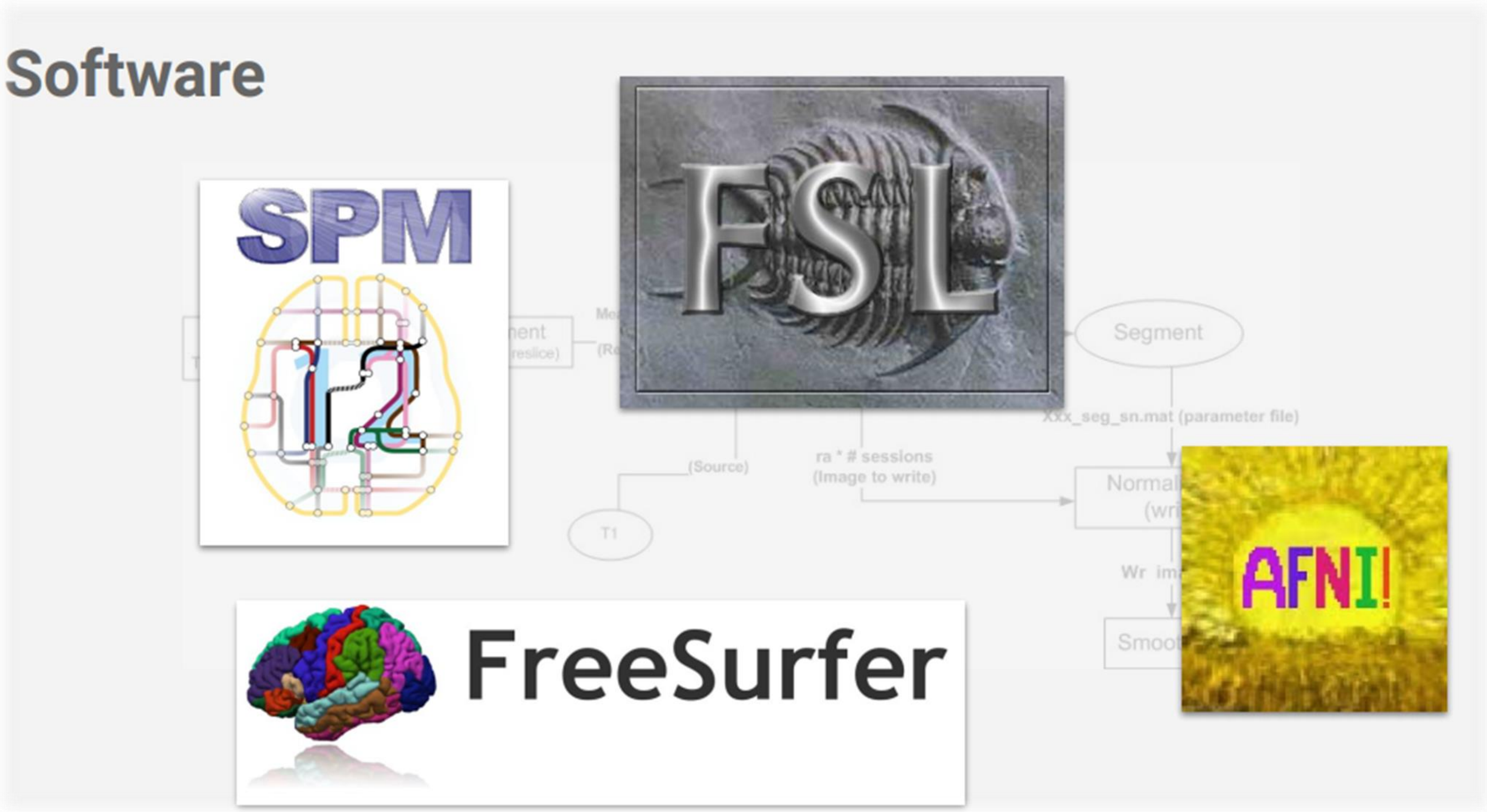






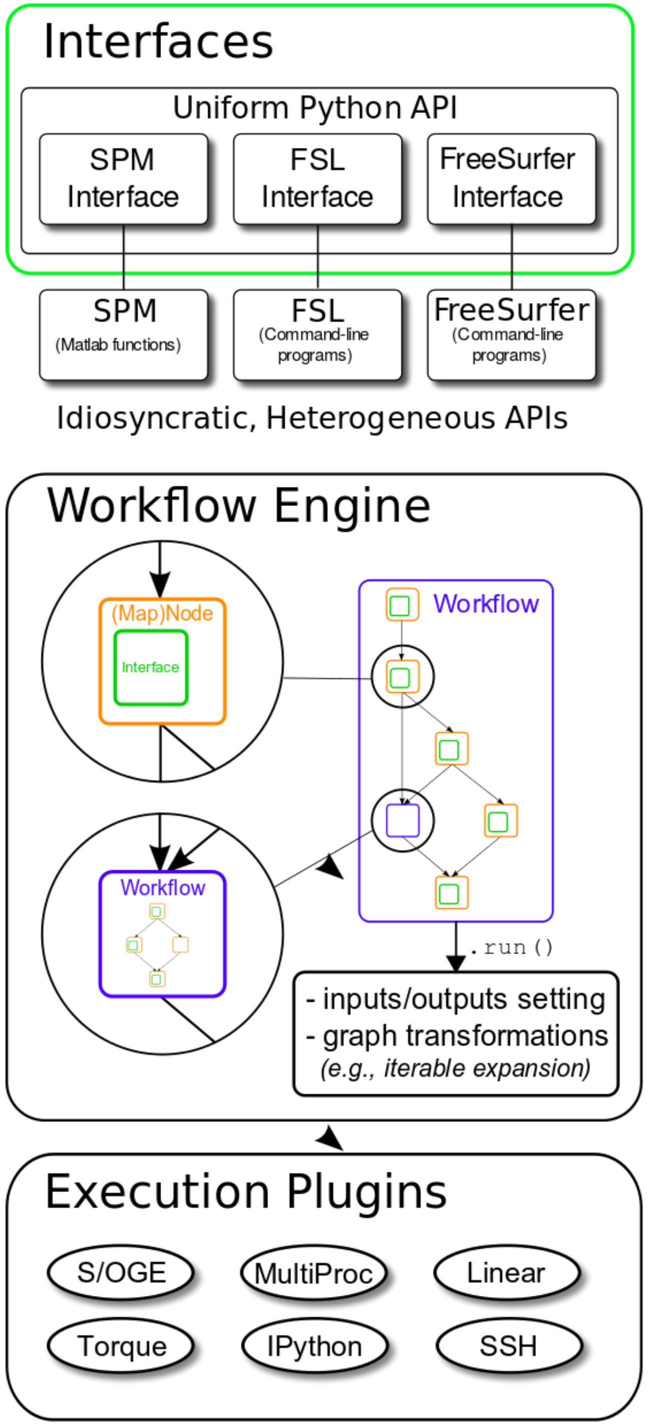


Pre-processing tools



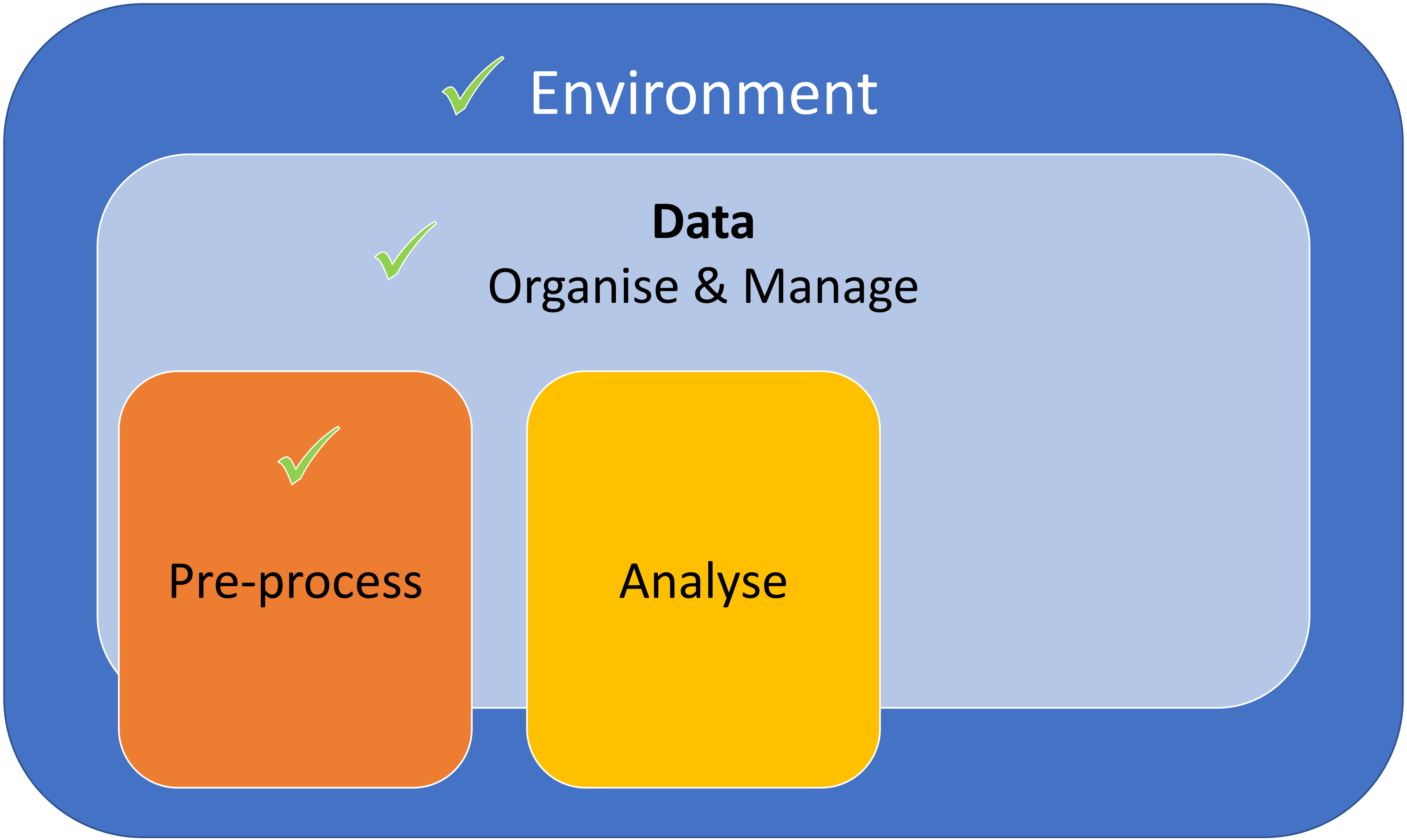
Pre-processing tools

Nipype:
Neuroimaging in Python
Pipelines and Interfaces



Pre-processing tools

- **fMRIPrep** https://fmriprep.org/en/stable/
  - Fully automated fMRI data pre-processing tool
  - The workflow is based on Nipype and encompasses a large set of tools from well-known neuroimaging packages, including FSL, ANTs, FreeSurfer, AFNI, and Nilearn. This pipeline is designed to provide the best software implementation for each state of pre-processing.
  - **Robustness** - The pipeline adapts the pre-processing steps depending on the input dataset and should provide results as good as possible independently of scanner make, scanning parameters or presence of additional correction scans (such as fieldmaps).
  - **Ease of use** - Thanks to dependence on the BIDS standard, manual parameter input is reduced to a minimum, allowing the pipeline to run in an automatic fashion.
  - **"Glass box" philosophy** - Automation should not mean that one should not visually inspect the results or understand the methods. Thus, fMRIPrep provides visual reports for each subject, detailing the accuracy of the most important processing steps.


