LOCATE – LOCALLY ADAPTIVE THRESHOLD ESTIMATION
User Manual

Contents:

1. 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 any method (not just BIANCA), provided there is some available training data with manual lesion masks.

LOCATE has the potential to improve the subject-level lesion segmentation, compared to the use of a global threshold, by making the threshold vary spatially across the brain and therefore also differing between subjects. Results can improve with respect to detecting more deep lesions, and being less sensitive to variability in lesion load without requiring additional training data, beyond that used in BIANCA classifier for LPM estimation. For the detailed information about the LOCATE method and validation, please refer our article:
https://www.biorxiv.org/content/early/2018/10/08/437608.

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

2. Initial setup

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

To run LOCATE on your machine, clone the git repository in your working directory. To do this, open your terminal and type the following command in your working directory:

```
git clone https://git.fmrib.ox.ac.uk/vaanathi/LOCATE-BIANCA
```

Now change your working directory to LOCATE-BIANCA:

```
cd LOCATE-BIANCA/
```

Also, please ensure that you have FSL installed in your machine and the following environment variables/paths are correctly set in your system by typing the following commands in your terminal and check if they provide similar output:

```
echo $FSLDIR
```
The output should be the path where FSL is installed, e.g. /usr/local/fsl

```
echo $FSLOUTPUTTYPE
```
The output should be: NIFTI_GZ

3. 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 provide data in a specific, standardised manner (which can typically be achieved by some moving and renaming of files).

First, create your LOCATE working directory, e.g. MyLOCATE/. Within that, images for the subjects used to train the model should be grouped in a directory (e.g. <Training_imgs_directory>), while the images for the subjects on which LOCATE is applied need to be in a directory (e.g. <Test_imgs_directory>). To use the same subjects for training and testing (with leave-one-out validation), images for all subjects need to be in the same directory (e.g. <LOO_imgs_directory>).

Within each directory, LOCATE requires the following images for each subject. In the current version all the images need to have the standard name specified below (except 2, all the other images are compulsory):

1. The base image modality used in BIANCA
   <subject_name>_feature_<base_modality_name>.nii.gz

It is essential that you provide at least one image modality (preferably the base image modality used in BIANCA). For example, if you used FLAIR as your base image modality in BIANCA, your base_modality_name will be 'FLAIR'. So the image has to be named as <subject_name>_feature_FLAIR.nii.gz

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

   <subject_name>_feature_<modality_name>.nii.gz

   (eg. <subject_name>_feature_T1.nii.gz, please note that it is mandatory to add _feature_ in the filename before the modality name for it to be considered as a feature)

3. Lesion Manual mask (for training subjects only): binary mask (values of 0 and 1) indicating lesion voxels, based on manual segmentation

   <subject_name>_manualmask.nii.gz

4. Ventricle distance map: image where each voxel intensity represents the distance from ventricles within the brain mask (this can be calculated using the FSL tool *distancemap* – see the wiki 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

5. BIANCA output: unthresholded lesion probability map (LPM) obtained from BIANCA

   <subject_name>_BIANCA_LPM.nii.gz

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

   <subject_name>_brainmask.nii.gz

7. BIANCA mask: binary mask, as 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

4. Running LOCATE

LOCATE can be run in 3 main ways: 1) train and test the LOCATE model on data with manual mask available (with leave-one-out validation, LOO) in the directory (<LOO_imgs_directory>); 2) train the LOCATE model on data with manual mask available, in the <Training_imgs_directory>; 3) test the LOCATE model on data without manual mask, in <Test_imgs_directory>.

The following three subsections outline the various options available within each of these main ways of running LOCATE. Under each subsection, we provide the example function call and the main outputs generated by running LOCATE. Additionally, we provide the list of optional inputs that could be provided to the

LOCATE function call to make LOCATE more specific to the data and/or feature sets. Finally, we provide the complete list of outputs generated by LOCATE.

