BCP preprocess pipeline

**1. Introduction**

The BCP fMRI data processing pipeline is in the form of a bash script. It is easy to run for someone with knowledge of the BASH scripting language. It is an infant-dedicated pipeline[1] which include the following steps: head-motion and distortion correction, infant-dedicated spatial registration, one-time resampling of fMRI data and individual independent component analysis (ICA)-based extensive denoising[2], similar to the state-of-the-art HCP pipeline.

**2. Pre-requisite**

The pipeline can be run on a Linux system computer (My own PC is Ubuntu system and the pipeline runs well), including the longleaf server. One must install ANTS (ver. 2.3.1, <http://stnava.github.io/ANTs/>), FSL (ver. 5.0.10/5.0.11, ver. 6 may causes some mistakes, <https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/FslInstallation>), AFNI (ver. 17.2.09, [https://afni.nimh.nih.gov](https://afni.nimh.nih.gov/)) and python2 to run the script. If you run it on longleaf, simply just add these modules each time you run the code (using ‘module add’ command). Note that on longleaf, you can use command ‘SLURM\_ARRAY\_TASK\_ID’ to run hundreds of subjects simultaneously. If run it on your own PC, use the for loop to preprocess data one by one. The estimated time for running one sample is approximately 6 hours in my own PC.

**3. Input and Output**

The directory structure of the input folder has to be arranged as shown in Fig. 1. As shown in Fig. 1, there are two main folders in the input folder, i.e., “All” and “noise\_comp\_detection”[2] folders, along with some other bash scripts. The main code is in the “BCP\_preprocess.sh” file. “mask\_transform.sh” and “transform.sh” are used to transform the seed-based correlation generated brain networks[3] from native space into MNI space.

电脑屏幕的照片上有文字

描述已自动生成

Fig. 1 Directory structure of the input folder.

Under the “All” folder, there are two folders: “After\_Sorted” and “MNI\_Template”. What’s in “MNI\_Template” are some templates in MNI space which we use to extract the time series or generate the seed-based correlation maps in the last few steps.

In the “After\_Sorted” folder, there are the data input folders which store the data to be preprocessed along with two configuration files: “acqparams.txt” and “b02b0.cnf”. For example, as shown in Fig. 1, we have folder “Sample988”. Note that in our current scripts, it’s better to name the folder as “SampleXXXX” which will make it convenient for latter analysis. We can put “Sample989”, “Sample990”, “Sample991” and so on in “After\_Sorted” folder if we have many samples.

In the “Sample988” folder, the following files should be prepared in advance.

1. REST.nii.gz is the raw fMRI data file which can be directly obtained from the /BCPspace/BCPshare/BCP\_NIFTI folder in our lab’s servers by anyone who has access to the /BCPspace path;
2. SBRef.nii.gz is the single band reference file for each sample. Note that not all samples have this reference file. If we cannot find the reference file for one sample, we use the 10th volume of the REST.nii.gz as the reference file;
3. T1stripped.nii.gz and T2stripped.nii.gz are the structural files which can be directly obtained from /BCPspace/BCPshare/BCP\_NIFTI folder in our lab’s servers;
4. tissuelabel.nii.gz is generated by Dr. Wangli’s algorithms ([https://ibeat.wildapricot.org](https://ibeat.wildapricot.org/)/), Zhengwang can provide this result. This file is used to register EPI image to T1 image.
5. FieldMapAP.nii.gz and FieldMapPA.nii.gz are essential for each sample. These are the phase encoding direction files which can also be directly obtained from /BCPspace/BCPshare/BCP\_NIFTI folder in our lab’s servers;
6. sbref.txt is a file generated when sorting. There is a number stored in a text file (‘0’ or ‘1’). ‘0’ represents there is no SBRef.nii.gz for this sample, then consider the 10th volume of the REST.nii.gz as the reference file. ‘1’ represents there is a SBRef.nii.gz for this sample in the server;
7. Phase.txt is the file indicated the phase direction (‘x’ or ‘y’). ‘x’ denotes AP while ‘y’ denotes PA;
8. ICANormalLabel.txt can be ignored though it can be generated during sorting.

In the “BCP\_preprocess.sh” file, some directories need to be stated at beginning (Red represents the input while purple denotes the generated folders).

DATAPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/After\_Sorted"

OUTPUTPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/Functional\_Preprocessed"

FILEPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/MNI\_Template"

STRUCPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/Structural\_Preprocessed"

ATLASPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/NativeSpace\_atlas"

OUTPUTRegPic="/media/zz/data/test\_BCPpipeline/test\_data/All/Structural\_Preprocessed/PICforRegCheckMNItoT2"

ICANORMPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/ICA\_Normalized"

TIMECOURSEPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/TIMECOURSE"

NETWORK="/media/zz/data/test\_BCPpipeline/test\_data/All/NETWORK"

CLEANDATAPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/AFTERDENOISE"

MINIOUTPUTPATH="/media/zz/data/test\_BCPpipeline/test\_data/All/Minimal\_Preprocessed"

We can find all the following directories in the same subdirectory under the “All” directory. Except “After\_Sorted” and “MNI\_template”, all other directories are generated during preprocessing to store all the middle results.

1. After\_Sorted is the folder which stores the data after sorting;
2. Functional\_Preprocessed stores the preprocessed functional data;
3. MNI\_Template stores the initial MNI templates, including the Harvard 112, Zalesky980, Zalesky1024 and HOA1024 four atlases;
4. Structural\_Preprocessed stores the preprocessed structural data, mainly the middle results of registration from structural images to MNI space using ants;
5. NativeSpace\_atlas stores the initial MNI templates in each sample’s native space;
6. Structural\_Preprocessed/PICforRegCheckMNItoT2 stores the pictures to check the registration quality from structural images to MNI space using ants;
7. ICA\_Normalized stores the ICA result;
8. TIMECOURSE stores the time series extracted by the four atlases respectively;
9. NETWORK stores the seed-based correlation brain maps. We may also need the seed-based correlation brain maps in MNI space. Use ‘transform.sh’ and ‘mask\_transform.sh’ to convert the brain maps from native space to MNI space for further analysis and the results will be stores in the MNI\_NETWORK folder;
10. AFTERDENOISE stores final data cleaned by TaeEui’s[2] denoising framework;
11. Minimal\_Preprocessed stores the minimal preprocessed functional data.

Please note that it’s better to save all these middle results because you may need to use them in future analysis. All the unnecessary middle results have been deleted during preprocess. The output folders including TIMECOURSE, NETWORK and AFTERDENOISE are the output of this pipeline and most of the following analysis will be using the results from these three folders.

**4. Demo code**

As I mentioned before, if running on longleaf, one can use command ‘SLURM\_ARRAY\_TASK\_ID’ to run hundreds of subjects simultaneously. If running on PC, one needs to use the for loop to preprocess the data one by one. For running the script, simply use “sbatch BCP\_preprocess.sh” on longleaf or “bash BCP\_preprocess.sh” on your own PC as shown below. Below is the sample code for running program on the longleaf. One needs to get familiar with the usage of the longleaf server if wanting to run the script on longleaf. There are also detailed comments in the script which can be referred to.

#!/bin/bash

#SBATCH --error=./result\_record/job.%J.err

#SBATCH --output=./result\_record/job.%J.out

#SBATCH --ntasks-per-node=1

#SBATCH -N 1

#SBATCH --time=02-00:00:00

run on longleaf

#SBATCH --mem=30g

#SBATCH -a 0-100

……

input=(1 2 3 4 5 … 101)

……

i=${input[$SLURM\_ARRAY\_TASK\_ID]}

#!/bin/bash

for i in 1 2 3 …. 101

run on PC

do

……

done

I have moved all the preprocessed data and middle results to /proj/ptyaplab/users/hanzhang/projects/zhouzhen/BCP\_All and /proj/ptyaplab/users/hanzhang/projects/zhouzhen/BCP\_All\_Second on longleaf server.

**References**

1. Zhou, Z., et al., *Multi-layer Temporal Network Analysis Reveals Increasing Temporal Reachability and Spreadability in the First Two Years of Life*, in *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*. 2019. p. 665-672.

2. Kam, T.-E., et al. *A Deep Learning Framework for Noise Component Detection from Resting-State Functional MRI*. 2019. Cham: Springer International Publishing.

3. Smith, S.M., et al., *Correspondence of the brain's functional architecture during activation and rest.* Proceedings of the National Academy of Sciences, 2009. **106**(31): p. 13040-13045.