pyGluCEST Documentation

1. Process\_GluCEST.py

* Parameter <type>: “syrp” is for sagittal slices, “hippocampus” is for axial slices and “3d” is for 3d slices.
* **Pay attention to this ^^**
* **7tglucestage data 🡪 hippo**
* **Longglucest data 🡪 syrp**
* **ALL 3D DATA IS ‘3D’ !!!**
* Parameter <mask>: Define directory that contains masks and segmentation maps or output for them.
* Parameter <mask\_type>: “mni” is not yet fully implemented.
* Line 268 – 305 is just checking that all argument are valid and setting default values if certain arguments were not specified.

1. C library (\*.so in <library>) compilation

* All the .so files are compiled in LINUX, so you won’t have to recompile the .c files to run pyGluCEST on PMACS
* To run on other OS, recompile the .c files by running this command:
  + First *cd* into pyGluCEST\_2.0/library
  + Then run: *cc -fPIC -shared -o calc\_b0\_map.so calc\_b0\_map.c*

1. Hd-bet\_latest.sif

* Docker image in <images> directory that pyGluCEST uses to skull-strip structural images.
* Can be updated/installed from <https://hub.docker.com/r/pennsive/hd-bet/tags>

1. ssh singularity 01
2. singularity pull docker://pennsive/hd-bet:latest
3. exit
4. move the .sif file to pyGluCEST\_2.0/images

Required files:

* 3d INV2 NIFTI & 3d UNI NIFTI OR MPRAGE NIFTI
* CEST B1 map DICOM
* CEST WASSR (B0) map DICOM
* CEST none map DICOM
* CEST none map NIFTI
* CEST DICOM
* CEST NIFTI
* CEST T1 DICOM for 3d CEST processing

\*\*\*\* The DICOM files (.dcm) must be ordered properly within the DICOM folders.

Dependencies:

* FSL (fsl/6.0.3)
* **ANTS (ANTs/2.3.5)**
* **AFNI (afni\_openmp/20.1)**
* **Add dcm2niix as dependency**
* Conda environment (conda version : 4.9.2 / python version : 3.8.5.final.0)
* Python packages:
  + Nibabel (3.2.2)
  + Pydicom (2.2.2)

How to run:

1. Copy\_rawdata.py

* Copies data 3d INV2, UNI, & CEST acquisition dicoms to a new data directory
  + If you want to copy over different dicoms, make sure to the ‘required\_files’ variable on line 27
* Expects rawdata folder to be in <subid>/<sesid> format and copies to <subid\_sesid> directory
* Make sure to modify the paths on line 4-6
* Line 10 & 17 are searching for regex matches ‘^\d+’; it means it is searching for strings that start with at least 1 digit

1. Convert2nifti.py

* Calls on convert2nifti.sh to create all the necessary niftis (INV, UNI, cest, none)
  + If you want different niftis, modify line 13-15 in convert2nifti.sh
* Make sure to modify the paths on line 3-6
* **Don’t move to /data folder until after running this**
* **Also don’t make any other folders until this step is done**
* **Replace melliott’s convert2nifti.sh with dcm2niix**

1. Preprocess\_cest.sh

* Runs pyGluCEST and submits a job per case (<subid>\_<sesid>)
* 2 positional arguments: input data directory and output directory
* Modify line 11 if you need to run a different version of pyGluCEST
* Modify the arguments!
* Example output folder name: <preprocessed>

1. Register\_to\_MNI.py

* **Example call: python scripts\_copy/register\_to\_MNI.py -d 7t\_cest\_analysis/preprocessed -o 7t\_cest\_analysis/preprocessed -s UNI -a 7t\_cest\_analysis/atlases -r 0.7**
* Has many arguments – make sure to look at the helper function!!
* Calls on register\_to\_MNI.sh
* The structural image that you want to register to the MNI\_brain will be bias corrected and masked if not already
* If your data is not in 1mm, make sure to resample the atlases and MNI\_brain to desired resolution by using *ResampleImage*
  + *ResampleImage 3 MNI152\_T1\_1mm\_brain.nii.gz MNI152\_T1\_0.8mm\_brain.nii.gz 0.8223684430X0.8223684430X0.8199999928 0 4*
  + *ResampleImage 3 HarvardOxford-cort-maxprob-thr25-1mm.nii.gz HarvardOxford-cort-maxprob-thr25-0.8mm.nii.gz 0.8223684430X0.8223684430X0.8199999928 0 1 4*
  + **Check fslinfo analysis/data/<sub\_id>-UNI.nii for dimensions**
* The <atlases> directory must be in this format
  + <atlases>
    - <MNI>
      * MNI162\_T1\_\*mm\*.nii.gz
    - <HarvardOxford>
      * HarvardOxford-cort-maxprob-thr25-\*mm.nii.gz
      * HarvardOxford-Cortical.xml
      * HarvardOxford-Subcortical.xml
* Set output directory as the output directory from step 3

1. Process\_cest.py

* **Example call: python scripts\_copy/process\_cest.py -d 7t\_cest\_analysis/preprocessed -p 7t\_cest\_analysis/preprocessed -o 7t\_cest\_analysis/postprocessed -a 7t\_cest\_analysis/atlases -l 7t\_cest\_analysis/logs -s scripts\_copy/process\_cest.sh -m python -r 1 -t UNI**
* Has many arguments – make sure to look at the helper function!!
* Create a directory called logs before running this script
* Calls on process\_cest.sh that does more correction on the CEST and outputs subject space 3d and CEST space atlases
* Example output folder name: <postprocessed>

1. Extract\_rois.sh

* Has 3 positional arguments: path to postprocessed CEST data, corrected INV2/mprage NIFTI, output path
  + Path to corrected INV2/mprage NIFTI would be the ‘-m’ argument you used in pyGluCEST (step 3), so probably ‘preprocessed’ directory
* Example output folder name: <output\_measures>
* **Looking for ‘postprocessed/91422\_11753/atlases/UNI/91422\_11753-2d-HarvardOxford-sub-bin.nii.gz’**
  + **Only have cest-HarvardOxford and HarvardOxford**
  + **Replace ‘2d’ with ‘cest’**

1. Compile\_all\_subjects.R

* Need to change the path on line 1 before running

1. Compile\_all\_subjects.py

* Make sure you look at the helper function
* *Ex) python scripts/compile\_all\_subjects.py -i output\_measures -o output\_measures -x atlases/HarvardOxford -g CEST\_Groups.csv*