# FMRI processing pipeline guide for diffusion embedding

Here is a little guide on the whole process from data preprocessing to data analysis. :)

Data path : /dycog/meditation/ERC/Analyses/MINCOM

Step 1: Conversion into BIDS format.
For us to be able to use FMRIPREP pipeline (Step 3), we first need to convert the whole dataset into BIDS (=Brain Imaging Data Structure) format( https://bids.neuroimaging.io/). For more information on how to specify directory structure and file names under BIDS format, please refer to **bids_spec1.0.2.pdf**

Prior to performing this step, be sure to download **dcm2niix** package.

Then you can edit the **rs_conversion_to_bids.sh** bash script for it to be coherent to the raw data structure you are about to convert and run the script on a bash shell typing :
./rs_conversion_to_bids.sh

You can also run all subjects in parallel, if you have a large data set. In order to do so, you will need to log on the cluster. In order to log on the cluster type:
“
>ssh loic.daumail@10.69.168.62 (replace loic.daumail by your own inserm ID)
> password (the one you use to log on your account)
“

The number 27 (last number on the first commandline) can vary from 25 to 35 (or something like this), which will get you on either of the 10-12 nodes of the cluster.
Then create your own directory to store your scripts and data in ‘/mnt/data/{your_own_folder}’

Then adapt the script so that it’s calling the list of subjects numbers (with ‘{1}’ to replace ‘002 004 005….’) in a separate .txt file, on a one column structure) and run the script with the following command :
‘sbatch ./rs_conversion_bids.sh subject_list.txt’

!!!
**Make sure the raw data is already stored on the cluster database** and **not in the dycog/meditation database**, and to adapt the directories accordingly in the script so they refer to the data stored in the cluster.
If you refer to the data stored in /dycog/meditation, the whole network **might crash** because of that.
!!!

Step 2: data quality check step using MRIQC:
(https://mriqc.readthedocs.io/en/stable/)

Please, go to the following directory: ./mriqc_quality_check_step/
In this directory, you will see two output folders and two bash/slurm scripts.
The MRIQC quality check must be performed prior to FMRIPREP preprocessing, in order to reject potential outliers and you should do it on the cluster, to speed up the process.

Hence, make sure to import the BIDS data from step 1 in this directory, in order to use it accordingly.
There are 2 ways to run MRIQC: either making an individual quality check for every subject data (using **mriqc_pre_fmriprep.sh** bash script), or performing a group level quality check that is the lazy way to do it but also that takes less time (with **mriqc_group_pre_fmriprep.sh** script).
To run it on the cluster, just type:
 'sbtach ./{script.sh}'
You can check the real-time advancement of the jobs with the bash command line:
'watch -n1 squeue'
or just 'squeue' but this will not be real-time.
Duration: 1:30-2 hours

Step 3: FMRIPREP preprocessing step:
(https://fmriprep.readthedocs.io/en/stable/)
FMRIPREP is a fully automated preprocessing pipeline. In order to use it, IT engineers here at CRNL installed a singularity container on the cluster.
The singularity container calls all the dependencies necessary to FMRIPREP. Hence, all dependencies must be updated (if any issue, ask the IT engineers), in order to use the latest version of FMRIPREP, for which a singularity image is created (make sure the singularity image uses the latest version of FMRIPREP).

Go to the ./fmriprep_preprocessing_step/
One folder is the output folder that should look like when you will run the **neurostars_fmriprep_success.sh** script on the cluster.
Just run sbatch ./neurostars_fmriprep_success.sh

Duration: If all subjects are launched in parallel, this should take no more than 6-7 hours.

Step 4:  Diffusion embedding step
Go to the ./diffusion_embedding_step directory
For the packages necessary to this step, please, follow the README.md

a) Project the volume on the surface using mri_vol2surf

In the ./scripts directory, **batch_vol2surf.sh** script launches jobs in parallel that call x.mri_vol2surf_test.sh on every subject listed in the former script that will project the grey matter volume onto the surface.

b) Perform diffusion embedding

This step requires the use of **pipeline.py** script on the cluster with **batch_pipeline.sh** script (pipeline.py calls **fs_load.py**).
As it is a python script, you will need to install locally python3.6 (or higher) with the following command :
'pip install python3.6 –user'
You can also check the python version with 'python -V'

Similarly, you will have to install all the required packages (README.md):
Example:
'python -m pip install numpy --user'
Then make sure you call the latest python version with :

‘alias python = ‘python3.6’’
Then launch the jobs with:
‘batch batch_pipeline.sh’

c) Rotate and gather all subjects into a matrix

You can run this script with : ‘python combine_subjects_all.py’, located in the ./matrices_script folder

All the other scripts are similar, except the “separate_*” scripts that were intended to create independent matrices for every state and subject.

d) Visualization of first results

In ./scripts/visualize_emb_output_step, you can edit the scripts that will allow to save an image for every state and every subject, for you to have a first glance on your data.
An easy way to use these scripts is to use the user interface kate.

Step 5: data analysis with SurfStat

Go to ./statistical_analysis_step/diffusion_embedding_analysis
At this stage, the scripts are based on the previously formed matrix called **comp_om_rs_group_embedding_new.mat** that contains all diffusion maps over the 5 principle dimensions.

a) Checking for outliers

A correlation analysis can be performed in order to detect outliers.
In ./exclude_outliers, you will find the script **exclude_outliers_fisher.m** that aims to exclude the diffusion maps of the 1st dimension that least correlate with the mean diffusion map across all subjects and states.
Once the correlation coefficients are computed, they are fisher z-transformed, to obtain a normal distribution of the correlation coefficients, in order to exclude the outliers below 1.64*SD.

b) Plotting the dimensions against each other

Go back in ./diffusion_embedding_analysis.
In the script entitled **scatter_plots.m** you will be able to plot the different dimensions of the matrix against each other, and get similar graphs as in Hong et al, 2019.

c) Linear mixed model fitting and clusterwise analysis

(tutorial on how to use SurfStat: http://www.math.mcgill.ca/keith/surfstat/)
You can open the script EmbRoiAnalysisComp.m in matlab and use surfstat toolbox (take the folder ‘./surfstat’, where hard code editions were performed in some functions by Daniel S. Margulies for the functions to work well on our data).

**EmbRoiAnalysis.m** script aims to perform clusterwise analyses on a linear-mixed model of our data.
Some rois were determined and then registered for visualization.

A simpler version of the script can be found in **linear_mixed_surfstat_design.m** script.