ANTs-LONGLEAF Manual ~

Tengfei Li @ UNC

May 9, 2020

1. Login on the department Longleaf cluster. For Windows users, using x-win32 or putty, and connect to the host “longleaf”; for Mac users, open a terminal to type the following command.

ssh username@longleaf.unc.edu

And then type you UNC usrname and password. Install software Winscp or sshClient for windows, or filezilla or cyberduck for Mac. These softwares are for **uploading/downloading** files from longleaf. Install Mango and Mricron software to check the result quality.

1. The related T1 images with subject ID= SubID1, SubID2, SubID3,... is collected at

/PATH/T1\_raw/SubID1/T1.nii.gz

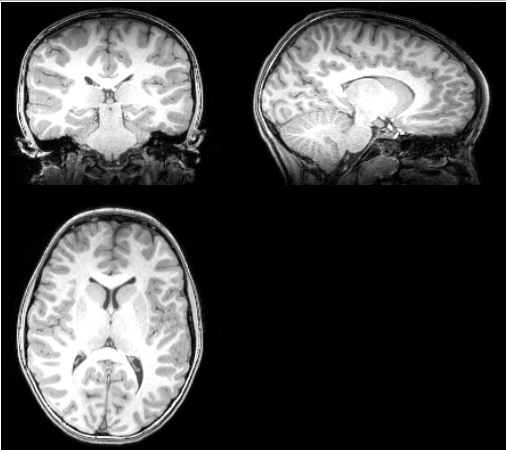
/PATH/T1\_raw/SubID2/T1.nii.gz

… …

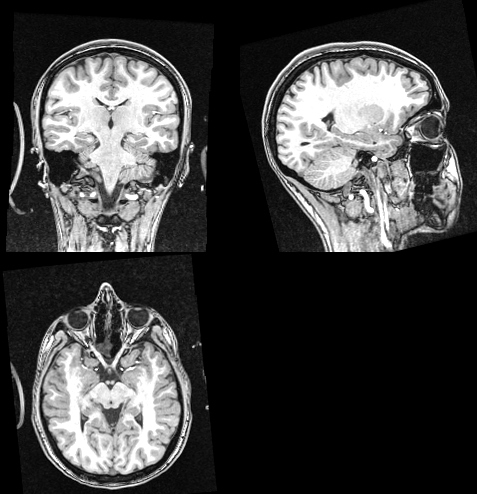
If you want to process all SubID start with “1” (similar for other numbers), you can copy files at /OtherPATH/SubID/T1.nii.gz to /PATH/T1\_raw using the command

cp -r /OtherPATH/1\* /PATH/T1\_raw/

The T1 image example opened by Mricron or Mango software for PNC dataset:



The T1 image example opened by Mricron or Mango software for PING dataset:



Upload files/folders "ANTs\_longleaf.m", “Scripts”, ”Template\_30”,”label.csv”, “label.txt”, “label1.txt” to “/PATH/” using uploading softwares. Modify the string “PATH” in "ANTs\_longleaf.m" to your specific PATH. (Remember: double check to make sure you save your modification on server). Type the following command at the terminal to run the matlab file ANTs\_longleaf.m.

cd /PATH/

module load matlab

chmod 775 /PATH/Scripts/\*

matlab -nojvm

ANTs\_longleaf

exit

This will create a new folder “code” at /PATH/ and generate [AntsCorticalThickness](https://github.com/stnava/ANTs/blob/master/Scripts/antsCorticalThickness.sh) pipeline scripts for all subjects in “code”.

1. **Run the** [AntsCorticalThickness](https://github.com/stnava/ANTs/blob/master/Scripts/antsCorticalThickness.sh) **Pipeline in this step.**

**Procedure:**

First open the script “/code/ANTs\_batAll.sh” , which reads:

“

#!/bin/bash

sbatch ./ANTS\_bat1.pbs

sbatch ./ANTS\_bat2.pbs

sbatch ./ANTS\_bat3.pbs

sbatch ./ANTS\_bat4.pbs

sbatch ./ANTS\_bat5.pbs

sbatch ./ANTS\_bat6.pbs

sbatch ./ANTS\_bat7.pbs

… …

sbatch ./ANTS\_batN.pbs

”

This script will submit N jobs at one time.

If N>1000, modify the scripts by deleting the line “sbatch ./ANTS\_batj.pbs” for all j: j>1000. It is recommended to run at most 1000 jobs at a time on LONGLEAF cluster.

“

#!/bin/bash

sbatch ./ANTS\_bat1.pbs

sbatch ./ANTS\_bat2.pbs

sbatch ./ANTS\_bat3.pbs

sbatch ./ANTS\_bat4.pbs

sbatch ./ANTS\_bat5.pbs

sbatch ./ANTS\_bat6.pbs

sbatch ./ANTS\_bat7.pbs

… …

sbatch ./ANTS\_bat500.pbs

”

After modification of this shell script, open and check ANTS\_bat1.pbs file which should be found in “/code”:

“

#PBS -l nodes=1:ppn=3 -l walltime=23:59:59

#PBS -N ANTs\_dqshtc\_1

#PBS -o ANTs\_dqshtc\_1.out

#PBS -j oe

#!/bin/bash

setenv ANTSPATH /software/x86\_64/ANTs/2.1.0/

setenv ITK\_GLOBAL\_DEFAULT\_NUMBER\_OF\_THREADS 3

bash ${ANTSPATH}antsCorticalThickness.sh -d 3 -n 3 -w 0.25 \

-a /scratch/tli3/PNC/T1\_raw/600009963128/T1.nii \

-o /scratch/tli3/PNC/T1\_raw/600009963128/Output/ants \

-e /scratch/tli3/PNC//Template\_30/T\_template0.nii.gz \

-t /scratch/tli3/PNC//Template\_30/T\_template0\_BrainCerebellum.nii.gz \

-m /scratch/tli3/PNC//Template\_30/T\_template0\_BrainCerebellumProbabilityMask.nii.gz \

-f /scratch/tli3/PNC//Template\_30/T\_template0\_BrainCerebellumExtractionMask.nii.gz \

-p /scratch/tli3/PNC//Template\_30/priors%d.nii.gz

”

Here ppn=3 uses 3 threads for each job which will speed up 4 times. This should be consistent with the number 3 in the line

setenv ITK\_GLOBAL\_DEFAULT\_NUMBER\_OF\_THREADS 3

A job with threads 4 will have a longer pending time; you can also change this to ppn=1 and setenv ITK\_GLOBAL\_DEFAULT\_NUMBER\_OF\_THREADS 1

This can be changed in "ANTs\_dqshtc.m" and rerun step 2.

