LST: A Lesion Segmentation Tool For SPM

manual for version 1.2.3
March 12, 2013

Paul Schmidt
Mark Mühlau
Christian Gaser
Lucie Wink

Contents

1 Introduction

The toolbox "LST: Lesion Segmentation Tool" was developed, see Schmidt et al. (2012), by a coorperation of the following organizations: Morphometry Group[1], Department of Neurology, Technische Universität München (TUM), Munich, Department of Statistics[2], Ludwig-Maximilians-University, Munich, Germany, and Structural Brain Mapping Group[3], Departments of Neurology and Psychiatry, Friedrich-Schiller-University, Jena, Germany. The reason for its development was the need for an implemented open source tool for the segmentation of lesions in Multiple Sclerosis (MS). Even though it is still under construction, we hope that the tool will be able to contribute to current MS research.

At this point in time, only a lesion growth algorithm is implemented, see Schmidt et al. (2012). We may include further lesion segmentation algorithms and other utility functions to this toolbox in the future.

2 Getting started

2.1 License

The LST toolbox is available to the scientific community under the terms of the GNU General Public License[4]. A copy of the GNU General Public License is received along with this toolbox.

2.2 Installation

To install the LST toolbox, visit the toolbox-website[5] 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 SPM8 `toolbox` folder.

The toolbox requires the newest version of the vbm8 toolbox for SPM that is available online[6] for free. Although the installation of vbm8 should be rather simple, please consult the vbm8 manual if you have any questions or problems.

[1] http://www.neurokopfzentrum.med.tum.de/neurologie/425.html
[2] http://www.stat.uni-muenchen.de
[3] http://dbm.neuro.uni-jena.de
[4] http://www.gnu.org/licenses/gpl.html
[5] http://www.applied-statistics.de/lst.html
[6] http://dbm.neuro.uni-jena.de/vbm/

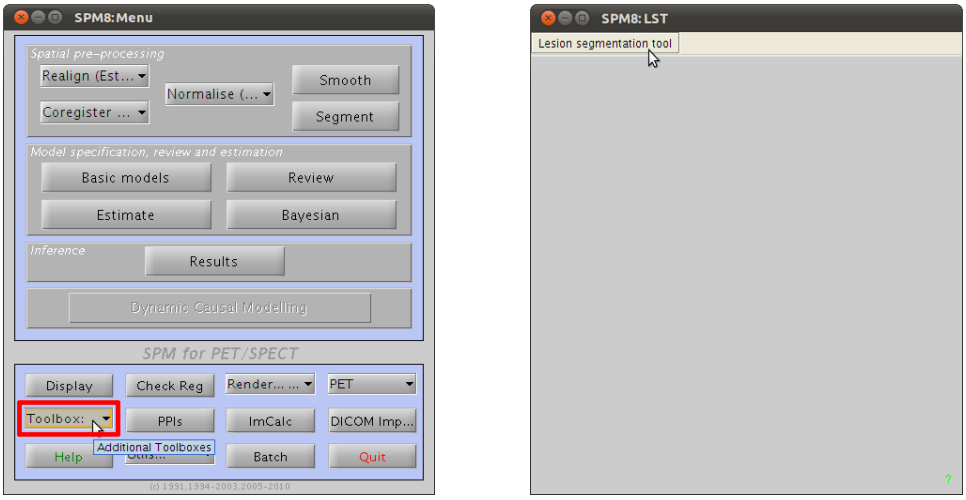

Figure 1: Menu for SPM8 and LST.

2.3 Starting the toolbox

To start the toolbox, run MATLAB and SPM8. Select LST from the drop-down menu of the toolbox-button (left panel in Figure 1). The toolbox will open in SPM's second window (right panel in Figure 1). By clicking on "Lesion segmentation tool", access to different parts of the toolbox is given. At this point in time only a lesion growth algorithm is implemented.

3 Lesion segmentation via a lesion growth algorithm

This section presents an algorithm for the segmentation of MS lesions which is based on lesion growth. For details on this algorithm, see Schmidt et al. (2012).

One of the following situations will be appropriate for the needs of most users.

