

1. **XNAT and its main features**

XNAT is an open-source imaging informatics software platform dedicated to helping you perform imaging-based research.

XNAT's core functions manage importing, archiving, processing and securely distributing imaging and related study data.

- Account Management

All access to any XNAT site is restricted to valid users of the site.

User accounts determine the data and pages which are accessible to visitors of the site.

- Creating and Managing Projects

Data stored within XNAT is organized according to projects.

To manage data within your XNAT you will need to create projects which your data can be assigned to.

- Creating and Managing Subjects

Most experiment data in XNAT is associated directly with a subject. In turn, a subject can have multiple experiments. Subject information can be as brief and simple as an identifier. Or, it can contain in-depth demographic information. Either way, creating a subject is required to add imaging data (as well as subject assessment data).

There are multiple ways to store subjects in XNAT:

"Create Subject" form in the website UI

StoreXML

Upload Spreadsheet

REST

-Defining, creating and managing experiments

"Experiments" is the term used in XNAT to describe data that is gathered or measured from direct interaction with a subject.

These types of data can range from a simple consent form signed by a subject at the beginning of a study to demographic data gathered about the subject, their medication list, blood test results,

behavioral assessments for cognitive psychology measures, to MR or PET sessions taken when the subject is in the scanner.

Each experiment corresponds to a data type in XNAT.

- User roles and permissions

By default, XNAT defines three common project roles: Owners, Members, and Collaborators. As a project owner, you have the ability to add specific users to your project within these roles.

These users will then have the permissions associated with their defined role (which overrules the default accessibility permissions).

-Uploading imaging data

The most common purpose for XNAT usage is to store imaging data. The list of imaging types which XNAT supports is continually expanding. Currently DICOM, Analyze, ECAT, IMA, and NRD are supported in varying degrees. XNAT supports several modalities, including MRI, CT, and PET.

If you have a scanner on site, imaging data can be uploaded directly from the scanner via a DICOM C-Store connection. Otherwise, you will commonly be receiving imaging data via the Upload Applet or zipped archives of scan data, and using the ZIP uploader.

Advanced users may use REST scripts to upload data and files

- Viewing image files

XNAT comes bundled with a Java applet that allows users to view images from an MR Session in the browser window.

- Downloading files

There are three modalities for downloading the files:

- download applet - XNAT includes a download applet for downloading files from multiple imaging sessions
- download with REST API - A common mechanism for downloading data is to use the REST API. Using this tool usually requires some mild scripting, but it gives you the most flexibility for choosing what to download.
- download script - most common script is ArcGet

- Pipelines

Pipelines in XNAT are powerful mini-applications that can be run on your project data, to aid in complex processing or leverage the power of large computing clusters. Some pipeline-enabled workflows are carried out automatically without any human intervention, like auto-QC'ing images as they are added to your project archive.

Others require a person to do manual steps, such as drawing a region of interest.

Pipeline Engine

Pipeline Engine is a Java-based framework that links sequential activities, human and computer, into a defined process flow and manages how data moves from step to step in that flow based on the results of each step. In most laboratories, some processes (or pipelines) are carried out automatically without any human intervention while others require a person to do manual steps, such as drawing a region of interest. Pipeline Engine facilitates both fully automated and semi-automated workflows. Pipelines can be executed up to a step, then notify a user(s) to perform the manual task, and finally restart the pipeline from the next step. The process flow is defined in an XML document called the *pipeline descriptor* and the executables are defined in a separate XML document called *resource descriptors*.

Pipeline engine is a standalone tool. However, when used with XNAT one can:

- set up project-based workflows with project specific and experiment specific parameters,

- track a pipeline and send email notifications
- capture provenance information as the pipeline executes

IMPORTANT:

XNAT can be used in order to process data needed as input for TVB.

More details here: <https://wiki.xnat.org/display/XNAT/Pipeline+Engine>

The creation of a pipeline section examples the two needed XML document types: pipeline descriptor (SampleAutoRunPipeline.xml) and resource descriptor (AntCopy.xml). These can be found in the "doc" directory.

- Searching and browsing XNAT data

Once you have project data stored in your XNAT installation, users need to be able to find it. XNAT provides a number of ways of accessing and locating stored data.

2.Scan formats

- T1 MRI

One of the most important features of MRI is that it can generate images depicting a number of different contrast characteristics.