Now run the following commands in terminal to submit parallel jobs.

cd code/

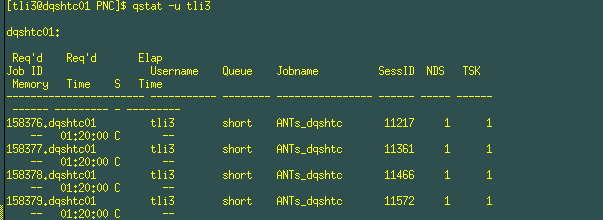
chmod 775 ANTs\_batAll.sh

./ANTs\_batAll.sh

Then min (1000, N) .pbs jobs will be submitted to the cluster. After 12 hours type commands:

squeue –u usrname

to check if all jobs are done.



You can also use

squeue –u usrname | grep R –c

to calculate how many jobs are still running, and

squeue –u usrname | grep Q –c

to calculate how many jobs are still pending.

Wait until all jobs are done and then repeat the procedure in this step to run another

N0=min (1000, N-1000)

jobs by modifying the shell script “/code/ANTs\_batAll.sh”:

“

#!/bin/bash

sbatch ./ANTS\_bat1001.pbs

sbatch ./ANTS\_bat1002.pbs

sbatch ./ANTS\_bat1003.pbs

sbatch ./ANTS\_bat1004.pbs

sbatch ./ANTS\_bat1005.pbs

sbatch ./ANTS\_bat1006.pbs

sbatch ./ANTS\_bat1007.pbs

… …

sbatch ./ANTS\_batN.pbs

”

and submit parallel jobs.

Repeat the procedure until all jobs are done.

**Details of Pipeline:**

Main function of ANTs: antsCorticalThickness.sh

Input: T1.nii (target raw T1; see image examples in Step 2)

Input: T\_template0.nii.gz (T1 template)

Input: T\_template0\_BrainCerebellum.nii.gz (T1 template with skull off)

Input: T\_template0\_BrainCerebellumExtractionMask.nii.gz (mask)

Input: T\_template0\_BrainCerebellumProbabilityMask.nii.gz (prob mask)

Input: priors1.nii.gz, priors2.nii.gz, priors3.nii.gz, priors4.nii.gz, priors5.nii.gz, priors6.nii.gz (label 1 -> csf; label 2 -> gm; label 3 -> wm; label 4 -> deep gm; label 5 -> brain stem; label 6 -> cerebellum)

Parameters –n: number of segmentation iterations. We use the default =3.

Parameters –w: Atropos prior segmentation weight. The weight of priors for the segmentation. We use the default = 0.25.

See all inputs in the following images.

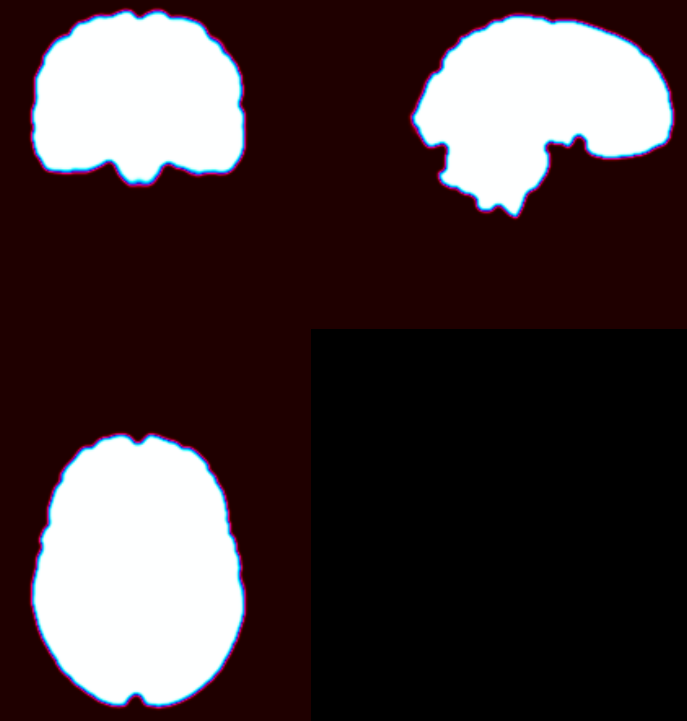
T1 template:



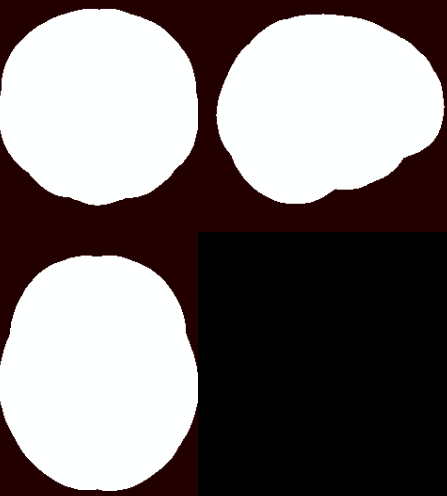
T1 template with skull off:



Probability Mask

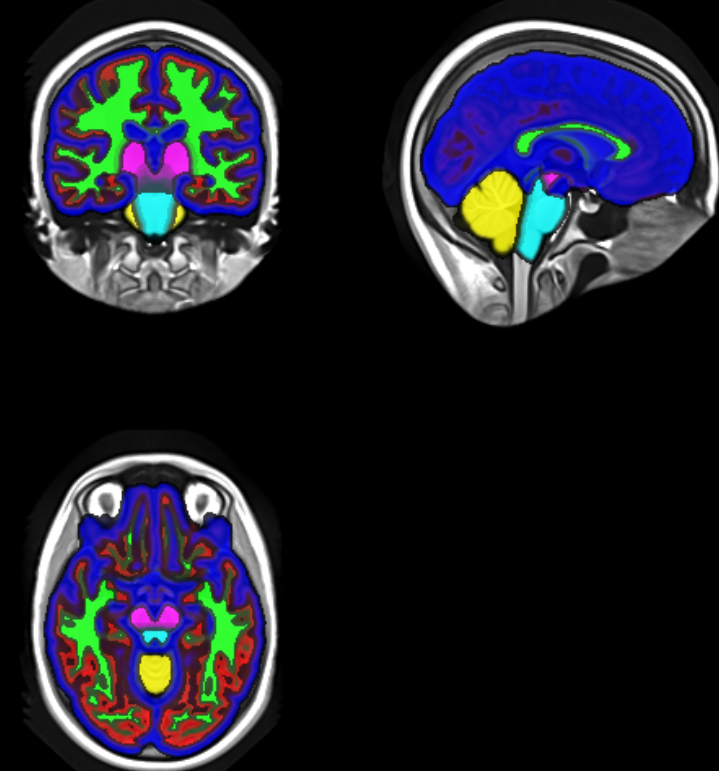


Extracted Mask



Priors of 6 labels

(blue:CSF; red: GM; green: WM; pink: deep GM; cyan: brainstem; yellow: cerebellum)



After within twelve hours, we will get all output files.

