

Data analysis for Neuroimaging - PSYG4043 / C84DAN

Overview

Denis Schluppeck

What's the plan?

1. Acquire some [functional] MRI data in a simple, but real experiment
2. Analyze the data with `fsl` ([FMRI B webpage](#))
3. Learn a bit about `UNIX` and version control, in particular `git` and `github`
4. Use `Matlab` to inspect and visualise some data
5. [optional] anatomical, diffusion weighted +/- multi-echo data (T2*)

Learning objectives

1. Build a better understanding of principles (and practice of) magnetic resonance imaging
2. Learn about state of the art analysis tools for (f)MRI
3. Appreciate usefulness of scripting, `unix`, version control for reproducibility / transparency of work
4. Discover `matlab` for data visualisation

Timeline

Unit	Topic
1 ★	Introduction, Administrivia, computers, ...
2	Data acquisition (scanning on 3T at SPMIC)
3	Inspecting & analysing data in FSL
4	Version control (<code>git</code> and <code>github.com</code>)
5	Images in Matlab , display, analyze
6	Timeseries signals in Matlab
7	wrapup + Reading/writing text, CSV, data files Matlab

What's the assignment?

A short, written report

Summarise the experimental setup, analysis methodology and results. Need to have clearly written abstract (250w), methods, results and discussions (and **figures**).

Aim: Get you thinking about journal-style writing, rather than essays. Plus: presenting your own data, identifying key points, a story/pitch.

When to work on this?

Start as soon as we have the data

- explore your analysis ideas
- talk to us about questions you could address
- think about plots + data visualisations you'd like to make

Submission details

Currently w/ Student Services, *date to-be-confirmed*

- turn-it-in submission on moodle page
- **deadline: end of March** (the week after last class of this module - check moodle page for details)

What's the assignment (2)

- 250w abstract
- plus a main document (max 1500w)
- references / citations as for standard written work
- **max 5 figures¹** illustrating
 - details of the experimental setup
 - analysis methodology
 - results

¹figures can have sub-panels or subplots

! For next time (lab 2)

- sign up for 1 of 3 groups (max 7 people) - moodle
- complete visitor screening form
- we also need 3 volunteers (~40 min in scanner)

Setting up computers, logins

If you have to do this again on another machine, you can use this shortcut. It's located in `/Volumes/practicals/ds1/`

Cheat: Double-click `Set up My Machine – version 6` icon



Note

Each user (at a particular machine) needs to make sure that `Terminal/shell` is set up correctly – if you move machines between labs (or want to do things elsewhere, repeat the above step!)