4.1 Leave-one-out (LOO) validation

This function call is used to perform leave-one-out validation (i.e. the LOCATE model is iteratively trained on data from N-1 subjects present in the <LOO_imgs_directory> and applied on the remaining subject).

Example function call

LOCATE_LOO_testing(LOO_imgs_directory_name);

- LOO_imgs_directory_name - Name of the directory where the images are located. (e.g. '<LOO_imgs_directory>')

Note: If you call LOCATE_LOO_testing.m from where the images are located, you can leave the input argument empty and call the function as LOCATE_LOO_testing();

Main outputs:

- **LOCATE_LOO_results_directory** - directory inside <LOO_imgs_directory> containing all the LOCATE output files.
- **LOCATE_LOO_results_directory/<subjectname>_BIANCA_LOCATE_binarylesionmap.nii.gz** – binary lesion map obtained as the final output of LOCATE.

Additional/optional inputs to the function call

1. LOCATE_LOO_testing(LOO_imgs_directory_name, feature_select);
   If you want to select specific features for training and testing
   - feature_select - vector with elements indicating if the feature has to be included or not (1 – to be included, 0 to be discarded). Current order is: distance from ventricles, size of individual lesions (automatically calculated from the LPM) and mean intensity in various modalities (e.g. If <base_modality_name> is the only modality provided and distance from ventricles is not needed, then the function call would be LOCATE_LOO_testing(LOO_imgs_directory_name, [0, 1, 1]);

2. LOCATE_LOO_testing(LOO_imgs_directory_name, feature_select, verbose);
   - verbose (default = 0; 1 if the comments for all the steps need to be displayed on the screen)

Complete list of outputs

The 'LOCATE_LOO_results_directory' contains:

1. LOCATE_test_features.mat' – an intermediate MATLAB data file containing features for all the test images in a single file.

2. <subjectname>_indexmap.nii.gz – image showing the Voronoi polygons obtained from the 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 (main output).
5. <subjectname>_LOCATE_thresholds.mat – array of thresholds (the same thresholds shown in the <subjectname>_thresholdsmap.nii.gz in step 3). This format can be useful for further analysis of the threshold values (e.g., histogram or stats).
6. Consolidated_LOCATE_output.mat – outputs from 2, 3, 4 and 5 for all the images available in a single .mat file.

4.2 LOCATE_training:

This function call is used to train and save the LOCATE model using images in the <Training_imgs_directory> (you will then need to run LOCATE_testing function, explained in section 4.3, to get the final thresholded maps for the subjects in the <Test_imgs_directory>).

Example function call

LOCATE_training(train_image_directory_name);

- train_image_directory_name - Name of the directory where the images are located. (e.g. '<Training_imgs_directory>')

Note: If you call LOCATE_training.m from where the images are located, you can leave the input argument empty and call the function as LOCATE_training ();

Main outputs:

- **LOCATE_training_files** - directory inside <Training_imgs_directory> containing all the LOCATE training files including the trained model.
- **LOCATE_training_files/'RF_regression_model_LOCATE.mat'** – the MATLAB data file with the regression model that has to be used in LOCATE testing.

Additional/optional inputs to the function call

1. LOCATE_training(train_image_directory_name, feature_select);
   If you want to select specific features for training
   - feature_select - vector with elements indicating if the feature has to be included or not (1 – to be included, 0 to be discarded). Current order is: distance from ventricles, size of individual lesions (automatically calculated from the LPM) and mean intensity in various modalities (e.g. If

<base_modality_name> is the only modality provided and distance from ventricles is not needed, then the function call would be LOCATE_training (train_image_directory_name, [0, 1, 1]);

Note: Unlike LOO, here you will be performing training alone, and the created model can be evaluated on any set of images using LOCATE_testing. It is essential that the image modalities and feature_select values remains consistent for both training and testing. For example, if you run LOCATE_training on FLAIR, T1 and PD images with feature_select option [1,0,1,1,1], the same modalities and feature_select options need to be provided for the test data while testing LOCATE (explained in section 4.3) as well.