T1 is when they programme the machine to only look at the longitudinal movement of protons. T1 images are usually used to look at normal anatomical details.

Anatomical MRI is T1 weighted.

T1 is best for looking at brain structure because fat appears very bright and bone marrow contains a great deal of fat.

- dMRI (diffusion magnetic resonance imaging)

Diffusion MRI (or dMRI) is a magnetic resonance imaging (MRI) method. It allows the mapping of the diffusion process of molecules, mainly water, in biological tissues, in vivo and non-invasively.

Molecular diffusion in tissues is not free, but reflects interactions with many obstacles, such as macromolecules, fibers, and membranes.

Water molecule diffusion patterns can therefore reveal microscopic details about tissue architecture, either normal or in a diseased state.

- DWI (Diffusion weighted imaging) synonym: diffusion weighted MRI

DWI derives its image contrast from differences in the motion of water molecules between tissues. Such imaging can be performed quickly without the need for the administration of exogenous contrast medium.

The technique yields qualitative and quantitative information that reflects changes at a cellular level and provides unique insights about tumor cellularity and the integrity of cell membranes.

- bvals , bvecs

bvals and bvecs, contain the information on the diffusion-sensitising magnetic field gradients (note that some nifti conversion tools will create bvals and bvecs information and some will not).

bvals contains a scalar value for each applied gradient, corresponding to the respective b-value. bvecs contains a 3x1 vector for each gradient, indicating the gradient direction.

The entries in bvals and bvecs are as many as the number of volumes in the dwidata file. So the ith volume in the data corresponds to a measurement obtained after applying a diffusion-sensitising gradient with a b-value given by the ith entry in bvals and a gradient direction given by the ith vector in bvecs.

3.FSL

FSL is a comprehensive library of analysis tools for FMRI, MRI and DTI brain imaging data. It could be used either by FSL main GUI or the command-lines. The advantage of the GUI lies in its simplicity but the drawback is the less flexibility comparing to the command-lines. FSLView as the display tool could be completely separate use from processing and analysis. Currently FSL only accepts the input files in NIFTI format, and the DICOM files need to be converted to NIFTI after acquisition and before processing. The conversion from DICOM to NIFTI could be fulfilled by several methods, such as dcm2nii from mricon or dcmstack. In the GUI there are nine popular modules defined, and depending on the input file types, they fall into functional-oriented, structural-oriented or diffusion-oriented modules.

The *structural-oriented* functions include:

-BET brain extraction

-BET (Brain Extraction Tool) deletes non-brain tissue from an image of the whole head. It can also estimate the inner and outer skull surfaces, and outer scalp surface, if you have good quality T1 and T2 input images.

-FAST segmentation - tissue segmentation

-FAST (FMRIB's Automated Segmentation Tool) segments a 3D image of the brain into different tissue types (Grey Matter, White Matter, CSF, etc.), whilst also correcting for spatial intensity variations (also known as bias field or RF inhomogeneities). The underlying method is based on a hidden Markov random field model and an associated Expectation-Maximization algorithm. The whole process is fully automated and can also produce a bias field-corrected input image and a probabilistic and/or partial volume tissue segmentation. It is robust and reliable, compared to most finite mixture model-based methods, which are

sensitive to noise.

–FLIRT linear registration

–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.

The *functional-oriented* functions involve :

–SUSAN noise reduction

–SUSAN noise reduction uses nonlinear filtering to reduce noise in an image (2D or 3D) whilst preserving the underlying structure. It does this by only averaging a voxel with local voxels which have similar intensity.

–FEAT FMRI analysis

–FEAT automates as many of the analysis decisions as possible, and allows easy (though still robust, efficient and valid) analysis of simple experiments whilst giving enough flexibility to also allow sophisticated analysis of the most complex experiments. The data modelling which FEAT uses is based on general linear modelling (GLM), otherwise known as multiple regression. It allows you to describe the experimental design; then a model is created that should fit the data, telling you where the brain has activated in response to the stimuli.

– MELODIC ICA

–MELODIC (Multivariate Exploratory Linear Optimized Decomposition into Independent Components) 3.0 uses Independent Component Analysis to decompose a single or multiple 4D data sets into different spatial and temporal components. For ICA group analysis, MELODIC uses either Tensorial Independent Component Analysis (TICA, where data is decomposed into spatial maps, time courses and subject/session modes) or a simpler temporal concatenation approach. MELODIC can pick out different activation and artefactual components without any explicit time series model being specified .