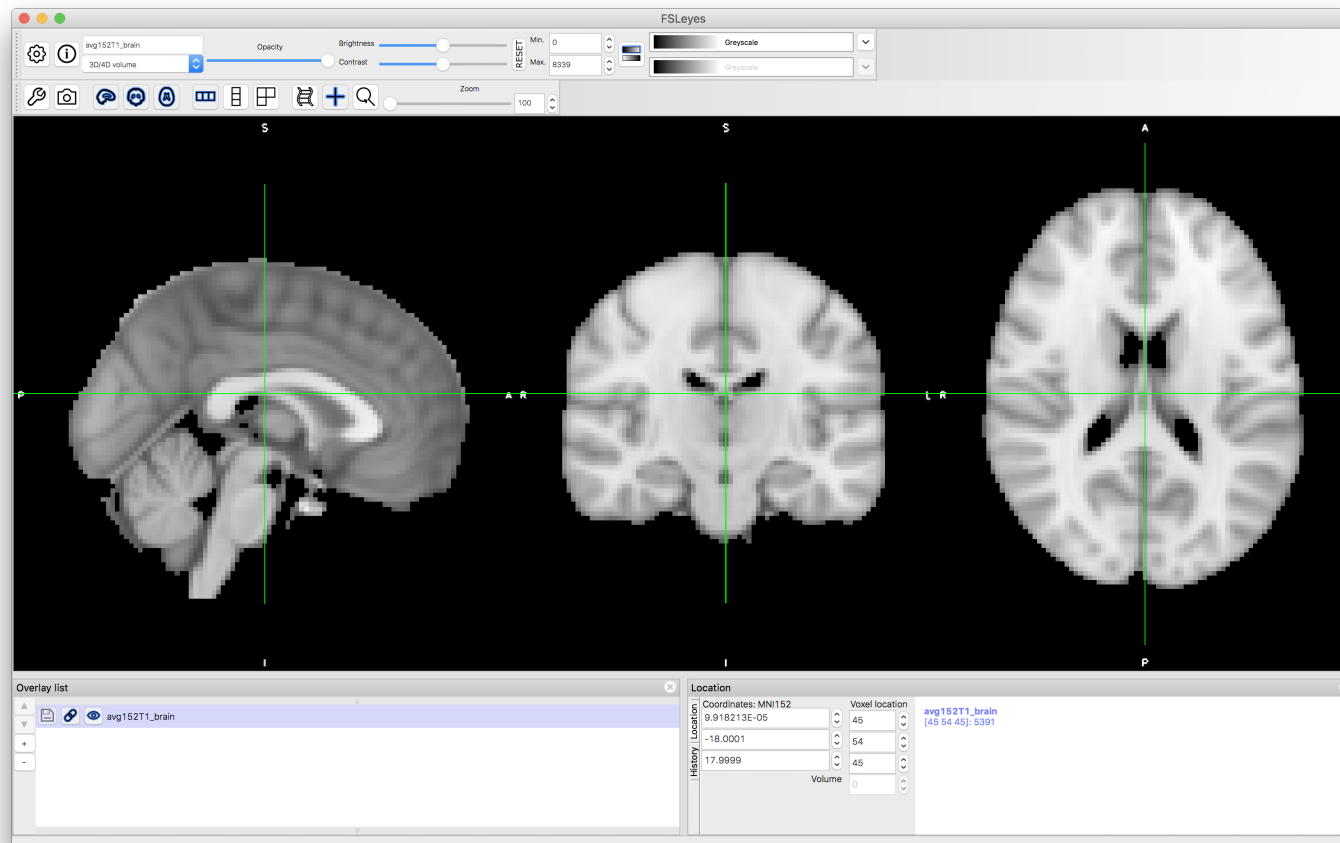
Has setup worked? Reality check.



1. If you see `[ran custom .bash_profile]` in Terminal ✓
2. Also: look at some existing anatomies with `fsleyes` ✓

`which fsl # see anything?`

`fsleyes & # File -> Add Standard -> Pick 1st or 2nd`



FSL analysis

- get data from sessions S001 to S004 into a common folder data
- make folders, copy files by "drag & drop"
- point & click version (like some of you have already done)
- digging into the details of how this is implemented
- inspecting analysis output, intermediate files, ...

```
cd ~/data/S001/ # for example  
# run FSL analysis
```

Some UNIX

- only basics needed for running `fs1`
- lots of functionality is available through point-and-click
- **but** command line is helpful for organising data!
- more complex analysis, e.g. `freesurfer`, require some working knowledge

```
# navigate file system
# cd, ls, pwd, which, ...

# some powerful commands for organising your data
# cp, rm, touch, mkdir, rmdir

# some stuff to show of how powerful
# grep, "lists", "wildcards (*, ., ?)"
# "regular expressions"
```

Version control `git`

- 30min [lecture on principles of version control](#) (`git`)
- start using your (free) `github.com` id by working on a simple project
- make your first modifications to a local copy of code and get it into a repo.

```
mkdir test && cd test # what does this do?
git init
# [[ create, edit a file, say my_first.md ]]
git add my_first.md # add it to "staging area"
git commit # enter commit message
# - OR -
git commit -m 'adds first version of file'
git log
```

Version control (v2.0) 😊

Everyone should sign up for a free `github` account, so we can work together on this from session 4 onwards:

<https://github.com/join>

- it's free and useful
- we'll want to play with this in lab #4
- once you have an username (pick one that I will recognise!), go to our github classroom at

<https://classroom.github.com/a/7ZwbkqLi>

matlab - reading images (1)

- we'll learn/revisit how to read imaging data into matlab (nifti files)

```
% > R2017b  
data = niftiread('file_from_scanner.nii');
```

matlab - reading images (2)

- revisit indexing of arrays, "slicing", etc.

