Read Me Information for do\_analysis\_edit.sh

**Template**

Inputs:

Outputs:

Other Info:

If in doubt of input/output ignore what is in ‘echo’ and go by fsl format – note the order is not consistent for all tools

# !/bin/bash %Is a bash script

# do\_analysis\_edit.sh sourcedata subj\_no %basic input

#Prints out usage of the code and input parameters

if [ $# -lt 1 ]; then

echo "Usage: $0 <sourcedata folder> <SubjectNumber> e.g. ./do\_analysis\_edit.sh /Users/colette/sourcedata/ sub-01";

#exit 1 ;

fi

Prints out the usage if you enter the script name into the command line.

Usage is: path to sourcedata folder, subject number in format sub-01

src=$1 Input 1 = path to sourcedata folder

subj=$2 Input 2 = subject number in format sub-01

PVthresh=0.5 Partial Volume threshold is set at 0.5

srcout=/Users/colette/sourcedata/derivatives/ Source output path is /Users/colette/sourcedata/derivatives/

MRIParam='default paramA paramB' MRI parameters input to loops are default, paramA, paramB

## Make CVR maps ##

## Make directory for Percentage CVRmaps ##

if ! [ -d "${srcout}/StatsOutput/CVRmaps\_percentage" ]; then #check if the output directory exists, if not make it

mkdir ${srcout}/StatsOutput/CVRmaps\_percentage Make Directory {source output} StatsOutput/CVRmaps\_percentage

echo "Making CVRmaps\_all folder:${srcout}/StatsOutput/CVRmaps\_percentage" prints to screen “Making CVRmaps\_all folder {source output} StatsOutput/CVRmaps\_percentage”

fi end of if statement

echo "Making CVRmaps percentatges for $2..." Prints to screen "Making CVRmaps percentatges for $2..."

## Loop to make CVRmaps percentages

MRIParam='default paramA paramB'

for MRIParam in ${MRIParam}

do

${FSLDIR}/bin/fslmaths ${srcout}/StatsOutput/CVRmaps\_all/CVRmaps\_concat\_${subj}\_${MRIParam}.nii.gz -mul 100 ${srcout}/StatsOutput/CVRmaps\_percentage/CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz

echo "... CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz"

done

Loop that uses FSLmaths to take input: {source out}/StatsOutput/CVRmaps\_all/CVRmaps\_concat\_{subject number}\_{MRI Parameter}.nii.gz and multiplies it by 100 to create a percentage CVR map, output called CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz

It does this for all three MRI parameters

## Anatomical analysis ##

echo "Running anatomical analysis for $2..."

Prints to screen "Running anatomical analysis for {subject number}..."

echo "Running BET..." Prints to screen "Running BET..."

${FSLDIR}/bin/bet ${src}/${subj}/anat/${subj}\_T1w ${srcout}/${subj}/${subj}\_T1w\_brain -f 0.3 -g 0

From FSL directory, applies FSL BET (Brain Extraction)

(-f) Fractional intensity threshold set at 0.3 (0->1) (default=0.5; smaller values give larger brain outline estimates)

(-g) vertical gradient in fractional intensity threshold set at 0, (-1->1); default=0; positive values give larger brain outline at bottom, smaller at top

**Path to input**: {sourcedata folder}/{subject number}/anat/{subject number}\_T1w – Subject’s anatomical structural T1 weighted scan

**Gives output**: {source out}/{subject number}/anat/{subject number}\_T1w\_brain

echo "Running Fast for $2...only run once then comment out" Prints to screen "Running Fast for {subject number}...only run once then comment out"

${FSLDIR}/bin/fast -t 1 -n 3 -H 0.1 -I 4 -l 20.0 -o ${srcout}/${subj}/${subj}\_T1w\_brain ${srcout}/${subj}/${subj}\_T1w\_brain

From FSL directory, applies FAST (FMRIB's Automated Segmentation Tool)

*Zhang, Y. and Brady, M. and Smith, S. Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm. IEEE Trans Med Imag, 20(1):45-57, 2001.*

(-t) selected as a T1 weighted image (type of image (n=1 for T1, n=2 for T2, n=3 for PD))

(-n) number of tissue-type classes, set at 3 (GM, WM, CSF)

Advanced options:

(-H) MRF beta value for main segmentation phase (increasing this gives spatially smoother segmentations), set at 0.1

(-I, for Indigo) Number of main-loop iterations, set at 4

(-l, for Lima) Bias field smoothing : n is iterations, m is FWHM in mm - rough guide is m ≅ 2 √n, set at 20

(-o) basename for outputs, is given base name {subject number}\_T1w\_brain

**Input**: {source out}/{subject number}/anat/{subject number}\_T1w\_brain

**Output**: {source out}/{subject number}/anat/{subject number}\_T1w\_brain x6

{subject number}\_T1w\_brain\_mixeltype

{subject number}\_T1w\_brain\_pve\_0

{subject number}\_T1w\_brain\_pve\_1 (Grey matter)

{subject number}\_T1w\_brain\_pve\_2

{subject number}\_T1w\_brain\_pveseg

{subject number}\_T1w\_brain\_seg

* **Partial volume maps**: A (non-binary) partial volume image for each class, where each voxel contains a value in the range 0-1 that represents the proportion of that class's tissue present in that voxel. This is the default output.
* **Binary segmentation**: single image: This is the "hard" (binary) segmentation, where each voxel is classified into only one class. A single image contains all the necessary information, with the first class taking intensity value 1 in the image, etc.
* **Binary segmentation**: One image per class: This is also a hard segmentation output; the difference is that there is one output image per class, and values are only either 0 or 1.
* **Restored input**: This is the estimated restored input image after correction for bias field.
* **Bias field**: This is the estimated bias field.

## Make reference images for registrationfunc ##

echo "Making reference images" Prints to screen “Making reference images"

echo "Making reference images for $2 Mzeroscan..." Prints to screen "Making reference images for {subject number} Mzeroscan..."

${FSLDIR}/bin/mcflirt -in ${src}/${subj}/func/${subj}\_MZeroScan -out ${srcout}/${subj}/${subj}\_MZeroScan\_mc

From FSL directory, runs MCFLIRT

MCFLIRT is an intra-modal motion correction tool designed for use on fMRI time series and based on optimization and registration techniques used in [FLIRT](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FLIRT) (FMRIB's Linear Image Registration Tool), a fully automated robust and accurate tool for linear (affine) inter- and inter-modal brain image registration.

*Jenkinson, M., Bannister, P., Brady, J. M. and Smith, S. M. Improved Optimisation for the Robust and Accurate Linear Registration and Motion Correction of Brain Images. NeuroImage, 17(2), 825-841, 2002.*

**Input**: {sourcedata folder}/{subject number}/func/{subject number}\_MZeroScan – the original M0 scan

**Output**: {source out folder}//{subject number}/{subject number}\_MZeroScan\_mc

* Extension \_MC stands for motion correction

MCFLIRT loads the time-series in its entirety and will default to the middle volume as an initial template image. A coarse 8mm search for the motion parameters is then carried out using the cost function specified followed by two subsequent searches at 4mm using increasingly tighter tolerances. All optimizations use trilinear interpolation.

As part of the initial 8mm search, an identity transformation is assumed between the middle volume and the adjacent volume. The transformation found in this first search is then used as the estimate for the transformation between the middle volume and the volume beyond the adjacent one. This scheme should lead to much faster optimization and greater accuracy for the majority of studies where subject motion is minimal. In the pathological cases, this scheme does not penalise the quality of the final correction.

If mean registration is used, the current motion correction parameters are applied to the time-series, the volumes are averaged to create a new template image and the same 3-stage correction is carried out using this new mean image as a template.

Finally, if a 4-stage correction has been specified, a further optimization pass is carried out using sinc interpolation (internally) for greater accuracy. This step is significantly more time-consuming than the previous part of the correction, which should take in the order of 10 minutes for a 100 volume timeseries. This internal interpolation is independent of the final resampling interpolation (as specified by sinc\_final or spline\_final).

The scheme has the added advantage of `sensible' handling of the end slices in a volume. In other schemes, voxels which move out of the Field of View (FOV) due to (often slight) movement of the head are either excluded from further calculations or treated as zero-value. In order to retain as much useful information in the data as possible, the volumes are padded by doubling the first and last slices in the z-plane so that we can interpolate from locations outside the FOV using appropriate values.

