Image Processing with fslr

John Muschelli

Johns Hopkins Bloomberg School of Public Health

April 22, 2015

FSL and fslr

- FSL is a comprehensive library of analysis tools for fMRI, MRI and DTI brain imaging data.
    - Collection of routines in C, C++
- fslr: port of FSL into R
- The two functions we focus on are:
    1. Image inhomogeneity correction (using FAST [2])
    2. Image registration

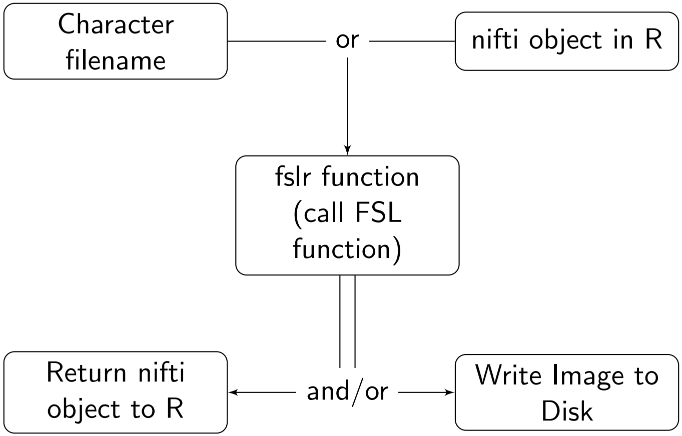
Installing fslr

First, you must Install FSL
http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation.

`fslr` is installed on CRAN, but the development arm of `fslr` is most likely the best to install, using the `devtools` package:

```
if (!require(devtools)){
        install.packages('devtools')
}
devtools::install_github("muschellij2/fslr")
```

Structure of fslr functions



Interactive/GUI vs. Terminal R

In general, GUI-based apps do not inherit the shell environment (aka if `FSLDIR` is defined in your Terminal, RStudio doesn't see it). For fslr to work, it must know where the directory FSL was installed. If `FSLDIR` is found, it will be used. You can check this by 2 ways:

```
Sys.getenv("FSLDIR")
[1] ""
library(fslr)
have.fsl()
[1] FALSE
```

If `have.fsl()= FALSE` then you must specify the path using:

```
options(fsl.path="/my/path/to/fsl")
```

fslmaths: Math with FSL

`fslmaths` (in fslr) calls `fslmaths` from FSL (see `fslr::fslmaths.help()` for help): Let's read an image in using `readNIfTI` from `oro.nifti`:

```
library(oro.nifti)
nim = readNIfTI("Output_3D_File.nii.gz",
reorient=FALSE)
```

fslstats: Stats with FSL

`fslstats` (in fslr) calls `fslstats` from FSL (see `fslr::fslstats.help()` for help):
We can do statistics (e.g. mean) in R and fslr:

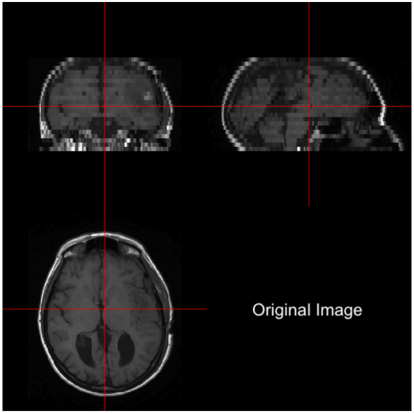
```
mean(nim)
[1] 102.4701
fslstats(nim, opts="-m")
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
[1] "102.470113"
fslstats("Output_3D_File.nii.gz", opts = "-m")
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
[1] "102.470113"
```