- For applying the algorithm for the first time to new data, the module presented in section 3.1 can be used with different values for the initial threshold κ (e.g. in the range of 0.05 to 1.0 with an increament of 0.05). Then, see section 3.2 for strategies to find the optimal κ for these data.
- If an optimal κ has already been estimated (e.g. with one of the strategies presented in section 3.2), new images can be segmented with the module presented in section 3.1 by choosing the opimal κ as the initial threshold.
- If a PVE-label of the T_1 image and a bias-corrected version of the FLAIR image is already available (e.g. by an earlier run of this tool, vbm8 or other software), the module in section 3.3 can be used to segment lesions with either the optimal value of κ (derived from a former analysis) or different values of κ (e.g. in the

Name	Type	Explanation
`T1`	input	T_1 image
`F2`	input	FLAIR image
`mT1.nii`	output	bias-corrected T_1 image, needed for lesion filling, section 4.1
`mF2.nii`	output	bias-corrected FLAIR image
`rmF2.nii`	output	coregistered bias-corected FLAIR image, i.e. bias-corrected FLAIR image in the space of the T_1 image
`y_T1.nii`	output	forward deformation field, needed for warping images into MNI space
`iy_T1.nii`	output	backward deformation field, needed for warping MNI white matter template into native space
`p0T1.nii`	output	PVE-label of the T_1 image
`lesion_lbmz_xxx_rmF2.nii`	output*	probability lesion map, where `xxx` is replaced with the corresponding value of κ and `lbmz` indicates the lesion belief map that was used for initialization
`b_yyy_lesion_lbmz_xxx_rmF2.nii`	output*	binary lesion map, where `yyy` is replaced by the binary lesion threshold
`wlesion_lbmz_xxx_rmF2.nii`	output*	probablity lesion map in MNI space
`wb_yyy_lesion_lbmz_xxx_rmF2.nii`	output*	binary lesion map in MNI space
`indx.mat`	output	position of brain tissue, needed for determining the optimal κ, see section 3.2

Table 1: Input and output files for the module "PVE-label estimation and lesion segmentation". Optional files are marked with an asterisk.

range of 0.05 to 1.0 with an increament of 0.05). In the latter case, see section 3.2 for strategies to find the optimal κ.

3.1 PVE-label estimation and lesion segmentation

For this module, only the original T_1 and FLAIR images are needed. The input and output files for this module are listed in Table 1. The user needs to select the FLAIR images in the same order as the T_1 images. Furthermore, the user has to choose one or more (up to twenty) values for the initial threshold (κ) for the lesion growth algorithm. We have had good experience with values of κ in the range of 0.25 to 0.35, so we set 0.3 as the default value. However, since the choice of this parameter is critical we recommend the segmentation of lesions with different values for κ (e.g. in the range of 0.05 to 1.0 with an increament of 0.05) and choosing one of the strategies presented in

section 3.2 to find the optimal κ. If the optimal κ is already known, choose this value for the initial threshold.

Further options includes the choice of the Markov Random Field (MRF) parameter, the maximum number of iterations and the choice of a threshold for producing binary lesion maps. The latter is necessary for saving binary lesion maps which are thresholded versions of the probability lesion maps.

The **writing options** allow to control which images should be saved (see output files in Table 1). To this end probability lesion maps, binary lesion maps and the normalized versions of these images can be saved. The forward deformation field of the PVE-label estimation is used for normalization. Per default, images that are needed for lesion segmentation are saved. To be precise, these images are the T_1 and FLAIR images corrected for bias, the coregistered bias-corrected FLAIR image, the estimated PVE label as well as the forward and inverse deformation field. These images are needed to run the lesion growth algorithm again (e.g. with a different initial threshold) without estimating the PVE label and the bias correction of the FLAIR image, see section 3.3.

To **run** the algorithm, press the green triangle at the top of the batch editor. The algorithm will start with the estimation of the PVE labels for all specified T_1 images and will proceed with the bias correction of the selected FLAIR images. When these processes have been succesfully terminated, the lesion growth algorithm will start. On a computer with 3.2 GHz processor and 8 GB RAM the first two steps take about 10 minutes in total for a single subject. The lesion growth algorithm takes another 2-3 minutes depending on the resolution of the T_1 image, the total lesion volume and thus the number of iterations.