2. LOCATE_training(train_image_directory_name, feature_select, verbose);
   - verbose (default = 0)

Complete list of outputs

The 'LOCATE_training_files' contains:

1. LOCATE_features_<subjectname>.mat – a MATLAB data containing features of individual training subjects
2. LOCATE_features.mat – containing features of all the subjects in a single .mat file
3. **RF_regression_model_LOCATE.mat** – Trained Random Forest regression model for LOCATE in the training phase. This model could be applied to any test dataset (it need not be the same dataset or have the same degree or amount of lesions) with image dimensions matching the image training dataset.

4.3. LOCATE_testing:

This function call is used to evaluate the LOCATE model on images in <Test_imgs_directory>. Please note that for running this function, you should have already run LOCATE training (explained in section 4.2) and have a trained model named 'RF_regression_model_LOCATE.mat' available. This function call performs the testing and provides the final thresholded maps and the threshold values.

Example function call

LOCATE_testing(test_image_directory_name , train_image_directory_name);
   - test_image_directory_name - Name of the directory where the test images are located (e.g. '<Test_imgs_directory>')
   - train_image_directory_name - Name of the directory where the images are located. (e.g. '<Training_imgs_directory>')

Main outputs:

- **LOCATE_results_directory** - directory inside <Test_imgs_directory> containing all the LOCATE output files.

- **LOCATE_results_directory/<subjectname>_BIANCA_LOCATE_binarylesionmap.nii.gz** – binary lesion map obtained as the final output of LOCATE.

Additional/optional inputs to the function call

1. LOCATE_testing(test_image_directory_name, train_image_directory_name, feature_select)
   If you want to select specific features for testing
   - feature_select - vector with elements indicating if the feature has to be included or not (1 – to be included, 0 to be discarded). Current order is: distance from ventricles, size of individual lesions (automatically calculated from the LPM) and mean intensity in various modalities (e.g. If <base_modality_name> is the only modality provided and distance from ventricles is not needed, then the function call would be LOCATE_testing(test_image_directory_name, train_image_directory_name, [0, 1, 1]);

   Note: The feature_select option values must consistent with the values used for training LOCATE (explained in section 4.2).

2. LOCATE_testing(test_image_directory_name, train_image_directory_name, feature_select, verbose);
   - verbose (default = 0)

Complete list of outputs

The 'LOCATE_results_directory' contains:

1. LOCATE_test_features.mat' – an intermediate file containing features for all the test images in a single file
2. <subjectname>_indexmap.nii.gz – image showing the Voronoi polygons obtained from the 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. <subjectname>_LOCATE_thresholds.mat – array of thresholds (the same thresholds shown in the <subjectname>_thresholdsmap.nii.gz in step 3). This format can be useful for further analysis of the threshold values (e.g., histogram or stats).
6. Consolidated_LOCATE_output.mat – The outputs from 2, 3, 4 and 5 for all the images available in a single .mat file.

5. Viewing results with FSLeyes

The sample output maps and images are shown overlaid on the base modality image (e.g. the FLAIR image).

Example commands (one single line)

1) Base modality image and thresholded binary map

```
fsleyes
MyLOCATE/Test_imgs_directory/LOCATE_results_directory/<subjectname>_
BIANCA_LOCATE_binarylesionmap.nii.gz --cmap yellow
MyLOCATE/Test_imgs_directory/<subject_name>_feature_FLAIR.nii.gz --
cmap greyscale &
```

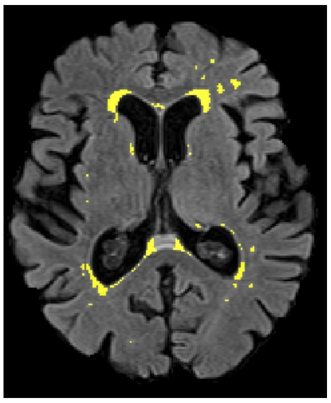

Figure 1: Output binary lesion map <subjectname_BIANCA_LOCATE_binarylesionmap.nii.gz>