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_MZeroScan\_mc -Tmean ${srcout}/${subj}/${subj}\_MZeroScan\_ref

From FSL directory, uses FSLmaths to create the temporal mean

**Input:** {source out folder}//{subject number}/{subject number}\_MZeroScan\_mc

**Output:** {source out folder}//{subject number}/{subject number}\_MZeroScan\_ref \_ref stands for reference

${FSLDIR}/bin/bet ${srcout}/${subj}/${subj}\_MZeroScan\_ref ${srcout}/${subj}/${subj}\_MZeroScan\_brain -Z

From FSL directory, applies FSL BET (Brain Extraction) to the \_ref and

-(Z) can improve the brain extraction if only a few slices are present in the data (i.e., a small field of view in the Z direction). This is achieved by padding the end slices in both directions, copying the end slices several times, running bet2 and then removing the added slices.

**Input:** {source out folder}//{subject number}/{subject number}\_MZeroScan\_ref

**Output:** {source out folder}//{subject number}/{subject number}\_MZeroScan\_brain

MRIParam='default paramA paramB'

for MRIParam in ${MRIParam}

do

if ! [ -d "${srcout}/${subj}/${subj}\_${MRIParam}\_analysis" ]; then #check if the output directory exists, if not make it

mkdir ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis

echo "Making folder: ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis"

fi

Loop that checks to see if the folder {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis exists, and creates one for each MRI parameter

echo "MCFLIRT for $2 ${MRIParam}..."

Prints to screen “MCFLIRT for {subject number} {MRI parameter}

${FSLDIR}/bin/mcflirt -in ${src}/${subj}/func/${subj}\_task-hyper\_acq-${MRIParam}\_asl -out ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_mc

From FSL directory, runs MCFLIRT on the task-hyper\_acq file

**Input:** {sourcedata folder}/{subject number}/func/{subject number}\_task-hyper\_acq-{MRI parameter}\_asl

**Output:** {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis//{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_mc

echo "FSLmaths Tmean for $2 ${MRIParam}..."

Prints to screen “FSLmaths Tmean for {subject number} {MRI parameter}”

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_mc -Tmean ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_ref

From FSL directory, uses FSLmaths to create the temporal mean

**Input:** {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis//{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_mc

**Output:** {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis//{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_ref

echo "BET for $2 ${MRIParam}..."

Prints to screen “BET for {subject number} {MRI parameter}”

${FSLDIR}/bin/bet ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_ref ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_brain -Z

done

From FSL directory, applies FSL BET (Brain Extraction) to the \_ref and

-(Z) can improve the brain extraction if only a few slices are present in the data (i.e., a small field of view in the Z direction). This is achieved by padding the end slices in both directions, copying the end slices several times, running bet2 and then removing the added slices.

**Input:** {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis//{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_ref

**Output:** {source out}/{subject number}/{subject number}\_{MRI parameter}\_analysis//{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_brain

Does this for each MRI parameter

## Registration of functional and anatomical images ##

echo "Registering functional and anatomical images..."

Prints to screen "Registering functional and anatomical images..."

## T1w to MNI ##

echo "T1w to MNI..."

Prints to screen "T1w to MNI..."

${FSLDIR}/bin/flirt -in ${srcout}/${subj}/${subj}\_T1w\_brain -ref ${FSLDIR}/data/standard/MNI152\_T1\_1mm\_brain -out ${srcout}/${subj}/${subj}\_T1w\_MNI -omat ${srcout}/${subj}/${subj}\_T1w2MNI.mat -v

From FSL directory uses FLIRT (FMRIB's Linear Image Registration Tool) a fully automated robust and accurate tool for linear (affine) intra- and inter-modal brain image registration.

[*1.*](http://www.ncbi.nlm.nih.gov/pubmed/11516708)*M. Jenkinson and S.M. Smith. A global optimisation method for robust affine registration of brain images. Medical Image Analysis, 5(2):143-156, 2001.*

[*2.*](http://www.ncbi.nlm.nih.gov/pubmed/12377157)*M. Jenkinson, P.R. Bannister, J.M. Brady, and S.M. Smith. Improved optimisation for the robust and accurate linear registration and motion correction of brain images. [NeuroImage](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/NeuroImage), 17(2):825-841, 2002.*

(-v) Registration of Tw1 to MNI space.