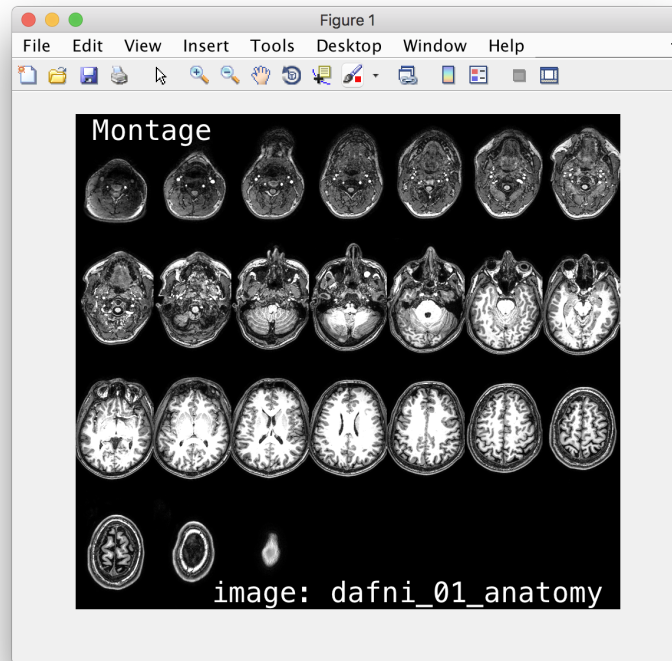
```
data(12, 24, :, 1) % what is this?  
data(:, 24, 24, 1) % ... and this?  
data(34, 44, 12, :) % ... or that?
```

- we built a `returnSlice()` function, to complete imageviewer:

```
% function signature  
s = returnSlice(array, sliceNum, orientation);
```