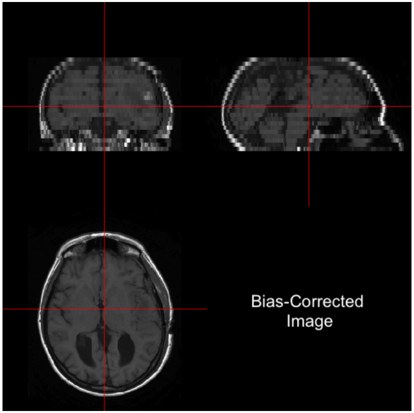
fslr: Bias Field Correction

`fslr::fsl_biascorrect` calls `fast` from FSL which incorporates the bias field correction by Guillemaud and Brady [1]:

```
fast_img = fsl_biascorrect(nim,
        retimg=TRUE)
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

fslr: Bias Field Correction



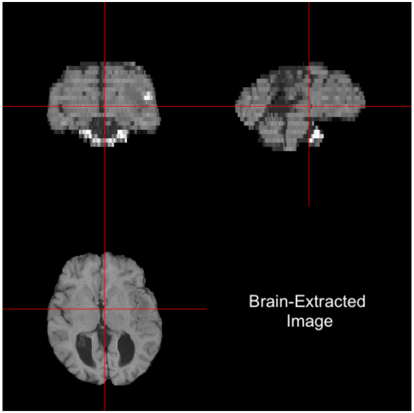


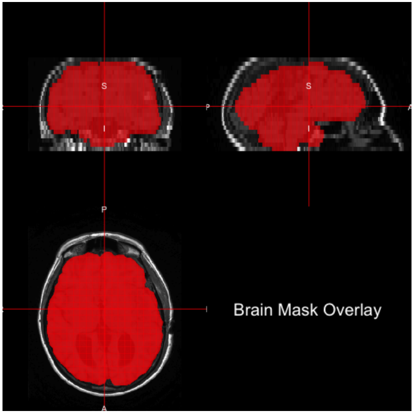
fslr: Brain Extraction

FSL's Brain Extraction Tool (BET) can be used for skull stripping. It is fast, robust, and one of the most popular for this task. `fslr::fslbet` is used to call the FSL commands `bet2`, which does brain extraction or `bet`, which does brain extraction with additional options.

```
bet_fast = fslbet(infile=fast_img, retimg=TRUE)
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

fslr: Brain Extraction Results





fslr: Better Brain Extraction

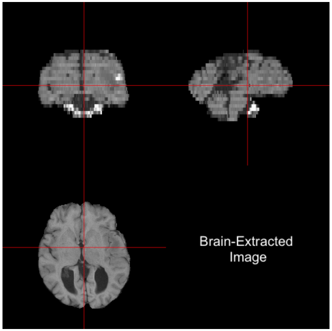
There are some parts of the brain not segmented in the image. We can estimate the center of gravity (COG) from the brain extracted image, and then re-run bet with the new COG to get a better result

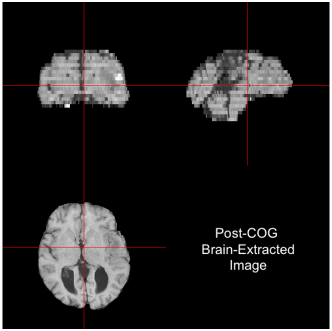
```
cog = cog(bet_fast, ceil=TRUE)
bet_fast2 = fslbet(infile=fast_img, retimg=TRUE,
                   opts =
                     paste("-c", paste(cog, collapse= " "))
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/
```

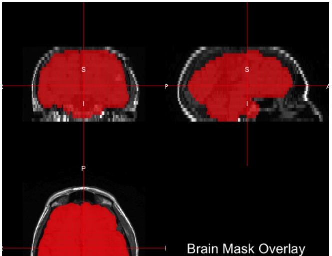
fslr: Better Brain Extraction Results

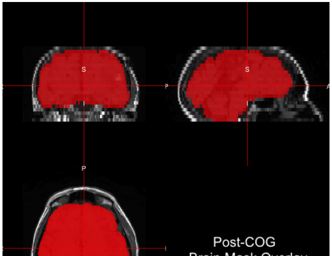
Before COG

After COG







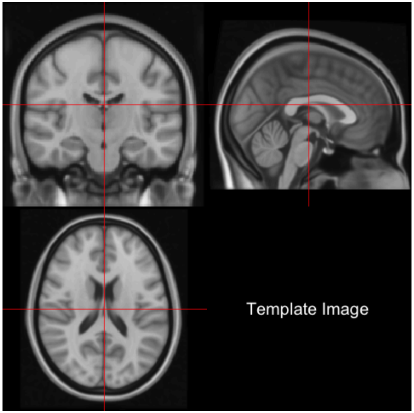


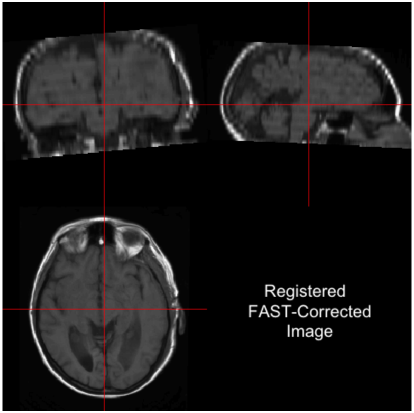
fslr: Image Registration (Linear)