2) Base modality image, lesion probability map and thresholded binary map

```
fsleyes
MyLOCATE/Test_imgs_directory/LOCATE_results_directory/<subjectname>_
BIANCA_LOCATE_binarylesionmap.nii.gz --cmap yellow
MyLOCATE/Test_imgs_directory/<subjectname>_BIANCA_LPM.nii.gz --cmap
blue-lightblue
MyLOCATE/Test_imgs_directory/<subject_name>_feature_FLAIR.nii.gz --
cmap greyscale &
```

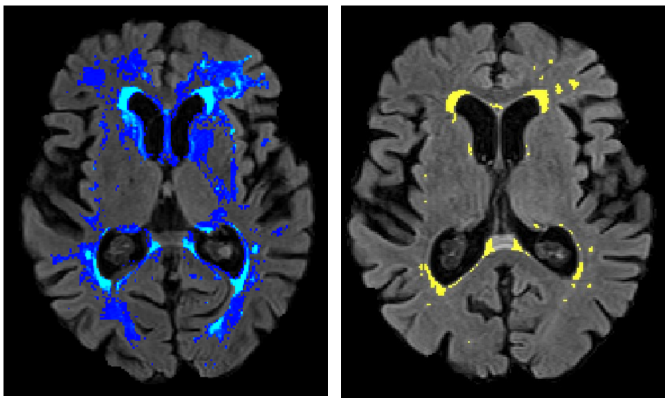

Figure 2: Lesion probability map <subjectname_BIANCA_LPM.nii.gz>

Output binary lesion map <subjectname_BIANCA_LOCATE_binarylesionmap.nii.gz>

3) Base modality image, lesion probability map and intermediate files (Voronoi tessellation and local thresholds map)

```
fsleyes
MyLOCATE/Test_imgs_directory/LOCATE_results_directory/<subjectname>_
BIANCA_LOCATE_binarylesionmap.nii.gz --cmap yellow
MyLOCATE/Test_imgs_directory/LOCATE_results_directory/<subjectname>_i
ndexmap.nii.gz --cmap random
MyLOCATE/Test_imgs_directory/LOCATE_results_directory/subjectname>_thr
esholdsmap.nii.gz --cmap red-yellow
MyLOCATE/Test_imgs_directory/<subject_name>_feature_FLAIR.nii.gz --
cmap greyscale &
```

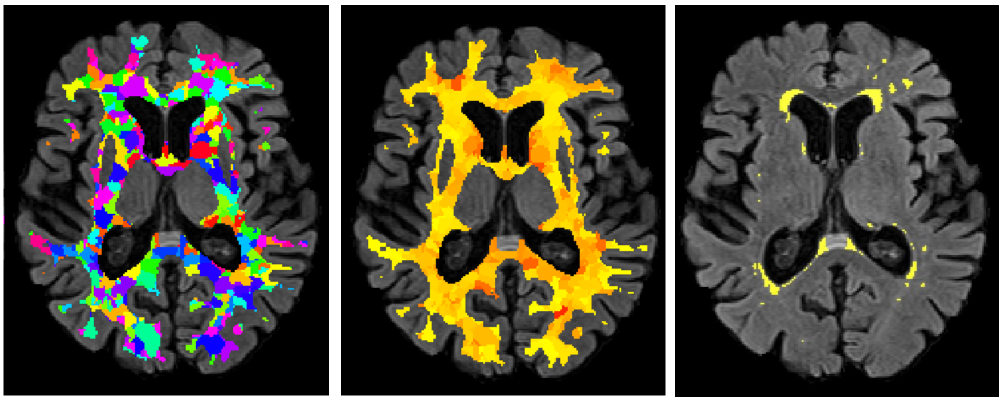

Figure 3: Voronoi tessellations <subjectname_indexmap.nii.gz>

Map showing local thresholds <subjectname_thresholdsmap.nii.gz>

Output binary lesion map <subjectname_BIANCA_LOCATE_binarylesionmap.nii.gz>