**LOCATE – LOCALLY ADAPTIVE THRESHOLD ESTIMATION**

**User Manual**

**Contents:**

1. Introduction
2. Dataset preparation and input files required
3. Running LOCATE
4. Correcting the geometric information of the output files
5. Viewing results on FSLeyes
6. **Introduction**

LOCATE (LOCally Adaptive Thresholds Estimation) is a supervised method to determine thresholds for binarising the subject-level lesion probability map (LPM) with specific applicability to BIANCA (https://fsl.fmrib.ox.ac.uk/fsl/fslwiki/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.

With respect to the use of a global threshold (same for the whole brain and all subjects), LOCATE has the potential to improve the subject-level lesion segmentation, by detecting more deep lesions, and being less sensitive to variability in lesion load without requiring additional training data, other than that used in BIANCA classifier for LPM estimation.

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:

<https://git.fmrib.ox.ac.uk/vaanathi/LOCATE-BIANCA>

1. **Dataset preparation and input files required**

LOCATE is a supervised method and hence requires data for training. In the current version the user needs to prepare the data by renaming folders and images in a specific standardised manner.

Images of the subjects used for training and testing need to be grouped in separate directories, named ‘Training\_imgs’ and ‘Test\_imgs’ in your working directory (eg. Myfolder).

LOCATE requires the following images for each subject (either belonging to the training or testing set). In the current version all the images need to have the standard name specified below:

1. FLAIR image or the base image modality used in BIANCA

<subject\_name>\_feature\_FLAIR.nii.gz

1. Any other additional images that were used as a intensity features in BIANCA (--featuresubset) (optional)

<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)

1. Lesion Manual mask (for training subjects only): binary mask indicating lesion voxels manually segmented

<subject\_name>\_manualmask.nii.gz

1. Ventricle distance map: image where each voxel intensity represents the distance from ventricles within the brain mask (refer distancemap in FSL for more details, example call:distancemap -i <ventricle\_mask\_image\_in\_FLAIR\_space> -m <Brain\_mask\_in\_FLAIR\_space> -o <subject\_name>\_ventdistmap.nii.gz)

<subject\_name>\_ventdistmap.nii.gz

1. BIANCA output: Lesion probability map (LPM) obtained from BIANCA

<subject\_name>\_BIANCA\_LPM.nii.gz

1. Brain mask: obtained from FSL-BET or any other method

<subject\_name>\_brainmask.nii.gz

1. 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 the brain mask and rename it as BIANCA mask.

<subject\_name>\_biancamask.nii.gz

1. **Running LOCATE**

LOCATE can be run in three main ways for performing following operations: Leave-one-subject-out validation, train on a specific set of subjects and test on another independent set, without reusing any of the training subjects. You want to use LOSO testing evaluation of the LOCATE performance on the dataset having manual segmentation for all the subjects. On the other hand, if you do not have ground truth for all subjects, you can train using the ones with ground truth and test on the rest. Training on testing on separate set of subjects is also logical if you have done the same for BIANCA.

* 1. **Leave-one-subject-out (LOSO) validation**

**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 separate directory

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

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

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

- 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])

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

* 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.

**Outputs**

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 main output is highlighted in bold text in red colour.

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.
   1. **LOCATE\_training:**

**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

- train\_image\_directory\_name - Name of the directory where the training images for feature extraction are located (if not in the same folder). In our case, it would be ‘Myfolder/Training\_imgs’

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

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

- 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])

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

- verbose (default – 0)

**Outputs**

Once the training is done, the following files will be created additionally in the **Training\_imgs** directory. The main output is highlighted in bold text in red colour.

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.
   1. **LOCATE\_testing:**

**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

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

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

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

- 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])

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

**-** 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.

**Outputs**

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 main output is highlighted in bold text in red colour.

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.

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

The output nifti files will have voxel dimensions of 1mm x 1mm x 1mm, irrespective of the input FLAIR image dimensions. Therefore, the correct geometric information needs to be copied 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.

