LST: A lesion segmentation tool for SPM

Manual/Documentation for version 2.0.15

April 2016

Paul Schmidt

Lucie Wink

Contents

Introduction

The toolbox "LST: Lesion Segmentation Tool" is an open source toolbox for SPM that is able to segment T2 hyperintense lesions in FLAIR images. Originally developed for the segmentation of MS lesions it has has also been proven to be useful for the segmentation of brain lesions in the context of other diseases, such as diabetes mellitus or Alzheimer's disease.

Currently, there are two algorithms implemented for lesion segmentation. The first, a lesion growth algorithm (LGA), requires a T1 image in addition to the FLAIR image. The second algorithm, a lesion prediction algorithm (LPA), requires a FLAIR image only but is currently still in beta mode. As a third highlight a pipeline allowing the longitudinal segmentation is implemented. In addition, the toolbox can be used to fill lesions in any image modility. We hope that these algorithms will be able to contribute to current research in MS and other disciplines.

The toolbox was developed by a cooperation of the following organizations: Morphometry Group, Department of Neurology, Technische Universität München (TUM), Munich, Germany; Department of Statistics, Ludwig-Maximilians-University, Munich, Germany; and Structural Brain Mapping Group, Departments of Neurology and Psychiatry, Friedrich-Schiller-University, Jena, Germany.

1 Getting started

1.1 License

The LST toolbox is available to the scientific community under the terms of the GNU General Public License. A copy of the GNU General Public License is received along with this toolbox.

1.2 Installation

To install the LST toolbox, visit the toolbox-website and download the zipped folder LST_x.x.x.zip, where x.x.x may be replaced by the newest version of the toolbox. Unzip the folder into the SPM12 toolbox folder.

1.3 Starting the toolbox

To start the toolbox, run MATLAB and SPM12. Select LST from the drop-down menu of the toolbox-button. The toolbox will open in SPM's second window. By

clicking on "LST", access to different parts of the toolbox is given.

1.4 Preparation

Before segmenting lesions for several subjects it is advisable to put some thought in the organization of your data and the structure of folders on your disk. In our experience it is best to create one folder for each subject (e.g. `subj01`, `subj02`, ...). Keep the names of the images simple (e.g. `t1.nii` and `flair.nii`) and free from any personal information of the patient. If images for multiple time points are available they may be all in the same folder but clearly distinguishable (e.g. `t1_01.nii`, `t1_02.nii`, ...) or in seperate folders.

1.5 Referencing LST

We put a lot of effort in the creation of this toolbox, so please cite it accordingly.

Minimal version If you used the LGA, please use something similar to the following paragraph:

> Lesions were segmented by the lesion growth algorithm (Schmidt et al., 2012) as implemented in the LST toolbox version x.x.x (www.statistical-modelling.de/lst.html) for SPM.

In addition, mentioning how the optimal initial threshold was determined is advised. If this was done by visual inspection, then just add this information. If you compared the segmentations to reference images, then please provide some details about how these images were obtained and what metric you used for comparison.

If lesions have been segmented by the LPA instead, please use something like the following paragraph:

> Lesions were segmented by the lesion prediction algorithm as implemented in the LST toolbox version x.x.x (www.statistical-modelling.de/lst.html) for SPM.

Here, please add information about the images you used, i.e. did you use the FLAIR images only or did you also use reference images?

If you used the longitudinal pipeline then it is best to first explain the independent lesion segmentation by one of the suggestions given above. You can then add something like the following:

> Segmented lesion maps of different time points were compared using the longitudinal pipeline implemented in the LST toolbox.

More detailed version Feel free to use the following paragraphs if you want or need to explain the algorithms in more detail.

For the LGA use a paragraph such as

> Lesions were segmented by the lesion growth algorithm (Schmidt et al., 2012) as implemented in the LST toolbox version x.x.x (www.statistical-modelling.de/lst.html) for SPM. The algorithm first segments the T1 images into the three main tissue classes (CSF, GM and WM). This information is then combined with the coregistered FLAIR intensities in order to calculate lesion belief maps. By thresholding these maps with a pre-chosen initial threshold (κ) an initial binary lesion map is obtained which is subsequently grown along voxels that appear hyperintense in the FLAIR image. The result is a lesion probability map.

In addition, please mention how the optimal initial threshold was determined. If this was done by visual inspection, then just add this information. If you compared the segmentations to reference images, then please provide some details about how these images were obtained and what metric you used for comparison.

For the LPA use a paragraph such as

> Lesions were segmented by the lesion prediction algorithm as implemented in the LST toolbox version x.x.x (www.statistical-modelling.de/lst.html) for SPM. This algorithm consists of a binary classifier in the form of a logistic regression model trained on the data of 53 MS patients with severe lesion patterns. Data were obtained at the Department of Neurology, Technische Universität München, Munich, Germany. As covariates for this model a similar lesion belief map as for the lesion growth algorithm (Schmidt et al., 2012) was used as well as a spatial covariate that takes into account voxel specific changes in lesion probability. Parameters of this model fit are used to segment lesions in new images by providing an estimate for the lesion probability for each voxel.

