

---

# **SOTNIR handbook**

***Release 0.1.0***

**Huneke NTM, Varatharaj A, Zou K**

**Dec 16, 2023**



## GETTING STARTED

<b>1</b>	<b>Linux Workstations</b>	<b>3</b>
<b>2</b>	<b>Unix Basics</b>	<b>9</b>
<b>3</b>	<b>Introduction</b>	<b>11</b>
<b>4</b>	<b>Cloud Computing</b>	<b>19</b>
<b>5</b>	<b>Convert from DICOM to NIfTI</b>	<b>23</b>
<b>6</b>	<b>Pseudonymise your dataset</b>	<b>31</b>
<b>7</b>	<b>Brain Imaging Data Structure (BIDS)</b>	<b>33</b>
<b>8</b>	<b>DataLad: Distributed Data Management</b>	<b>35</b>
<b>9</b>	<b>fMRI Analysis</b>	<b>43</b>
<b>10</b>	<b>dMRI Analysis</b>	<b>69</b>
<b>11</b>	<b>QSM Analysis</b>	<b>73</b>



This handbook is a resource for those carrying out neuroimaging projects.

This resource contains quickstart guides, example scripts, and standardised workflows, developed in Southampton.

Contributors are credited on each page.

---

**Important:** This site is under construction and active development. Contents of this site are subjected to change. *You can be a part of this project as well!* If you would like to contribute to this handbook, please contact Nathan ([N.Huneke@soton.ac.uk](mailto:N.Huneke@soton.ac.uk)), Ara ([A.Varatharaj@soton.ac.uk](mailto:A.Varatharaj@soton.ac.uk)), or Kai ([Y.Zou@soton.ac.uk](mailto:Y.Zou@soton.ac.uk))

---



## LINUX WORKSTATIONS

Contributors: Nathan TM Huneke, Harry Fagan, Nick Hedger, Yukai Zou  
Maintainers: Nathan TM Huneke

---

### 1.1 Why Linux?

Linux is a lightweight (small size) platform which consumes not as much memory as a Windows OS. This feature is particularly useful when it comes to computing-intensive environment, where a lot of memory and storage spaces are in need. For neuroimaging analysis, it is almost essential that you use a UNIX - based operating system (Mac or Linux), since popular software packages such as FSL, AFNI and FreeSurfer do not run on a Windows operating system.

A number of workstations at the University of Southampton run a *flavour* of Linux operating system known as Red Hat Enterprise Linux (RHEL) 7. Some of these have been configured for neuroimaging analysis. Current machines that can be used for this purpose include:

- uos-15263
- uos-210028
- uos-211997

---

**Note:** For more information on the University's RHEL7 machines, see <http://linuxdesktops.soton.ac.uk/>

---

### 1.2 Setting Up the Machine

To use the above machines effectively, you need to go through a number of initial set up steps.

### 1.2.1 0. Grant Access

First of all, your access to the specific University linux workstation needs to be granted. This is typically done by the admin (i.e. you would need to find out who that person is) of that workstation, who will raise an iSolution ticket on behalf of you.

---

**Note:** Please provide your UoS username to the admin.

---

### 1.2.2 1. Log in

These machines can be accessed remotely via remote desktop apps or SSH.

#### Remote Desktop Apps

For MacOS, you can use “Microsoft Remote Desktop”. If you already have it installed for connecting to “Win 10 Student” at the University, then this would come handy. For Windows, you can use “Remote Desktop Connection”, which is automatically installed on University machines running Windows. You should be able to download and install either app on your personal machines.

---

**Note:** If you are off site when logging in then you will need to connect to the University’s [VPN](#) first.

---

1. On your machine, click on *Start* and search for Remote Desktop Connection
2. Type in the computer name as above followed by `.clients.soton.ac.uk` (e.g. `uos-210028.clients.soton.ac.uk`) and click connect
3. Input your University of Southampton username and password when prompted

Alternatively, if you would like to connect via remote desktop app on Linux, you can use Remmina by following an instruction [here](#).

#### SSH

To connect remotely via SSH, launch a Terminal window, and enter the following command:

---

**Note:** If you are off site when logging in then you will need to connect to the University’s [VPN](#) first.

---

`ssh username@uos-#.clients.soton.ac.uk` - Replace `username` with your University of Southampton username, and `#` with the specific workstation ID.

Please enter your University of Southampton password when prompted, and then hit Return.

To display application GUI’s from the remote machine on your computer (e.g. the GUI for FSL or Matlab) you will need to connect with X-forwarding enabled by using the `-X` flag (or on Mac, `-Y`):

```
ssh -X username@uos-#.clients.soton.ac.uk
```

This is particularly useful when using Matlab remotely. Click [here](#) for more information on common problems with X and how to solve them.



**Tip: What if I am using a Windows machine?**

You can install a terminal emulator and use the commands as above. For example, [MobaXterm](#) has all the features needed to achieve X-forwarding on a Windows machine. Alternatively, you can also use [Git BASH](#), a BASH emulation used to run Git from the command line, which is already installed on all University Windows machines.

**Tip: I can't connect via SSH and I got a warning message: "REMOTE HOST IDENTIFICATION HAS CHANGED!" What should I do?**

The issue here is most probably due to a new (randomised) host ID has been generated, which is usually a result of system upgrade.

*Why is it causing a warning?*

ssh is warning you that this host ID is different to what it previously saw when you connected to the computer, since it has no way to know whether (as in this case) it is a legitimate change made by the machine owner, or whether someone has maliciously tampered and may be trying to steal information by getting you to connect to a different computer entirely.

*How to solve?*

There are several ways to work around:

1. If this is the only computer that you use 'ssh' or related commands (e.g. rsync, scp, sftp) to connect to, then you can safely just delete the file where it keeps its log of host key fingerprints:

```
$ rm /h/.ssh/known_hosts
```

then try the ssh command again, confirm you wish to store the host ID, and proceed as usual.

2. Alternatively, if you wish to preserve the old file, you can rename it to a different filename:

```
$ mv /h/.ssh/known_hosts /h/.ssh/known_hosts_old
```

then try the ssh command again. In this case, a new known\_hosts file will be generated.

3. Alternatively, if you use ssh for other purposes as well (e.g. connect to other computers) and want to keep their host key fingerprints intact, then you will need to delete just the relevant lines from the file.

You could edit the file with `vi` or `nano` if you are comfortable with those editors, or, you could use Wordpad or Notepad++ (but not notepad since it is likely not to 'understand' the line endings) in Windows, finding the file at `h:\.ssh\known_hosts`

## 1.2.3 2. Install Conda

Most of the software you will need has been pre-installed on the machine. However, Conda (a distribution of Python) needs to be installed on a per-user basis.

To install conda, first navigate to your home directory:

```
$ cd ~
```

Then download the installer:

```
$ wget https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

Then run the installer:

```
$ bash Miniconda3-latest-Linux-x86_64.sh
```

Follow the prompts on the installer screens. Use **q** to skip to the end of the license agreement and accept. When asked, save conda in a hidden directory in your home directory to prevent other software interfering with it. We suggest:

```
$ ~/.conda/Miniconda3
```

Once the installer has finished, close and reopen the terminal window. Then type `conda --version` to check it has installed successfully.

## 1.2.4 3. Create a conda environment for your software and analyses

### What is a conda environment?

An environment is a self-contained collection of conda packages. If you change one environment, your other environments and other installed software are not affected. Some software, particularly DataLad, seems to interfere with other software on the Linux machines. It is therefore safest to use this software in a self-contained environment.

Conda environments can be easily activated and deactivated as needed.

### Create your environment

We have created a ready-to-use conda environment that you can clone.

To create your environment, open a terminal. If conda is installed, you should see that you are currently in the `base` environment, which is signified like so:

```
(base) [nh6g15@uos-211997 ~] $
```

Next, install `mamba`:

```
$ conda install -c conda-forge mamba
```

Next, clone the environment description into your home directory with:

```
$ git clone https://github.com/nhuneke/imagingenv ~/imaging-env
```

And now create the environment:

```
$ cd ~/imaging-env  
$ mamba env create --file imaging-env.yml
```

## Activating and deactivating your environment

To activate your environment use the following command:

```
$ conda activate imaging-env
```

You should now see this environment is active in the terminal, like so:

```
(imaging-env) [nh6g15@uos-211997 ~] $
```

You will now be able to use all the software present in this environment.

To deactivate your environment, use the following command:

```
$ conda deactivate
```

### 1.2.5 4. Set up FSL

If this is your first time logging in on the Linux machine then you will need to set up your shell environment to use FSL. Each user's shell setup is stored in a file called `.bash_profile`. Open this file in a text editor:

```
$ gedit ~/.bash_profile
```

At the end of the file copy and paste the following lines:

```
FSLDIR=/usr/local/fsl      # NOTE: This is default; modify this line to match your local_
↪path
. ${FSLDIR}/etc/fslconf/fsl.sh
PATH=${FSLDIR}/bin:${PATH}
export FSLDIR PATH
```

Then logout and log back in. FSL will now be ready for use.

## 1.3 Using Space Effectively

The Linux machines as set up by the University are made up of three broad partitions:

- The root partition /
- The home partition `/home/`, or `~` for your personal home directory
- The local partition `/local/`

The root and home partitions are very small. You will not have access to the root partition anyway as changing things here can be dangerous. Only administrators can access this and make changes. But your home directory is yours. However, this partition is only 50GB in size **in total, for all users!**.

For this reason, you should not store neuroimaging data here, as a single dataset can easily reach more than 50GB in size. Instead, you should use the `scratch` space which is in the local partition. Each user has their own scratch space at:

```
/local/scratch/<user id>
```

Replace `user id` with your username.

**Danger: The scratch space is not backed up!**

The `scratch` space is for current work, like a drafting space. It is not backed up. You should regularly make backups of your work here and once the project is finished store the data in a permanent store, such as the university research filestore. DataLad can help with this (add ref).

## UNIX BASICS

Contributors: Nathan TM Huneke, Harry Fagan, Yukai Zou  
Maintainers: Nathan TM Huneke

---

### 2.1 Unix

- Unix is an operating system (like Windows).
- It uses a command line interface (or command line) where you type commands you want to run into a terminal.
- Command line interpreters are called shells (two commonly used ones are bash [Bourne-again shell] and zsh [Z-shell])
- Most neuroimaging packages need to be run from a command line
- MacOS and Linux have Terminals installed. To use Unix commands on Windows, you would need to install a Terminal emulator, or to enable **Windows Subsystem for Linux (WSL)** <<https://docs.microsoft.com/en-us/windows/wsl/install>> feature.

---

**Note:** You can practice using Unix online with JupyterLab (see <https://neuroimaging-core-docs.readthedocs.io/en/latest/pages/unix.html> for an excellent tutorial).

An excellent tutorial on the use of the Unix Shell is available [here](#).

---

### 2.2 Basic Unix Commands

- `pwd` Print Working Directory, tells you the current directory you are in
- `cd directory` Change Directory, takes to you the directory specified
- `cd ~` Go to home directory
- `cd /` Go to root directory
- `ls` List, lists the contents of the directory you are in
- `ls <directory>` lists contents of the specified directory
- `ls -a` lists all files and directories in the directory you are in (including hidden ones which start with an “.” and don’t appear is you just you the “ls” command)

- `ls -l` Long list, lists all files and directory with ownership and user permissions
- `ls -al` Long list (including hidden files and directories)
- `mkdir <name>` creates a directory with that name
- `mkdir .<name>` creates a hidden directory with that name
- `history` lists all recently run commands
- `!<number>` re-runs command specified (by number from history list)
- `cp -r <from> <to>` copies a directory from one path to another path
- `mv <from> <to>` renames a file or directory
- `rm <file>` deletes a file
- `rm -rf <directory>` deletes a directory

## 2.3 Important Unix Directories

- `/bin` Where built-in Unix commands (e.g. `ls`, `mkdir`, etc...) are stored.
- `/etc` Where system profiles are stored (e.g. users and passwords).
- `/usr/local/bin` Where user-installed programmes are often stored, unless user specifies a different install location

## INTRODUCTION

Contributors: Nathan TM Huneke, Yukai Zou  
Maintainers: Nathan TM Huneke

---

Under construction

### 3.1 Trainings and Certifications

Contributors: Yukai Zou  
Maintainer: Yukai Zou

---

Often times, your onboarding process to carry out neuroimaging research at UHS requires several trainings beforehand, so make sure you have taken the appropriate trainings and obtain certifications. Some essential trainings are listed below.

---

**Note:** This is not a complete list and the instructions may be out of date. If you spot something needs to be added or changed, please contact the Maintainer of this page.

---

#### 3.1.1 MR Safety Training

- Basic training on VLE: <https://www.uhs-vle.co.uk/course/view.php?id=905>
- (optional) Full training on VLE: <https://www.uhs-vle.co.uk/course/view.php?id=904>

### 3.1.2 Introduction to GCP eLearning

1. Visit [NIHR Learning website](#), or search “NIHR learn good clinical practice” on Google then select “Good Clinical Practice (GCP | NIHR...)”
2. At the bottom of that page, click on [NIHR Learn](#) (would recommend bookmark this page) if already registered; if not, click on “Getting started with NIHR Learn” and create an account. Follow the instructions to complete the registration process (and fill in any necessary profile info).
3. At the Home panel, select “GCP And Other Research Delivery Training” under Explore Learning.
4. Click on “Introduction to Good Clinical Practice (GCP) eLearning” under GCP.

### 3.1.3 PACS at UHS

To access the Picture Archiving and Communication System (PACS) at UHS:

1. Complete training “PACS at UHS e-learning” on [VLE](#).
2. On a UHS workstation (e.g. via UHS Remote Access), request access via the Service Catalogue. This can be found on the IT Service Desk, from there, scroll down to the bottom of the page and select the system you wish to request access to.

## 3.2 Research MRI Scanners

Contributors: Yukai Zou

Maintainers: Yukai Zou

---

**Important:** Our LabArchives Scheduler subscription, which was funded by the University’s Research Concordat, supported up to 12 users until 2nd September 2023. We are currently in the process of renewing our Scheduler license, and we are actively exploring a more long-term solution to the management of research MRI slots.

---

### 3.2.1 Making Reservation

LabArchives is being used as the online booking system for the research MRI scanners. It allows both UoS researchers and UHS staff (radiographers and research nurses) to view the scanner availability and make, update, and cancel reservations.

Link: <https://scheduler.labarchives.com/>

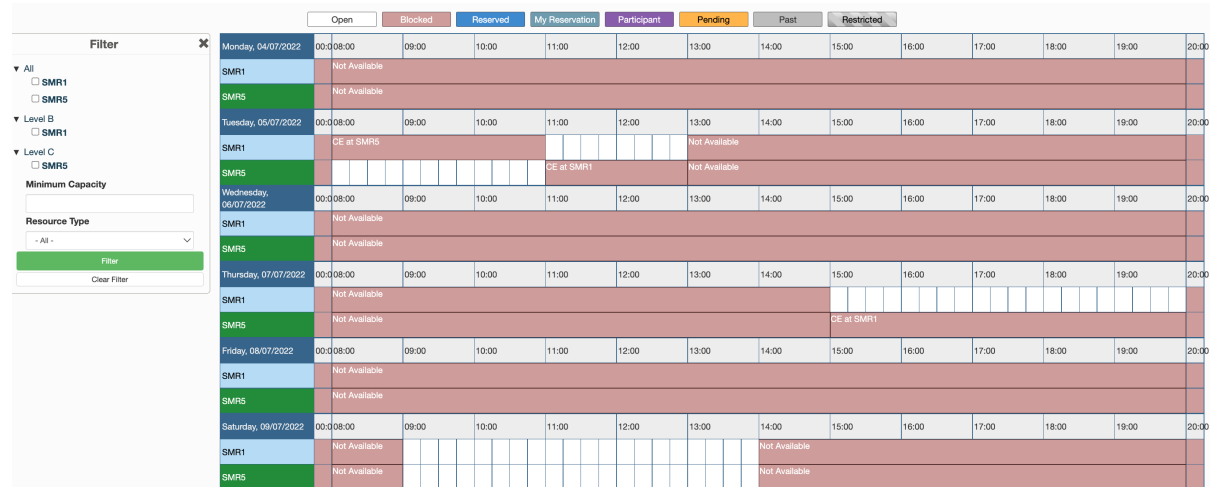


## How do I register?

You would need a web browser that is not Internet Explorer (e.g. Firefox, Chrome, Microsoft Edge), create a password that is at least 12-digit long, and selecting “United Kingdom” for the instance of creating account from the UK server.

## How does this work?

Here is how the calendar looks like:



**For Radiographer:** You, as the resource administrator of SMR1/SMR5, will get email notification every time a new reservation is proposed. You can then choose to approve/disprove depending on staff availability during that time. Our study coordinator will then confirm with the participant about the time and date to come.

The scheduler itself has an email subscription functionality - it will send you email to inform you a potential reservation to review and approve. You can adjust setting to turn on/off notifications to reduce email traffic, because they could get overwhelming (hopefully not).

**For Research Nurse/Coordinator:** You will have access to view the calendar of the research MRI scanner, where you can see the available scanners (SMR1/SMR5), when the scanner is available/unavailable/booked. You can propose new reservation on the calendar, which is pending approval (i.e. not confirmed yet) depending on radiographer availability during the proposed time. Once the reservation is approved, you would get an email notification, you can confirm with the participant about the time and date to come.

### Note: What are SMR1 and SMR5 exactly?

SMR1 and SMR5 are the names for the two MRI scanners in our hospital. SMR1 is located on Level B, and SMR5 is located on Level C. They are managed as two separate resources on the LabArchives, so that it helps radiographers to anticipate where to assist research scans. Make sure you know which scanner to book, and if unsure, please check with the project PI.

### What should I do after signing in?

After joining (once this step is done), you will need to click on your name at the top right and select Switch Organizations. This will open up a window where you can choose which organization to view. Select “Southampton Research MRI”, and you should be able to view the calendar.

### How to look up for the reservations?

To get there you need to click on “Application Management” on top and then select “Reservations”. From there you can specify the filter to, e.g., a certain period of time, user (who made the reservations), and search for any keywords in the Title/Description (i.e. subject ID, hospital ID) of the reservations.

## 3.2.2 Your feedback matters

If you wish to share your thoughts and feedbacks on how to improve the running and management of research MRI slots, they are more than welcome and appreciated. Below you may find the link to a survey for submitting anonymous feedback:

- Link to the survey: <https://forms.office.com/r/ZS6LrG3dde>

## 3.3 High Performance Computing

Contributors: Yukai Zou

Maintainer: Yukai Zou

---

**Note:** Under construction

---

High Performance Computing (HPC) accelerates large workflows of *highly-interdependent* sub-tasks, which can effectively make the processing and analysis of neuroimage data (especially large dataset) more efficient. The University of Southampton has one of the largest computational facilities in the UK. [The Iridis Compute Cluster](#) is one of the world’s top supercomputers, which is now in the 5th generation.

---

**Important:** There is a learning curve ahead. It’s important that you first get familiar with operating within a command line interface, and know how to write basic scriptings. More contents and resources of these topics will be added to this handbook.

---

---

**Note:** If you are off site when planning to use Iridis, you will need to connect to the University’s [VPN](#) first.

---

### 3.3.1 Request Access

Once you obtain UoS username and password, you will be able to apply for Iridis access after completing an [online application form](#). The form will ask for a brief justification of usage.

---

**Note:** If you are uncertain how to fill out the form, please contact your research advisor or line manager.

---

Once your access has been granted, you are also subscribed to the HPC mailing list. Make sure to keep an eye on any notice regarding power outage and scheduled maintenance, to ensure your workflow can complete successfully.

Here are a few links to the resources:

- HPC Community Wiki: <https://sotonac.sharepoint.com/teams/HPCCommunityWiki>
- **Submit a job:**  
<https://sotonac.sharepoint.com/teams/HPCCommunityWiki/SitePages/Submitting-Jobs-Slurm.aspx>
  - Specify job resources: <https://sotonac.sharepoint.com/teams/HPCCommunityWiki/SitePages/Submitting-Jobs-Slurm.aspx#specifying-job-resources>
- Job extension policy: <https://sotonac.sharepoint.com/teams/HPCCommunityWiki/SitePages/Fair-usage-and-Job-extension-policy.aspx>

### Login nodes

---

**Important:** Overloading login nodes can cause issues for other users. Login nodes are intended for short interactive processing only. For longer, interactive work, please utilize `sinteractive` sessions.

---

### Compute nodes

Initiating an interactive session on a compute node using the `sinteractive` command allows for interactive computing and the use of the GUI over X11, e.g. for RStudio, without the risk of overloading the login nodes.

Check out the following page for more details on how to use `sinteractive`: <https://sotonac.sharepoint.com/teams/HPCCommunityWiki/SitePages/Submitting-Jobs-Slurm.aspx#interactive-jobs>

### Using job arrays

To run jobs in parallel can greatly accelerate your workflow and save time.

- <https://sotonac.sharepoint.com/teams/HPCCommunityWiki/SitePages/Submitting-Jobs-Slurm.aspx#job-arrays>

## Visualisation on Iridis 5

Although mainly operated in a command line interface, Iridis 5 provides options for visualisation through graphical interface. This section concerns with the option of using [NICE Desktop Cloud Virtualisation](#).

- Setting up EngineFrame VNC: <https://sotonac.sharepoint.com/teams/HPCCCommunityWiki/SitePages/Data-Visualisation.aspx#engineframe-vnc>

## Visualization Expansion Portal (Linux Orange)

In addition to Linux Desktop, Iridis 5 visualisation expansion portal Linux Orange has become available since October 2023.

Nodes	orange02 and orange03
Sessions	4
CPU cores	12
Memory	240 GB
GPU available?	No

## (Optional) Mount Research Filestore to Iridis

The University Research Filestore is a secure research data storage service provided by iSolutions, for the research community to store active research data. Data on the research filestore is backed up and with an extra copy stored in a secure location (for disaster recovery purposes). The newest research filestore supports mounting onto Iridis 5, and this allows a more direct access to the research data via a shared filesystems and network.

---

**Note:** Due to the age and the nature of the setup, some research filestore cannot be mounted onto Iridis 5. This is mainly for security reasons. Older research filestore can't set up an NFS export that allows the directory to be mounted in a way that protects the sensitive nature of the data when retrieved via other users on Iridis 5.

**Solution:** If the data owner requests a new research filestore space, then it can be set up with the old data copied into it. To do this, make a request by filling out a form [here](#). iSolution should be able to mount the new filestore to Iridis 5 allow direct access of data.

---

## 3.3.2 A100 Scavenger Nodes

An A100 scavenger queue has been added to Iridis 5 in December 2023. This queue contains one node and has a twelve hour time limit, and is available to all Iridis 5 users. This may be of interest to users who are interested in testing the A100 GPUs before their full release to the user community.

## Specification

GPU hardware	Dual A100 NVLinked GPUs (160GB VRAM total across two GPUs)
CPU hardware	48 intel CPUs (24 cores x 2 sockets, Intel(R) Xeon(R) Gold 6336Y CPU @ 2.40GHz)
Memory	480GB of RAM

## How to access the node

Please use the partition `a100_scavenger` to access the node.



## CLOUD COMPUTING

Contributors: Yukai Zou  
Maintainer: Yukai Zou

---

**Note:** Under construction

---

### 4.1 Cloud Computing: Basics

According to [Amazon Web Services](#), cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider.

**Important:** There is a learning curve ahead. Mastering cloud computing requires dedicated effort and practice. To become proficient, you will need to follow a few key steps. First, learn how to request a worker that can do the computational job. Next, set up the software environment and transfer data onto the worker prior to the computation. Finally, specify what computations to run on the worker. If you are stuck, remember to take a break and get back later when your mind is refreshed, or seek assistance from a knowledgeable friend!

---

#### 4.1.1 Working on local machine vs. on cloud

	Working locally	Working on cloud
Who is the owner?	You	Cloud service provider
Who manages software environment?	You	Cloud service provider
Who installs software packages?	You	You
Availability of apps and code	Yes	Transfer needed
Availability of data	Yes	Transfer needed

### 4.1.2 Instance Types

`mem2_ssd1_v2_x16`

- `mem2`: Memory Available per core, ranging from 3.8 GB to 1.9 TB.
- `ssd1`: Disk Space Available per core, ranging from 40 GB to 60 TB.
- `v2`: Versioning. *Always use version 2 of an instance type.*
- `x16`: Cores/GPUs available, ranging from 2 to 126 cores.

---

**Note:** What are the differences between `ssd` and `hdd`?

---

---

**Note:** What are the key things to consider when choosing instance types?

- Whether your software requires more memory (`mem2` vs. `mem3`)
  - Whether your software requires large storage space (`ssd1` vs. `ssd2`)
  - Whether your software utilises multiple cores (`x16` vs. `x32`)
  - Whether your software is GPU optimised (`gpu_x32`)
- 

## 4.2 UK Biobank Research Analysis Platform (RAP)

---

**Important:** You must first have access to a project in UK Biobank. This typically comes with an Application ID, so please reach out to your PI to confirm this.

---

UK Biobank partners with [DNAnexus](#) to provide their cloud computing service, also known as Research Analysis Platform (RAP).

### 4.2.1 Request Access

1. Register at the UK Biobank Access Management System at <https://ams.ukbiobank.ac.uk/ams/>, which will take up to 10 working days for approval.

### 4.2.2 Image Processing on RAP

DXJupyterLab App is a DNAnexus extension that allows JupyterLab server to directly view and interact with DNAnexus projects, and allows analyses to be carried out in a Jupyter notebook running on the DNAnexus platform. You can simply connect to the Jupyter environment using a local web browser.

JupyterLab app with `IMAGE_PROCESSING` feature provides access to the following Python libraries:

- Nipype (1.8.4): click [here](#) for a very good Nipype tutorial.
- FreeSurfer (7.3.2)
- FSL (3.2.3)



A full list of individual apps and tools is available at the [Tools Library](#) documentation.

To launch DXJupyterLab on RAP's user interface:

1. Click on Projects on the navigation bar, and select a project in which you wish to run this app.
2. Click the Start Analysis button.
3. From the list of tools, select “JupyterLab with Python, R, Stata, ML, Image Processing”.

To launch DXJupyterLab from the command line, run:

```
dx run dxjupyterlab
dx run dxjupyterlab -h # List instructions for specifying inputs
```

**:: important::**

You will first need to set up the `dx` command-line client. Click [here](#) for a quickstart.

### 4.2.3 Costs and Billing

You can use the AWS estimator [here](#) to obtain an estimated cost. Detailed information about rates of different instance types is available [here](#).

## 4.3 NeuroImaging Tools & Resourcees Collaboratory (NITRC)

NeuroImaging Tools & Resourcees Collaboratory (NITRC, <https://nitrc.org>) is a free web-based resource that offers comprehensive information on neuroinformatics software and data. To date, more than 1,000 neuroimaging tools and resources have been registered, and NITRC has become the *de facto* standard mechanism for sharing neuroimaging tools and resources. NITRC contains three main components: Resource Registry (NITRC-RR), Imaging Repository (NITRC-IR), and Computational Environment (NITRC-CE).

### 4.3.1 NITRC-CE: the Computational Environment

NITRC-CE provides a cloud-based, pay-as-you-go virtual computing platform. It is pre-configured with popular neuroimaging tools, including FSL, FreeSurfer, ANTs, C-PAC, MRICron, etc. A full list of the installed packages is available [here](#). Additionally, you can also add your own commercial or open source tools.

### 4.3.2 Setting up NITRC-CE on Amazon Web Services (AWS)

---

**Note:** You can [build your own NITRC-CE instance](#).

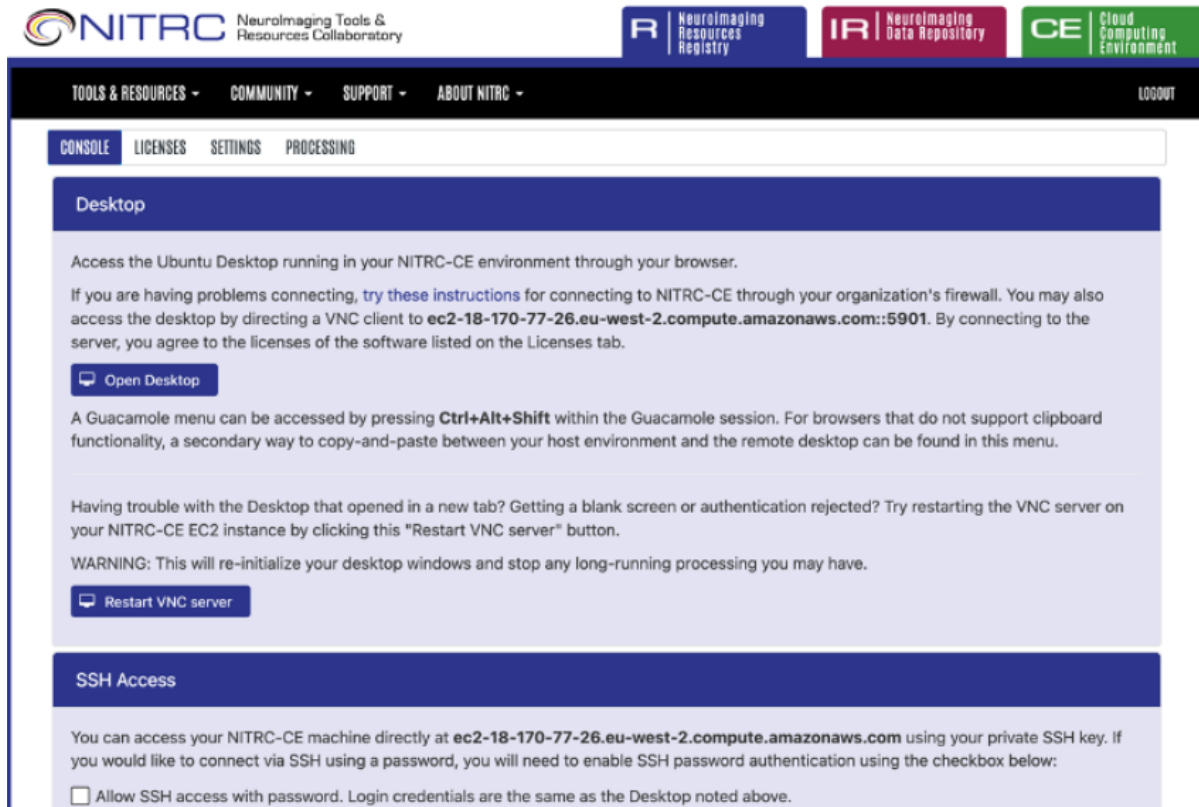
---

Using NITRC-CE on AWS is a straightforward process and can save substantial time setting up a computational environment for neuroimaging data analysis.

1. Visit <https://www.nitrc.org/>;
2. Click on “CE: Cloud Computing Environment” on top right;
3. Click on “Access NITRC-CE”, and select “Find a NITRC-CE AMI” in the drop-down menu;
4. Follow the instructions to set up EC2 instance.

5. (Optional) Under Advanced details, selecting “Request Spot Instances” can take advantage of spare/unused EC2 instances, which significantly reduces cost compared to On-Demand instances.
6. After the instance is launched and status checked, from your EC2 Console Dashboard, copy the Instance ID, visit the Public IPv4 address, and paste the Instance ID into the interface.

You will see this screen after login successfully:



### 4.3.3 AWS EC2 Pricing

Pricing information for using AWS EC2 instances is available [here](#).

## CONVERT FROM DICOM TO NIFTI

Contributors: Nathan TM Huneke, Harry Fagan, Yukai Zou  
Maintainers: Nathan TM Huneke

---

A common preprocessing step in neuroimaging data analysis is converting DICOM (Digital Imaging and Communications in Medicine) format output by the scanner to NIfTI (Neuroimaging Informatics Technology Initiative) format. This comes with several benefits, including a more standardised format for neuroimaging data, more consistent data format across multiple imaging modalities and scanners, enhanced compatibility with a wide range of neuroimaging software, and a more streamlined workflow for data analysis and sharing.

### 5.1 Dcm2niix

