**LOCATE – LOCALLY ADAPTIVE THRESHOLD ESTIMATION**

**User Manual**

**Contents:**

1. Introduction
2. LOCATE dependencies
3. Dataset preparation
4. Running LOCATE for leave-one-subject-out (LOSO) validation
5. Running LOCATE for separate training and test images
6. Outputs
7. Correcting the geometric information of the output files
8. Viewing results on FSLeyes
9. **Introduction**

LOCATE (LOCally Adaptive Thresholds Estimation) is a supervised method to determine thresholds for binarising the subject-level LPM with specific applicability to BIANCA. In principle, LOCATE can be applied to the LPM obtained by a different method, provided the availability of a training data with manual lesion masks.

LOCATE is more adaptive to the variation in lesion load and has the potential to improve the subject-level lesion segmentation, by detecting more deep lesions, without requiring additional training data, other than that used in BIANCA classifier for LPM estimation. Moreover, LOCATE can be used when manual segmentation is unavailable for the test dataset, or the number of subjects in the dataset is not enough to generate an accurate training dataset.

Thanks for trying our Beta version of LOCATE and we look forward to your feedback and comments to improve LOCATE.

LOCATE is currently available in MATLAB and the scripts are available in the following Github repository:

**/vols/Scratch/vaanathi/LOCATE/**

* 1. **Main Contents of the folder:**

**(You do not need to know the details; it helps us to identify the possible errors)**

LOCATE\_training

LOCATE\_testing

LOCATE\_LOSO\_testing

LOCATE\_Voronoi\_tessellation

LOCATE\_feature\_extraction

LOCATE\_test\_feature\_extraction

Modify\_voronoi\_features

MATLAB <directory containing i/o console functions>

Of all the functions, LOCATE\_training and LOCATE\_testing are the main functions for training and testing LOCATE on separate training and test images.

LOCATE\_LOSO\_testing is the function for performing leave-one-subject-out validation for LOCATE.

1. **LOCATE dependencies**
2. FLAIR image or the base image modality used in BIANCA
3. Any other additional images that were used as a main feature in BIANCA (optional)
4. Lesion Manual mask: binary mask indicating lesion voxels manually segmented
5. Ventricle distance map: Grayscale image showing distance from ventricles within the brain mask (refer distancemap in FSL for more details)

**distancemap -i <ventricle\_mask\_image\_in\_FLAIR\_space> -m <Brain\_mask\_in\_FLAIR\_space> -o <Output\_distancemap\_in\_FLAIR\_space>**

1. BIANCA output: Lesion probability map obtained from BIANCA
2. BIANCA mask: Mask obtained from make\_bianca\_mask (white matter mask excluding sub-cortical regions) - If you are not using BIANCA mask in your analysis, make a copy of brain mask and rename it as BIANCA mask using specification described in the next section.
3. Brain mask: obtained from BET or any other method
4. **Dataset preparation**

LOCATE is a supervised method and hence requires data for training. In a future release of BIANCA, LOCATE will be integrated as an additional option in BIANCA. However, currently for testing, the user needs to prepare the training data by renaming in a specific standardised manner.

Step 1: Create a directory named ‘Training\_imgs’ and ‘Test\_imgs’ in your working directory (eg. Myfolder).

Step 2: Rename the training files for the training data as follows and copy them in into the Training\_imgs:

1. <subject\_name>\_feature\_FLAIR.nii.gz
2. <subject\_name>\_feature\_ modality name (eg. <subject\_name>\_feature\_T1.nii.gz, please note that it is mandatory to add \_feature\_ in the filename for it to be considered as a feature)
3. <subject\_name>\_manualmask.nii.gz
4. <subject\_name>\_ventdistmap.nii.gz
5. <subject\_name>\_BIANCA\_LPM.nii.gz
6. <subject\_name>\_biancamask.nii.gz (if you are using brain mask as BIANCA mask, you still need to rename a copy of it as specified)
7. <subject\_name>\_brainmask.nii.gz

Step 3: Similarly rename the test files, except the manual mask file (2 in step 2), with the same convention and copy them into the ‘Test\_imgs’ directory

1. **Running LOCATE for leave-one-subject-out (LOSO) validation**

**4.1 Example function calls**

1. LOCATE\_LOSO\_testing();

- If you have the training images (all the modalities) in the same folder

2. LOCATE\_LOSO\_testing(train\_image\_directory\_name);

- If you have the training images are in the seperate directory

3. LOCATE\_LOSO\_testing(train\_image\_directory\_name, feat\_select);

- If you want to select specific features for training and testing

4. LOCATE\_LOSO\_testing(train\_image\_directory\_name, feat\_select, verbose);

**4.1 Optional inputs (in the order)**

- train\_image\_directory\_name - Name of the directory where the training images for feature extraction are located (if not in the same folder)