From FSL: "FLIRT (FMRIB's Linear Image Registration Tool) is a fully automated robust and accurate tool for linear (affine) intra- and inter-modal brain image registration"
`fslr::flirt` takes in a input filename (or nifti) and a reference filename (or nifti) to transform the infile to:

```
registered_fast = flirt(infile=fast_img,
        reffile = "MNI152_T1_1mm.nii.gz",
        dof = 6,
        retimg = TRUE)
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

fslr: Image Registration (Linear) Results



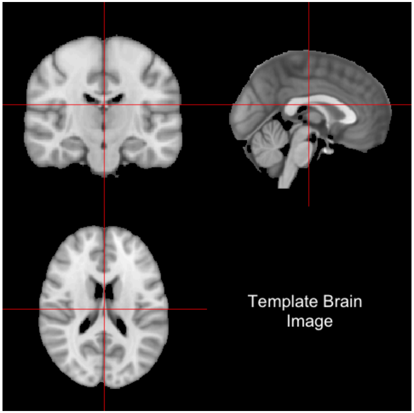


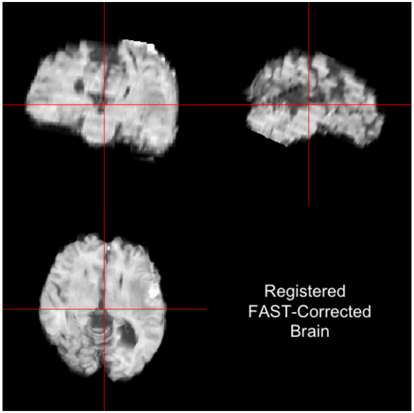
fslr: Image Registration (Linear) Brain

Let's use linear registration with brains only:

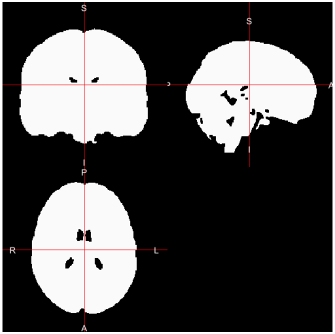
```
registered_fast_brain = flirt(infile=bet_fast2,
  reffile = "MNI152_T1_1mm_brain.nii.gz",
        dof = 6,
        retimg = TRUE)
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

fslr: Image Registration (Linear) Results





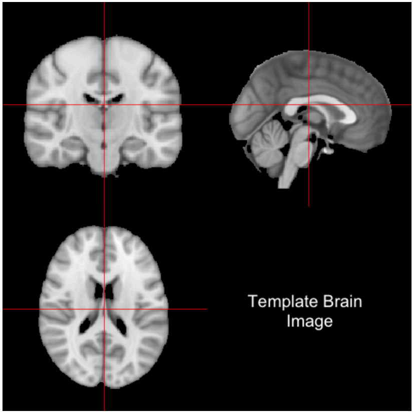
```
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

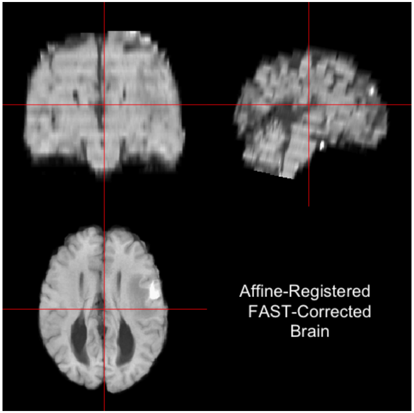


```
RStudioGD
        2
```

fslr: Image Registration (Affine) Results

Instead of
a rigid-body registration, let us try an affine (still linear) registration:



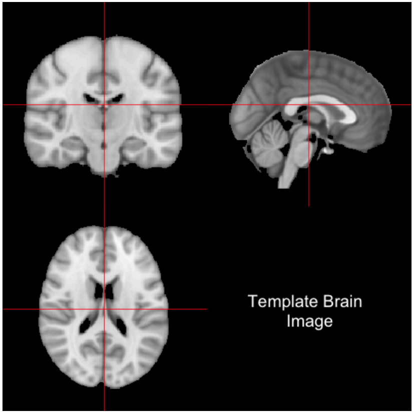


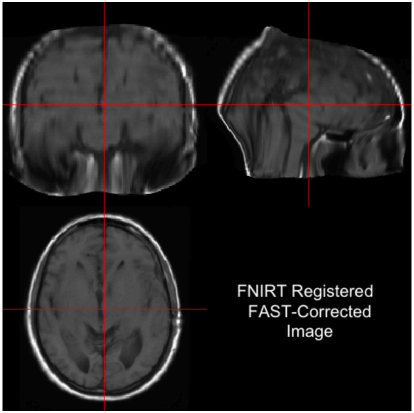
fslr: Image Registration (Non-Linear)

FNIRT performs non-linear registration. An affine registration must be performed before using FNIRT.
FLIRT can also do Affine registrations (DOF = 12), and `fslr::fnirt_with_affine` will perform an affine registration than FNIRT. You want to perform this on skull-stripped images.

```
fnirt_fast = fnirt_with_affine(infile=bet_fast2,
        reffile = "MNI152_T1_1mm_brain.nii.gz",
        outfile = "FNIRT_to_Template", retimg=TRUE)
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
FSLDIR='/usr/local/fsl'; export FSLDIR; sh "${FSLDIR}/etc/f
```

fslr: Image Registration (Non-Linear)





References I

Régis Guillemaud and Michael Brady. “Estimating the bias field of MR images”. In: *Medical Imaging, IEEE Transactions on* 16.3 (1997), pp. 238–251.

Yongyue Zhang, Michael Brady, and Stephen Smith. “Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm”. In: *Medical Imaging, IEEE Transactions on* 20.1 (2001), pp. 45–57.