Environment
Data
Organise & Manage
Pre-process
Analyse

Finding a Face area in the brain

Which brain regions are engaged when people look at faces

- With fMRI the meaningful questions are questions that **compare two conditions**
  - We need some sort of control condition –> **Scrambled** condition

- **Which brain regions respond more to looking at face images than scrambled images**
  - The control question hopefully helps to **wash out all the regions we are not interested in**. Because regions that we are NOT interested in should activate both conditions to the same extend (e.g. visual areas)

GLM: $Y = X\beta + \varepsilon$

BOLD signal	=	X * b	+	errors
		explained variation		unexplained variation
		task-related activity changes		noise (other changes)

- **What we know?**
  - BOLD signal: we collect this from the brain (functional data)
  - X: the design matrix (each column is a predictor that we build ourselves)
- **What we want to find?**
  - b: vector of beta-weights (one weight for predictor in X) that give the best approximation of the BOLD signal
- **How we find it?**
  - By minimising the sum of squared errors. In practice, the **GLM has a formula, which guarantees to find these beta-weights**

1. **Extract the signal time-series** from a given voxel

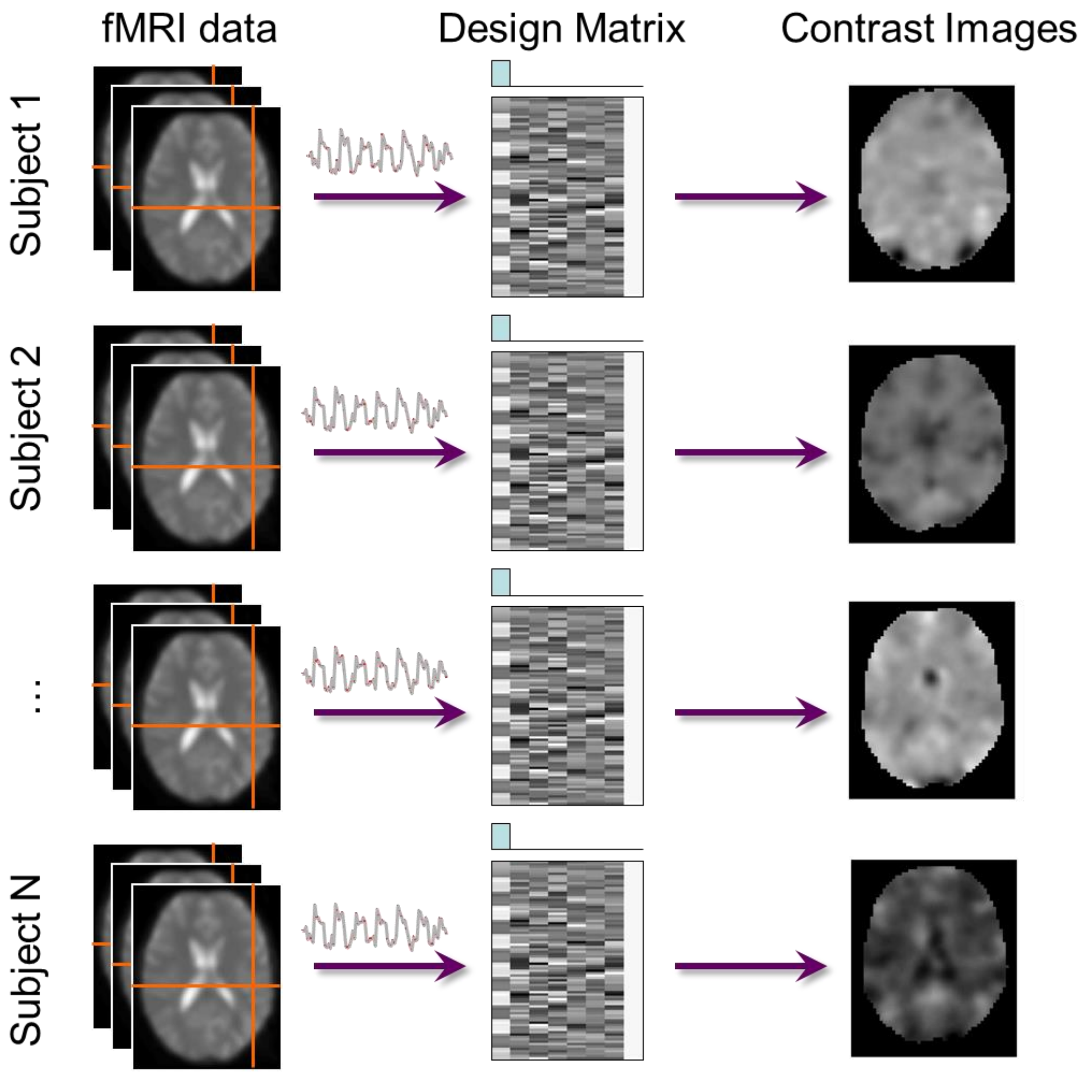
2. **Run GLM** (the signal and the design matrix are the inputs) to **find beta-weights** that best approximate the true signal