The *DTI-oriented* functions incorporate "FDT diffusion".

FDT (FMRIB's Diffusion Toolbox) is a software tool for analysis of diffusion weighted images. FDT includes tools for data preprocessing, local diffusion modelling and tractography. Each stage in FDT is run separately.

Besides these functional modules, there is a simulator called "POSSIUM MRI simulator", which is a software tool to produce realistic simulated MRI and FMRI images or time series. POSSUM (Physics-Oriented Simulated Scanner for Understanding MRI) includes tools for the pulse sequence generation, signal generation, noise addition and image reconstruction.

4.Mrtrix 3

MRtrix provides a set of tools to perform diffusion-weighted MRI white matter tractography

in the presence of crossing fibres, using Constrained Spherical Deconvolution and a probabilistic streamlines algorithm.

Common tasks:

- Basic DWI processing
- Get from the raw DW image data to performing some streamlines tractography.
- Anatomically Constrained Tractography (ACT)
- SIFT: Spherical-deconvolution informed filtering of tractograms
- SIFT, or 'Spherical-deconvolution Informed Filtering of Tractograms', is a novel approach for improving the quantitative nature of whole-brain streamlines reconstructions. By producing a reconstruction where the streamlines densities are proportional to the fibre densities as estimated by spherical deconvolution throughout the white matter, the number of streamlines connecting two regions becomes a proportional estimate of the cross-sectional area of the fibres connecting those two regions. We therefore hope that this method will attract usage in a range of streamlines tractography applications.
- Structural connectome construction
- Generating a connectome from HCP diffusion & structural data

5.MNE

MNE is a software package for processing magnetoencephalography (MEG) and electroencephalography (EEG) data.

–Magnetoencephalography (MEG) is a functional neuroimaging technique for mapping brain activity by recording magnetic fields produced by electrical currents occurring naturally in the brain, using very sensitive magnetometers.

–Electroencephalography (EEG) is typically a non-invasive (however invasive electrodes are often used in specific applications) method to record electrical activity of the brain along the scalp. EEG measures voltage fluctuations resulting from ionic current within the neurons of the brain. In clinical contexts, EEG refers to the recording of the brain's spontaneous electrical activity over a period of time, as recorded from multiple electrodes placed on the scalp.

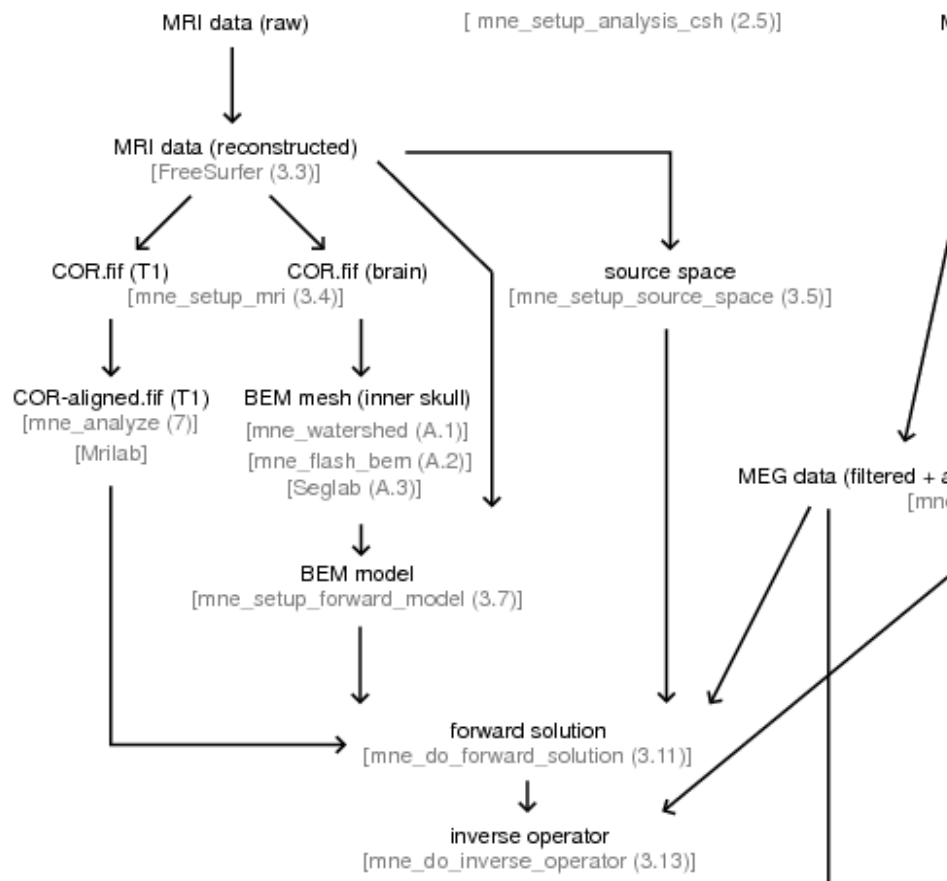
The MNE software computes cortically-constrained L2 minimum-norm current estimates and associated dynamic statistical parametric maps from MEG and EEG data, optionally