```
dcm2niix -b o -r y -w 1 -o /path/to/destination -f sub-$sID/%t_s%2s_%d/%d_%5r.dcm /path/  
→to/inputdata
```

- `-b` refers to BIDS sidecar, which is set to `o` to indicate there is no NIfTI.
- `-r` is set to `y`, to rename instead of convert DICOMs.
- `-w` is set to `1` to overwrite wherever name conflicts occur.
- `-f` specifies how the output filename will look like.

### 5.2 Dcm2bids

It would be a good practice to organise the NIfTI data in compliance with the BIDS (Brain Imaging Data Structure) standard. To do this, we suggest using an app called `dcm2bids`.

If you are using my *neuroimaging conda environment* then `dcm2bids` and its dependencies will already be installed.

### 5.2.1 The Dcm2bids Helper

The scanner will output a number of original and derived series, most of which are not required for further analysis and do not need to be stored in the BIDS tree. Those that are required need to be placed in the correct part of the directory tree, along with their JSON sidecar files. How is this achieved? The Dcm2bids helper will run a dummy conversion, allowing you to see all the series in NIfTI format their JSON sidecar files. You can then decide which ones to keep and build a configuration file that will automatically place the correct images in their corresponding places in the directory tree.

Here is an example of some source data from the scanner for a single subject:

```
study/
  sourcedata/
    sub-01/
      2/
        IM-0001-0001.dcm
        IM-0001-0002.dcm
        IM-0001-0003.dcm
        ...
      3/
      4/
      5/
      6/
      7/
      8/
      9/
      10/
      11/
      12/
      13/
      14/
      15/
      16/
      17/
      18/
      19/
      20/
```

Each directory within sub-01/ is a series containing a number of DICOM images. At the moment they don't mean much to us. Now if we run the Dcm2bids helper with

```
dcm2bids_helper -d sub-01 -o ../bids/
```

This creates a directory within the specified output directory bids/ called tmp\_dcm2bids/helper/. Let's have a look inside this directory:

```
study/
  sourcedata/
    sub-01/
      bids/
        tmp_dcm2bids/
          helper/
            002_study_t1_mprage_sag_p2_iso_1.0_20180512142245.json
            002_study_t1_mprage_sag_p2_iso_1.0_20180512142245.nii.gz
```

(continues on next page)

(continued from previous page)

```

003_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.bval
003_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.bvec
003_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.json
003_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.nii.gz
004_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.json
004_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.nii.gz
005_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.bval
005_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.bvec
005_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.json
005_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.nii.gz
006_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.json
006_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.nii.gz
007_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.json
007_study_ep2d_diff_b1300_iso2p4_phase_AP_36dir_20180512142245.nii.gz
009_study_gre_field_mapping_2.5mm_20180512142245_e1.json
009_study_gre_field_mapping_2.5mm_20180512142245_e1.nii.gz
009_study_gre_field_mapping_2.5mm_20180512142245_e2.json
009_study_gre_field_mapping_2.5mm_20180512142245_e2.nii.gz
010_study_gre_field_mapping_2.5mm_20180512142245_e2_ph.json
010_study_gre_field_mapping_2.5mm_20180512142245_e2_ph.nii.gz
011_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_20180512142245.json
011_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_20180512142245.nii.gz
012_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ json
012_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ nii.gz
013_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_20180512142245.json
013_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_20180512142245.nii.gz
014_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ json
014_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ nii.gz
015_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ json
015_study_ep2d_bold_moco_p4_2.5mm_TR2500_378_REST_MoCo_20180512142245.
↪ nii.gz
016_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.json
016_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.nii.gz
017_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_MoCo_20180512142245.
↪ json
017_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_MoCo_20180512142245.
↪ nii.gz
018_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.json
018_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.nii.gz
019_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_MoCo_20180512142245.
↪ json
019_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_MoCo_20180512142245.
↪ nii.gz
020_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.json
020_study_ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES_20180512142245.nii.gz

```

This makes a bit more sense. We can see this study is made up of a T1-weighted anatomical scan, DTI, field mapping, fMRI at rest and during a faces task. Not all of these series are original scans, some are derived by the scanner. It is

the original scans we need for further analysis. Opening up the JSON sidecar files can give more information.

## 5.2.2 Configuring Dcm2bids

To tell Dcm2bids which images are anatomical, DTI, etc. we need to create a configuration JSON file. This file should be kept in your dataset in a directory called `code/`. Here is an example configuration file based on this study:

```
{
  "descriptions": [
    {
      "dataType": "anat",
      "modalityLabel": "T1w",
      "criteria": {
        "SeriesDescription": "t1_mprage_sag_p2_iso_1.0"
      }
    },
    {
      "dataType": "dwi",
      "modalityLabel": "dwi"
      "criteria": {
        "SidecarFilename": "003_*"
      }
    },
    {
      "dataType": "fmap",
      "modalityLabel": "magnitude1",
      "IntendedFor": [
        5,
        6
      ],
      "criteria": {
        "SeriesDescription": "gre_field_mapping_2.5x2.5x3",
        "EchoNumber": 1
      }
    },
    {
      "dataType": "fmap",
      "modalityLabel": "magnitude2",
      "IntendedFor": [
        5,
        6
      ],
      "criteria": {
        "SidecarFilename": " *_e2.json"
      }
    },
    {
      "dataType": "fmap",
      "modalityLabel": "phasediff",
      "IntendedFor": [
        5,
        6
      ],
    },
  ],
}
```

(continues on next page)

(continued from previous page)

```

    "criteria": {
      "SidecarFilename": "*_e2_ph.json"
    },
    {
      "dataType": "func",
      "modalityLabel": "bold",
      "customLabels": "task-rest",
      "criteria": {
        "SeriesDescription": "ep2d_bold_moco_p4_2.5mm_TR2500_378_REST",
        "ImageType": [
          "ORIGINAL",
          "PRIMARY",
          "FMRI",
          "NONE",
          "ND",
          "MOSAIC"
        ]
      },
      "sidecarChanges": {
        "TaskName": "rest"
      }
    },
    {
      "dataType": "func",
      "modalityLabel": "bold",
      "customLabels": "task-faces",
      "criteria": {
        "SeriesDescription": "ep2d_bold_moco_p4_2.5mm_TR2500_178_FACES",
        "ImageType": [
          "ORIGINAL",
          "PRIMARY",
          "FMRI",
          "NONE",
          "ND",
          "MOSAIC"
        ]
      },
      "sidecarChanges": {
        "TaskName": "faces"
      }
    }
  ]
}

```

The idea behind this configuration file is that Dcm2bids will search through all of the JSON sidecar files during the conversion and look for matches with the `criteria` specified in the configuration file. Wherever a match is found, that NIFTI file and its JSON sidecar will be moved to the correct place in the BIDS tree. It is important therefore that the `criteria` you specify apply *to one scan only*. Otherwise Dcm2bids will show an error.

The `customLabels` fields are used to name the files after conversion. Use `customLabels` to add information, e.g. task name, after the modality label. Dcm2bids will use information in `sidecarChanges` to write lines into the converted NIFTI file's corresponding JSON sidecar. This is important for adding task names for BOLD runs, a BIDS requirement.

You can also add other extra information if needed.

The least intuitive field is the `IntendedFor` field. In the example above, this field is a list describing which scans the field map applies to. In this case it is each of the BOLD runs, which are the 6th and 7th scans described in the file. The configuration file counts up from 0, so in our file above, the number for each scan is as follows:

```
T1w (0)
DTI (1)
Fieldmap 1st echo (2)
Fieldmap 2nd echo (3)
Fieldmap Diff (4)
BOLD rest (5)
BOLD faces (6)
```

### 5.2.3 Running Dcm2bids

---

**Note:** Prior to running `Dcm2bids` for the first time you can optionally run `dcm2bids_scaffold` to pre-populate your BIDS folder with some key files, like so:

```
dcm2bids_scaffold -o OUTPUT_DIR
```

Once you have written your configuration file, `Dcm2bids` can be run as follows:

```
dcm2bids -d sourcedata/sub-01 -p 01 -c code/bids_config.json --forceDcm2niix
```

There's a few arguments to note here:

- `-d` refers to the DICOM directory
- `-p` refers to participant ID
- `-c` refers to the location of your configuration file
- `--forceDcm2niix` forces the use of `Dcm2niix` for the dicom to NIfTI conversion

If this were a multi-session study, you would need to add a session label argument (`-s`) to the command above. For example:

```
dcm2bids -d sourcedata/sub-01 -p 01 -s 02 -c code/bids_config.json --forceDcm2niix
```

This would convert the data for sub-01, session 02.

### 5.2.4 Example scripts

Here is an example script that loops through participants to convert dicoms to NIfTI for a single session experiment:

```
#!/bin/bash

set -e -u

# You would run this script from the directory you want your BIDS dataset contained in

for id in `seq -w 1 20` ; do # seq -w creates a list from 01 to 20
```

(continues on next page)



(continued from previous page)

```

    subj="sub-id" # puts "sub-" in front of each id in turn, eg. "sub-01" "sub-02" etc.
    echo "=====> converting subj..." # in bash variables are recognised with the $
↪symbol
    dcm2bids -d sourcedata/subj -p id -c code/bids_config.json --forceDcm2niix # the
↪dcm2bids command
    echo
    echo "Done"
done

```

Here is an example script using DataLad to convert either a single session or both sessions for a multi-session experiment:

```

#!/bin/bash

set -e -u

echo "Enter subject IDs to add MRI data for: "

read -a SUBS # create array from input

prefix="sub-"

datalad update -d sourcedata -r --merge
datalad save -r

for sub in "${SUBS[@]}" # loop through array
do
    label=${sub}/${prefix} # remove prefix to define participant label
    # Check which session to convert
    read -p "For sub, convert dicoms for session1, session2 or both?
(Enter session1, session2, or both): " sessn
    # Run conversion
    case "$sessn" in
        session1) # commands for session1 only
            echo "Converting session 1 only..."
            echo
            datalad run \
                -i sourcedata/sub/ses-01/ \
                -o sub/ses-01/ \
                -m "Convert sub/ses-01 to nifti" \
                "dcm2bids -d sourcedata/sub/ses-01 -p label -s 01 -c code/bids_config.
↪json --forceDcm2niix" ;;
        session2) # commands for session2 only
            echo "Converting session 2 only..."
            echo
            datalad run \
                -i sourcedata/sub/ses-02/ \
                -o sub/ses-02/ \
                -m "Convert sub/ses-02 to nifti" \
                "dcm2bids -d sourcedata/sub/ses-02 -p label -s 02 -c code/bids_config.
↪json --forceDcm2niix" ;;
        both) # commands for both sessions

```

(continues on next page)

(continued from previous page)

```

echo "Converting both sessions..."
echo
sessions=("01" "02")
for sesh in "${sessions[@]}"
do
    datalad run \
        -i sourcedata/${sub}/ses-${sesh}/ \
        -o ${sub}/ses-${sesh}/ \
        -m "Convert ${sub}/ses-${sesh} to nifti" \
        "dcm2bids -d sourcedata/${sub}/ses-${sesh} -p ${label} -s ${sesh} -c
↳code/roar_bids_config.json --forceDcm2niix"
    done ;;
*) # else
    echo "$(tput setaf 1)$(tput bold)ERROR$(tput sgr 0)" # "ERROR" red &
↳bold
    echo "Please enter session1, session2, or both" ;;
esac
done

```

## PSEUDONYMISE YOUR DATASET

Contributors: Nathan TM Huneke  
Maintainers: Nathan TM Huneke

---

If you plan to share your dataset then it is important to pseudonymise/anonymise your dataset in line with data protection laws. You should ensure JSON sidecar files do not include identifiable information and that anatomical images have facial structures removed.

It is best to store pseudonymised data in a separate dataset from the source identifiable data. The reason for this is if you are using version control such as DataLad then it might be possible to retrieve identifiable information from the version history.

### 6.1 Removing facial structures with PyDeface

[PyDeface](#) is a BIDS app that allows easy removal of facial structures from anatomical MRI scans.

#### 6.1.1 Install PyDeface

Install with:

```
pip install pydeface
```

#### 6.1.2 How to use PyDeface

Assuming you will be storing pseudonymised data in a new dataset, you can use the following command:

```
pydeface \  
--outfile pseudonymised_dataset/path/to/T1w.nii.gz \  
--force \  
sourcedata/path/to/T1w.nii.gz
```

The `--outfile` argument specifies the output path. The `--force` argument will overwrite the output if it already exists. It is useful to leave this on to prevent errors. Finally, the path to the original anatomic image is the only positional argument required.

## 6.2 Pseudonymisation with BIDSonym

---

**Todo:** Add info re: BIDSonym

---

## BRAIN IMAGING DATA STRUCTURE (BIDS)

Contributors: Harry Fagan  
Maintainers: Nathan TM Huneke

---

Under construction

<https://bids.neuroimaging.io/>  
<https://www.nature.com/articles/sdata201644>



## DATALAD: DISTRIBUTED DATA MANAGEMENT

Contributors: Nathan TM Huneke  
Maintainers: Nathan TM Huneke

---

The following text is adapted from the [DataLad handbook](#).

[DataLad](#) is a free and open source command line tool, available for all major operating systems, and builds up on Git and [git-annex](#) to allow sharing, synchronizing, and version controlling collections of large files in repositories known as DataLad datasets. You can find information on how to install DataLad at [handbook.data-lad.org/en/latest/intro/installation.html](http://handbook.data-lad.org/en/latest/intro/installation.html).

---

**Note:** If you are using my *neuroimaging conda environment* then datalad will be installed.

---

### 8.1 Get the dataset

A DataLad dataset can be cloned by running:

```
datalad clone <url>
```

Once a dataset is cloned, it is a light-weight directory on your local machine. At this point, it contains only small metadata and information on the identity of the files in the dataset, but not actual *content* of the (sometimes large) data files.

### 8.2 Retrieve dataset content

After cloning a dataset, you can retrieve file contents by running:

```
datalad get <path/to/directory/or/file>
```

This command will trigger a download of the files, directories, or subdatasets you have specified.

DataLad datasets can contain other datasets, so called *subdatasets*. If you clone the top-level dataset, subdatasets do not yet contain metadata and information on the identity of files, but appear to be empty directories. In order to retrieve file availability metadata in subdatasets, run:

```
datalad get -n <path/to/subdataset>
```

Afterwards, you can browse the retrieved metadata to find out about subdataset contents, and retrieve individual files with `datalad get`. If you use `datalad get <path/to/subdataset>`, all contents of the subdataset will be downloaded at once.

## 8.3 Stay up-to-date

DataLad datasets can be updated. The command `datalad update` will *fetch* updates and store them on a different branch (by default `remotes/origin/master`). Running:

```
datalad update --merge
```

will *pull* available updates and integrate them in one go.

## 8.4 Find out what has been done

DataLad datasets contain their history in the `git log`. By running `git log` (or a tool that displays Git history) in the dataset or on specific files, you can find out what has been done to the dataset or to individual files by whom, and when.

## 8.5 Saving changes you make to a dataset

If you make a change to a dataset, you will need to save it into the history so that you or other users in the future can see what was done when and by whom. The command `datalad status` will show you whether any changes need to be saved. If you run this command you might see either the message:

```
nothing to save, working tree clean
```

In which case the log is up to date, or:

```
modified: file1.txt (file)
```

There is a file (`file1.txt`) that has been modified. This modification needs to be saved in the dataset's history. To do so run a `datalad save` command. This will produce the following output:

```
add(ok): file1.txt (file)
save(ok): . (dataset)
action summary:
  add (ok: 1)
  save (ok: 1)
```

`file1.txt` was added to the repository and then the dataset was saved. It is useful to add a message regarding what was saved, so that you can keep track of what has been done to the dataset. To do so use the `-m` argument, e.g.:

```
datalad save -m "Add file1.txt"
```

If you now look at the `git log` you will see that this entry is accompanied by the message that `file1.txt` was added:



```
commit 7a286d45195b7ac6a167fe7b2a2229fa87af2425
Author: nh6g15 <n.huneke@soton.ac.uk>
Date:   Mon Jun 21 14:49:43 2021 +0100
```

```
    Add file1.txt
```

```
commit a736983ec90a2094ce401105173122f9a9033824
Author: nh6g15 <n.huneke@soton.ac.uk>
Date:   Thu Jun 17 14:34:17 2021 +0100
```

```
    Apply YODA dataset setup
```

```
commit 8bf3c337ef7c06852ffe07ee738eae7f44b1f46c
Author: nh6g15 <n.huneke@soton.ac.uk>
Date:   Thu Jun 17 14:34:15 2021 +0100
```

```
    [DATALAD] new dataset
```

## 8.6 DataLad Run

Possibly the most useful feature of DataLad for computationally intensive analyses (e.g. neuroimaging) is the `datalad run` command. Using this command allows you to capture your command(s), fetch relevant files, do something with them, and then save the results.

For example, the following `datalad run` command, runs a script on a file called `anonymised_dataset.csv` to convert it to long format:

```
datalad run \
-m "Save long format dataset" \
-i anonymised_dataset.csv \
-o dataset_long_format.csv \
"code/convert2long.R"
```

After running this, checking the `git log` will show the following:

```
commit 1eac06986726b3f98c61b0b7eab0964ca54c2e0b (HEAD -> master)
Author: nh6g15 <n.huneke@soton.ac.uk>
Date:   Fri Jun 25 16:17:16 2021 +0100
```

```
    [DATALAD RUNCMD] Save long format dataset
```

```
=== Do not change lines below ===
{
  "chain": [],
  "cmd": "code/convert2long.R",
  "dsid": "9663676d-5ac4-4071-9406-6ee778f7d49e",
  "exit": 0,
  "extra_inputs": [],
  "inputs": [
    "anonymised_dataset.csv"
  ],
```

(continues on next page)

(continued from previous page)

```
"outputs": [
  "dataset_long_format.csv"
],
"pwd": "."
}
^^^ Do not change lines above ^^^
```

Because the command and files needed are all saved in the log, we can even re-run this command if needed! To do so, we use `datalad rerun <SHASUM>` using the SHASUM of the commit in question. For example:

```
datalad rerun 1eac06986726b3f98c61b0b7eab0964ca54c2e0b
```

I strongly suggest you read the Chapter on `datalad run` in the [DataLad handbook](#) as this command is so important.

## 8.7 Dataset Storage and Backup

DataLad includes tools for easily managing dataset storage and backup. These work very nicely with the University's research filestore. A suggested workflow is described below.

### 8.7.1 DataLad Siblings

Before understanding how DataLad can be used for storage and backup, it is important to understand the concept of a [DataLad sibling](#). A datalad sibling is essentially a 'copy' of a dataset stored in another location. Each sibling will have its own `git` log, and changes made in either dataset can be incorporated into the other with a `datalad update` command. For dataset storage and backup, we will be using a special kind of sibling known as a `special remote`. We will use two types of `special remote`: `remote indexed archives` and a `gitlab remote`.

### 8.7.2 Remote Indexed Archives

[RIA stores](#) can be easily created or extended from within any dataset. The advantage of using an RIA store is that the remote machine does not need datalad to be installed. Nevertheless, datalad will still be able to find and retrieve dataset contents through a `datalad get` command. The RIA store is therefore perfect for content storage and backup, particularly as the University filestore is regularly backed up itself.

The RIA store is created with the following command:

```
datalad create-sibling-ria -s ria-backup ria+<URL>
```

If using the university filestore you would replace `<URL>` with the path to access your research filestore via SSH. This takes the following form:

```
ssh://ssh.soton.ac.uk:/research/absolute/path/to/ria-store
```

The final command therefore looks like this:

```
datalad create-sibling-ria -s ria-backup ria+ssh://ssh.soton.ac.uk:/research/absolute/
↪path/to/ria-store
```

**Note:** To access your research filestore via SSH you need to ask iSolutions for the directory to be NFS and you need to be added to the list of SSH gateway users.

To backup your dataset contents in the RIA store, use the following command:

```
datalad push --to ria-backup
```

### Tip: Accessing a directory via SSH without password

It can be useful to set up access to your filestore via SSH without the need for a password. This way you can automate operations like getting dataset contents or pushing dataset updates without needing to input your password every 5 seconds. Doing this is very straightforward through the use of an SSH key.

First on your machine, type the following command:

```
ssh-keygen
```

Just use the defaults when prompted by pressing return. Next, copy your key to the SSH server with:

```
ssh-copy-id user@server
```

You should now be able to login without using a password.

This key can be copied (using secure copy) to any other machine to allow access to the same server without using a password:

```
scp ~/.ssh/id_<key> user@machine:~/.ssh/id_<key>
scp ~/.ssh/id_<key>.pub user@machine:~/.ssh/id_<key>.pub
```

## 8.7.3 GitLab Siblings

RIA stores are great, but they have one problem: *they are not human-readable*. Here is an example of what an RIA store actually looks like:

```
/path/to/my_riastore
├── 946
│   └── e8cac-432b-11ea-aac8-f0d5bf7b5561
│       ├── annex
│       │   ├── objects
│       │   │   ├── 6q
│       │   │   │   └── mZ
│       │   │   │       ├── MD5E-s93567133--7c93fc5d0b5f197ae8a02e5a89954bc8.nii.gz
│       │   │   │       └── MD5E-s93567133--7c93fc5d0b5f197ae8a02e5a89954bc8.nii.gz
│       │   │   ├── 6v
│       │   │   │   └── zK
│       │   │   │       ├── MD5E-s2043924480--47718be3b53037499a325cf1d402b2be.nii.gz
│       │   │   │       └── MD5E-s2043924480--47718be3b53037499a325cf1d402b2be.nii.gz
│       │   │   ├── [...]
│       │   │   └── [...]
│       │   └── archives
│       │       └── archive.7z
```

(continues on next page)

(continued from previous page)



Cloning this dataset would involve finding out the dataset `id`, which is not trivial when you don't know where to look. Instead, we can store a human-readable version of the dataset on the University's [GitLab instance](#). GitLab is not suitable for storing dataset contents (other than code), so does need to be used in conjunction with the filestore. GitLab instead stores metadata about the dataset, allowing retrieval of contents from the RIA store using human-readable commands. This is very useful when collaborating with others on a project.

## Setup

Before a GitLab remote can be created, you need to complete a few setup steps:

1. Generate a personal access token for GitLab [here](#).
2. Copy and paste the following into a text file, inserting your personal access token in the appropriate field:

```
[soton]
url = https://git.soton.ac.uk
private_token = [insert token here]
```

3. Save this file in your home directory (`~`) as `.python-gitlab.cfg`.

## Create the GitLab remote

To create a GitLab remote, use the following command:

```
datalad create-sibling-gitlab -s gitlab --site soton --project <path/to/project>
```

The metadata and code can then be pushed to gitlab with:

```
datalad push --to gitlab
```

Human-readable metadata will now be visible at <https://git.soton.ac.uk/path/to/project>. The dataset can then be cloned with:

```
datalad clone https://git.soton.ac.uk/path/to/project .
```

assuming the cloner has permissions to view the dataset. Contents can then be retrieved with a `datalad get` as `datalad` will by default search the RIA store for contents.

**Note:** To reduce complexity, it helps to create a GitLab remote for the **superdataset only**. Any subdatasets can be backed up to an RIA store. As long as the superdataset is cloned, it is possible to then retrieve subdataset contents from the RIA store. One line of code needs to be run in the superdataset to configure this:

```
git config -f .datalad/config "datalad.get.subdataset-source-candidate-origin" "ria+<URL>
↪ #<id>"
```

## 8.8 Procedures for Setting Up Datasets

To help speed up the process of setting up a new datalad dataset (including backup siblings), I have written a number of [procedures](#). The github repository including installation and usage instructions is [here](#).

## 8.9 More information

More information on DataLad and how to use it can be found in the DataLad Handbook at [handbook.datalad.org](http://handbook.datalad.org). The chapter [What you really need to know](#) is particularly useful.



## FMRI ANALYSIS

Contributors: Nathan TM Huneke  
Maintainers: Nathan TM Huneke

---

Under construction

### 9.1 Quality Assurance Checks

Contributors: Nathan TM Huneke  
Maintainers: Nathan TM Huneke

---

It is important to check the quality of your fMRI data before embarking on analyses to ensure there are no systematic issues with your data.

#### 9.1.1 Standard deviation and Signal to Noise ratio maps

Looking at these maps derived from your functional EPI runs can be helpful for ‘eyeballing’ any clear artefacts or systematic issues with the acquisition of your data.

---

**Todo:** Add examples of artefacts on standard deviation and SNR images

---

#### Creating standard deviation and tSNR maps

Some scanners can produce these maps for you, however, you might want to create them offline. This can be achieved easily with `fslmaths` commands. You will need FSL to be installed.

**The formula for calculating temporal signal to noise (tSNR) is**

tSNR = temporal mean/temporal standard deviation

We can therefore use the following commands to create the maps:

```
# create mean map
fslmaths <input_bold.nii.gz> -Tmean <output_mean.nii.gz>

# create standard deviation map
fslmaths <input_bold.nii.gz> -Tstd <output_std.nii.gz>

# calculate temporal SNR (tSNR)
fslmaths <output_mean.nii.gz> -div <output_std.nii.gz> <output_tSNR.nii.gz>
```

Replace the filenames in < > above with the paths to your input images and chosen output paths.

## 9.1.2 Using MRIQC for Quality Assurance

MRIQC is a BIDS app that extracts quality metrics from structural and functional MRI data.

It is best to run and install MRIQC using Docker.

---

**Note:** If you are using a university machine then you will need to be given permission to use Docker. Please speak to the admin for the machine.

---

### Install MRIQC with Docker

Type the following to download the latest MRIQC image:

```
docker run -it poldracklab/mriqc:latest --version
```

### Run MRIQC with Docker

#### Participant-level

To prevent your machine's RAM from being overloaded, it is best to run 1 participant at a time. Use the following command:

```
docker run -it --rm -v \
  <path/to/bids/dataset>:/data:ro -v \
  <desired/path/to/outputs>:/out \
  nipreps/mriqc:latest \
  /data /out participant \
  --participant-label 01 \
  --user $( id -u )
```

**Warning:** The paths above must be *absolute* paths.

If you see potential artifacts on the reports, you can re-run the participant and output extra information:



```
docker run -it --rm -v \
  <path/to/bids/dataset>:/data:ro -v \
  <desired/path/to/outputs>:/out \
  nipreps/mriqc:latest \
  /data /out participant \
  --participant-label 01 \
  --user $( id -u ) \
  --ica --verbose-reports
```

This will provide a more detailed report as well as the results of independent component analysis.

## Group-level

MRIQC can also generate group-level reports to help you identify any outlying participants. Use the following command to run a group-level QC check:

```
docker run -it --rm -v \
  <path/to/bids/dataset>:/data:ro -v \
  <desired/path/to/outputs>:/out \
  nipreps/mriqc:latest \
  /data /out group \
  --user $( id -u )
```

## 9.2 fMRI Preprocessing with fMRIPrep

Contributors: Nathan TM Huneke, Nick Hedger

Maintainers: Nathan TM Huneke

---

fMRIPrep is a NiPreps application for preprocessing task-based and resting-state fMRI.

The idea is that this application provides a common, comprehensive pipeline that can be carried out across multiple studies to allow a reproducible neuroimaging analysis workflow.

fMRIPrep is installed on uos-210028 and uos-211997. If you are installing on a personal machine (not recommended) then you can find installation instructions [here](#). Please install as a docker image.

If you are new to fMRI preprocessing, then please see the [introductory guide](#).

### 9.2.1 Beginner's Guide to fMRI Preprocessing

Contributors: Nathan TM Huneke, Nick Hedger

Maintainers: Nathan TM Huneke

---

Functional neuroimaging aims to relate performance in some sort of task with neural activity in a region of the brain.

## Logic of fMRI

The basic logic underlying functional MRI is as follows:

- Firstly, the brain consumes around 20% of our oxygen uptake. This is usually in response to neural activity, which uses up energy and therefore creates a need for oxygen.
- Hemoglobin is a molecule in the blood that can transport this oxygen.
- The magnetic properties of hemoglobin differ depending on whether or not it is carrying oxygen.
- De-oxygenated hemoglobin has what are called paramagnetic properties, which can cause distortions to the local magnetic fields induced by the scanner. This means that protons receive slightly different magnetic field strengths, which interferes with magnetic resonance. By contrast, oxygenated hemoglobin is more diamagnetic and less paramagnetic than de-oxygenated hemoglobin.
- When there is neural activity in a region, metabolic demands increase and so more oxygenated hemoglobin is rushed to the area. As a result, the presence of oxygenated hemoglobin reduces the signal distortions caused by de-oxygenated hemoglobin.

This forms the basis of the blood oxygen level dependent (BOLD) response. It is sensible to think of the BOLD response as being the ratio of oxygenated to deoxygenated hemoglobin, which is measured via these distortions to magnetic resonance.

Unlike structural images, where the relative signal intensity in an image depends on the T1 relaxation of the tissue, signal intensity in functional images represents a different, independent component of the MR signal: T2\* relaxation. On the application of an RF pulse, the precessions of hydrogen protons are aligned in one direction (they are phase coherent), which produces a strong signal. However, immediately afterwards, this arrangement is lost and the precessions lose coherence. Spatially proximate spins may interact like bumper cars on a track, causing some spins to precess faster than others. This dephasing of precessions leads to an exponential decay in the MR signal that is described by the time constant T2. This dephasing can be partly explained by the contribution of paramagnetic artefacts that cause local magnetic field variations, and this contribution is described by the time constant T2\*. Dephasing in a paramagnetic environment (near deoxygenated hemoglobin) is faster than dephasing in a diamagnetic environment (near oxygenated hemoglobin). Thus, pulse sequences sensitive to T2\* relaxation show more signal intensity in locations where blood is highly oxygenated.

## A Simple Experiment

To clarify, it is perhaps useful to think of things in terms of a simple experiment (see figure below). Suppose we are interested in recording from the visual cortex, which is at the posterior part of the brain. We want to contrast two conditions. In the rest condition (upper row), the observer views a blank screen and in the active condition (lower row), the observer views a salient visual stimulus.

In the rest condition, since there is no visual stimulation, there is little neural activity in the visual cortex and so there is a high concentration of deoxy-hemoglobin. As a result, the magnetic susceptibility of the nearby protons is distorted and proton dephasing is fast. This means that the visual cortex emits a weak T2\* signal.

However, in the active condition, the participant views a salient stimulus, this means that there is an increase in neural activity in the visual cortex, and so oxyhemoglobin rushes to this region. As a result, there is a reduction in the paramagnetic distortions to magnetic resonance and therefore T2\* increases. It's this increase in signal that is shown in the image on the right below.

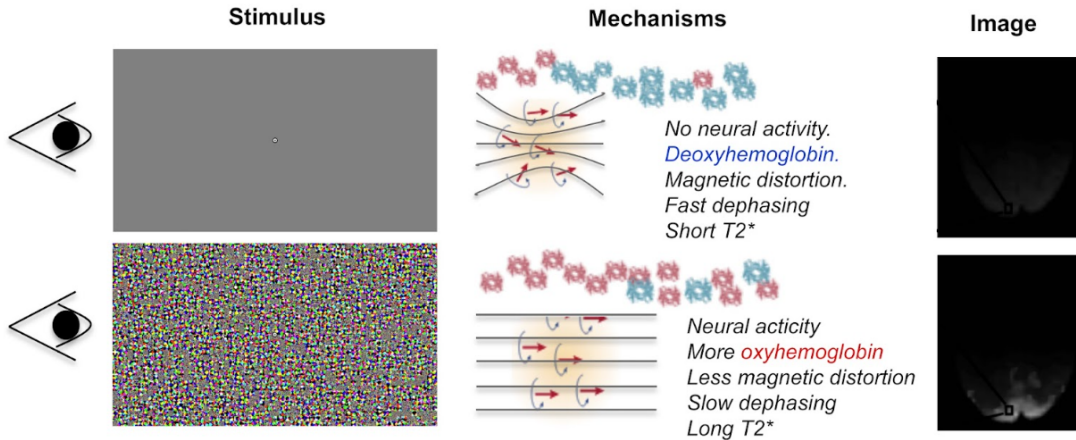


Fig. 1: Upon visual stimulation there is an increase in local neural activity in the visual cortex. This results in oxygenated hemoglobin rushing to the area. As a result, the paramagnetic distortions to magnetic resonance are reduced and the  $T2^*$  increases. This is reflected on the image on the right.

### Hemodynamic Response Function

When we record from a region of neural activity, the BOLD response resembles a hemodynamic response function (HRF). We can explain the shape of this function in terms of the concentration of oxygenated and de-oxygenated hemoglobin.

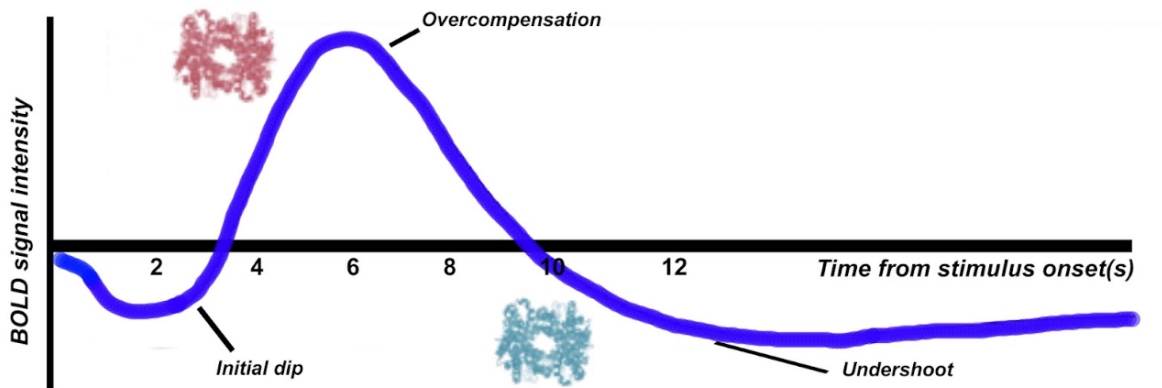


Fig. 2: Firstly, there tends to be an initial dip in the function, which reflects the neurons consuming oxygen. Therefore temporarily, the concentration of deoxygenated hemoglobin is higher. As a compensatory mechanism, the vascular system rushes more oxygenated hemoglobin to the area, at a faster rate than it can be consumed, giving rise to local blood oxygen levels that are higher than necessary. This results in an elevated response (an overcompensation) that typically peaks after around 6 seconds. The third component is an undershoot. This probably reflects the vascular system tiring, before oxygen consumption returns to normal again, as a result there is temporarily more de-oxyhemoglobin again.

The shape of the HRF is not just worth learning about for purely theoretical reasons. It has a number of practical

applications. Most notably, the canonical shape of the HRF is an important component of the statistical models that are used to analyze functional imaging data.

## A Typical Scanning Session

This next section is designed to give you a more tangible idea of what happens during a typical scanning session. A typical scanning session consists of at least 3 separate scans. A *localiser scan*, a *high resolution structural scan* and a *functional scan*.

### Localiser

Firstly, a localizer scan is conducted. This consists of a 1-2 minute low resolution scan, that allows the radiographer to localize the brain for further scans. The basic idea of this scan is to allow the radiographer to determine where the brain is located in scanner coordinates and use these coordinates to guide further scans.

### High Resolution Structural scan

Next, a high resolution T1 structural scan is collected. Because this is a high resolution image, it takes a long time—usually 8-10 minutes. But why, you may ask, are we bothered with collecting a static, structural image for a functional imaging study?

The reason this structural scan is collected is because the functional data that follows is much lower resolution. We need this higher resolution scan so that we can register the functional data to an image that has more precise spatial co-ordinates.

By way of illustration, some functional data is plotted in the figure below (a). As you can see, this is very low resolution and it's hard to differentiate between structures. In fact, it's hard for us to even tell what part of the brain we are recording from. However, if we superimpose this on the high resolution structural scan (b), this all becomes a lot easier. We can see that we are recording from the posterior part of the brain, and we can better differentiate between parts of the subjects' anatomy.

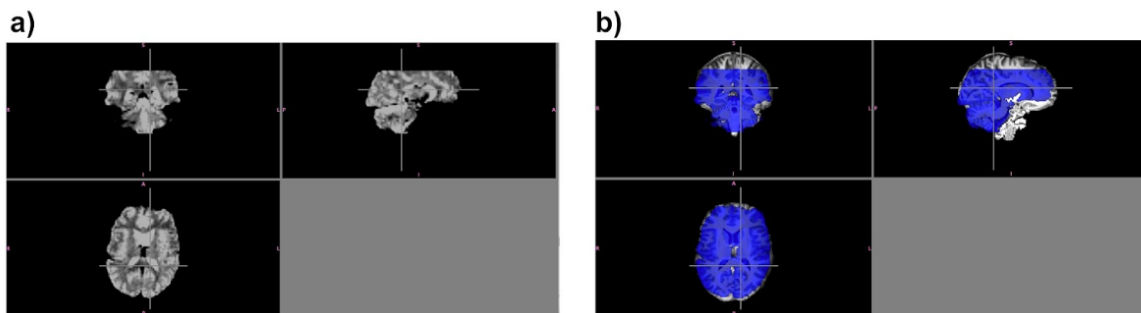


Fig. 3: a) Shows low resolution functional data (74\*74\*36). b) Shows the same data (translucent blue) superimposed on a high resolution image (144\*198\*200). The functional data has been upsampled and spatially registered to the same space as the high resolution structural data. This registration process improves the ability to make inferences about regions of task-related activation.

## Functional Scan

Next, the functional scan itself is collected. This consists of a series of low-resolution scans, or *volumes* that are collected while a task is being performed by the participant. It is important to note that functional data are 4 dimensional. First there is the 3 dimensional image of the brain and the fourth dimension is the volume number in the time dimension. Its perhaps useful to think of functional data as being like a 3 dimensional video recording of the brain, with each volume being like a ‘frame’ of a video. Obviously, the length of a functional scan will vary depending on the complexity of the particular experimental design and related factors, but a typical functional scan will be around 30 minutes long and is usually broken into a series of discrete *functional runs* of approximately 10 minutes.

## Functional Scan: Important Parameters

There are two parameters of a functional scan that are important to understand. First is the *repetition time*, which is abbreviated to *TR*. This is the length of time between successive functional volumes. If the whole brain is scanned, a TR is usually 2-3 seconds. Secondly, there is the *size of the voxels* (resolution), or the 3 dimensional units of space that are recorded from. You can think of these in terms of the brain being broken down into as cubes (or more precisely - pyramidal shapes). From each voxel there is a corresponding data point. If the voxel size is large, we have a low resolution image, whereas if the voxel size is small the representation of space is more precise and the image has higher resolution.

There is an inherent trade off between these two parameters, for instance if we want small voxels, we then have to record more data points per volume and thus it takes longer to scan the whole brain. However, if we have large voxels, we only need to record a few data points and thus our TR can be shorter. In other words, we can sacrifice spatial resolution for temporal resolution and vice versa. Conceptually, this is the same trade off associated with cathode ray tube (CRT) monitors: low resolutions support higher refresh rates than higher resolutions.

Of course, not all functional scans require each volume to be a recording of the entire brain. It is perfectly viable to obtain *partial brain* functional volumes to decrease the TR and length of the experiment. In the functional data we saw *earlier*, much of the parietal lobe was sacrificed so that better spatial resolution of the occipital and temporal lobes could be obtained.

## Functional Scan: Example

Let’s think about some hypothetical functional data for a moment. The main point of this is to illustrate the 4 dimensional nature of the data and the vast amount of data handling involved with functional imaging experiments.

As described above, a functional scan consists of a series of 3 dimensional volumes, each of which is composed of voxels. A typical voxel might be *3mm cubed* in size. To scan an entire human brain once, *33 separate slices* may be required, each containing a 64\*64 grid of voxels. Per individual volume, we are therefore recording *135,168* data points. However, we don’t just obtain one volume in a functional scan, we record very many successively. In a 30 minute functional scan with a TR of 2 seconds, each of these 135,168 data points would need to be recorded *900* separate times. This gives us a total of *121,651,200 data points*. Thus, even making fairly standard assumptions about parameters, the amount of data involved in functional imaging is somewhat intimidating.

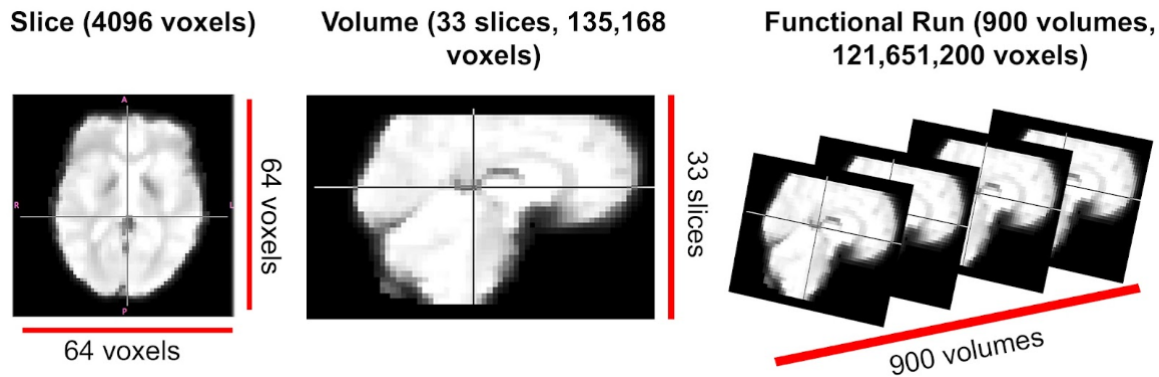


Fig. 4: A functional run is composed of successive volumes, each of which contains slices. It is intuitive to think of functional data like a 3D video recording of the brain.

### Other Scans: Shimming and Field Map

Under optimal conditions, a scanner would have an entirely uniform magnetic field. Even if this were possible, we unfortunately have to place humans inside of the scanner, which distorts the magnetic field. Two scans are occasionally conducted to characterise and partially correct for this distortion. After the participant enters the scanner, all inhomogeneities can be corrected via a process known as *shimming*. Often however, these distortions will eventually reappear and it is impractical and time consuming to keep repeating this process. As a result, a *field map scan* is often collected to characterise the inhomogeneity in the magnetic field, so that it can later be corrected in data preprocessing.

### Preprocessing Steps

As described above, the canonical shape of the hemodynamic response function (HRF) is important for the statistical models used to analyze fMRI data. However, as a result of the way functional MRI scans are acquired, a number of issues can interfere with the normal shape of the HRF. The data therefore need to be *preprocessed* to attempt to approximately 're-create' the HRF before statistical models can be run.

The following steps below are discussed in no particular order and to help with understanding.

#### 1. Slicetiming Correction

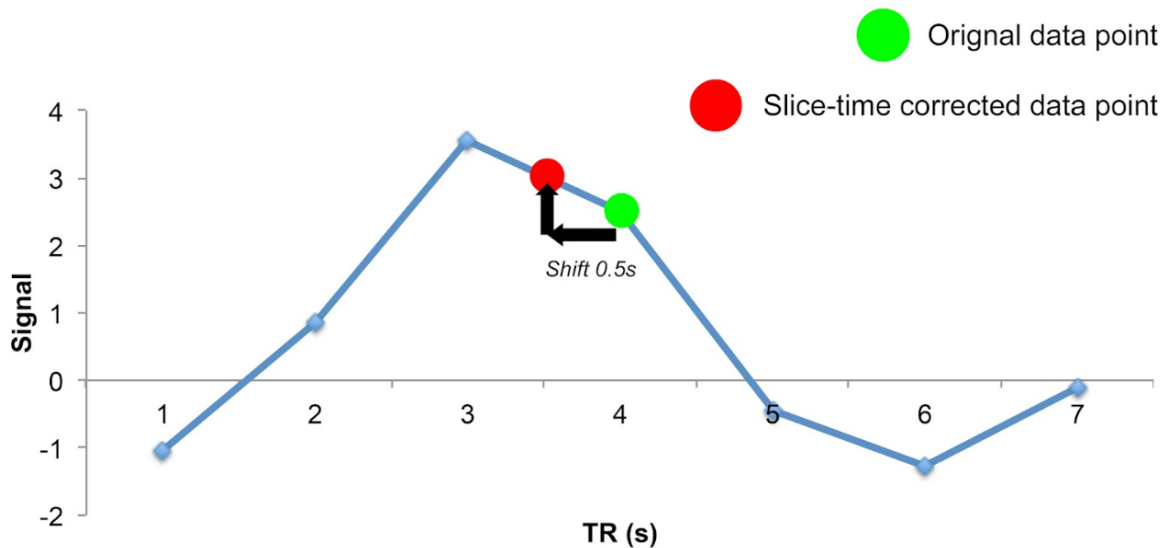
##### Problem

A standard echo planar imaging (EPI) sequence for fMRI data acquisition involves functional volumes being collected in successive 2D slices. For instance, assuming a repetition time (TR) of 1 second and 35 slices being collected in ascending order, this entails that the top slice is collected approximately 0.97 seconds after the bottom slice. Clearly this presents a problem, since we want all data points in our volume to correspond to the same point in time (the timing of a 'reference' slice).

## Solution

Fortunately, the solution to this problem is relatively straightforward. Conceptually, all slice timing algorithms operate in the same way: There is some form of *interpolation* between the data points we did acquire, to allow prediction of the data points we *would have* acquired if the data from all slices were collected at the same time.

For instance, let's suppose that the figure below shows the response (blue points) recorded from a voxel in the top slice in successive volumes (TR = 1s). Let's also suppose that our slices were acquired in an ascending order (from bottom to top). For this data in the top slice to correspond to the timing of the middle slice (our *reference slice*, in this example) we simply interpolate between the data points (blue line), navigate ~ 0.5 seconds back along the x axis and use the interpolation line to give us the response corresponding to the acquisition of the middle slice. Then, we simply repeat this process of interpolation for all slices to give us our completed set of slice-time corrected data.



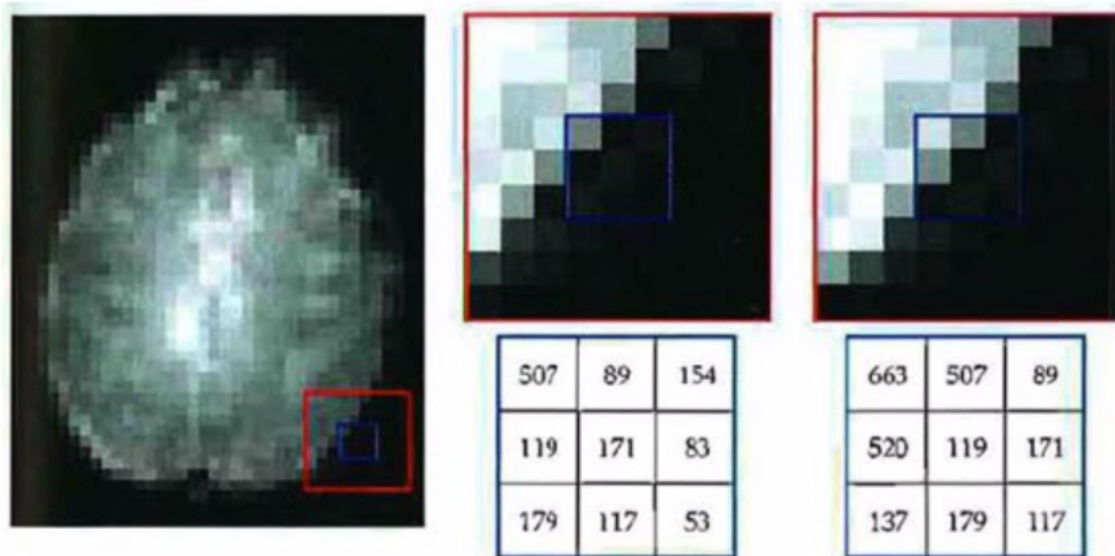
The slice acquisition order can be found in the JSON sidecar associated with the functional run in a BIDS compliant dataset.

## 2. Motion Correction

### Problem

If a subject moves their head during an fMRI experiment (they will), their head will move to different locations within the scanner. This can mean that the same voxel records from a different region of the brain on successive volumes (below). Clearly, we don't want this - even small movements can badly corrupt fMRI data - we want the same voxel to be recording from the same location in the brain throughout the functional run.





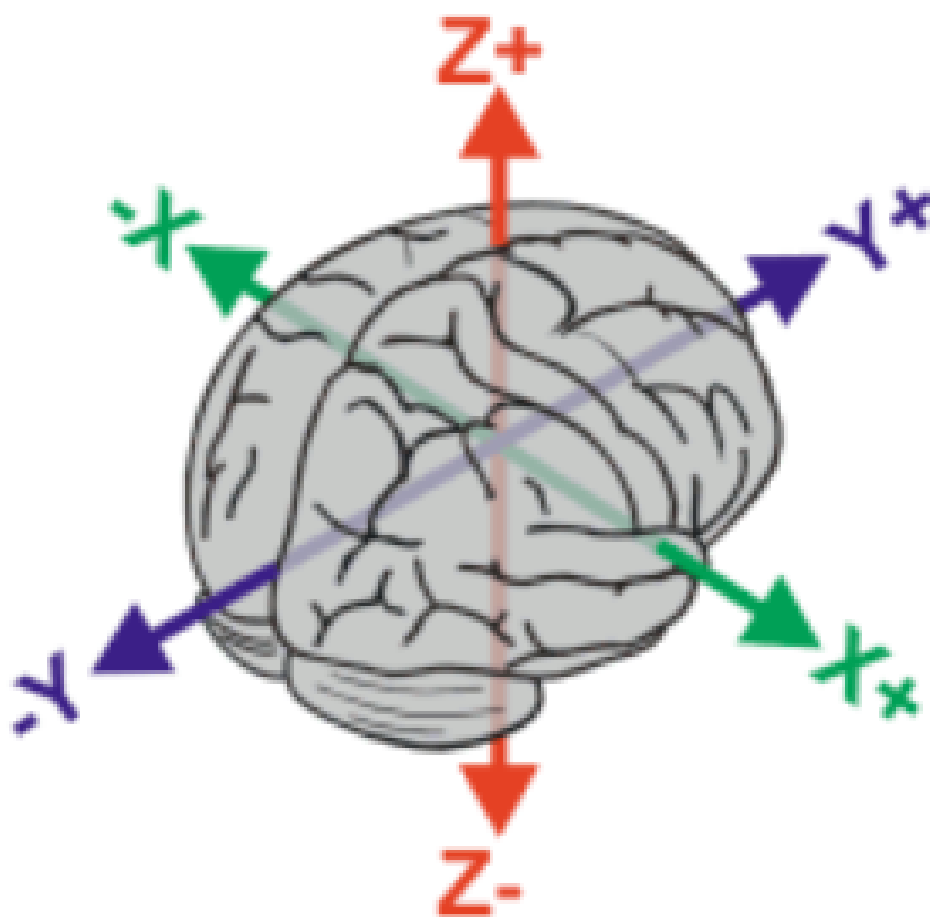
### Solution

The solution to correcting for head movement is quite intuitive. All corrections for motion essentially use modifications of the same algorithm, which involves the minimization of a *least-squares cost function*, or maximization of *mutual information*. The algorithms attempt to find the movements to apply to each volume that minimize the voxel-by-voxel intensity difference from a reference volume (usually the middle volume in the functional run).

Since the head does not change shape or size, it can be treated as a *rigid body*. Any rigid body movement can be described by 6 parameters. 1) A *translation* along the x axis, 2) a *translation* along the y axis, 3) a *translation* along the z axis 4) a *rotation* about the x axis, 5) a *rotation* about the y axis, 6) a *rotation* about the z axis.

Thus, motion correction algorithms perform these rigid body movements until a set of parameters are obtained that ensure each volume agrees as closely as possible with a reference volume (e.g. the middle volume) at each x y and z coordinate. Since it is not viable to characterise all possible ways in which the head could move during a functional run, motion correction is an *iterative* process - an initial rough estimation followed by more precise refinements. Once an optimal set of realignment parameters has been determined, the original data is then resampled to estimate the values that would have been obtained if there had been no head motion. Similar to slice-timing correction this involves interpolation - but in the spatial domain rather than the temporal domain. Unlike temporal interpolation, which only involves 1 dimension, this spatial interpolation involves 3 dimensions. *Tri-linear interpolation* assumes that each interpolated data point should be a weighted average of all adjacent data points.