Here, please add information about the images you used, i.e. did you use the FLAIR images only or did you also use reference images?

A more detailed version for the longitudinal pipeline is given next:

Segmented lesion maps of different time points were compared using the longitudinal pipeline implemented in the LST toolbox. This pipeline consists of the following steps: First, lesion maps and FLAIR images are coregistered to the images of the first time point. In the next step, relative differences of FLAIR intensities are calculated along all voxels that were segmented as lesions in at least one time point. Finally, significant increase and decrease of lesion voxels are identified if their differences exceed or fall below a certain threshold that is obtained by analyzing healthy white matter. As a final result, lesion change labels are produced for all consecutive time points. In these images the three possible cases decrease, no change and increase are labeled by the numbers 1, 2, and 3, respectively.

2 Lesion segmentation

Currently, there are two algorithms implemented for the segmentation of lesions, the LGA and the LPA. In addition, a longitudinal pipeline is available. The usage of these methods is explained in the following subsections.

2.1 Lesion growth algorithm (LGA)

This section deals with the lesion growth algorithm (LGA) as presented in Schmidt et al. (2012).

2.1.1 Segmentation

Lesion segmentation by the LGA requires a T1 and a FLAIR image. Furthermore, the user has to specify an initial threshold, κ. If the algorithm is applied for the first time to new data we recommend to choose different values for κ (e.g. in the range of 0.05 to 1.0 with an increment of 0.05) and to obtain the best value by either visual inspection of the results or by the comparison with reference segmentations, if available. For the latter strategy we implemented a special routine, see section 2.1.2.

Input The user needs to select the original T1 and FLAIR images for each subject. The FLAIR images must be in the same order as the T1 images.

Options   The main parameter that needs to be set by the user is κ (Initial threshold). In addition, the user can specify the strength of the Markov random field (MRF parameter) as well as the maximum number of iterations (Maximum iterations) for the LGA. While we do not recommend to change the former parameter the maximum number of iterations should be increased if it reaches its limit during the segmentation process. Finally, the user can choose if the results of the segmentation should be summarized in a HTML report. Although it needs a while to produce this report we recommend its usage as it makes it easier to check the results. See section 3.4 on how to merge multiple HTML reports.

Output   The LGA returns a bias corrected version of the FLAIR image (`rm[FLAIR]`
`.nii`) that is coregistered to the T1 image as well as a lesion probability map
(`ples_lga_[`κ`]_rm[FLAIR].nii`). In addition, a .mat-file (`LST_lga_[`κ`]_rm[FLAIR]`
`.mat`) is produced which contains all necessary components that are needed for a
re-run of the algorithm, the longitudinal pipeline, or for lesion filling. If the user decided to do so a HTML report (`report_LST_lga_[`κ`]_rm[FLAIR].html`) is saved
as well as a folder that contains images that are displayed in the report.

2.1.2   Determination of the optimal initial threshold

With this module we offer the opportunity to determine the optimal κ based on the
Dice coefficient, see Schmidt et al. (2012) for details. This requires the existence
of reference segmentations to be compared with the lesion maps. These reference
images are a binary images in the space of the T1 images where a 1 indicates a
lesion.

Input   As input the user needs to choose the binary reference images. The algorithm automatically searches the folder of each reference image for lesion probability
maps obtained by LGA, thus for each patient the reference images need to be in
the same folder as the lesion probability maps.

Options   The threshold for the computation of binary lesion maps must be specified. This must be a value between 0 and 1.

Output   This routine saves a CSV file (`LST_doit_[date]_[time].csv`) in MAT-LAB's current directory. The CSV file contains columns for the folder of the reference images, the name of the FLAIR images, the value of κ, the coresponding Dice
coefficients as well as values for sensitivity and specifity.

2.2   Lesion prediction algorithm (LPA)

Lesion segmentation by the LPA requires a FLAIR image only. However, the user is free to choose an additional image that serves as a reference image during a coregistration step before the main lesion segmentation. This may be helpful if the dimension of the FLAIR image is low or if the goal of the lesion segmentation is to fill lesions in T1 images. Beside that no additional parameters need to be set.

Input   As input the user needs to choose the original FLAIR images. Additional reference images can be chosen but must be in the same order as the FLAIR images.

Options   The user can choose if the results of the segmentation should be summarized in a HTML report. Although it needs a while to produce this report we recommend its usage as it makes it easier to check the results. See section 3.4 on how to merge multiple HTML reports.