Output: antsBrainExtractionMask.nii.gz (brain mask for the raw T1 image)

Output: antsBrainSegmentation0N4.nii.gz (N4 bias-corrected T1 image)

Output: antsBrainSegmentation.nii.gz (segmentation results)

Output: antsBrainSegmentationPosteriors1.nii.gz, antsBrainSegmentationPosteriors2.nii.gz, antsBrainSegmentationPosteriors3.nii.gz, antsBrainSegmentationPosteriors4.nii.gz, antsBrainSegmentationPosteriors5.nii.gz, antsBrainSegmentationPosteriors6.nii.gz (Posteriors for the 6 labels)

Output: antsSubjectToTemplate1Warp, antsSubjectToTemplate0GenericAffine.mat (Transforms to be used when warping images from the subject space to the template space

antsApplyTransforms -d 3 -i imageInSubjectSpace.nii.gz -o imageInTemplateSpace.nii.gz \

-t outputPrefixSubjectToTemplate1Warp.nii.gz -t SubjectToTemplate0GenericAffine.mat -r template.nii.gz

)

Output: antsTemplateToSubject0Warp, antsTemplateToSubject1GenericAffine.mat

(Transforms to be used when warping images from the template to the subject space

antsApplyTransforms -d 3 -i imageInTemplateSpace.nii.gz -o imageInSubjectSpace.nii.gz \

-t outputPrefixTemplateToSubject1Affine.mat -t outputPrefixTemplateToSubject0Warp.nii.gz -r template.nii.gz

)

Output: antsCorticalThickness.nii.gz, antsCorticalThicknessNormalizedToTemplate.nii.gz (cortical thickness map on subject space and template space, respectively. Used for voxelwise analysis)

Output: antsSubjectToTemplateLogJacobian.nii.gz (Log of Jacobian matrix of the transformation. Used for voxelwise analysis)

Output: antsExtractedBrain0N4.nii.gz (N4 bias-corrected T1 image with skull off)

Output: antsRegistrationTemplateBrainMask.nii.gz (brain mask on the template space)

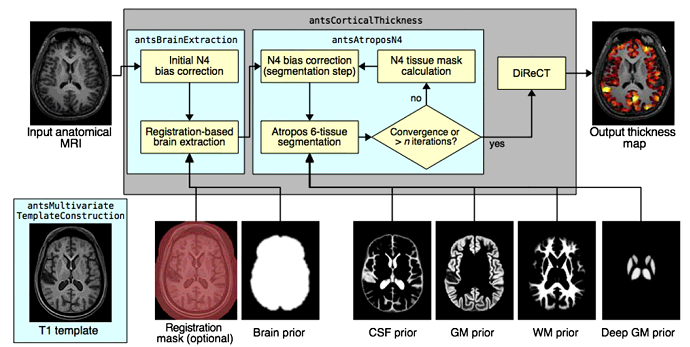
Output: antsBrainNormalizedToTemplate.nii.gz (extracted brain registered onto the template space)

Output: antsbrainvols.csv (a .csv file including PearsonCorrelation---correlation between antsBrainNormalizedToTemplate.nii.gz and the brain template, BVOL---whole brain volume, GVol---grey matter volume, WVol---white matter volume and ThicknessSum---a sum of all intensities of the corticalthickness map (before registered).

Output: antsCorticalThicknessTiledMosaic.png, antsBrainSegmentationTiledMosaic.png (These two inputs are slices of antsCorticalThickness.nii.gz and brain segmentation (antsBrainSegmentation.nii.gz or posterior not sure) overlaid on the template and they are for quality control purposes)

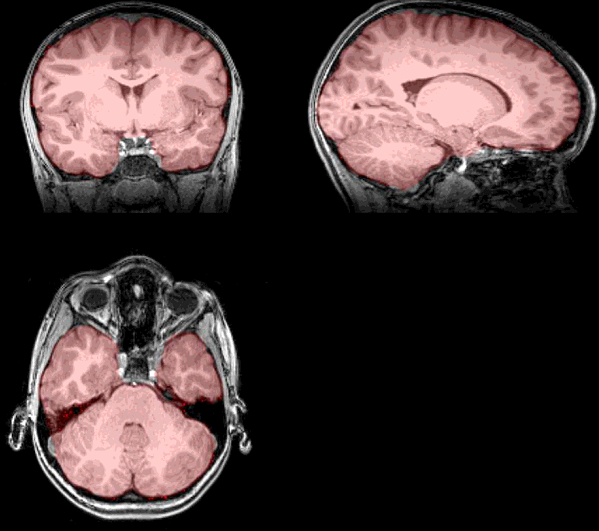
Output: antsBrainSegmentationConvergence.txt (iteration and posterior value)

See the following pipeline illustration for reference. [(Tustison, et.al 2014)](http://ac.els-cdn.com/S1053811914004091/1-s2.0-S1053811914004091-main.pdf?_tid=e8169170-c7ea-11e6-9e90-00000aab0f27&acdnat=1482372421_829c5017cfcff8e5a6fce7dec2ee6150)

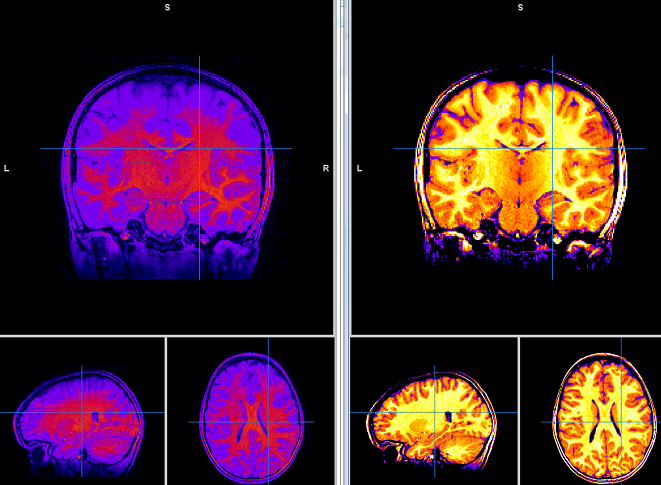


Output images are shown in the following.