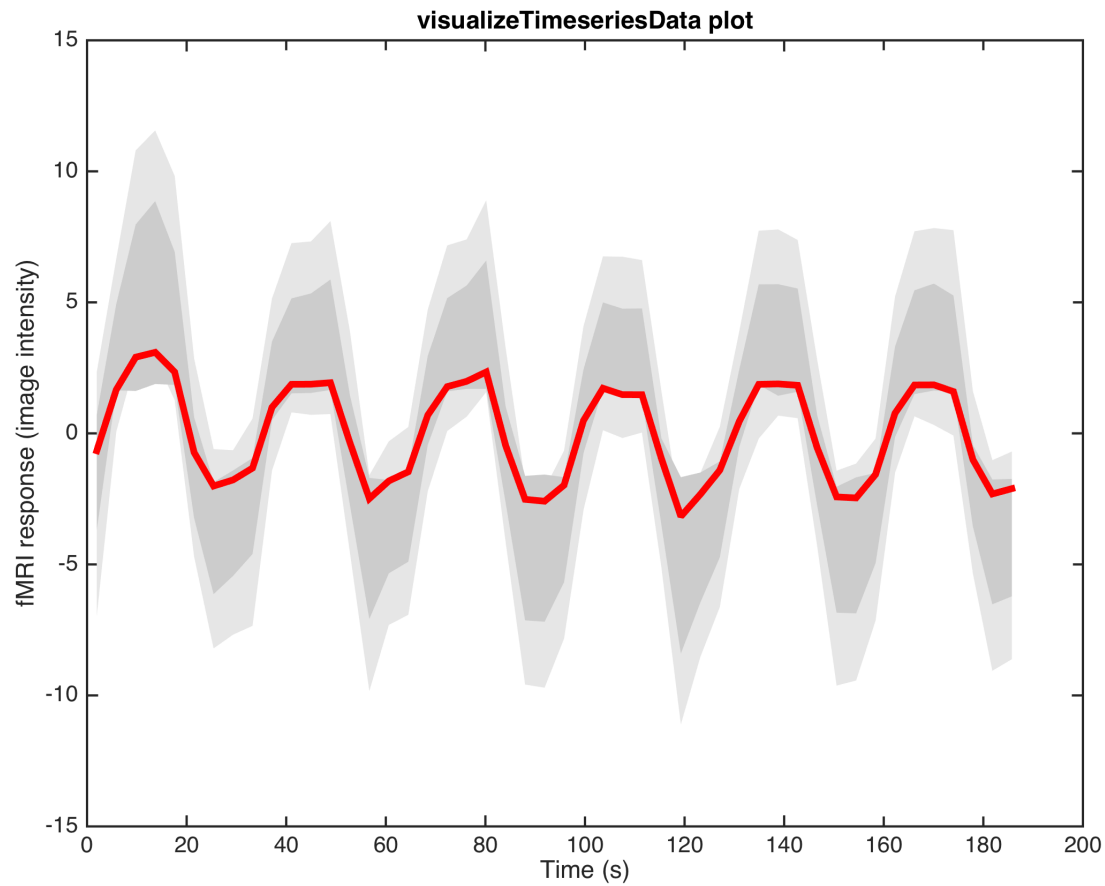
- some more coding along these lines

```
makeMontage('dafni_01_anatomy.nii', 25)
```



```
function [ ] = makeMontage(fname, nSamples)
%makeMontage - make a montage from 3d/4d image
```

matlab - timeseries and subplots



matlab - text / csv / other data

- think about data formats / interop with other analysis & tools (R , python , ... even UNIX tools). Sometimes a text file is best!

```
% read realistic behavioural data (mix of numbers, text)
table(), writetable(), ...

% read in a simple CSV file, skipping first row (r=0)
% csvread(file, R, C) % row R, column C (starting at 0!)
d = csvread('timecourse.csv', 1, 0)
```

Wrap-up (Lab 7)

- recap what have we covered in the last 7 weeks?
- where to go to from here (unleash your inner coding 🐱)
- try to approach each new problem, project with lots of repetition (analysis, writing, coding, ...):
 - there must be a better way!
 - what's the smallest unit that gets repeated all the time?
 - can I use `bash/unix`, `matlab` or another tool to automate?
- just try things out – you'll learn tons in the process

For next time

- make sure you pick a timeslot on **moodle** page.
- make sure you complete screening form (and/or get in touch with any concerns)
- to volunteer for a scan, send me an e-mail:
denis.schluppeck@nottingham.ac.uk
- next week: see you at SPMIC for your timeslot – if you don't know where on campus, google SPMIC ... first hit is imaging centre w/ directions

Random technical stuff

Some notes of manipulating text files in the shell (using `awk`)
and (viz timing files, etc.)

Notes - text hacking in shell

Small `awk` program for adding a counter `n` and time `t` and turn one column `txt` file into `csv` file:

```
awk 'NF {print NR-1 ", " (NR-1)*1.5 ", " $1}' \  
timecourse.txt > timecourse.csv
```

- with a headerline (matlab's `csvread()` doesn't like!)

```
awk 'BEGIN {print "n, t, response"}  
NF      {print NR-1 ", " (NR-1)*1.5 ", " $1}' \  
timecourse.txt > timecourse.csv
```

Solution in `matlab` ?

Turn `timecourse.txt` (column of y-values), into `timecourse.csv` :

- where first column is a counter that goes from `1...n` ,
- the second column is `t` (in s), which goes up from `0..1.5s..` and
- the third column is the `y` values

What about something else?

- `Excel` ? `R` ? Another cool idea that's worth having in your set of tools?