3.2 Strategies for the determination of the initial threshold

As the choice of the initial threshold κ has a huge influence on the resulting segmentation, the user has to choose an optimal value of this threshold that fits best to the data. This sections presents two methods for determining the optimal κ:

- If no reference image is available, the user can perform a visual inspection of the different segmentations.
- If binary reference images are available, the user can compare the segmented lesion maps to these reference images based on the Dice Coefficient (DC) with the module “Determination of the optimal initial threshold”.

Visual comparison

If no reference images for the considered subjects are present, visual inspection (e.g. with SPMs `Check coreg` function) can be used to perform a visual comparison of the

Name	Type	Explanation
`lesion_lbmz_xxx_F2.nii`	input	probability lesion maps, where `xxxx` is replaced with the corresponding value of κ and `lbmz` indicates the lesion belief map that was used for initialization
`reference`	input	binary reference segmentations
`kappa_dice_se_sp.mat`	output	matrix with the calculated values of the DC, sensitivity and specificity
κ vs. DC	plot	jittered plot of κ vs. the calculated Dice Coefficients

Table 2: Input and output files for the module "Determination of the optimal initial threshold".

segmented lesion maps. One of the most important aspects of the visual comparison is to make sure that the algorithm caught most (hopefully all) lesions without segmenting non-lesion areas.

Comparative Analysis based on the Dice Coefficient

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 a reference segmentation to be compared with the lesion map. This reference image is a binary image in the space of the T_1 image where a 1 indicates a lesion. Other software such as the free FSL, the commercial Amira or any other suitable software can be used to construct these reference images.

Table 2 shows the input and output files of this module. After selecting the probability lesion maps, the reference images have to be chosen. The number of chosen lesion probabilty maps for one subject is of lesser importance than the fact that for every subject one reference image is present in the same folder as the probablity lesion maps. As the module expects a binary reference image, the user has to choose a binary threshold to produce a binary lesion map that can be compared to the reference image.

The algorithm calculates the Dice Coefficients, sensitivity and specificity. Finally, the results are saved in the file `kappa_dice_se_sp.mat` in the user's current working directory and a plot of the Dice Coefficients along the values of κ will pop up.

3.3 Lesion segmentation (PVE-labels already available)

This module offers the opportunity to segment MS lesions without the time consuming estimation of the PVE-label and bias correction of the FLAIR image given that these images are already available. The input, output and required files are listed in Table

Name	Type	Explanation
`p0T1.nii`	input	PVE-label of the T_1 image
`mF2.nii`	input	bias-corrected FLAIR image
`iy_T1.nii`	required	backward deformation field, needed for warping MNI white matter template into native space
`lesion_lbmz_xxx_rmF2.nii`	output*	probability lesion map, where `xxx` is replaced with the corresponding value of κ and `lbmz` indicates the lesion belief map that was used for initialization
`b_yyy_lesion_lbmz_xxx_rmF2.nii`	output*	binary lesion map, where `yyy` is replaced by the binary lesion threshold
`wlesion_lbmz_xxx_rmF2.nii`	output*	probablity lesion map in MNI space
`wb_yyy_lesion_lbmz_xxx_rmF2.nii`	output*	binary lesion map in MNI space
`indx.mat`	output	position of brain tissue, needed for determining the optimal κ, see section 3.2

Table 3: Input, output and required files for the module "Lesion segmentation (PVE-labels already available)". Optional files are marked with an asterisk.

3. Besides the PVE-label and the (coregistered) bias-corrected FLAIR image, the backward deformation field must also be present in the same folder. The batch editor of this module is the same as the one from the previous section. The user only needs to choose the PVE-labels and bias-corrected FLAIR images and can perform the lesion segmentation with up to twenty different initial thresholds. If the bias-corrected FLAIR images are not already coregistered to the T_1 images, they will be coregistered at the beginning of the algorithm.

3.4 How to proceed further

We expect that the refilled native T_1-weighted images can be used to analyze grey matter with available tools. Yet our own experience is restricted to vbm8. Here, normalization including high-dimensional warping is feasible. To further analyze white-matter lesions, lesion maps should be normalized. We suggest the use of the normalization matrix derived from the normalization of the native refilled T_1-weighted image. Dependent on the issue under investigation, lesion maps can be subjected in a binary or non-binary manner. In our opinion thresholding should only be applied with good reason. It is critical as it will inherently become effective at the edges of the lesion

resulting in considerable changes in the estimated volume. E.g. if different thresholds result in a spherical lesion of 9mm or 10mm in diameter. Volumes will differ by factor 1.37 ($= 1000/729 = 10^3/9^3$).