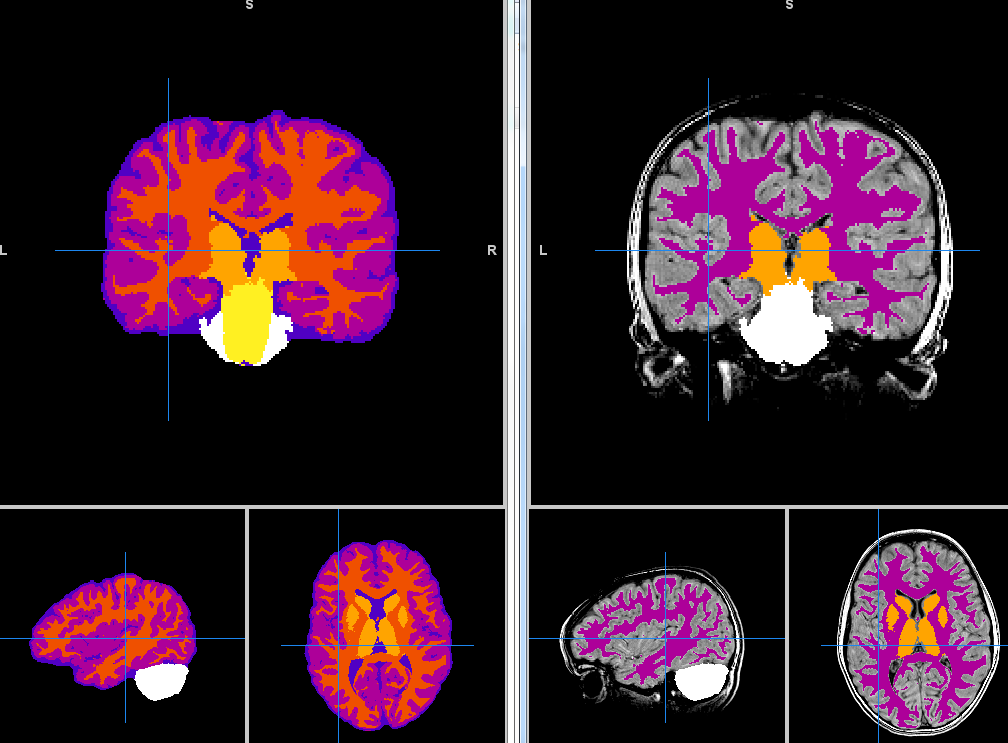
T1.nii overlaid by antsBrainExtractionMask.nii.gz



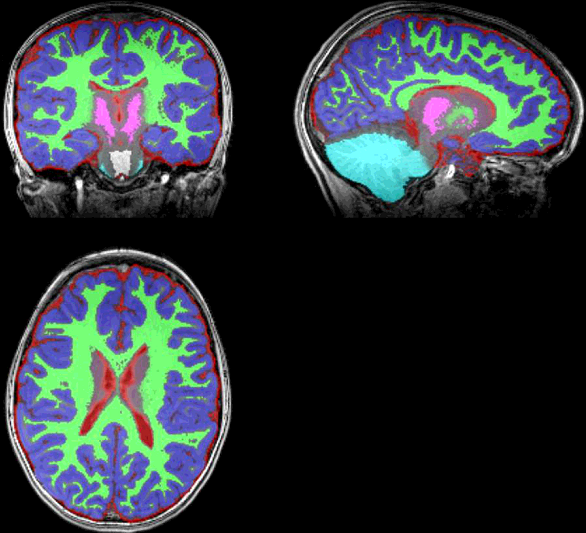
T1.nii and antsBrainSegmentation0N4.nii.gz



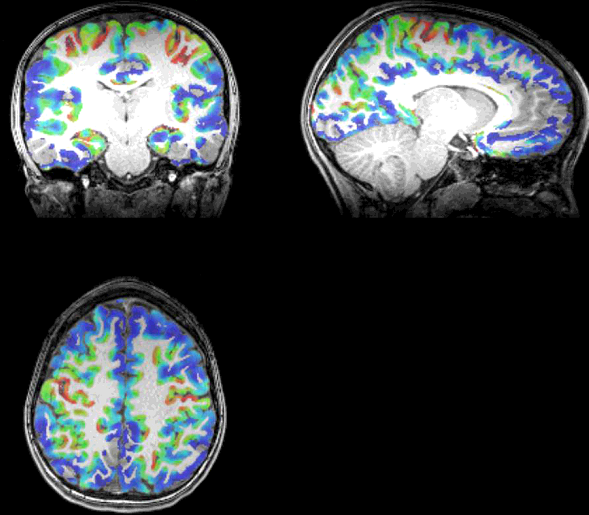
antsBrainSegmentation.nii.gz and it overlaid on antsBrainSegmentation0N4.nii.gz



T1.nii overlaid by antsBrainSegmentationPosteriors%d.nii, %d=1, 2, …6



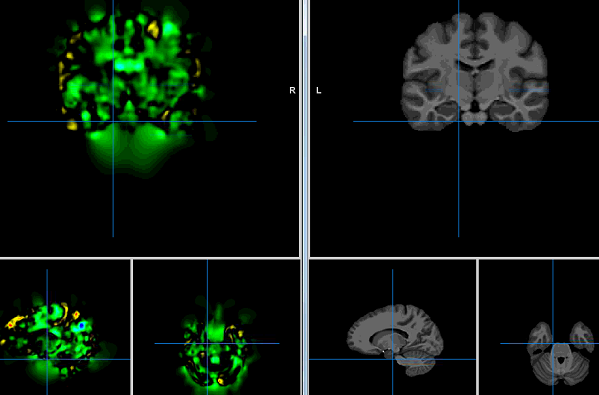
antsCorticalThickness.nii.gz overlaid on T1.nii



antsCorticalThicknessNormalizedToTemplate.nii.gz overlaid on T\_template0.nii.gz



antsSubjectToTemplateLogJacobian.nii.gz and T\_template0\_BrainCerebellum.nii.gz displayed together



1. **Extract the Output this step**.