constrained by fMRI.

This software includes MEG and EEG preprocessing tools, interactive and batch-mode modules for the forward and inverse calculations, as well as various data conditioning and data conversion utilities. These tools are provided as compiled C code for the LINUX and Mac OSX operating systems.

The software depends on anatomical MRI processing tools provided by the FreeSurfer software.

The typical workflow needed to produce the minimum-norm estimate movies using the MNE software:



6. Pipeline

In order to run the "main_surface.sh" file you have to give right for execution for some files and this is the reason for we have created a script to do this – "rights_for_files.sh" and we have called the execution of it in the main file, right after "importing the configuration" section.

After the execution of the "main_surface.sh" file, the following directories and files are created in the input directory.

- the directory "118730" which contains:

- "connectivity" directory with the files: areas.txt, average_orientation.txt, centres.txt, cortical.txt, tract_lengths.txt, weights.txt.

The "centres.txt" file contains 88 connectivity regions.

- "connectivity_<number>" directories, where number = {1,2,4,8,16,32}.

These directories contain the same files as the "connectivity" directory, EXCEPT the "ares.txt" and "cortical.txt" files.

"centres.txt" file contains for each directory a different number of connectivity regions, as follows: connectivity_1 – 87, connectivity_2 – 157, connectivity_4 – 297, connectivity_8 – 577, connectivity_16 – 1137, connectivity_32 – 2257.

- "region_mapping.txt" file

- "surface" directory with the 9 files: inner_skull_(triangles.txt)/(vertices.txt), outer_skin_(triangles.txt)/(vertices.txt), outer_skull_(triangles.txt)/(vertices.txt), region_mapping.txt, triangles.txt, vertices.txt.

- There are also zip files for every directory: for connectivity and surface.

The matrix from "tract_lengths.txt" file is symmetric.

– "surface" directory with:

– "subcortical" directory with 66 "aseg" files.

– the following files: (lh/rh)info.txt, (lh/rh)_high.off, (lh/rh)_low.off, (lh/rh).pial.asc, (lh/rh)_region_mapping_low_not_corrected.txt, (lh/rh)_region_mapping_low, (lh/rh)_triangles_high.txt, (lh/rh)_triangles_low.txt, (lh/rh)_vertices_high.txt, (lh/rh)_vertices_low.txt .

– in the "connectivity" input folder, contains ,beside the files needed for input (lowb.nii.gz, mask.mif, T1_2_diff.nii.gz) the following files:

- act.mif – generated if "act" flag from the configuration file is "yes"
- aparcaseg_2_diff_<number>.(mif/nii.gz), where number = {1,2,4,8,16,32} in both file formats: mif and nii.gz
- aparcaseg_2_diff_cortical_only.nii, aparcaseg_2_diff.(mif/nii.gz) in both file formats
- aparc+aseg.nii.gz, aparc+aseg_reorient.nii.gz
- bvals, bvecs
- corr_mat_<number>.txt, where number = {1,2,4,8,16,32}
- CSD8.mif
- diffusion_2_struct.mat, diffusion_2_struct_inverse.mat
- dwi.(mif/nii.gz)
- gmwmi_mask.mif - generated if "act" flag is "yes"
- lowb_2_struct.nii.gz
- response.txt
- T1.nii.gz
- tract_lengths.csv , tract_lengths_<number>.csv, where number = {1,2,4,8,16,32}
- weights.csv, weights_<number>.csv, number = {1,2,4,8,16,32}
- whole_brain.tck, whole_brain_post.tck


```

root@pipeline:/opt/software/Pipeline_Input_118730/118730/connectivity# ls
act.mif                                bvals                                T1_2_diff.nii.gz
aparcaseg_2_diff_16.mif               bvecs                                T1.nii.gz
aparcaseg_2_diff_16.nii.gz            corr_mat_16.txt                     tract_lengths_16.csv
aparcaseg_2_diff_1.mif                corr_mat_1.txt                      tract_lengths_1.csv
aparcaseg_2_diff_1.nii.gz             corr_mat_2.txt                      tract_lengths_2.csv
aparcaseg_2_diff_2.mif                corr_mat_32.txt                    tract_lengths_32.csv
aparcaseg_2_diff_2.nii.gz             corr_mat_4.txt                     tract_lengths_4.csv
aparcaseg_2_diff_32.mif               corr_mat_8.txt                     tract_lengths_8.csv
aparcaseg_2_diff_32.nii.gz            CSD8.mif                           tract_lengths.csv
aparcaseg_2_diff_4.mif                diffusion_2_struct_inverse.mat       weights_16.csv
aparcaseg_2_diff_4.nii.gz             diffusion_2_struct.mat              weights_1.csv
aparcaseg_2_diff_8.mif                dwi.mif                            weights_2.csv
aparcaseg_2_diff_8.nii.gz             dwi.nii.gz                         weights_32.csv
aparcaseg_2_diff_cortical_only.nii    gmwmi_mask.mif                     weights_4.csv
aparcaseg_2_diff.mif                  lowb_2_struct.nii.gz                weights_8.csv
aparcaseg_2_diff.nii.gz               lowb.nii.gz                        weights.csv
aparc+aseg.nii.gz                     mask.mif                           whole_brain_post.tck
aparc+aseg_reorient.nii.gz            response.txt                        whole_brain.tck
root@pipeline:/opt/software/Pipeline_Input_118730/118730/connectivity# █