3. Define your **contrast** and test it

4. Repeat for **all voxels**
   - Produces an image file with contrast values for each voxel: **contrast-maps**

First-level analysis

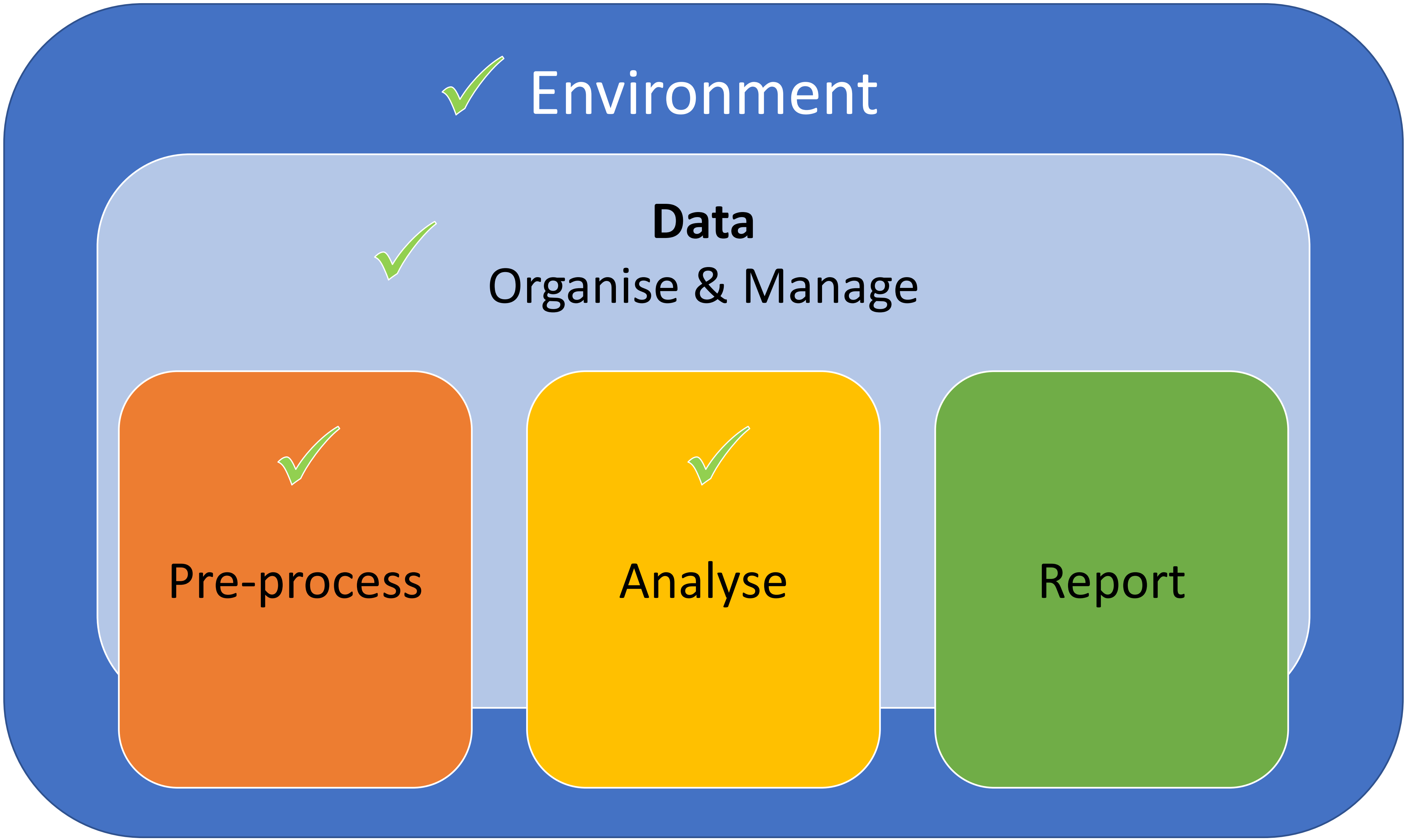
- Run the GLM for each subject



Group level (2nd level) analysis is across subjects

- Which voxels are showing significant activation differences between our conditions consistently **within a group**

- Importantly, all subject brains need to be in a common space, e.g. MNI, to perform voxel-wise group analyses


Environment
Data
Organise & Manage
Pre-process
Analyse
Report