Upload file getcsv.m to your PATH. Wait until all jobs in step 3 is done. You can use

squeue –u username

to check the job status. Then, modify PATH in your getcsv.m to your own path and set Finalmove=0. Then type the following commands:

cd /PATH/

module load matlab

matlab -nojvm

getcsv

exit

This will create 4 new folders

csv

antsCorticalThicknessNormalizedToTemplate antsSubjectToTemplateLogJacobian

QC

The first three folders include

antsbrainvols.csv

antsCorticalThicknessNormalizedToTemplate.nii.gz antsSubjectToTemplateLogJacobian.nii.gz

for each subject, respectively; folder QC include two subfolders

brainsegmentation

corticalthickness

which include

antsBrainSegmentationTiledMosaic.png

antsCorticalThicknessTiledMosaic

respectively. Type

squeue –u username

to check this job status. After this job is done, type

cd /PATH/

module load matlab

matlab -nojvm

ANTs\_longleaf

exit

again, and open and check the following “sh” file

/code/ANTs\_batAll.sh

If it reads

“

#!/bin/bash

”

This means all jobs are completed successfully; otherwise, e.g.,

“

#!/bin/bash

sbatch ./ANTS\_bat101.pbs

sbatch ./ANTS\_bat210.pbs

“

It means some jobs are not corrected executed due to time limit, i.e. ANTS\_bat101.pbs and ANTS\_bat210.pbs. If this happens, type the following commands:

cd code/

chmod 775 ANTs\_batAll.sh

./ANTs\_batAll.sh

to re-run the jobs with error message. After all jobs are completed, run the beginning of this step (getcsv and ANTs\_longleaf) again and check ANTs\_batAll.sh file again. Repeat this again and again until ANTs\_batAll.sh reads

“

#!/bin/bash

”

which means all jobs are successful.

Then you can open matlab to run summ.m by

matlab -nojvm

summ(‘csv’)

exit

to combine all csv files into a large csv table vol.csv.

1. **Do the quality control in this step.**

Download the folder QC into your local computer. Check each

Brain segmentationimage in

/QC/brainsegmentation

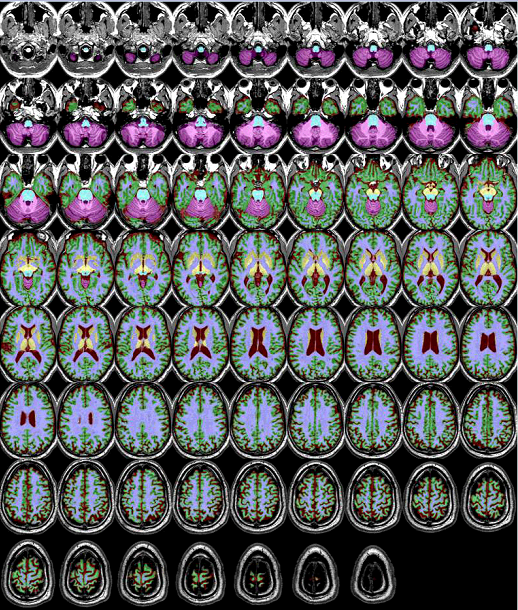
And check brain corticalthickness in

/QC/corticalthickness

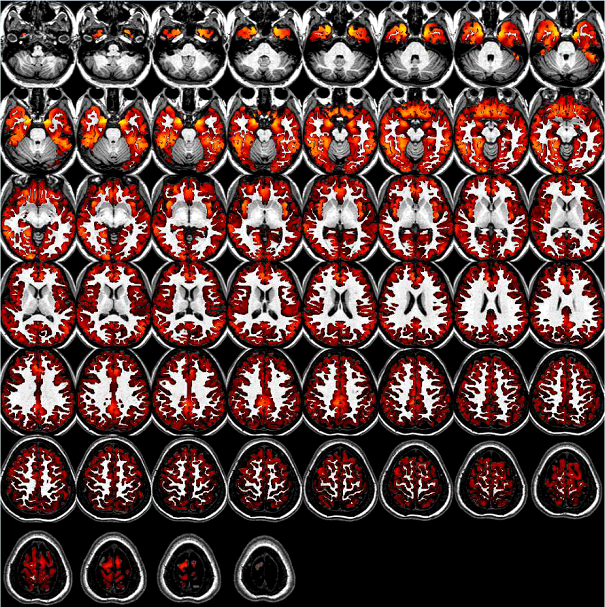
If the results is far from the following standard segmentation

and corticalthickness results, email me.

Standard segmentation results

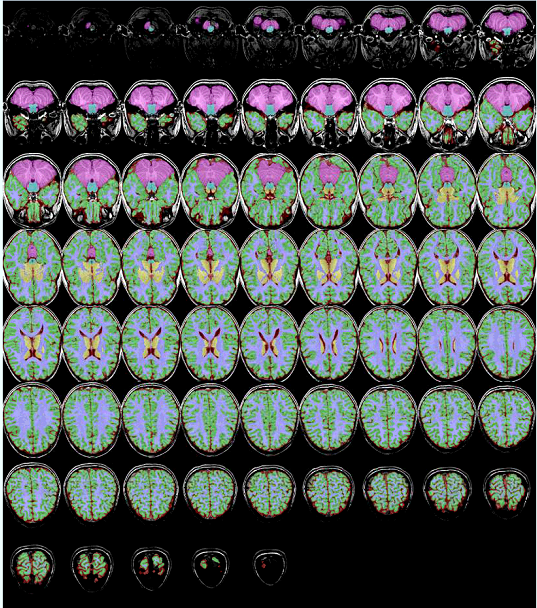


Standard corticalthickness result

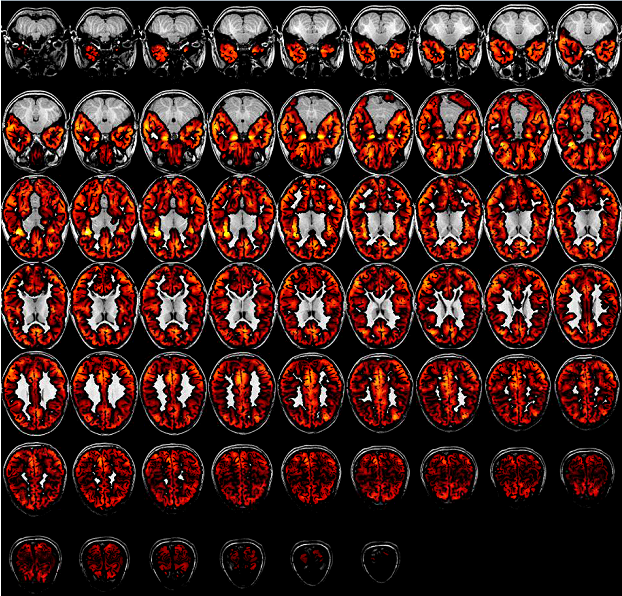


One example as follows can be considered OK.