```

- in the folder given by the SUBJECTS_DIR path (freesurfer processed data):
- 118730 directory with the following directories: ascii, bem, label, mri, scripts, src, stats, surf, tmp, touch, trash

```

root@pipeline:/opt/data/freesurfer/118730# ls
ascii hem label mri scripts src stats surf tmp touch trash
root@pipeline:/opt/data/freesurfer/118730# ls ascii/
aseg_004.srf aseg_007.srf aseg_010.srf aseg_012.srf aseg_014.srf aseg_016.srf aseg_018.srf aseg_028.srf aseg_044.srf aseg_047.srf aseg_050.srf aseg_052.srf aseg_054.srf aseg_060.srf aseg_252.srf aseg_254.srf
aseg_005.srf aseg_008.srf aseg_011.srf aseg_013.srf aseg_015.srf aseg_017.srf aseg_026.srf aseg_043.srf aseg_046.srf aseg_049.srf aseg_051.srf aseg_053.srf aseg_058.srf aseg_251.srf aseg_253.srf aseg_255.srf
root@pipeline:/opt/data/freesurfer/118730# ls hem/
118730-5120-5120-5120-hem.fif 118730-head.fif 118730-inner_skull-5120.surf 118730-outer_skin-5120.surf 118730-outer_skull-5120.surf inner_skull.surf outer_skin.surf outer_skull.surf
118730-5120-5120-5120-hem-sol.fif 118730-inner_skull-5120.pnt 118730-outer_skin-5120.pnt 118730-outer_skull-5120.pnt inner_skull.asc outer_skin.asc outer_skull.asc watershed
root@pipeline:/opt/data/freesurfer/118730# ls label/
aparc.annot.a2009s.ctab lh.BA1.label lh.BA44.label lh.BA6.label lh.perirhinal.label rh.BA1.label rh.BA44.thresh.label rh.BA6.thresh.label rh.perirhinal.label
aparc.annot.ctab lh.BA1.thresh.label lh.BA44.thresh.label lh.BA6.thresh.label lh.V1.label rh.BA1.thresh.label rh.BA44.thresh.label rh.BA6.thresh.label rh.V1.label
aparc.annot.DKTatlas40.ctab lh.BA2.label lh.BA45.label lh.BA.annot lh.V1.thresh.label rh.BA2.label rh.BA45.thresh.label rh.BA.annot rh.V1.thresh.label
BA.ctab lh.BA2.thresh.label lh.BA45.thresh.label lh.BA.thresh.annot lh.V2.label rh.BA2.thresh.label rh.BA45.thresh.label rh.BA.thresh.annot rh.V2.label
BA.thresh.ctab lh.BA3a.label lh.BA4a.label lh.cortex.label lh.V2.thresh.label rh.BA3a.label rh.BA4a.label rh.cortex.label rh.V2.thresh.label
lh.aparc.a2009s.annot lh.BA3a.thresh.label lh.BA4a.thresh.label lh.entorhinal_exvivo.label rh.aparc.a2009s.annot rh.BA3a.thresh.label rh.BA4a.thresh.label rh.entorhinal_exvivo.label
lh.aparc.annot lh.BA3b.label lh.BA4p.label lh.MT.label rh.aparc.annot rh.BA3b.label rh.BA4p.label rh.MT.label
lh.aparc.DKTatlas40.annot lh.BA3b.thresh.label lh.BA4p.thresh.label lh.MT.thresh.label rh.aparc.DKTatlas40.annot rh.BA3b.thresh.label rh.BA4p.thresh.label rh.MT.thresh.label
root@pipeline:/opt/data/freesurfer/118730# ls mri/
aparc.a2009s+aseg.mgz aseg.auto_nocCseg.mgz brainmask.mgz lh.ribbon.mgz nu.mgz orig_nu.log ribbon.mgz talairach.log wm.asegedit.mgz
aparc+aseg.mgz aseg.mgz brain.mgz mri_nu_correct.mni.log nu_noneck.mgz orig_nu.mgz segment.dat talairach_with_skull_2.log wm.mgz
aseg.auto.mgz brain.finalsurfs.mgz ctrl_pts.mgz mri_nu_correct.mni.log.bak orig rawavg.mgz T1.mgz talairach_with_skull.log wmparc.mgz
aseg.auto_nocCseg.label.intensities.txt brainmask.auto.mgz filled.mgz norm.mgz orig.mgz rh.ribbon.mgz talairach.label.intensities.txt transforms wm.seg.mgz
root@pipeline:/opt/data/freesurfer/118730# ls scripts/