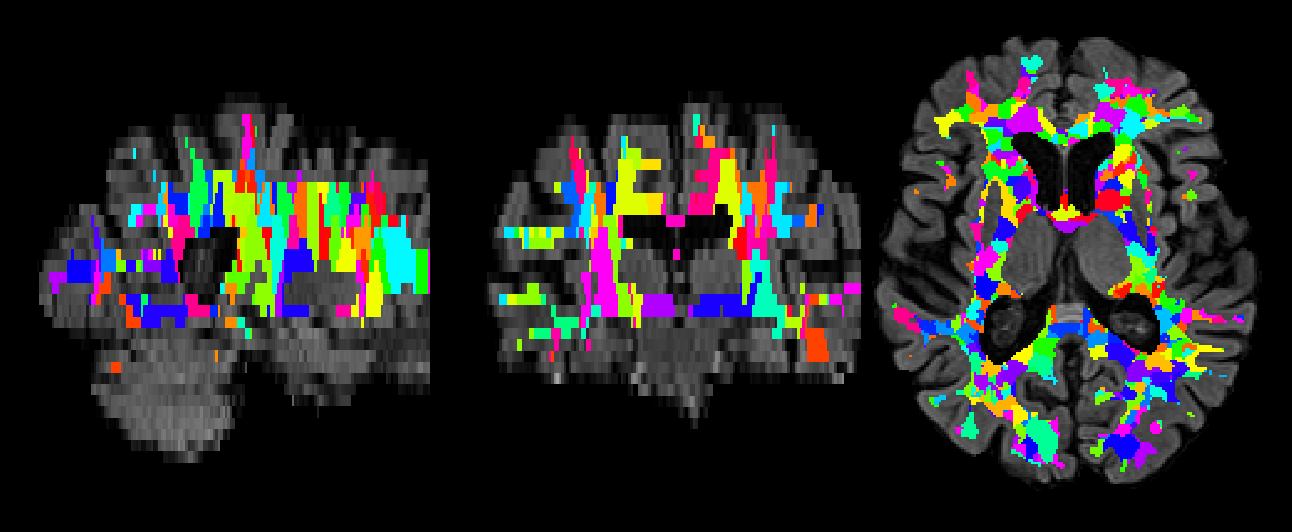
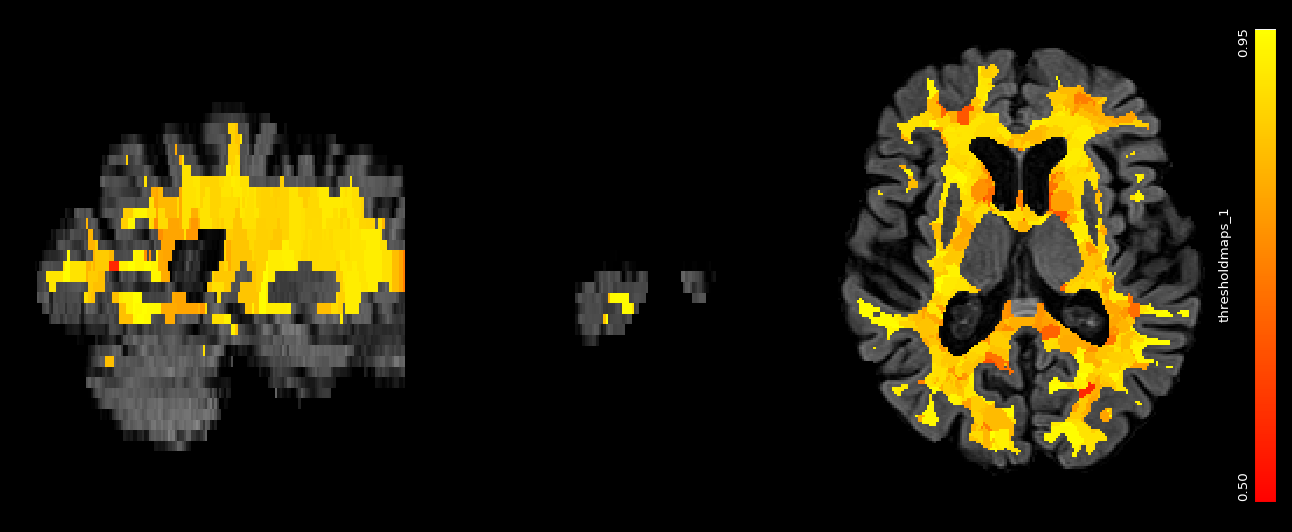
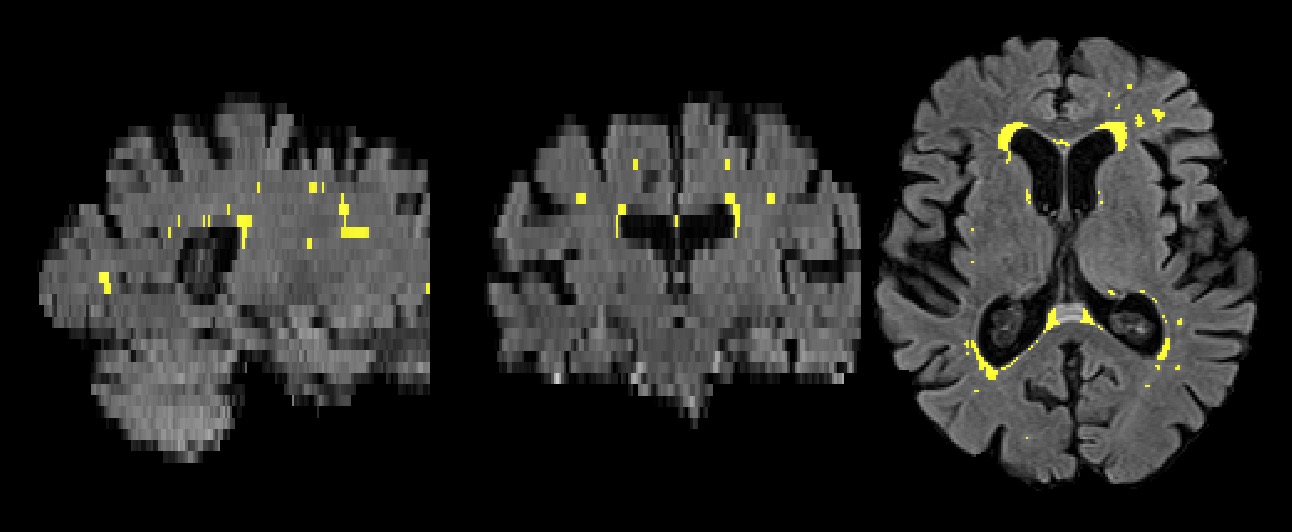
Eg. fslcpgeom Myfolder/Training\_imgs<subjectname>\_FLAIR.nii.gz Myfolder/Test\_imgs/LOCATE\_results\_directory/<subjectname>\_BIANCA\_LOCATE\_binarylesionmap.nii.gz

**5. Viewing results on FSLeyes**

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

**Example command**

fsleyes myfolder/Test\_imgs/<subject\_name>\_feature\_FLAIR.nii.gz myfolder/Test\_imgs/LOCATE\_results\_directory**/**<subjectname>\_indexmap.nii.gz myfolder/Test\_imgs/LOCATE\_results\_directory/subjectname>\_thresholdsmap.nii.gz myfolder/Test\_imgs/LOCATE\_results\_directory/<subjectname>\_BIANCA\_LOCATE\_binarylesionmap.nii.gz &



**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>**