An example that can be considered OK

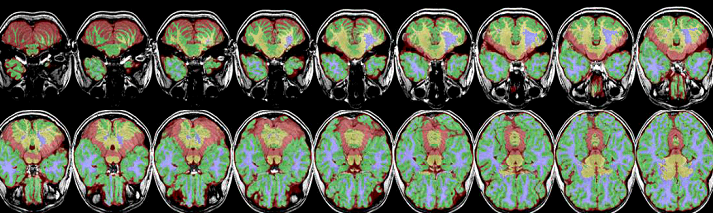


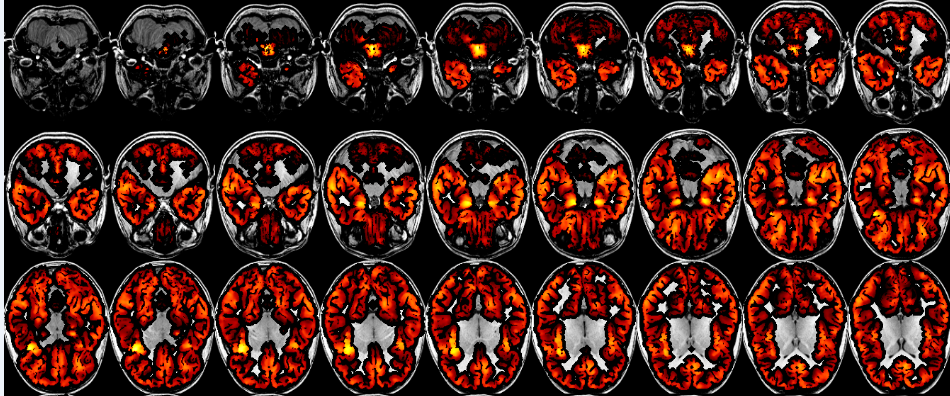
An example that can be considered OK



**The following example is a bad one, and needs to be re-run or deleted.**

**A bad case**





1. **Run the** [antsJointLabelFusion.sh](https://github.com/stnava/ANTs/blob/master/Scripts/antsJointLabelFusion.sh) **Pipeline in this step.**

**Procedure:**

Upload folder oasis\_label, and file ANTs\_longleaf1.m to your PATH. Modify the PATH in "ANTs\_longleaf1.m" to your specific PATH. Type the following command at the terminal.

cd /PATH/

module load matlab

matlab -nojvm

ANTs\_longleaf1

exit

Check the status of this job using

squeue –u usrname

If it is done, open and check file

/code/ANTS\_bat1\_1.pbs

which reads

“

#PBS -l nodes=1:rhel7:ppn=1 -l walltime=23:59:59

#PBS -N ANTs\_dqshtc\_1

#PBS -o ANTs\_dqshtc\_1.out

#PBS -j oe

echo 'Job ran on '

cat $PBS\_NODEFILE

#!/bin/bash

setenv ANTSPATH /software/x86\_64/ANTs/2.1.0.post681-g09854/

setenv ITK\_GLOBAL\_DEFAULT\_NUMBER\_OF\_THREADS 1

setenv inputPath /scratch/tli3/PNC/

bash ${ANTSPATH}/antsJointLabelFusion.sh -d 3 -c 0 -x or \

-o /scratch/tli3/PNC/T1\_raw/SubID2/labeloutput/SubID2\_ \

-t /scratch/tli3/PNC/T1\_raw/SubID2/Output/antsExtractedBrain0N4.nii.gz \

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-1\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-1\_DKT31\_CMA\_labels.nii.gz \

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-2\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-2\_DKT31\_CMA\_labels.nii.gz \

… …

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-20\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-20\_DKT31\_CMA\_labels.nii.gz

rm /scratch/tli3/PNC/T1\_raw/SubID1/labeloutput/SubID1\_Intensity.nii.gz

bash ${ANTSPATH}/antsJointLabelFusion.sh -d 3 -c 0 -x or \

-o /scratch/tli3/PNC/T1\_raw/SubID1/labeloutput/SubID1\_ \

-t /scratch/tli3/PNC/T1\_raw/SubID1/Output/antsExtractedBrain0N4.nii.gz \

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-1\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-1\_DKT31\_CMA\_labels.nii.gz \

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-2\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-2\_DKT31\_CMA\_labels.nii.gz \

… …

-g ${inputPath}/oasis\_label/brain/OASIS-TRT-20-20\_brain.nii.gz -l ${inputPath}/oasis\_label/label/OASIS-TRT-20-20\_DKT31\_CMA\_labels.nii.gz

rm /scratch/tli3/PNC/T1\_raw/SubID1/labeloutput/SubID1\_Intensity.nii.gz

”

If it is correct, type the following commands from terminal:

cd code/

chmod 775 ANTs\_batAll.sh

./ANTs\_batAll.sh

You can use

squeue –u usrname | grep R –c

to calculate how many jobs are still running, and

squeue –u usrname | grep Q –c

to calculate how many jobs are still pending.

Wait until all jobs are done which will cost less than 24 hours. Repeat the above by typing

cd /PATH/

module load matlab

matlab -nojvm

ANTs\_longleaf1

exit

Then open and check file

/code/ANTs\_batAll.sh

If it has only one line, and reads

“

#!/bin/bash

” ,

this step is finished. Otherwise, type

cd code/

chmod 775 ANTs\_batAll.sh

./ANTs\_batAll.sh

Repeat this step until the generated

/code/ANTs\_batAll.sh

Has only one line, and reads

“

#!/bin/bash

”

Then this step is finished. This will guarantee the label file /labeloutput/SubID\_Labels.nii.gz

for each individual is correctly generated.

**Details of Pipeline:**

Main function of ANTs: antsJointLabelFusion.sh

Input: antsExtractedBrain0N4.nii.gz (see image examples in Step 3)

Input: (brain MRI in the training set)

OASIS-TRT-20-1\_brain.nii.gz

OASIS-TRT-20-2\_brain.nii.gz

… …

OASIS-TRT-20-20\_brain.nii.gz

Input: (labeled brain MRI in the training set)

OASIS-TRT-20-1\_DKT31\_CMA\_labels.nii.gz

OASIS-TRT-20-2\_DKT31\_CMA\_labels.nii.gz

… …

OASIS-TRT-20-20\_DKT31\_CMA\_labels.nii.gz

Parameter: -d (dimension)

Parameter: -c (Control for parallel computation (default 0) -- 0 == run serially, 1 == SGE sbatch, 2 == use PEXEC (localhost), 3 == Apple XGrid, 4 == PBS sbatch, 5 == SLURM.)

Parameter: -x (Target mask image (default = 'otsu'); otsu: use otsu thresholding to define foreground/background; or: 'or' all the warped atlas images to defined foreground/background; <filename>: a user-specified mask; none: don't use a mask)

Output: (log file)

SubID\_OASIS-TRT-20-1\_brain\_0\_log.txt

SubID\_OASIS-TRT-20-1\_brain\_1\_log.txt

… …

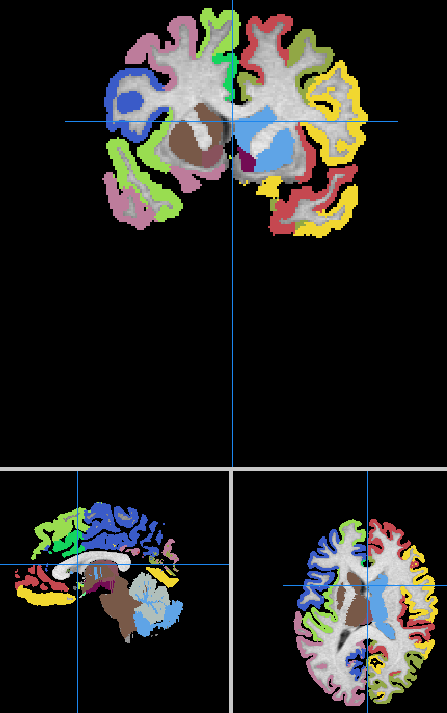
SubID\_OASIS-TRT-20-1\_brain\_19\_log.txt

Output: SubID\_Labels.nii.gz (Segmentation of 101 ROIs)

Output: SubID\_TargetMaskImageOr.nii.gz (Mask generated for all ROIs)

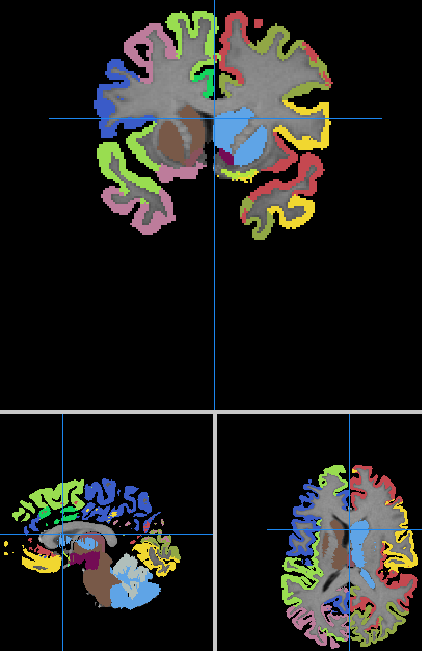
The standard segmentation results

(OASIS-TRT-20-1\_brain overlaid on OASIS-TRT-20-1\_DKT31\_CMA\_labels)



Another standard segmentation results

(OASIS-TRT-2-20\_brain overlaid on OASIS-TRT-20-20\_DKT31\_CMA\_labels)



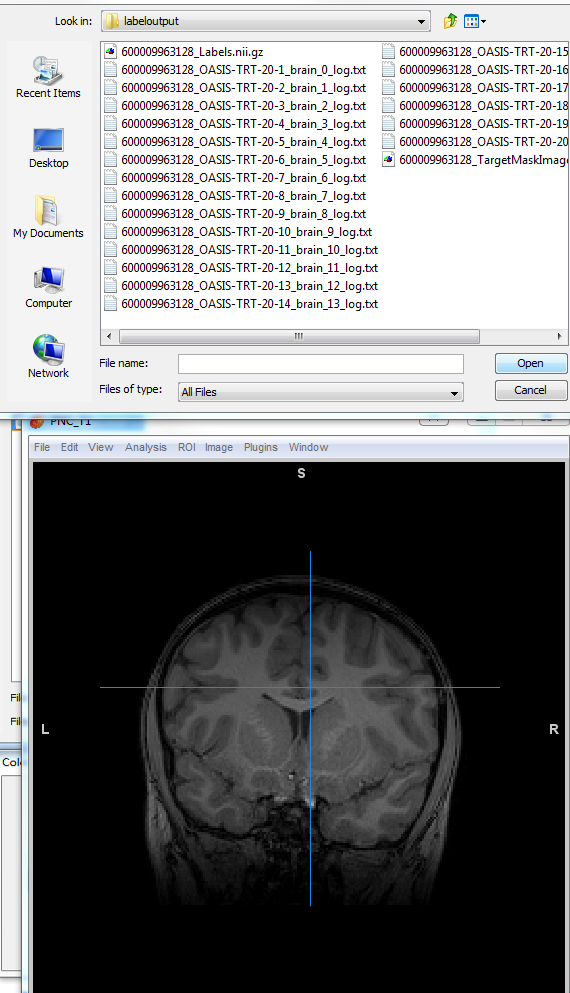
**Quality Control:**

1. Download one your raw data T1.nii.gz and segmentation result

/labeloutput/SubID\_Labels.nii.gz

To your local computer.

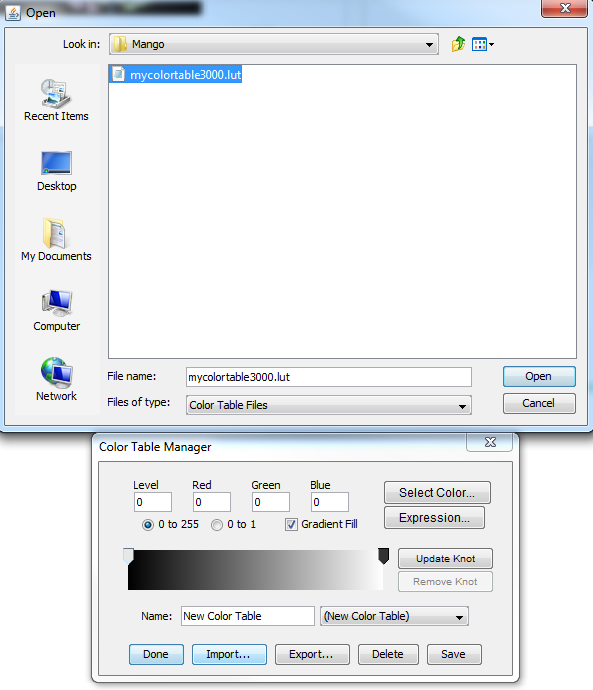
1. Use Mango to open T1.nii.gz. Then push the button Ctrl+Shift+O, the following window will pop up. Then select SubID\_Labels.nii.gz and the label segmentation and T1 MRI are overlaid together.



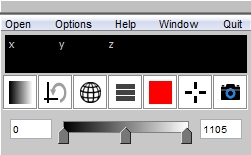
1. Click the menu

Options--Color Table Manager--Import-- select file mycolortable3000.lut

Click Save and then click Done.

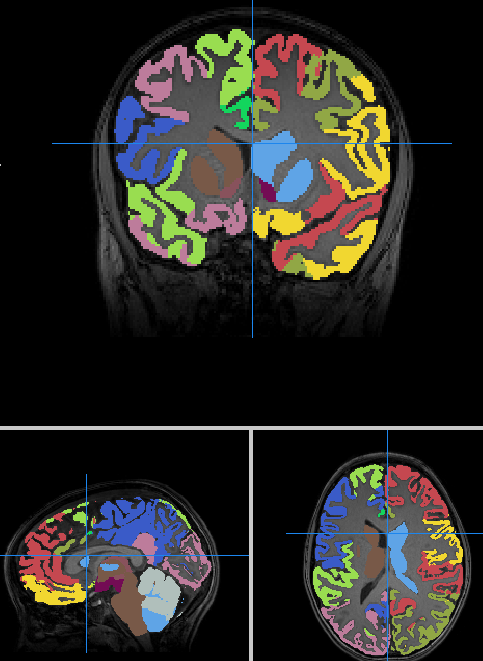


Then tune the color range to [0,3100] as bellows.



We will obtain a 101-ROI segmentation image as below.

One example of SubID\_Labels.nii.gz overlaid on T1.nii



Compare this segmentation with the standard segmentation results illustrated in Details of Pipeline of this step to check the segmentation quality.

1. **Summarize the volume and sum of cortical thickness for all ROIs**.

Upload files Surf.py, extract\_UKB.m to /PATH/. Set Finalmove=1 and run getcsv again to extract all output into specific folders by typing the following commands from terminal:

cd /PATH/

module load matlab

matlab -nojvm

getcsv

exit

When this job is finished, modify the PATH in extract\_UKB.m to your own PATH, and then change extent1, extent2, extent3 to subject ID range you processed. Then run extract\_UKB.m by typing the following commands from terminal:

cd /PATH/

module load matlab

matlab -nojvm

extract\_UKB

exit

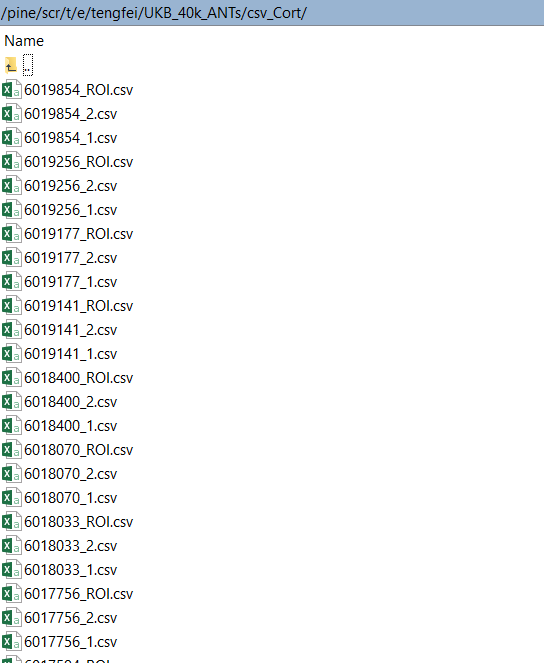
Then pbs job codes will be generated in the folder code/. Type the following command to submit jobs:

cd code/

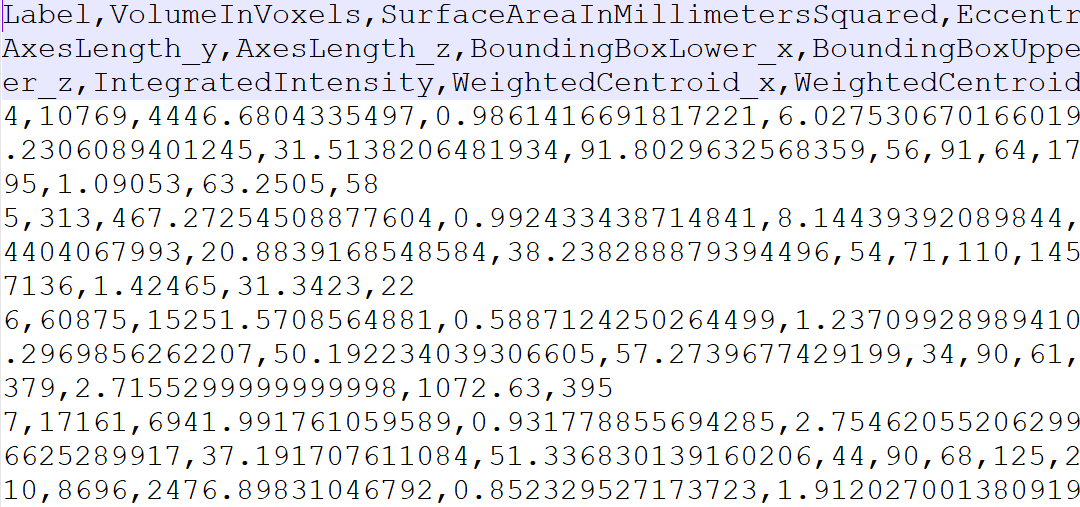
chmod 775 ANTs\_batAll.sh

./All.sh

After 3~4 hours, check whether this job has been finished. If it is finished, check if a new folder PATH/csv\_Cort is created and feature for all



subjects are generated in PATH/csv\_Cort. The files \*\_ROI.csv reads as



which include the ROI lables, each ROI volume, surface area, etc.

**YOU ARE DONE HERE!**