- feat\_select - vector with elements indicating if the feature has to be included or not. Current order is distance from ventricles, lesion volume and other modalities in alphabetical naming order

(e.g. If FLAIR is the only modality provided and distance from ventricles is not needed then feat\_select = [0, 1, 1])

- verbose (default – 0; 1 if the steps need to be displayed on the screen)

The above command performs LOSO testing for all the images (with <subjetname>\_BIANCA\_LPM.nii.gz) in the ‘Training\_imgs’ directory using the selected features.

1. **Running LOCATE for separate training and test images**
   1. **LOCATE\_training:**

**5.1.1 Example funtional calls:**

1. LOCATE\_training();

- If you have the training images (all the modalities) in the same folder

2. LOCATE\_training(train\_image\_directory\_name);

- If you have the training images are in the seperate directory

3. LOCATE\_training(train\_image\_directory\_name, feat\_select);

- If you want to select specific features for training and testing

4. LOCATE\_training(train\_image\_directory\_name, feat\_select, verbose);

**5.1.2 Optional inputs (in the order):**

- train\_image\_directory\_name - Name of the directory where the training images for feature extraction are located (if not in the same folder)

- feat\_select - vector with elements indicating if the feature has to be included or not. Current order is distance from ventricles, lesion volume and other modalities in alphabetical naming order

(e.g. If FLAIR is the only modality provided and distance from ventricles is not needed then feat\_select = [0, 1, 1])

- verbose (default – 0)

* 1. **LOCATE\_testing:**

**5.2.1 Example funtional calls:**

1. LOCATE\_testing(test\_image\_directory\_name);

- If you have the training images (all the modalities) in the same folder

2. LOCATE\_testing(test\_image\_directory\_name, train\_image\_directory\_name);

- If you have the training images are in the seperate directory

3. LOCATE\_testing(test\_image\_directory\_name, train\_image\_directory\_name, feat\_select);

- If you want to select specific features for training and testing

4. LOCATE\_testing(test\_image\_directory\_name, train\_image\_directory\_name, feat\_select, verbose);

**5.2.2 Optional inputs (in the order):**

- train\_image\_directory\_name - Name of the directory where the training images for feature extraction are located (if not in the same folder)

- feat\_select - vector with elements indicating if the feature has to be included or not. Current order is distance from ventricles, lesion volume and other modalities in alphabetical naming order

(e.g. If FLAIR is the only modality provided and distance from ventricles is not needed then feat\_select = [0, 1, 1])

- verbose (default – 0)

The above command performs testing for all the images (with <subjetname>\_FLAIR.nii.gz) in the ‘Test\_imgs’ directory, using the model trained using all the images from ‘Training\_imgs’ directory.

1. **Outputs**

Outputs from training and testing stages are listed in this section. The main outputs are highlighted in bold text in red colour.

**6.1 LOSO Testing:**

During LOSO testing, a new directory named ‘**LOCATE\_LOSO\_results\_directory**’ will be created in the ‘Training\_imgs’ directory. All the LOCATE output files will be saved in the ‘LOCATE\_LOSO\_results\_directory’.

The contents of LOCATE\_results\_directory are:

1. LOCATE\_test\_features.mat’ – an intermediate file containing test features for all the test images in a single file
2. <subjectname>\_indexmap.nii.gz – image showing the voronoi polygongs obtained from Voronoi tessellation step
3. <subjectname>\_thresholdsmap.nii.gz – image showing the local thresholds within the Voronoi polygons
4. **<subjectname>\_BIANCA\_LOCATE\_binarylesionmap.nii.gz** – binary lesion map obtained as the final output of LOCATE
5. <subjectnames>\_LOCATE\_thresholds.mat– array of thresholds (basically, the same thresholds shown in the <subjectname>\_thresholdsmap.nii.gz in step 3. If you want the threshold values directly for any further analysis or plotting, this utput could be useful.
6. Consolidated\_LOCATE\_output.mat – The outputs from 2, 3, 4 and 5 for all the images available in the single .mat file.

**6.2 Training and testing on separate data:**

**6.2.1 Training stage:**

Once the training is done, the following files will be created additionally in the **Training\_imgs** directory:

1. LOCATE\_features\_<subjectname>.mat – containing features of individual training subjects
2. LOCATE\_features.mat – containing features of all the subjects in a single .mat file (this is the file that will be needed in the testing phase)
3. **RF\_regression\_model\_LOCATE.mat** – Trained Random Forest regression model for LOCATE in the training phase. This model could be applied on any test dataset (need not be the same dataset or having same pathological conditions) with image dimensions matching the image training dataset.