**Input:** {source out}/{subject number}/{subject number}\_T1w\_brain + {source out}/{subject number}/{subject number}\_ T1w\_MNI

**Output:** {source out}/{subject number}/{subject number}\_T1w2MNI

## MZeroScan to T1w ##

echo "MZeroScan to T1w..."

Prints to screen "MZeroScan to T1w..."

${FSLDIR}/bin/flirt -in ${srcout}/${subj}/${subj}\_MZeroScan\_brain -ref ${srcout}/${subj}/${subj}\_T1w\_brain -out ${srcout}/${subj}/${subj}\_MZeroScan\_T1w -omat ${srcout}/${subj}/${subj}\_MZeroScan2T1w.mat -dof 6 -v

From FSL directory uses FLIRT to register from MZeroscan to T1w and creates this as a .mat file

Uses Mzeroscan\_brain as a reference

(-dof) uses 6 degrees of freedom

(-v) verbose output

(-omat) the calculated affine transformation that registers the input to the reference which is saved as a 4x4 affine matrix

**Input:** {source out}/{subject number}/{subject number}\_MZeroScan\_brain

**Output:** {source out}/{subject number}/{subject number}\_MZeroScan2T1w.mat

## PCASL to MZeroScan ##

echo "PCASL to MZeroScan..."

Prints to screen "PCASL to MZeroScan..."

MRIParam='default paramA paramB'

for MRIParam in ${MRIParam}

do

${FSLDIR}/bin/flirt -in ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_brain -ref ${srcout}/${subj}/${subj}\_MZeroScan\_brain -out ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_task-hyper\_acq-${MRIParam}\_asl\_MZeroScan -dof 6 -omat ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_${MRIParam}\_2MZeroScan.mat -v

echo "Creating ... ${subj}\_task-hyper\_acq-${MRIParam}\_asl\_brain, ${subj}\_MZeroScan\_brain, ${subj}\_task-hyper\_acq-${MRIParam}\_asl\_MZeroScan, ${subj}\_${MRIParam}\_2MZeroScan.mat"

done

From FSL directory uses FLIRT to register from the registers from the {MRI parameter}\_asl\_brain to the MZero+brain space creating a {subject number}\_{MRI parameter}\_2MzeroScan as a .mat file

Uses Mzeroscan\_brain as a reference

(-ref) specifies the reference file

(-dof) uses 6 degrees of freedom

(-v) verbose output

(-omat) the calculated affine transformation that registers the input to the reference which is saved as a 4x4 affine matrix

**Input:** {subject number}\_task-hyper\_acq}-{MRI parameter}\_asl\_brain

{subject number}\_MZeroScan\_brain

{subject number}\_task-hyper\_acq-{MRI parameter}\_asl\_MZeroScan

**Output:** {subject number}\_{MRI parameter}\_2MZeroScan.mat

\*\* check this one\*\*

## Create transform from MNI space to MZeroScan space ##

echo "Create transform from MNI space to MZeroScan space..."

Prints to screen "Create transform from MNI space to MZeroScan space..."

${FSLDIR}/bin/convert\_xfm -omat ${srcout}/${subj}/${subj}\_T1w2MZeroScan.mat -inverse ${srcout}/${subj}//${subj}\_MZeroScan2T1w.mat

From FSL directory uses convert\_xfm

**convert\_xfm** is a utility that is used to convert between different transformation file formats. It can read and write ascii 4x4 matrices. In addition, it can be used to concatenate two transforms (using -concat with the second transform) or to find the inverse transformation (using -inverse).

Creates MZero transformed to T1w space

(-omat) the calculated affine transformation that registers the input to the reference which is saved as a 4x4 affine matrix

(-inverse) find inverse transformation

**Input:** {source out}/{subject number}/{subject number}\_T1w2MZeroScan.mat

**Output:** {source out}/{subject number}/{subject number}\_MZeroScan2T1w.mat

echo "T1w2MNI ==> MNI2T1w..."

Prints to screen "T1w2MNI ==> MNI2T1w..."