build-stamp.txt csurfdir potsurfcon.log potsurfcon.log.old ponsscc.cut.log recon-all.cmd recon-all.done recon-all.env recon-all.local-copy recon-all.log recon-all-status.log
root@pipeline:/opt/data/freesurfer/118730# ls src/
root@pipeline:/opt/data/freesurfer/118730# ls stats/
aseg.stats lh.aparc.DKTatlas40.stats lh.BA.stats lh.curv.stats lh.w-g.pct.stats rh.aparc.DKTatlas40.stats rh.BA.stats rh.curv.stats rh.w-g.pct.stats
lh.aparc.a2009s.stats lh.aparc.stats lh.BA.thresh.stats lh.entorhinal_exvivo.stats rh.aparc.a2009s.stats rh.aparc.stats rh.BA.thresh.stats rh.entorhinal_exvivo.stats wmparc.stats
root@pipeline:/opt/data/freesurfer/118730# ls surf/
lh.area lh.defect_borders lh.inflated.nofix lh.smoothwm lh.smoothwm.K2.crv lh.sulc rh.area.mid rh.defect_chull rh.jacobian.white rh.smoothwm.BE.crv rh.smoothwm.K.crv rh.thickness
lh.area.mid lh.defect_chull lh.jacobian.white lh.smoothwm.BE.crv lh.smoothwm.K.crv lh.thickness rh.area.pial rh.defect_labels rh.orig rh.smoothwm.C.crv rh.smoothwm.nofix rh.volume
lh.area.pial lh.defect_labels lh.orig lh.smoothwm.C.crv lh.smoothwm.nofix lh.volume rh.avg_curv rh.inflated rh.orig.nofix rh.smoothwm.FI.crv rh.smoothwm.S.crv rh.w-g.pct.mgh
lh.avg_curv lh.inflated lh.orig.nofix lh.smoothwm.FI.crv lh.smoothwm.S.crv lh.w-g.pct.mgh rh.curv rh.inflated.H rh.pial rh.smoothwm.H.crv rh.sphere rh.white
lh.curv lh.inflated.H lh.pial lh.smoothwm.H.crv lh.sphere lh.white rh.curv.pial rh.inflated.K rh.qsphere.nofix rh.smoothwm.K1.crv rh.sphere.reg
lh.curv.pial lh.inflated.K lh.qsphere.nofix lh.smoothwm.K1.crv lh.sphere.reg rh.area rh.defect_borders rh.inflated.nofix rh.smoothwm rh.smoothwm.K2.crv rh.sulc
root@pipeline:/opt/data/freesurfer/118730# ls tmp/
root@pipeline:/opt/data/freesurfer/118730# ls touch/
aparc2aseg.touch conform.touch lh.aparcstats2.touch lh.inflatel.touch lh.smoothwm2.touch mri_remove_neck.touch rh.avgcurv.touch rh.pial_surface.touch rh.tessellate.touch talairach.touch
aparc.a2009s2aseg.touch cortical_ribbon.touch lh.aparcstats3.touch lh.inflate2.touch lh.sphmorph.touch nu.touch rh.curvstats.touch rh.qsphere.touch rh.topofix.touch wmaparc.stats.touch
asegmerge.touch em_register.touch lh.aparcstats4.touch lh.jacobian.white.touch lh.sphreg.touch rh.aparc2.touch rh.final_surfaces.touch rh.smoothwm1.touch rh.white_surface.touch wmaparc.touch
ca_label.touch fill.touch lh.aparc.touch lh.potsurfcon.touch lh.surfvolume.touch rh.aparcstats2.touch rh.inflatel.touch rh.smoothwm2.touch segstats.touch wmsegment.touch
ca_normalize.touch inorm1.touch lh.avgcurv.touch lh.pial_surface.touch lh.tessellate.touch rh.aparcstats3.touch rh.inflate2.touch rh.sphmorph.touch skull_2.lta.touch
ca_register_inv.touch inorm2.touch lh.curvstats.touch lh.qsphere.touch lh.topofix.touch rh.aparcstats4.touch rh.jacobian.white.touch rh.sphreg.touch skull.lta.touch
ca_register.touch lh.aparc2.touch lh.final_surfaces.touch lh.smoothwm1.touch lh.white_surface.touch rh.aparc.touch rh.potsurfcon.touch rh.surfvolume.touch skull_strip.touch
root@pipeline:/opt/data/freesurfer/118730# ls trash/
root@pipeline:/opt/data/freesurfer/118730#