Note that application of a deformation matrix to a binary map will usually result in a non-binary normalized map necessitating subsequent thresholding. Alternatively, the interpolation method of nearest neighbor (as implemented in vbm8 as an option) can be used. This option yields binary maps given that binary maps had been subjected. Further note that the Display function of SPM8 also applies some interpolation by default. This interpolation must be switched off to evaluate whether the map is binary or not (pull-down menu in the lower right corner: change bilin interp to NN interp).

4 Utilities

Name	Type	Explanation
`T1`	input	PVE-label of the T_1 image
`(b_yyy_)lesion_lbmz_xxx_F2.nii`	input	binary probability lesion map
`mT1.nii`	required	bias-corrected T_1 image
`p0T1.nii`	required	PVE-label of the original T_1 image
`y_T1.nii`	required*	forward deformation field
`T1_filled.nii`	output*	T_1 image with filled lesions
`mT1_filled.nii`	output*	bias-corrected T_1 image with filled lesions
`wT1_filled.nii`	output*	T_1 image with filled lesions in MNI space
`wmT1_filled.nii`	output*	bias-corrected T_1 image with filled lesions in MNI space

Table 4: Input, output and required files for the module "Lesion filling". Optional files are marked with an asterisk.

4.1 Lesion filling

This module can be used to fill the segmented lesions in the T_1 image with estimated healthy white matter tissue. The underlying algorithm is inspired by a method proposed by Chard et al. (2010). The input, output and required images are listed in Table 4. In this module, the user has to select the original T_1 images and the corresponding lesion probability maps. Either the probability lesion map or the binary lesion map can be used. Further, the bias-corrected T_1 image is required in order to fill the lesions. As

output one can choose between the original filled T_1 image, the bias-corrected filled T_1 image and the warped versions of these two images. For warping, the forward deformation field is required.

Name	Type	Explanation
`lesion_lbmz_xxx_F2.nii`	input	probability lesion map
`b_yyy_lesion_lbmz_xxx_F2.nii`	output	binary probability lesion map

Table 5: Input and output files for the module "Create binary lesion map"

4.2 Create binary lesion map

This function computes binary lesion maps out of probability lesion maps. The user has to set a threshold in the interval of $[0, 1]$, where voxels with lesion probabilities lower than this threshold are set to zero. Otherwise the values of voxels are set to one. Beside the threshold, the user has to select the lesion probability maps (see Table 5).

4.3 Compute total lesion volume

This function computes the total lesion volume for given lesion maps. The user can either select probability lesion maps or binary lesion maps (see Table 6). All voxels with values greater than zero will count as lesions, therefore the final total lesion volume will depend on the choice of the lesion map, see section 3.4. The result of a single image is written to a `.txt`-file, the results for all images can be written to a `.csv`-file.

5 Acknowledgement

The authors would like to thank Michael Schneider for helpful comments about the interface and Matthew Senjem for detecting a bug and providing a solution to solve it.

Name	Type	Explanation
`(b_yyy_)lesion_lbmz_xxx_F2.nii`	input	probability lesion map
`tlv_(b_yyy_)lesion_lbmz_xxx_F2.txt`	output	Total lesion volume in ml
`tlv_all.csv`	output*	Total lesion volume in ml for all images

Table 6: Input and output files for the module "Compute total lesion volume". Optional files are marked with an asterisk.

References

D.T. Chard, Jackson J.S., Miller D.H., and Wheeler-Kingshott C.A. Reducing the impact of white matter lesions on automated measures of brain gray and white matter volumes. *Journal of Magnetic Resonance Imaging*, 32(1):223–228, 2010.

P. Schmidt, Gaser C., Arsic M., Buck D., Förschler A., Berthele A., Hoshi M., Ilg R., Schmid V.J., Zimmer C., Hemmer B., and Mühlau M. An automated tool for detection of FLAIR-hyperintense white-matter lesions in Multiple Sclerosis. *NeuroImage*, 59:3774–3783, 2012.