${FSLDIR}/bin/convert\_xfm -omat ${srcout}/${subj}/${subj}\_MNI2T1w.mat -inverse ${srcout}/${subj}/${subj}\_T1w2MNI.mat

From FSL directory uses convert\_xfm to create T1 weighted image to MNI space

(-omat) the calculated affine transformation that registers the input to the reference which is saved as a 4x4 affine matrix

(-inverse) find inverse transformation

**Input:** {source out}/{subject number}/{subject number}\_MNI2T1w.mat

**Output:** {source out}/{subject number}/{subject number}\_T1w2MNI.mat

echo "T1w2MZeroScan + MNI2T1w ==> MNI2MZeroScan..."

Prints to screen "T1w2MZeroScan + MNI2T1w ==> MNI2MZeroScan..."

${FSLDIR}/bin/convert\_xfm -omat ${srcout}/${subj}/${subj}\_MNI2MZeroScan.mat -concat ${srcout}/${subj}/${subj}\_T1w2MZeroScan.mat ${srcout}/${subj}/${subj}\_MNI2T1w.mat

From FSL directory uses convert\_xfm to create MNI space to M0 space

convert\_xfm -omat AtoC.mat -concat BtoC.mat AtoB.mat

(-omat) the calculated affine transformation that registers the input to the reference which is saved as a 4x4 affine matrix

(-concat) concatenate two transforms

**Input:** {source out}/{subject number}/{subject number}\_T1w2MZeroScan.mat

{source out}/{subject number}/{subject number}\_MNI2T1w.mat

**Output:** {source out}/{subject number}/{subject number}\_MNI2MZeroScan.mat

## Apply transformations ##

echo "Apply Transformations..."

Prints to screen "Apply Transformations..."

echo "Creating ${subj}\_GM\_MZeroScan..."

Prints to screen "Creating {subject numbe\_GM\_MZeroScan..."

${FSLDIR}/bin/applywarp --in=${srcout}/${subj}/${subj}\_T1w\_brain\_pve\_1 --ref=${srcout}/${subj}/${subj}\_MZeroScan\_brain --out=${srcout}/${subj}/${subj}\_GM\_MZeroScan --premat=${srcout}/${subj}/${subj}\_T1w2MZeroScan.mat --interp=trilinear --super --superlevel=4

From FSL directory uses **applywarp** to apply warps

(in) Name of file to which the warps should be applied.

(ref) Name of a file in target space. This typically the same file that was specified as --ref in fnirt.

(out) Name of output file, i.e. the --in file warped into the space of the --ref file.

(-premat) Specifies an affine transform that should be applied to the data prior to the non-linear warping

**Input:** {source out}/{subject number}/{subject number}\_T1w\_brain\_pve\_1

**Reference:** {source out}/{subject number}/{subject number}\_MZeroScan\_brain

**Output:** {source out}/{subject number}/{subject number}\_GM\_MZeroScan

echo "Creating ${subj}\_MNIatlas..."

Prints to screen "Creating ${subj}\_MNIatlas..."

${FSLDIR}/bin/flirt -in ${FSLDIR}/data/atlases/MNI/MNI-maxprob-thr0-1mm -ref ${srcout}/${subj}/${subj}\_MZeroScan\_brain -out ${srcout}/${subj}/${subj}\_MNIatlas -init ${srcout}/${subj}/${subj}\_MNI2MZeroScan.mat -interp nearestneighbour -applyxfm -v

From FSL directory uses FLIRT

(in) input

(ref) reference

(out) output

(init) apply a saved transformation to a volume

(interp) interpret using nearest neighbour model

(-applyxfm) apply a saved transformation to a volume, or to apply a transform that aligns the NIFTI mm coordinates

(-v) verbose

**Input:** {FSLDIRectory}/data/atlases/MNI/MNI-maxprob-thr0-1mm

Reference: {source out}/{subject number}/{subject number}\_MZeroScan\_brain

**Output:**

{source out}/{subject number}/{subject number}\_MNI2MZeroScan.mat

{source out}/{subject number}/{subject number}\_ MNIatlas

echo "Applying Transforms "

Prints to screen “Applying Transforms "

MRIParam='default paramA paramB'

for MRIParam in ${MRIParam}

do

${FSLDIR}/bin/flirt -in ${srcout}/StatsOutput/CVRmaps\_percentage/CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz -ref ${srcout}/${subj}/${subj}\_MZeroScan\_brain.nii.gz -out ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_${MRIParam}\_cvr\_MZeroScan -init ${srcout}/${subj}/${subj}\_${MRIParam}\_analysis/${subj}\_${MRIParam}\_2MZeroScan.mat -applyxfm -v

echo "CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz + ${subj}\_${MRIParam}\_2MZeroScan ==> ${subj}\_${MRIParam}\_cvr\_MZeroScan..."