## FAQ's

### Can motion correction algorithms tolerate very large head movements?

An exact answer is not straightforward, but standard motion correction algorithms can tolerate misalignments that do not exceed 10mm (summed in the x,y and z directions). One beneficial strategy is to examine the plot of the estimated motion parameters for a functional run and determine which volumes are misaligned by more than the size of a voxel. These volumes can then be defined as 'outliers' and excluded from the analysis (known as *scrubbing*). This may, depending on your experimental design, fundamentally distort your trial balancing and compromise your parameter estimates. If this is the case, you should be prepared to exclude subjects data altogether. Head movement is something to be taken very seriously and you should make every effort to ensure subjects do not move their head during data acquisition in the first place.

## 3. Brain extraction

### Problem

Due to the non-invasive nature of the imaging, a considerable amount of non-brain tissue is included in magnetic resonance images. This includes eyeballs, skull tissue, muscle and so on. Removing this redundant tissue can significantly reduce file size and aid the algorithms involved in registration and normalisation, almost all of which assume that non-brain tissue has been removed.

### Solution

Manual brain tissue segmentation is possible, although it can be incredibly time consuming and requires a large degree of training. A popular automated solution ([Brain Extraction Tool](#), implemented in FSL) is based on the following logic. Recall that in a standard T1-weighted image, differences in the water content between tissue are represented as differences in intensity (e.g. white matter = light, cerebrospinal fluid = dark). These regularities can therefore be exploited to segment brain from non-brain tissue based on local differences in image intensity. First, the intensity histogram of the image is obtained, from which 'robust' minimum and maximum values are determined (i.e. the 2nd and 98th percentile of intensity values). Next, an intensity threshold is determined to distinguish between the brain and background, determined by 10% of the way between the 2nd and 98th percentile. This threshold is then used to estimate the centre of the brain tissue. A small tessellated icosahedron (3D shape with 20 sides) is then initialised as a surface model at the brain centre estimate and a set of adaptive forces (described by Smith, 2002) iteratively modulate and deform each vertex so that the brain model moves outward until it reaches the brain surface, as defined by the intensity threshold.

## 4. Susceptibility Distortion Correction

As described above, it is likely that inhomogeneities and susceptibility distortions in the magnetic field will be present during scanning. These distortions can distort the apparent anatomy of the brain. Echo planar imaging (used for fMRI acquisitions) is particularly susceptible to this. The placement and size of these distortions can be mapped through a *field map scan*. The effect these distortions have on the echo planar images can then be estimated, and unwarping algorithms applied.

Fig. 5: Reproduced from the [fMRIPrep documentation](#).

## 5. Registration and Normalisation

### Problems

1. **Registration.** A first problem that we are presented with is that functional data is very low resolution compared to our structural data. As compared to structural data, functional data can appear something of an indistinct blob, particularly after spatial smoothing. From functional data alone, it may be difficult to tell whether task related activation is located in one brain region, or an adjacent brain region.
2. **Normalisation.** How can we determine common regions of neural activation between subjects? Moreover, how can we report the loci of neural activation in a standardised format? These tasks would be trivial if all human brains had identical size and structure. However, unfortunately, human brains vary massively in shape and size and these differences are present in virtually every region of the brain. These drastic anatomical differences make it impossible to know whether clusters of activation are in equivalent brain regions in every subject.

### Solutions

#### Registration

We can register our functional data to our high-resolution structural data to improve spatial localisation of task-related activation. Intuitively, it may seem as though this could be achieved via a similar method to that employed in motion correction. The critical differences though, are that

1. The resolutions of the structural and functional images are different
2. The structural and functional volumes are run with different imaging parameters and their contrasts are different (namely T1 and T2\* contrasts).

The most common current solution to this problem is as follows. A first step is to create an intensity histogram of each image. Next, the intensity value of each voxel is replaced with the bin number to which that intensity value was assigned in each histogram - thereby associating each voxel with two bin numbers - one from each image. Finally, a series of linear transformations are conducted on the functional volume until the association between the histogram bins of the voxels is maximised. The method works because if the images are aligned, you would observe such an association regardless of the contrasts of the images, or their resolutions.

#### Normalisation

We need to apply non-linear transformations to alter the shapes of our subjects brains, so they are the same size and shape as a *standard brain*, such as the MNI brain. The concept of normalisation may be familiar to anyone that has watched a video of ‘morphing’ software that can transform one person’s face into another. Normalisation is considerably more complex than registration. This is because registration can be achieved by simple linear transformations. By contrast, a number of nonlinear transformations are required to alter the shape of a subjects brain to match a standard brain. Although there are many different normalisation algorithms, they usually begin with an initial linear transformation that aligns a subject’s brain to a standard as closely as possible (essentially, this is achieved by the same process detailed in the <guide-for-newbs:Registration> section). Then, the differences that remain between the two images are reduced by local stretching, shrinking and warping.

For instance, suppose that a subject had a particularly large visual cortex and a particularly small cerebellum. To normalise, we would need to shrink the visual cortex and stretch the cerebellum. This kind of operation is simply not possible with a linear transformation. Since normalisation methods perform a number of transformations in each small brain region, the algorithms often have millions of parameters.

## FAQ's

### What if my subject's brain is non-normal?

Many interesting fMRI experiments are conducted on children, or special adult populations. There is a large amount of literature available on conducting normalisation on lesioned brains (Brett et al, 2001), or child brains (Wilke et al., 2003) that you can consult.

### How do I check normalisation?

Pipelines like fMRIPrep will provide reports that you can check. Manual checking involves careful inspection in FSLeyes or similar application. Overlay the normalised image on top of the MNI 152 template. It is useful to check that 'landmark' anatomical features (e.g. the cerebellum, corpus collosum, grey and white matter boundaries) are closely overlapping. It is also informative to compare the normalised data and un-normalised data in terms of their similarity to the standard. This way, you can make more effective decisions about the appropriateness of the local warping that has occurred.

### What if I want to register a partial-brain functional volume?

This shouldn't pose too much of a problem. One can employ *Boundary Based Registration (BBR)* for registration of the functional volume to the structural data (Grieve et al, 2009). This method first depends on segmenting white and grey matter from the structural image. Next, the white matter is linearly registered to the functional volume using the histogram-based method we discussed before. Samples of the intensity of the functional volume are then taken from either side of the white matter boundary. The difference in these intensities is then used to create a function that is maximised by further transformations. In other words, this method exploits the reliable intensity differences between white and grey matter found in EPI images to guide registration. This method has been found to achieve excellent results for the registration of partial brain volumes. This method is used by default in fMRIPrep.

## 6. Spatial Smoothing

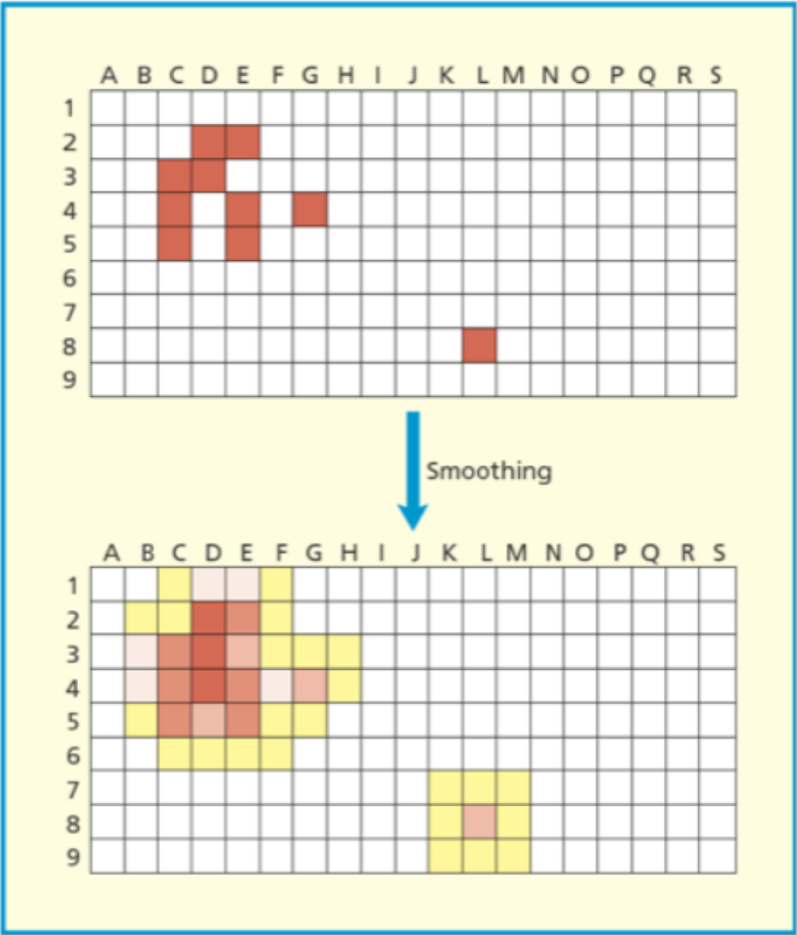
### Problem

Functional MRI data are noisy. This is partly due to factors that can be corrected to a large extent (e.g. head movement, magnetic inhomogeneities) and partly due to factors that cannot even be measured (thermal noise, system noise). One obvious negative consequence of this substantial noise is a decreased signal to noise ratio, which means that task-related effects can be hard to detect.

### Solution

Smoothing is based on the following assumptions: - Firstly, there is a good chance that task-related activity in an isolated voxel reflects noise, and that task-related activity in a group of voxels reflects a genuine signal. - Secondly, spatial noise is mostly Gaussian - mostly random, independent from voxel to voxel and has a mean of zero. Therefore, the average of noise is zero and the average of signal is non-zero.

By smoothing the data, we essentially spread the raw activation of a given voxel onto neighbouring voxels, using a Gaussian smoothing kernel. This has the effect of penalising the influence of what is probably noise, whilst increasing the influence of what is probably signal - in other words, it reduces the *false-positive rate*. By way of illustration, in the figure below (upper panel) there are two regions of activation. Since most task-related neural activity tends to take place in clusters rather than isolated voxels (though see caveats below), there is a good chance that the left region reflects signal and the right region reflects noise. Lets consider voxel D4 in the left region. Prior to smoothing it was inactive, but has become 'active' as a result of smoothing due to the spreading of activation from its neighbours. In contrast, voxel L8 was initially highly active, but its activity has been reduced since it has inactive neighbours. Thus, the influence of signal has been increased and the influence of noise has been diminished.



As well as enhancing the signal to noise ratio, smoothing increases the spatial extent of active regions. This increases the chances of finding common regions of activation when averaging across subjects.

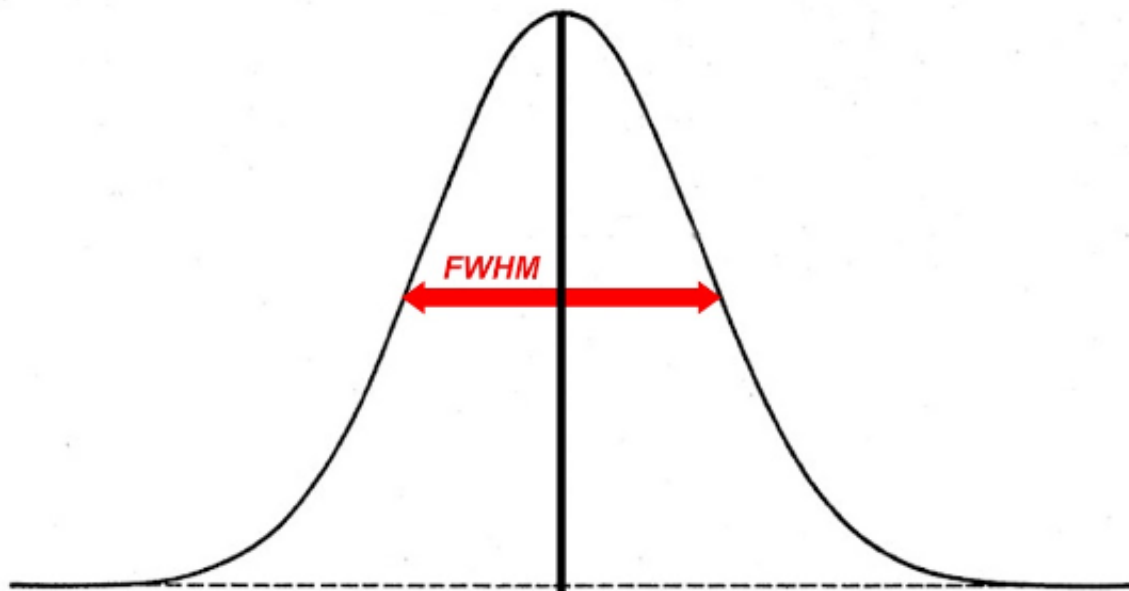
## FAQ's

### Under what circumstances should I smooth data?

Compared to other pre-processing stages, smoothing is relatively controversial (for this reason, smoothing is not included in fMRIPrep) and the FAQ's are associated with less precise answers. There are many circumstances where smoothing would be highly inappropriate. Smoothing is essentially 'blurring' the data and so if we are concerned with voxel by voxel resolution (for instance, in retinotopic mapping, or in multivoxel pattern analysis) we wouldn't want to smooth the data. Moreover, if you have reason to believe that your activations of interest will only extend a little more than 1 voxel, then smoothing would also be inappropriate. It is also the case that the supposed benefit of increasing the spatial extent of active regions may not be desirable in cases where individual differences are the focus of the study. In such cases, you could justifiably not smooth the data.

### How do you decide on the size of your smoothing kernel?

In fMRI analysis, the size of a Gaussian smoothing kernel is defined in terms of its *full width at half maximum (FWHM)* - as illustrated below. We know from matched filter theorem that we can best recover a signal from noisy data by employing a filter the same size and shape as the signal. In the context of fMRI, this would entail using a Gaussian kernel that is as close as possible in size to the region of activation. The problem is that, in most cases, we can only make vague assumptions about the size of active regions prior to an experiment. Even if we suspect a particular structure will be active (e.g. the amygdala) we may only obtain activity in a sub-region of the structure, not the whole structure.



We can at least apply a heuristic in deciding the size of a kernel. A kernel that is larger than the signal will be associated with more negative consequences than a kernel that is smaller than the signal. An oversized kernel will result in regions that contain noise being ‘averaged in’, thereby reducing the signal. Additionally, it is possible that two distinct signals can be fused together if a kernel is oversized. Skurdalski et al (1999) have determined that a kernel of 1-2 voxels worked best overall in the presence of unknown activation sizes.

## 7. Temporal Filtering

### Problem

Functional MRI data is characterised by noise in the temporal dimension. This noise is mostly concentrated in the low frequency range. Likely causes of these low frequency components are fluctuations in the magnetic field properties of the scanner (known as scanner drift), or generic changes in basal metabolism. In event-related designs, these changes are generally too slow to correspond to task-related neural activation and can distort the signal we are interested in (which tends to be much higher frequency).

### Solution

The most common solution to removing these low-frequency components is to apply a *high-pass temporal filter*. This results in low-frequency components being removed from the data and only the relevant, higher frequency components of the signal being preserved. However, this is controversial and would be inappropriate in certain designs. For example, in resting-state scans, the low-frequency components are *of interest* and so removing these will remove the signal we are interested in. For these reasons, temporal filtering is not applied in fMRIPrep.

### FAQ's

#### How do I determine a cut-off for a high pass filter?

This has no simple answer, but the application of the filter should ensure that task-related activation is not altered by the filtering process. The goal is to filter out as much low frequency noise as possible without altering anything corresponding to the ‘fundamental frequency’ of your experimental design. For instance, a typical hemodynamic response function (HRF) in response to a 500ms visual stimulus rises to a peak after about 6 seconds and decays back to baseline after around 15 seconds. In practice, the default strategy of many software packages is to only filter out frequencies with a period of ~ 120 seconds or longer, which will generally not alter signals relating to this HRF in an event related design. By contrast, in blocked designs, we may alternate between viewing one visual stimulus for 2 minutes and viewing a blank screen for 2 minutes, meaning that we can expect periodic changes in task-related activation that repeat every 240 seconds. In this case, the 120 second high pass filter would eliminate much of the signal that we would want to detect. Therefore, a common strategy is to apply the cut off at *twice the period at which the task related activation is expected to repeat*. Applying this rule will produce a cut off period of 480 seconds for this blocked design.

#### What about low pass filtering?

Low pass filtering is much more controversial than high pass filtering. The essence of the problem is that noise is not independent from volume to volume in fMRI and so smoothing the data in the temporal dimension can actually enhance the noise relative to the signal. Given i) mounting evidence that low pass filtering is counterproductive (Della Maggiore et al, 2002) and ii) the fact that low pass filtering is no longer a pre-processing option for many dominant software packages (SPM, AFNI, Brainvoyager) we will not discuss it any further.

## 9.2.2 Running fMRIPrep

Contributors: Nathan TM Huneke

Maintainers: Nathan TM Huneke

---

**Danger:** You should pre-process all participants in a study using the **same** version of fMRIPrep!

fMRIPrep is best run as a docker image and using the fmripred-docker wrapper.

If you are using my *neuroimaging conda environment* then the fmripred-docker wrapper will already be installed.

---

**Note:** If you are using a university machine then you will need to be given permission to use Docker. Please speak to the admin for the machine.

---

### The fMRIPrep-Docker command

**Warning:** Preprocessing each participant with fMRIPrep takes approximately 6-12 hours. Plan your time accordingly.

The fmripred-docker command has many available arguments, but for our purposes most are not relevant. I suggest you use the following:

```
fmripred-docker \
  path/to/bids/directory \
  path/to/output/directory \
  participant \
  --participant-label 01 \
  --user $( id -u ) \
  --skip-bids-validation \
  --fs-license-file /local/software/freesurfer/license.txt
```

There are a few arguments to explain here:

- replace path/to/bids/directory with the actual path to your BIDS directory
- replace path/to/output/directory with the path you would like for your outputs
- participant means do pre-processing at the participant as opposed to group level
- --participant-label is the subject ID (i.e. the label after sub-)
- --user defines which user should own the outputs. \$( id -u ) represents your username, so that you will own all the outputs. By default fmripred-docker will run as root which is unhelpful on a shared machine
- --skip-bids-validation prevents an error when using datalad datasets
- --fs-license-file specifies the location of the freesurfer license file, which is needed for surface reconstruction. The location above is correct for the university linux machines and does not need to be changed



## Inspecting the Outputs

The outputs of fMRIPrep will be saved in your output directory in two directories: `fmriprep` and `freesurfer`. Outputs are structured as follows within the `fmriprep` directory:

```
fmriprep/
  dataset_description.json
  desc-aparcaseg_dseg.tsv
  desc-aseg_dseg.tsv
  logs/
  sub-01.html
  sub-01/
    anat/
    figures/
    log/
    ses-01/
      anat/
      func/
        sub-01_ses-01_task-faces_desc-confounds_timeseries.json
        sub-01_ses-01_task-faces_desc-confounds_timeseries.tsv
        sub-01_ses-01_task-faces_from-scanner_to-T1w_mode-image_xfm.txt
        sub-01_ses-01_task-faces_from-T1w_to-scanner_mode-image_xfm.txt
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_boldref.nii.gz
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-aparcaseg_dseg.
        ↪nii.gz
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-aseg_dseg.nii.gz
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-brain_mask.nii.gz
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-brain_mask.json
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-preproc_bold.nii.
        ↪gz
        sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-preproc_bold.json
      ses-02/
      sub-02.html
      sub-02/
      sub-03.html
      sub-03/
      ...
    etc.
```

Your preprocessed BOLD data is `sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-preproc_bold.nii.gz`. Brain extraction is not performed on these data, but a brain mask (`sub-01_ses-01_task-faces_space-MNI152NLin2009cAsym_desc-brain_mask.nii.gz`) is created and can be used later.

fMRIPrep produces a very useful html report for each participant in the top level directory called `sub-id.html`. You should inspect this report to see whether there have been any errors and whether the preprocessing has been performed well. A guide for exploring these reports and interpreting the images is available [here](#).

## 9.3 fMRI Analysis with FSL

Contributors: Nathan TM Huneke, Mobeen Ali, Yukai Zou

Maintainers: Nathan TM Huneke

---

**Note:** Under construction

---

### 9.3.1 FEAT

#### Launch FSL FEAT GUI

In the terminal, type `fs1 &` to launch FSL, and click on the “FEAT FMRI Analysis” icon to launch the FEAT GUI.

---

**Tip:** Navigating to the data folder first

It is recommended that the command is called after navigating to the directory where the data is located, so that loading the image data will be easier.

---

---

**Tip:** Avoiding spaces in file and folder names

Avoiding having spaces in the file and folder names. This is considered a good practice in general when operating in Unix/Linux environment. Consider replacing spaces with underscores or dashes.

---

#### (Optional) Load FEAT Configuration

If you have a FEAT configuration file (`.fsf`) from a previous analysis, you can load it by clicking on the **Load** button. This will import all the settings from the configuration file, and you’ll be able to modify from there.

#### Slice timing correction

Click on the “Pre-stats” tab, and select “Slice timing correction” from the drop-down menu. Choose “Use slice timings file”, and select the file that corresponds to the input data. The file should be a text file with one number per line, corresponding to the slice acquisition order.

---

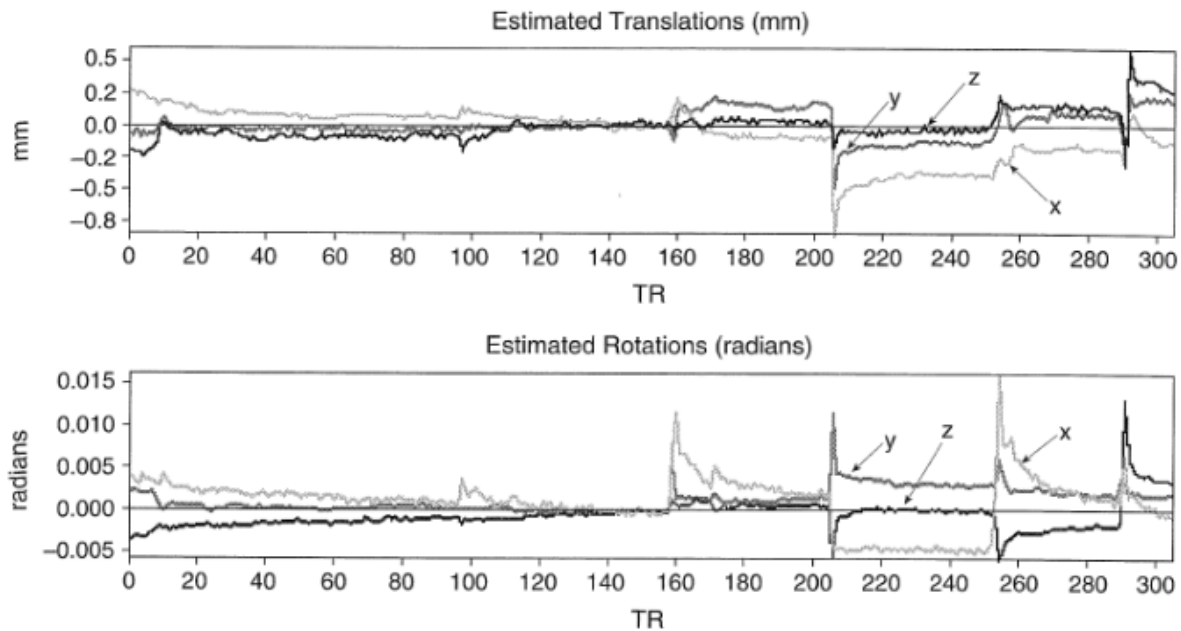
**Important:** Check whether your DICOM data contains slice timing information

It is important to check with radiographer whether the DICOM data contains slice timing information. Data collected UHS MRI scanners undergo anonymisation process, which can remove slice timing information before the data is transferred to the user. Typically, slice timing information can be preserved when the DICOM images are burnt onto a DVD from the scanners.

---

## Head motion correction

Under “Pre-stats” tab, select “MCFLIRT” from the drop-down menu of motion correction. This will enable motion correction.



**Note:** Motion correction is not always necessary

While head motion correction is critical, it is not always a mandatory preprocessing step. For instance, motion correction is not accurate on edge slices, where the cerebrospinal fluid (CSF) flows in and out of the imaging volume. Performing motion correction in this case can actually lead to inaccurate voxel's slice position. See [Fultz et al. 2019](#) for details.

## Coregistration

Click on the “Registration” tab, and under “Main structural image”, select the individual’s T1-weighted brain image.

**Tip:** Perform brain extraction on the T1-weighted image beforehand

Registration may be affected by large field of view (FOV) of the T1-weighted image and the skull. Therefore, performing brain extraction on the T1-weighted image beforehand can improve the registration accuracy. Even though BET brain extraction can be enabled in FEAT, it's good idea to provide a brain-extracted T1-weighted image directly if possible.

## Normalisation

You'll need to choose brain atlas for this step. In FSL there are a range of standard brain atlases available, in which we will be using the MNI152 T1 2mm atlas (`MNI152_T1_2mm_brain.nii.gz`). This is a commonly used standard brain in neuroimaging research.

---

**Tip:** How to find the path of FSL installation

In a terminal, type `whereis fsl`. This will return the path of the FSL installation.

---

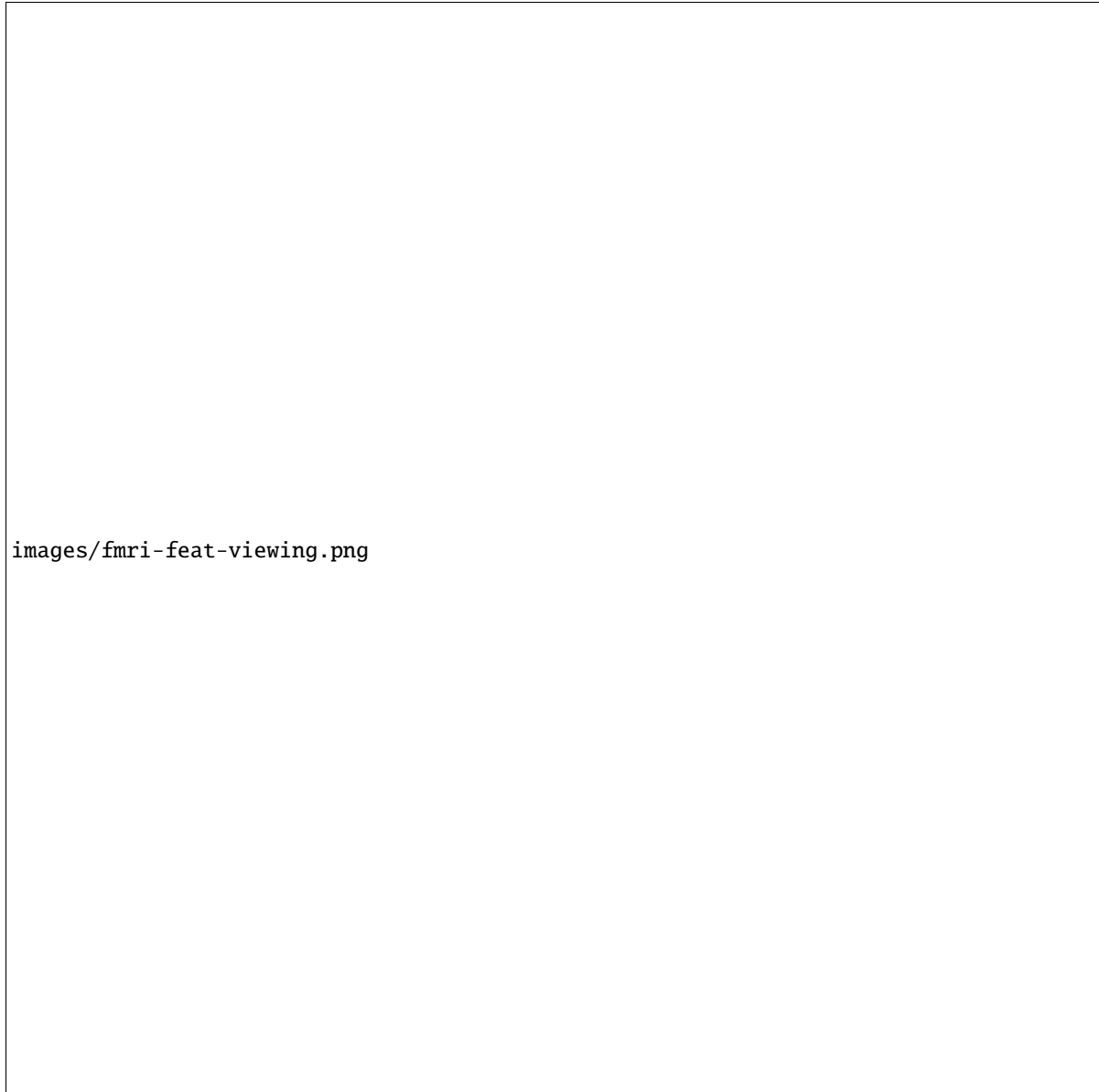
## Run FEAT

You are now ready to run FEAT. Press **Go** to start the analysis. FEAT should normally take a few minutes to complete.

### 9.3.2 Viewing FEAT Analyses Results

In a file browser (e.g. Windows File Explorer), navigate to the output folder of the FEAT analysis. This folder should end with the suffix `.feat`. Inside this folder, there should be a file called `report.html`. Double-clicking on this file, a web browser should open and display the status and results of the FEAT analysis, which is a convenient way to view the results.

While FEAT analysis is still running, it will display “STILL RUNNING” in the main page. When the analysis is complete, open FSLeyes and load the preprocessed fMRI image (`filtered_func_data`) from the `.feat` folder.



images/fmri-feat-viewing.png

### FSLEyes: a brief intro

FSLEyes is the image viewer released with FSL version 5.0.10. FSLEyes does not perform any processing or analysis on images.

A complete user guide of FSLEyes can be found [here](#). FSLEyes has the following features:

- Orthographic (3 orthogonal slicings) and lightbox (multiple slices) views
- Multiple simultaneous views (orthographic and/or lightbox)
- Timeseries display (via graphs or movie loops)
- Multiple semi-transparent colour-overlays
- Simple freehand image editing (mask drawing)
- 3D rendering

Starting FSLeves

Basic image viewing

Unlinking Cursors

Viewing multiple images

Viewing Timeseries (4D images)

Viewing Atlases

FSLeves - 3D mode

### 9.3.3 Brain Extraction Tool (BET)

Accurate brain extraction is crucial for carrying out structural analysis that involves segmentation. In FSL, it is straightforward to perform brain extraction by running BET, but obtaining accurate results will involve some skill and diligence.

For command-line version, you can type `bet` to learn about the usage description:

```
bet <input> <output> [options]
```

where `input` and `output` stand for filenames, and `options` can be many, or none, of the available extra options.

#### Varying the fractional intensity threshold parameter (-f)

The `-f` option in `bet` controls the fractional intensity threshold that distinguishes brain from non-brain. By default the value is set on 0.5, and when it is smaller, the brain estimate gets larger. In command line, try setting the `-f` option from 0.2 to 0.8, in turn, to see the effect it has. Save these outputs with different names and load them into FSLeves.

#### Troubleshooting Brain Extraction

The section describes some of the more problematic brain extraction cases, which are common with images that have large FOV and/or substantial bias field.

Using the gradient threshold option (-g)

Dealing with large FOV

1. crop image first to remove the neck
2. provide an estimate of the centre of the brain
3. use other BET options that might be more robust

### 9.3.4 (Optional) FSLUTILS

**fslinfo** and **fslhd**

**fslstats**

**fslmaths**

**fslmaths** is a very general image calculator and can be used to perform a variety of manipulations of images.

As an example, here we have extracted two images from a functional dataset, `image0` and `image1`. We'd like to calculate the difference between two consecutive timepoint images, which may be used as part of a quality assessment. We will call the output `imdiff`. To do so, run the following command:

```
fslmaths image0 -sub image1 imdiff
```

and view the output (*imdiff*).

Now, to calculate this as a percent difference image, run the following command:

```
fslmaths imdiff -div image0 -mul 100 imdiffpercent
```

which will first take the difference image (`imdiff`), divide by the first of the original images (`image0`), multiply by 100, and output `imdiffpercent`. View the output, and run **fslstats**.

**fslsplit** and **fslmerge**

**fslroi**

## 9.4 Functional Connectivity Analysis with CONN

Contributors: Nathan TM Huneke

Maintainers: Nathan TM Huneke

---

Under construction





## DMRI ANALYSIS

Contributors: Yukai Zou, Ikbeom Jang

Maintainers: Yukai Zou

---

**Warning:** Numerous sequences and parameters exist for diffusion MRI (dMRI) data. As a result, people often end up using different software and everyone has different preprocessing pipelines. The processing steps also vary depending on how your data were acquired.

---

**Note:** Under construction

---

This instruction focuses on the preprocessing and analysis of diffusion tensor imaging (DTI) processing using FSL libraries.

Here are some free educational materials that can help understand DTI:

- **Book:** <https://www.sciencedirect.com/book/9780123964601/diffusion-mri>
- <http://www.diffusion-imaging.com/2013/01/relation-between-neural-microstructure.html>
- <http://www.diffusion-imaging.com/2014/04/from-diffusion-weighted-images-to.html>
- <http://www.diffusion-imaging.com/2015/10/what-is-diffusion-tensor.html>
- [http://clinical-mri.com/wp-content/uploads/software\\_hardware\\_updates/Graessner.pdf](http://clinical-mri.com/wp-content/uploads/software_hardware_updates/Graessner.pdf)
- FSL course on YouTube
- DIPY 2021 workshop
- MRtrix3 B.A.T.M.A.N. on Open Science Framework
- ITK

## 10.1 DTI Preprocessing with FSL

Contributors: Yukai Zou, Ikbeom Jang

Maintainers: Yukai Zou

---

**Note:** Under construction

---

### 10.1.1 Download and Organize Data

Scripts to download and organize PNG's MRI data (e.g. DTI) can be accessed at <https://github.com/jibikbam/Download-and-organize-MRI-data-in-bash>.

### 10.1.2 Preprocess Data

Scripts to preprocess DTI can be accessed at <https://github.com/jibikbam/DTI-process-analysis>. You may use them if you'd like to try the easiest path.

- Make sure you have `acqparams.txt`, `index.txt`, and DICOM files. The number of values in `index.txt` should match the number of diffusion encoding directions of the data you use.
- Run the processing script with the command `bash process_DTI.sh`.
- The eddy current correction requires the longest time, therefore, it is recommended to process in parallel by using `eddy_openmp` (multi CPU core) or `eddy_cuda9.1` (GPU). Download one depending on the hardware resource available.
- DTI outputs should be ready including FA, MD, L1, L2, L3 images.
- To visualize output images, use FSL's `fsleyes`. You may also use FreeSurfer's `freeview` or AFNI viewer, etc. You may also visualize them in MATLAB or Python using some libraries (see DTI Analysis).

---

**Note:** Some parameters may need to change depending on the data you use. But for now, don't worry about it too much and focus on making it run.

---

---

**Note:** The processing can be improved in several ways. For example, use T1 or T2 for brain extraction rather than using b=0 volume of DTI (i.e. `vol0000.nii.gz`). Use the `topup` option in `eddy` for correcting susceptibility induced distortions if your DTI sequence collected two b=0 volumes with different phase encoding directions.

---

### 10.1.3 Tract-Based Spatial Statistics (TBSS)

---

**Important:** You can stop here if the purpose is to get FA and MD images of each individual. Below are further steps in case you need them.

---

TBSS extracts white matter tracts. It should be straightforward to follow the TBSS steps described in FSL's documentation page [here](#).

- If you just need skeletonized FA co-registered between all subjects, follow steps until “voxelwise statistics on the skeletonised FA data” section.
- If you need other metrics such as MD, AD, or RD, follow until “Using non-FA Images in TBSS”.
- Feel free to do further steps as needed.

## 10.2 DTI Analysis with FSL

Contributors: Yukai Zou, Ikbeom Jang

Maintainers: Yukai Zou

---

**Note:** Under construction

---

To analyze processed DTI images for research, it is recommended to use either of the libraries below:

- (MATLAB) [Tools for NIfTI and ANALYZE image](#)
- (Python) [NiBabel](#)

Feel free to try other libraries available online.



## QSM ANALYSIS

Contributors: Mobeen Ali, Yukai Zou  
Maintainers: Yukai Zou

---

---

**Note:** Under construction

---

### 11.1 Background and Theory

Susceptibility weighted imaging (SWI) is sensitive to both iron and myelin, and therefore SWI is used clinically to provide additional contrast to T1 and T2 weighted images.

#### 11.1.1 Role of iron in the brain

Iron is involved in regulating electrons in metabolic processes, such as the Krebs cycle and oxidative phosphorylation. In addition, iron is also involved in oxygen transport in the brain via haemoglobin.

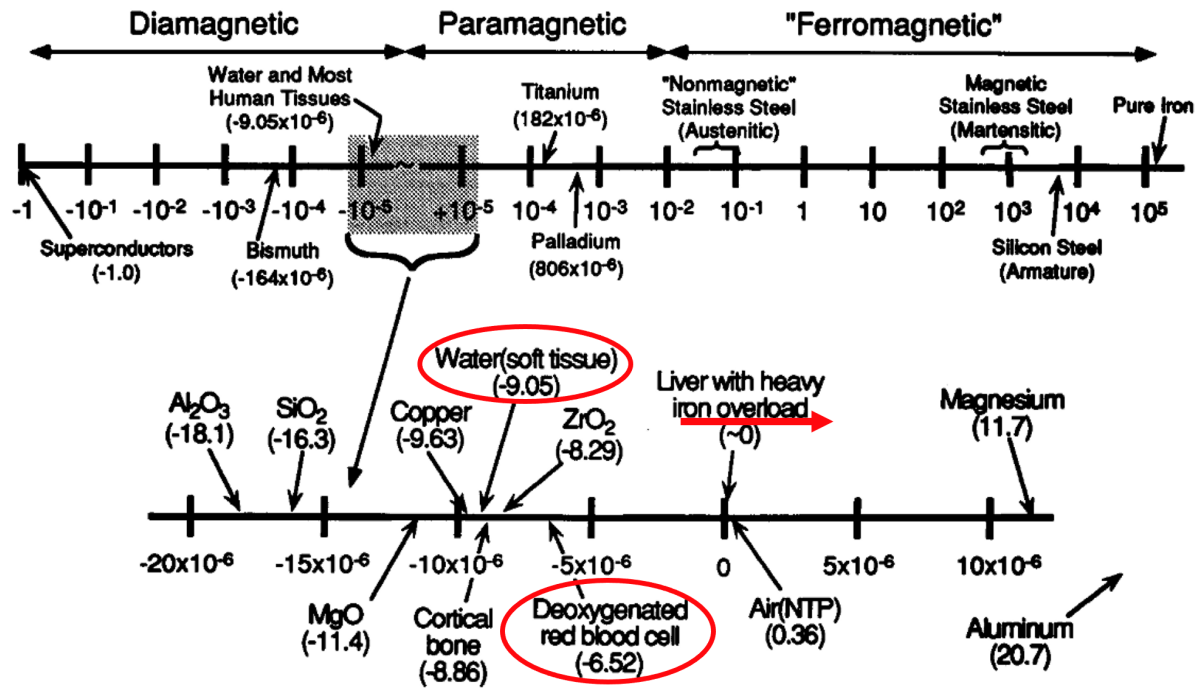
#### 11.1.2 Role of myelin in the brain

Myelin forms a sheath around axons of neurons, which insulates the axons and increases the speed of electrical signal transmission.

#### 11.1.3 Motivations for QSM

Iron accumulates in the brain with age and link to neurodegenerative diseases, such as Alzheimer's disease, Parkinson's disease, and multiple sclerosis.

### 11.1.4 Magnetic Susceptibility



## 11.2 Clinical Applications

### 11.2.1 Calcification

- Question: how to tell difference between iron and calcification in susceptibility weighted images?
- Higher concentrations would exhibit greater enhancement of the region.

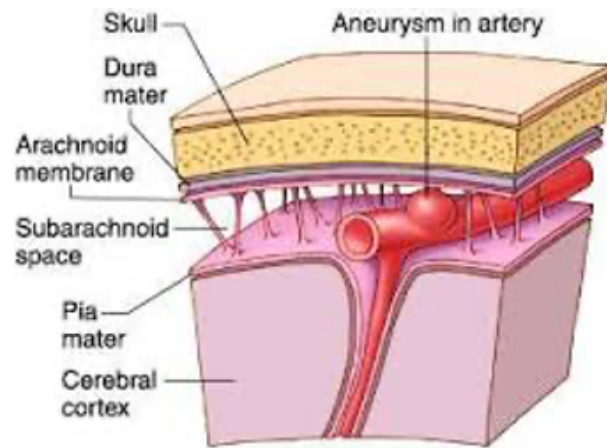
Case study of patient with both calcification and bleeds

- Depends on handedness of scanner, veins and bleeds may appear dark and calcifications may appear bright.
- Need to check for dipole field pattern in sagittal or coronal plane, which will be much more evident around paramagnetic regions, as these have a larger susceptibility difference compared to the diamagnetic tissue.

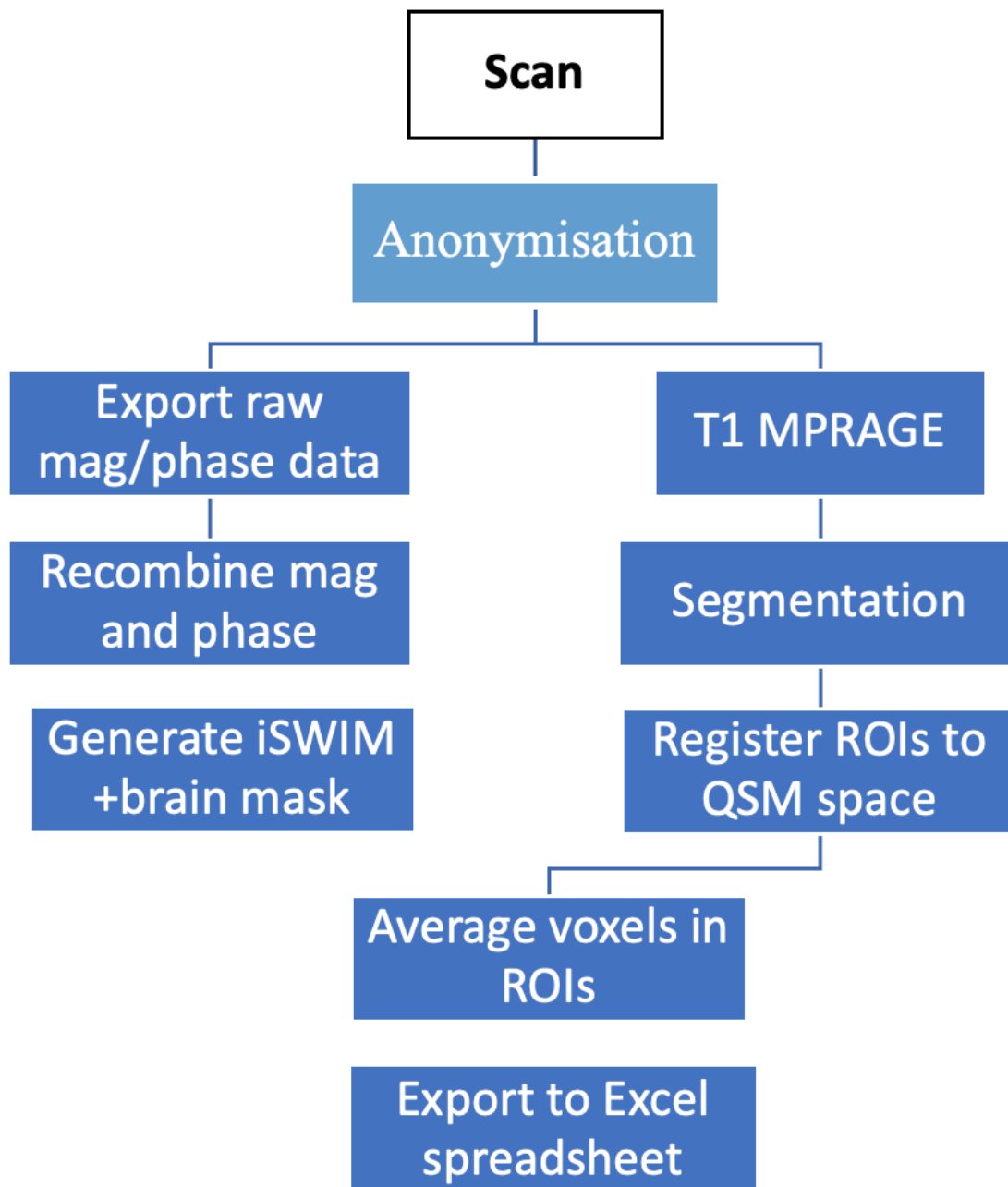
### 11.2.2 Hypoxic Ischemic Encephalopathy

### 11.2.3 Subarachnoid haemorrhage

- Subarachnoid haemorrhage refers to the bleeding in the subarachnoid space, usually from a ruptured aneurysm.



Workflow:



- Iron deposited in the brain after subarachnoid haemorrhage around the Sylvian fissures.
- Need to investigate the relationship of iron and brain volume changes.
- Future work includes studying whether clinical outcome is more closely associated with baseline blood volume or iron deposition.



### 11.2.4 Superficial Siderosis

- Hemosiderin deposited on the pial surface of the brain and/or spinal cord, as a result of chronic subarachnoid haemorrhage.
- Deferiprone is an iron chelator that can be used to treat superficial siderosis.
- There is a need to develop a workflow for rare clinical cases, such as superficial siderosis.

Workflow:

