# SNImagingLab\_STEP2\_ct\_segmentation\_t24.sh

# Brain Image Registration and Masking Script for Segmentation USING 0-24HU THRESHOLD FOR CSF SEGMENTATION

## Overview

This bash script applies the registration and masking techniques to brain-extracted images from preprocessed Head CT scans as in SNImagingLab\_STEP2\_ct\_segmentation.sh. However, this script utilizes CSF segmented using voxel threshold range of 0 – 24HU to capture more sulcal CSF in subarachnoid brain regions. The primary outputs include segmented cerebrospinal fluid (CSF) and ventricles.

### Output

The primary output of interest is:

- \*\*`\_restore\_native`\*\*: A bias field corrected brain image using the most complete brain extraction mask.

## Requirements

Before running the script, ensure that you have the following installed:

- FSL (FMRIB Software Library)

- Bash shell (for executing the script)

- Proper directory structure and input files as specified in the script

- Save required atlases and masks and update atlas directory

- Enable or disable optional code according to desired output

## Directory Structure

Before running the script, ensure your directory structure resembles the following:

```

Image\_dir/

├── restore/ # Input images

├── work/ # Working directory for intermediate files

├── omat/ # Transformation matrices

├── interim\_vent/ # Intermediate ventricle images

├── suptentcsf/ # Output CSF images

├── cluster/ # Cluster analysis output

├── thirdvent/ # Third ventricle output files

└── stdorient/ # Final vent images in standard orientation

```

## Usage

1. \*\*Set File Paths\*\*: Modify the script to set the correct paths for your input images, atlases, and output directories.

2. \*\*Run the Script\*\*: Execute the script from your terminal:

```bash

bash SNImaging\_STEP2\_ct\_segmentation\_t24.sh

```

##Script Workflow

Step 1: Image Registration

Aligns the bias field corrected "\_restore" brain image to the SRI24 standard brain using FSL's FLIRT, saving transformation matrices for later use.

Step 2: Create Supratentorial CSF

Multiple steps to supratentorial volume, extract CSF using thresholding, and create targeted masks to isolate ventricular CSF.

Step 3: Create Aligned Ventricle Atlases

Creates a target crude ventricle for alignment and registers both large and small ventricle atlases to this target. Two different ventricle atlases are used to accommodate variations in anatomy of both typical and enlarged ventricles.

Step 4: Create Ventricles

Combines large and small ventricles and generates a complete lateral ventricle.

Step 5: Extract Third Ventricle

Segments third ventricle CSF.

Step 6: Smooth and Threshold

Applies smoothing and thresholding to fill holes and clean edges of the ventricle images.

Step 7: Optional Third Ventricle Removal

Subtracts third ventricle contamination from the final lateral ventricle images, if necessary.

Step 8: Inverse Transformation

Uses the inverted transformation matrices to convert CSF and ventricle images back to native space.

Step 9: Cluster Analysis

Selects the largest contiguous clusters of ventricle images to target ventricular CSF and minimize contamination of non-ventricular CSF in final segmented ventricle.

Step 10: (Optional) Create Supratentorial Extraventricular CSF and/or Suprasylvian Subarachnoid CSF

Subtract lateral and third ventricles from supratentorial CSF to create extraventricular and suprasylvian subarachnoid CSF images.

Step 11: (Optional) Transform to Standard Orientation

Apply rigid body transformation to orient lateral ventricles to standard orientation.

Step 12: (Optional) Append Statistics to File

Use fslstats function to genearate and log volume statistics for various masks into a text file.

## Optional Steps

- Create suprasylvian subarachnoid CSF and/or supratentorial extraventricular CSF.

- Apply rigid body transformation for ventricle reorientation.

- Append volume measures to a output text file for further analysis.

## Notes

- Ensure that all required atlases are available and correctly referenced in the script.

- Parallel Processing

The script is designed to run multiple jobs in parallel to optimize processing time. Adjust the number of concurrent jobs by changing the value of N in the script based on your system's CPU and RAM availability.

## Troubleshooting

If you encounter issues, check the following:

- Verify that input image files are in the correct format and location.

- Ensure that all necessary tools and libraries are installed.

- Review script output for any error messages for guidance.

## License

This script is provided for educational and research purposes. Please attribute any usage to the original author.

Additionally, refer to the FSL (FMRIB Software Library) license posted at https://fsl.fmrib.ox.ac.uk/fsl/docs/#/license

## Authors

For questions or issues, please contact the authors of this script:

Kevin King [kking@sniweb.net](mailto:kking@sniweb.net)

Emily Foldes [emily.foldes@barrowneuro.org](mailto:emily.foldes@barrowneuro.org)