Prints to screen CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz + ${subj}\_${MRIParam}\_2MZeroScan ==> ${subj}\_${MRIParam}\_cvr\_MZeroScan...

done end of loop

From FSL directory uses FLIRT to loop through each MRI parameter and apply warp and transformations:

(in) input

(ref) reference)

(out) output

(init) apply a saved transformation to a volume

(-applyxfm) apply a saved transformation to a volume, or to apply a transform that aligns the NIFTI mm coordinates

(-v) verbose

**Input:** {source out}/StatsOutput/CVRmaps\_percentage/CVRmaps\_concact\_${subj}\_${MRIParam}\_percentage.nii.gz

**Reference:** {source out}/{subject number}/{subject number}\_MZeroScan\_brain.nii.gz

**Output:**

{source out}/{subject number}/{subject number}\_{MRI Parameter}\_analysis}/{subject number}\_{MRI Parameter}\_cvr\_MZeroScan

{source out}/{subject number}/{subject number}\_{MRI Parameter}\_analysis}/{subject number}\_{MRI Parameter}\_2MZeroScan.mat

echo "Make ROIs"

Prints to screen "Make ROIs"

## Make ROIs ##

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_MNIatlas -thr 3 -uthr 6 ${srcout}/${subj}/${subj}\_MNIatlas\_cortex

From FSL directory uses fslmaths to create an MNI atlas of the cortex

Lower threshold at 3

Upper threshold at 6

**Input:** {source out}/{subject number}/{subject number}\_MNIatlas

**Output:** {source out}/{subject number}/{subject number}\_MNIatlas\_cortex

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_MNIatlas -thr 8 -uthr 8 -add ${srcout}/${subj}/${subj}\_MNIatlas\_cortex ${srcout}/${subj}/${subj}\_MNIatlas\_cortex

From FSL directory uses fslmaths to create an MNI atlas of the cortex and adds the two together

Lower threshold at 8

Upper threshold at 8

**Input:** {source out}/{subject number}/{subject number}\_MNIatlas

{source out}/{subject number}/{subject number}\_MNIatlas\_cortex

**Output:** {source out}/{subject number}/{subject number}\_MNIatlas\_cortex

echo "Binarise MNIatlas ..."

Prints to screen "Binarise MNIatlas ..."

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_MNIatlas -bin ${srcout}/${subj}/${subj}\_MNIatlas\_cortex

From FSL directory uses fslmaths to create an MNI atlas that is binarized (bin) so any voxels in the cortex will appear as 1 in the mask and any outside will be 0

**Input:** {source out}/{subject number}/{subject number}\_MNIatlas

**Output:** {source out}/{subject number}/{subject number}\_MNIatlas\_cortex

echo "gmcortex ${subj}\_GM\_MZeroScan ${subj}\_MNIatlas\_cortex"

Prints to screen "gmcortex {subject number}\_GM\_MZeroScan {subject number}\_MNIatlas\_cortex"

${FSLDIR}/bin/fslmaths ${srcout}/${subj}/${subj}\_GM\_MZeroScan -thr ${PVthresh} -mas ${srcout}/${subj}/${subj}\_MNIatlas\_cortex -bin ${srcout}/${subj}/${subj}\_gmcortex

From FSL directory uses fslmaths to threshold (-thr) at the {Partial volume threshold} which is currently 0.5

(-bin) binarises the mask

(-mas) use (following image>0) to mask current image

**Input:** {source out}/{subject number}/{subject number}\_MNIatlas\_cortex

**Output:**

{source out}/{subject number}/{subject number}\_GM\_MZeroScan

{source out}/{subject number}/{subject number}\_gmcortex

echo "~~~ End of do\_analysis for $2 ~~~"

Prints to screen "~~~ End of do\_analysis for {subject number}~~~"