Output   If no reference image has been specified this algorithm saves a bias corrected version of the FLAIR image (`m[FLAIR].nii`). If a reference image has been chosen the bias corrected coregistered FLAIR image (`mr[FLAIR].nii`) is saved. Besides the lesion probability map (`ples_lpa_m[r][FLAIR].nii`) an additional .mat-file (`LST_lpa_m[r][FLAIR].mat`) is produced which contains all necessary components that are needed for the longitudinal pipeline or for lesion filling. If the user decided to do so a HTML report (`report_LST_lpa_m[r][FLAIR].html`) is saved as well as a folder that contains images that are displayed in the report.

2.3   Longitudinal pipeline

The longitudinal pipeline is able to compare segmented lesion maps for different time points. The segmentations can be derived either by the LGA or the LPA, but not both. Furthermore, it is required that the FLAIR images and .mat-files that are saved during the lesion segmentation process are available in the same folders as the lesions probability maps.

   The pipeline proceeds by comparing all consecutive time points in an iterative manner. It decides if changes in lesion structure are significant or due to natural variations of the FLAIR signal. Non-significant changes are labeled as lesions in both probability maps, thus, probability lesion maps are corrected within this procedure and may differ from the ones that served as input. As a final result, lesion change labels are produced for all consecutive time points. In these images the three

possible cases decrease, no change and increase are labeled by the numbers 1, 2, and 3, respectively.

In addition, a lesion change plot is constructed. This plot shows the lesion volumes for both time points of all segmented lesions. This way it is easy to recognize how the lesion structure has been changed, i.e. if the change occurred by the appearance of new lesions, the disappearance of old lesions, by the change of already existing lesions, or a combination of these possibilities.

Input   First, the user needs to create one item for each time point. Then, lesion probability maps for each time point need to be selected. For all time points the images must be in the same order.

Options   The user can choose if the results of the segmentation should be summarized in a HTML report. Although it needs a while to produce this report we recommend it as it makes it easier to check the results. See section 3.4 on how to merge multiple HTML reports.

Output   The longitudinal pipeline produces corrected lesion maps for each time point (`lples...`) and lesion change labels (`LCL_[nam_ples_t]_[nam_ples_t+1]` `.nii`) for consecutive time points. If the user decided to do so HTML reports (`report_LST_long_[#time_points]_....html`) are saved for the comparisons as well as a folder that contains images that are displayed in the report.

3   Utilities

3.1   Lesion filling

Lesion filling can be applied to any image that is in alignment with the lesion probability map. However, it is required that the .mat-files that are saved during the lesion segmentation process are available in the same folders as the lesions probability maps.

Input   The user needs to specify the images that need to be filled (Images in native space) and the probability lesion maps (obtained either by the LGA, LPA or longitudinal pipeline). The images must be selected in the same order.

Options   The user can choose if the results of the filling process should be summarized in a HTML report. Although it needs a while to produce this report we recommend its usage as it makes it easier to check the results. See section 3.4 on how to merge multiple HTML reports.

Output   This procedure saves the filled image in native space (`[image]_filled_lga _[`κ`]_rm[FLAIR].nii` for LGA or `[image]_filled_lpa_m[r][FLAIR].nii` for LPA). If the user decided to do so a HTML report (`report_LST_filled_m[r][FLAIR].html`) is saved as well as a folder that contains images that are displayed in the report.

3.2   Extract values of interest

This routine offers the possibility to extract some values that may be of interest. At this point in time this includes the total lesion volume (in ml) and the number of lesions.

Input   As input images this function requires the lesion probability maps obtained either by LGA, LPA, or the longitudinal pipeline.

Options   The threshold for the computation of binary lesion maps must be specified. This must be a value between 0 and 1.

Output   Extracted values are saved in a CSV file (`LST_tlv_[binthresh]_[date]_ [time].csv`) in MATLAB's current directory.

3.3   Create binary lesion maps

Binary lesion maps can be created using this utility.

Input   Lesion probability maps need to be specified by the user.

Options   The threshold for the computation of binary lesion maps must be specified. This must be a value between 0 and 1.

Output   This function saves the binary lesion map (`bles_[binthresh]_...` for LGA and LPA, and or `lbles_[binthresh]_...` for the longitudinal pipeline).

3.4   Merge HTML reports

This function allows to merge multiple HTML reports that have been obtained by different functions of this toolbox. The resulting report can be moved anywhere on your disk as long as it is provided that the folders for the original reports can be found in their original directory. A report that is readable on different platforms can be obtained by exporting/printing the HTML report as a PDF document, a function that is included in all modern browsers.

Input   The HTML reports must be selected.

Options   No options need to be specified.

Output   This function writes a HTML report (`report_LST_[number of reports]_[date]_[time].html`) into MATLAB's current directory.

References

Paul Schmidt, Christian Gaser, Milan Arsic, Dorothea Buck, Annette Förschler, Achim Berthele, Muna Hoshi, Rüdiger Ilg, Volker J Schmid, Claus Zimmer, et al. An automated tool for detection of FLAIR-hyperintense white-matter lesions in multiple sclerosis. *Neuroimage*, 59(4):3774–3783, 2012.