**6.2.2 Testing stage:**

During testing, a new directory named ‘**LOCATE\_results\_directory**’ will be created in the ‘Test\_imgs’ directory. All the LOCATE output files will be saved in the ‘LOCATE\_results\_directory’.

The contents of LOCATE\_results\_directory are:

1. LOCATE\_test\_features.mat’ – an intermediate file containing test features for all the test images in a single file
2. <subjectname>\_indexmap.nii.gz – image showing the voronoi polygongs obtained from Voronoi tessellation step
3. <subjectname>\_thresholdsmap.nii.gz – image showing the local thresholds within the Voronoi polygons
4. **<subjectname>\_BIANCA\_LOCATE\_binarylesionmap.nii.gz** – binary lesion map obtained as the final output of LOCATE
5. <subjectnames>\_LOCATE\_thresholds.mat– array of thresholds (basically, the same thresholds shown in the <subjectname>\_thresholdsmap.nii.gz in step 3. If you want the threshold values directly for any further analysis or plotting, this utput could be useful.
6. Consolidated\_LOCATE\_output.mat – The outputs from 2, 3, 4 and 5 for all the images available in the single .mat file.

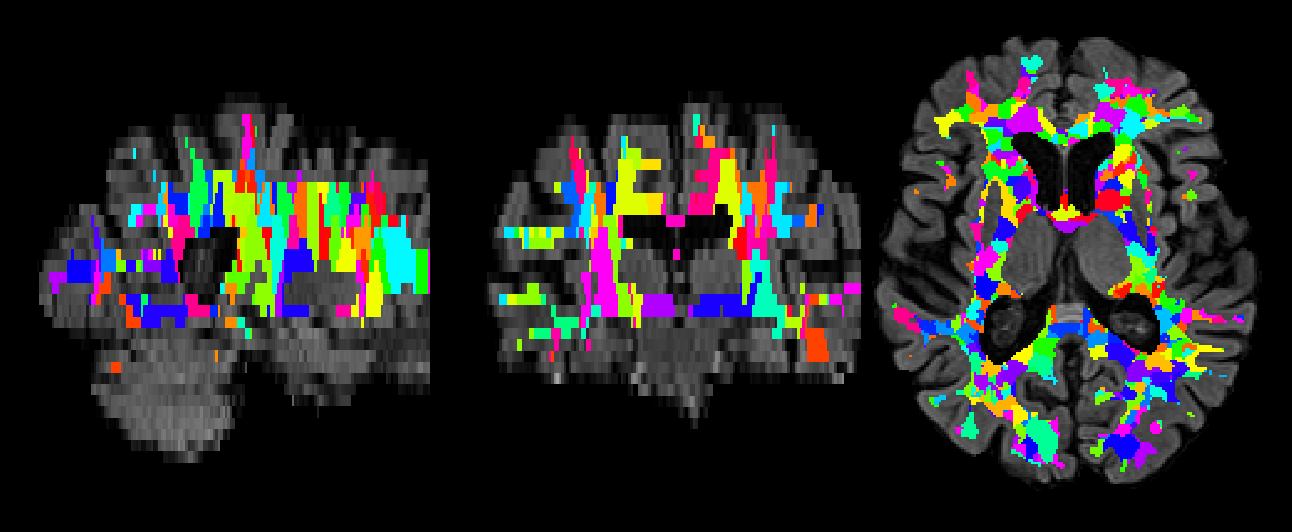
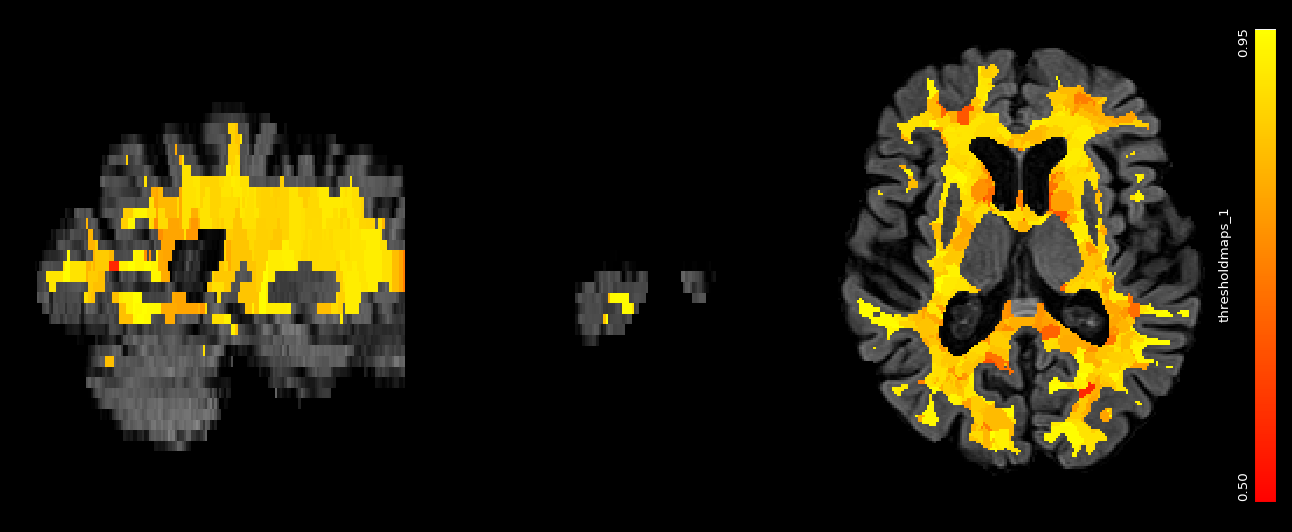
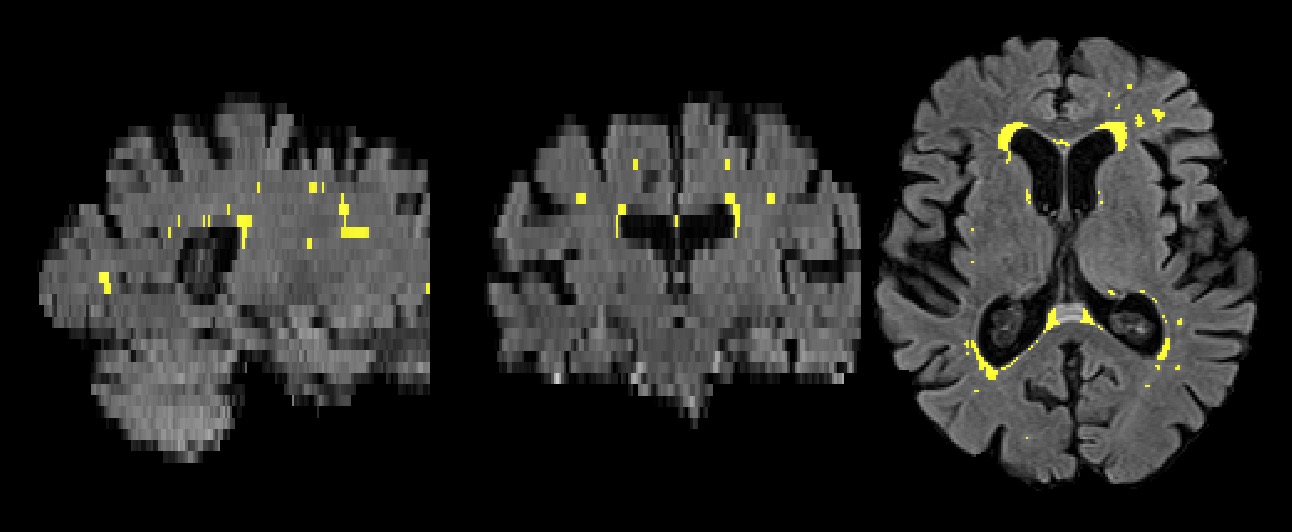
**7. Correcting the geometric information of the output files**

Currently the nifti files are saved with the voxel dimensions of 1mm x 1mm x 1mm, irrespective of the input FLAIR image dimensions. So please copy the geometric information from the FLAIR image of the subjects to the LOCATE outputs, thresholdmaps and indexmaps, before viewing the results on FSLeyes or calculating the lesion volumes. Please refer fslcpgeom (FSL) for further details.

**Eg. fslcpgeom Myfolder/Training\_imgs<subjectname>\_FLAIR.nii.gz Myfolder/Test\_imgs/LOCATE\_results\_directory/<subjectname>\_BIANCA\_LOCATE\_binarylesionmap.nii.gz**

**8. Viewing results on FSLeyes**

The sample output maps and images are shown overlaid on the base modality image (in our case, the FLAIR image)



**Figure 3: Output binary lesion map <subjectname\_BIANCA\_LOCATE\_binarylesionmap.nii.gz>**

**Figure 2: Map showing local thresholds <subjectname\_thresholdsmap.nii.gz>**

**Figure 1: Voronoi tessellations <subjectname\_indexmap.nii.gz>**