**Social Decision Neuroscience Lab (in collaboration with Dr Sanjay Manohar) guide to running lesion mapping using FSL & randomise**

see Manohar, S. G. & Husain, M. Human ventromedial prefrontal lesions alter incentivisation by reward. *Cortex* **76**, 104–120 (2016). <https://doi.org/10.1016/j.cortex.2016.01.005>.

Guide written by Dr Jo Cutler (2023) & tweaked by Zhilin Su (2025).

1. Prepare the imaging data. You will need the 4D data including all participants’ lesion masks for analysis as a nifi .nii.gz format file (called 4D\_data.nii.gz below and for shared files). This should have the length in the 4th dimension as the number of participants to include and the **order of files must match the order of behavioural regressors entered in the design file below**. This should already have applied mirroring, so each participant’s 3D binary image file is symmetrical, if you want to apply this.
2. Create a .nii mask file with where at least 5 patients (or whatever number to use as a minimum) have damage (called min5\_powerfilter.nii below and for shared files). *Note if you are checking these maps in MRIcroGL, you may need to turn off the (default) setting of loading smooth overlays before opening the images. For example, if checking your min5\_powerfilter.nii does select where 5 people have damage this will look very odd with the default smoothing.*
3. Prepare behavioural regressor design files. It might be easiest to first create a csv file with each participant’s z-scored behavioural regressor to include. These should be ranked before z-scoring if residuals are skewed and create a design file for both the positive and negative regressor if you want to examine both. See the script analysis.qmd to create and save these csv files for the mPFC lesion and social influence project and the files themselves on the OSF page within subfolders for each analysis.
4. See script analysis\_PM\_vmPFC.Rmd or for code to create and save these csv files for the vmPFC lesion and prosocial motivation project and the files themselves on the OSF page within subfolders for each analysis (e.g. recip\_k\_ranked.csv, recip\_choice\_ranked.csv, effort\_ranked.csv and reward\_ranked.csv)

To prepare the design files, use FSL’s glm function:

* 1. Open a terminal and change directory to the relevant folder for that analysis e.g.

cd FOLDER/all\_lesion\_patients/recip\_choice

* 1. type fsl (requires fsl installed on computer). Go to “Misc” dropdown menu and select GLM Setup. On the pop-up window drop down to Higher-level / non-timeseries design and set the # inputs to the number of participants you are including in the current analysis then make sure to press enter and see that the numbered input boxes on the General Linear Model window have updated to your n.
  2. If you are just including the behavioural regressor of interest keep Number of main EVs = 1. If not all showing Group 1 in the first column click Wizard on the GLM Setup box and single group average. Open the csv file of the regressors and click Paste on the fsl General Linear Model box. Clear the default input (ones) and then select the relevant values from the csv to paste into this box (without headings). What you are pasting should have as many rows as participants and as many columns as EVs, in this example just the one behavioural regressor. Click OK, the values you just pasted should now appear in the EV column(s). You can name the EV(s) (e.g. r\_choice).

1. Click save on the GLM Setup box, the Save Feat setup should default to the folder you changed to above. In the Selection box add the name of the regressor(s) at the end after / and click OK. This should save a number of different files with that name into the folder. Check you have one .con and one .mat. Repeat the fsl GLM setup steps for each regressor
2. Now you are ready to run randomise. In the terminal, run e.g.:

randomise -i ../4D\_data.nii.gz -o r\_choice\_pos -t r\_choice\_pos.con

-d r\_choice\_pos.mat -m ../min5\_powerfilter.nii -D -n 5000 --uncorrp -T

-i is the input, here the 4D data with all participants’ 3D lesion masks

-o is the name to use for all the output files

-t and -d are the design files created above in .con and .mat format respectively

-m is the mask of where at least 5 patients have damage – only voxels within this will be included

-D demeans the data as recommended

-n 5000 specifies 5000 permutations

--uncorrp generates maps that are not corrected voxelwise as we will manually correct below

-T applies threshold-free cluster enhancement (TFCE)

***This will run the analysis, taking several minutes.***

***Optional steps below for visualisation (followed for shared files):***

1. Copy the files for the p-values you want to report to ../results/pmaps and the files for the corresponding (TFCE) t-maps to ../results/tmaps\_unthr
2. Apply correction for multiple comparisons to threshold (here *p*<0.0125 to correct for 4 comparisons) and create a binary mask of the significant area(s) found in the lesion mapping:

cd ../results/pmaps

for f in \*\_tfce\_p\_tstat1.nii.gz ; do echo $f ; fslmaths $f -thr 0.9875 ../masks/$f ; fslmaths ../masks/$f -bin ../masks/$f ; done

1. Mask t-maps for visualisation

cd ../tmaps\_unthr

for f in \*\_tstat1.nii.gz ; do filename=`basename ${f%%\_tstat\*}`; echo $filename ; filename+=\_tfce\_p\_tstat1 ; echo $filename ; fslmaths $f -mas ../masks/$filename ../tmaps\_thr/$f ; done

1. Apply binary masks to each patients’ lesion map and save the results to a text file

cd ..

for m in masks/\*.nii.gz ; do filename=`basename ${m%%\_p\_tstat1\*}`; echo $filename ; for f in ../../../lesion\_masks/\*.nii ; do pn=`basename ${f%%.nii}` ; echo $pn >> textfiles/$filename.txt; done ; for f in ../../../lesion\_masks/\*.nii ; do fslstats $f -k $m -V >> textfiles/$filename.txt; done ; done

1. Convert text file to right format to read into R for plotting:
   1. Open .txt file with excel
   2. File > save as ../../../../PM\_R\_code/data/[filename].csv
   3. Move rows of values (second half of rows) to second column
   4. Apply text to columns (under Data tab) to second column with space delimeter to split
   5. Add headings in first row of “ID” then [variable]\_voxels and [variable]\_volume where variable is choice / recipient / reward
2. Extract MNI coordinates of peak voxels to report – this can be done either on p values from randomise (done below; need to specify scalarname=”1-p” so it knows high is sig not low values) with your p value threshold used above or on the t maps. To save the output to a text file instead of just printing to the terminal add > and the filename at the end e.g. > ../mni/r\_c\_neg\_tfce\_p

cluster -i r\_c\_neg\_tfce\_p\_tstat1.nii.gz -t 0.9875 --scalarname="1-p" –mm