

Registration

The purpose of image registration is to move images so that they align in the same space. The basic pipeline uses tools from the fsl toolbox and their wiki (<https://fsl.fmrib.ox.ac.uk/fsl/fslwiki>) has lots of information.

Affine

If the images are from the same participant then they likely do not need to be morphed to re-align so affine or linear registration is sufficient.

- `flirt -in regname(_brain) -ref refname -out outname -omat outname.mat -bins 256 -cost corratio -searchrx -90 90 -searchry -90 90 -searchrz -90 90 -dof 9 -interp sinc -sincwidth 7 -sincwindow hanning`

The function will move the -in image to align with the -ref image. Depending on the situation, I may use a brain extracted version of the image in this command in which case the ref image should also be brain extracted. **It should be noted that registration can go very wrong so it should always be checked.** If I find that the registration is very off then I try again with the -searchr parameters set to a lower limit (like -10 10 or -5 5) and this often works. I've also reduced the number of bins or changed the dof to see if this helps. Dof=degrees of freedom so 6 means translation in x, y and z only, 9 also includes rotation and 12 includes scaling.

If the image to register is collected with EPI (e.g. DTI or fMRI data) then there is a function called epi_reg which seems to do a better job than just flirt

- `epi_reg --epi=epi_image --t1=3DT1 --t1brain=3DT1_brain --out out_image`

Sometimes we want to apply the same registration matrix to another image (e.g. when doing registration for MWF, we use the first echo of the GRASE to register and then apply this same registration to the MWF map). This can be done with

- `flirt -in regfile.nii.gz -ref reference.nii.gz -applyxfm -init registration.mat -out outname.nii.gz -padding 0.0 -interp sinc -sincwidth 7 -sincwindow hanning`

If a participant has some white matter pathology (e.g. MS lesions) then this can affect the registration (and segmentation). We can fill in the lesions if we have a lesion mask.

- `Lesion_filling -i 3DT1.nii.gz -l lesion_mask.nii.gz -o outfilename`

The function will fill in the lesions with the intensity of surrounding non-lesional white matter.

Non-affine

Non-affine or non-linear registration will warp different brains together. If images are being moved into a standard space (e.g. MNI space) then you will need to use non-affine registration.

As a first step, do an affine registration of the 2 images to get an initial guess at the matrix.

- `fnirt --in=regname --ref=refname --aff=affine.mat --cout=fnirtname_coeff --iout=fnirtname`

For registration to a standard, fsl has some built-in atlas images (e.g. MNI152_1mm.nii.gz).

Multi-stage registration

If you're going to register in multiple steps (i.e. reg image A to image B and then reg that image B to image C), then to get image A into image C you should concatenate the registration matrices.

- `convert_xfm -omat <outmat_AtoC> -concat <mat_BtoC> <mat_AtoB>`

Grabbing ROIs from an atlas

Often, we're wanting to look at results from a particular brain region. One way to do this is to grab a predefined ROI from a brain atlas and move it back into our participant space. So we basically register the 3DT1 to standard space, grab the ROI from that space and inverse warp it back to 3DT1 space. Here is my little pipeline to do this

- `bet 3DT1 3DT1_brain -f 0.3`
- `flirt -in 3DT1_brain -ref fsldir/data/standard/MNI_T1_2mm_brain -out flirt_outname -omat flirt_outname.mat -bins 256 -cost corratio -searchrx -90 90 -searchry -90 90 -searchrz -90 90 -dof 9 -interp sinc -sincwidth 7 -sincwindow hanning`

check registration with fsleyes

- `fnirt --in=3DT1_brain.nii.gz --ref fsldir/data/standard/MNI152_T1_2mm_brain.nii.gz --aff=flirt_outname.mat --cout=fnirt_outname_coeff --iout=fnirt_outname`

check registration with fsleyes

- `invwarp -w fnirt_outname_coeff.nii.gz -o fnirt_outname_coeff_inv.nii.gz -r 3DT1_brain.nii.gz`
- `applywarp --in=atlas_ROI -o atlas_ROI_reg3DT1 -w fnirt_outname_coeff_inv.nii.gz -r 3DT1_brain.nii.gz`

The atlas ROI will need to be grabbed from one of the atlases (fsldir/data/atlas). You may need to threshold the atlas to get only the structure you're interested in. Finally, you will need to clean up the registered ROI by thresholding and binarizing it.

Note: I may or may not use the brain extracted version of the 3DT1 for this pipeline. Maybe test and see what works best.

Halfway reg

If there are 2 timepoints in a study, then it is best to move each image to a halfway space rather than registering one to the other. (If you only register one then you're biasing the data by smoothing one

slightly with the registration and not affecting the other.) This can be done with the following commands:

First, register Image A2 to A1

- `flirt -in A2 -ref A1 -out A2toA1 -omat A2toA1.mat -bins 256 -cost corratio -searchrx -90 90 -searchry -90 90 -searchrz -90 90 -dof 9 -interp sinc -sincwidth 7 -sincwindow hanning`
- `avscale A2toA1.mat A2.nii.gz > A2toA1_avscale`
- `extracttxt Forward A2toA1_avscale 4 1 > A2_half_A1.mat`
- `extracttxt Backward A2toA1_avscale 4 1 > A1_half_A2.mat`
- `flirt -out A1_2HW -applyxfm -init A1_half_A2.mat -ref A1 -in A1`
- `flirt -out A2_2HW -applyxfm -init A2_half_A1.mat -ref A1 -in A2`

Template Creation

ANTs (Advanced Normalisation Tools) can be used to create average templates from multiple images. We might want to do this if we have several timepoints and want to have an average space where all the timepoints are moved into. Or we might have several participants and we want an average space for all of them.

- `antsMultivariateTemplateConstruction2.sh -d 3 -o T_ -i 4 -g 0.2 -b 1 -e 1 -c 2 -j 21 -k 1 -w 1 -f 6x4x2x1 -s 3x2x1x0 -q 140x120x100x100 -n 1 -a 1 -y 1 -r 1 -m CC[2] -l 1 -t SyN P*_3DT1.nii.gz`

Here I've called defined each 3DT1 as a different participant PXX_3DT1 but you can change this up.

To then apply a transformation matrix to other images use

- `antsApplyTransforms -d 3 -i PXX_other.nii.gz -r T_template0.nii.gz -t T_PXX_3DT1*1Warp.nii.gz -t T_PXX_3DT1*GenericAffine.mat -o PXX_other_WarpedToTemplate.nii.gz`