Sharing & Reporting

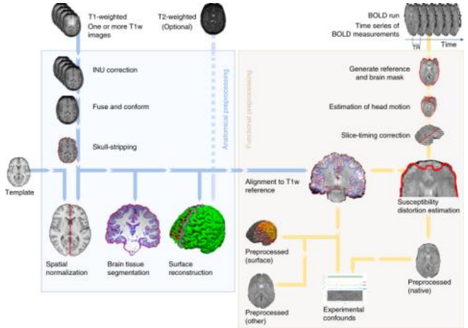
- Share your **code** and notebooks on GitHub
- Make it **citable** with Zendono
  - https://docs.github.com/en/repositories/archiving-a-github-repository/referencing-and-citing-content
- If you have consent from participants, share the **BIDS data** on OpenNeuro.
- Add your **contrast maps** to NeuroVault

The Plan

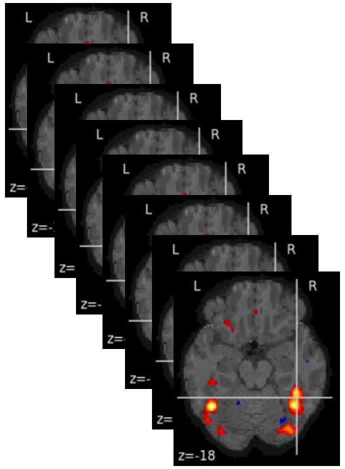
1
Data Organisation

2
MRI Data Manipulation

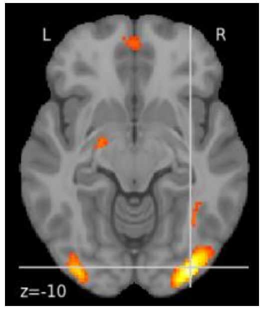
3
Quality Control & Pre-processing



4
Subject-Level Analysis



5
Group-Level Analysis



6
ROI Analysis