```

– fsaverage directory with the directories: label, mri, scripts, surf and the file mri.2mm

```

root@pipeline:/opt/data/freesurfer/fsaverage# ls
label mri mri.2mm scripts surf
root@pipeline:/opt/data/freesurfer/fsaverage# ls label/
lh.aparc.a2005s.annot lh.MT.label rh.aparc.a2005s.annot rh.MT.label
lh.aparc.a2009s.annot lh.MT.thresh.label rh.aparc.a2009s.annot rh.MT.thresh.label
lh.aparc.annot lh.oasis.chubs.annot rh.aparc.annot rh.oasis.chubs.annot
lh.aparc.label lh.oasis.chubs.ifc.label rh.aparc.label rh.oasis.chubs.ifc.label
lh.BA1.label lh.oasis.chubs.ipc.label rh.BA1.label rh.oasis.chubs.ipc.label
lh.BA1.thresh.label lh.oasis.chubs.ips.label rh.BA1.thresh.label rh.oasis.chubs.ips.label
lh.BA2.label lh.oasis.chubs.lateraltemporal.label rh.BA2.label rh.oasis.chubs.lateraltemporal.label
lh.BA2.thresh.label lh.oasis.chubs.medialpfc.label rh.BA2.thresh.label rh.oasis.chubs.medialpfc.label
lh.BA3a.label lh.oasis.chubs.mtl.label rh.BA3a.label rh.oasis.chubs.mtl.label
lh.BA3a.thresh.label lh.oasis.chubs.retrospenial.label rh.BA3a.thresh.label rh.oasis.chubs.retrospenial.label
lh.BA3b.label lh.oasis.chubs.tp.label rh.BA3b.label rh.oasis.chubs.tp.label
lh.BA3b.thresh.label lh.PALS_B12_Brodmann.annot rh.BA3b.thresh.label rh.PALS_B12_Brodmann.annot
lh.BA44.label lh.PALS_B12_labels.gii rh.BA44.label rh.PALS_B12_labels.gii
lh.BA44.thresh.label lh.PALS_B12_Lobes.annot rh.BA44.thresh.label rh.PALS_B12_Lobes.annot
lh.BA45.label lh.PALS_B12_OrbitoFrontal.annot rh.BA45.label rh.PALS_B12_OrbitoFrontal.annot
lh.BA45.thresh.label lh.PALS_B12_Visuatopic.annot rh.BA45.thresh.label rh.PALS_B12_Visuatopic.annot
lh.BA4a.label lh.perirhinal.label rh.BA4a.label rh.perirhinal.label
lh.BA4a.thresh.label lh.V1.label rh.BA4a.thresh.label rh.V1.label
lh.BA4p.label lh.V1.thresh.label rh.BA4p.label rh.V1.thresh.label
lh.BA4p.thresh.label lh.V2.label rh.BA4p.thresh.label rh.V2.label
lh.BA6.label lh.V2.thresh.label rh.BA6.label rh.V2.thresh.label
lh.BA6.thresh.label lh.Yeo2011_17NetworksConfidence_N1000.mgz rh.BA6.thresh.label rh.Yeo2011_17NetworksConfidence_N1000.mgz
lh.cortex.label lh.Yeo2011_17Networks_N1000.annot rh.cortex.label rh.Yeo2011_17Networks_N1000.annot
lh.entorhinal.label lh.Yeo2011_7NetworksConfidence_N1000.mgz rh.entorhinal.label rh.Yeo2011_7NetworksConfidence_N1000.mgz
lh.Medial_wall.label lh.Yeo2011_7Networks_N1000.annot rh.Medial_wall.label rh.Yeo2011_7Networks_N1000.annot
root@pipeline:/opt/data/freesurfer/fsaverage# ls mri
aparc.a2005s+aseg.mgz aseg.mgz lh.ribbon.mgz orig.mgz ribbon.mgz subcort_prob.log transforms
aparc.a2009s+aseg.mgz brainmask.mgz mni305.cor.mgz p.aseg.mgz subcort_mask.1mm.mgz subcort_prob.mgz
aparc+aseg.mgz brain.mgz orig rh.ribbon.mgz subcort_mask.1mm.README T1.mgz
root@pipeline:/opt/data/freesurfer/fsaverage# ls scripts/
build-stamp.txt make_average_volume.log mris_inflate rh.log recon-all.env recon-all.log
cvs_log_pre_31May2011.txt mris_inflate_lh.log recon-all.cmd recon-all.env.bak recon-all-status.log
make_average_surface.log mris_inflate.log recon-all.done recon-all.local-copy
root@pipeline:/opt/data/freesurfer/fsaverage# ls surf/
lh.area lh.orig lh.sulc rh.avg_curv rh.orig_avg rh.thickness
lh.avg_curv lh.orig_avg lh.thickness rh.avg_sulc rh.orig_avg.area.mgh rh.white
lh.avg_sulc lh.orig_avg.area.mgh lh.white rh.avg_thickness rh.pial rh.white_avg
lh.avg_thickness lh.pial lh.white_avg rh.cortex.patch.3d rh.pial_avg rh.white_avg.area.mgh
lh.cortex.patch.3d lh.pial_avg lh.white_avg.area.mgh rh.cortex.patch.flat rh.pial_avg.area.mgh rh.white_avg.H
lh.cortex.patch.flat lh.pial_avg.area.mgh lh.white_avg.H rh.curv rh.smoothwm rh.white_avg.K
lh.curv lh.smoothwm lh.white_avg.K rh.inflated rh.sphere
lh.inflated lh.sphere mris_preproc.surface.lh.log rh.inflated_avg rh.sphere.reg
lh.inflated_avg lh.sphere.reg mris_preproc.surface.rh.log rh.inflated_pre rh.sphere.reg.avg
lh.inflated_pre lh.sphere.reg.avg rh.area rh.orig rh.sulc
root@pipeline:/opt/data/freesurfer/fsaverage#

```

- the files: lh.EC_average, rh.EC_average .

Issues: The pipeline does not generate the volume mapping file and the normals for the surface which are needed for the TVB.

As input for the TVB it is also needed a T1 background nifti file format.

The "main_region.sh" file cannot be executed with Mrtrix3. We had to change the commands from the old version (Mrtrix0.2) to the new one, but even doing this, the file cannot be executed because of the differences between these two versions.