SynapseLocator application

Repeated imaging of brain tissue at high spatial resolution is faced with minute movements and non-concordant positional changes of synapses. SynapseLocator is a versatile tool that combines many steps of image processing, non-rigid image registration, and spot localisation. Input data are typically 3D image stacks from pre- and post-stimulation (square input images in form of interleaved two-channel data, ScanImage format, scanimage.vidriotechnologies.com, 16-bit greyscale) for which numerical and graphical result data are produced. Input images are filtered and deconvolved which considerably improves registration and spot detection. By default, results are reported based on processed images. However, channel intensities for minimally (or non) pre-processed images are also reported.

SynapseLocator Installation

SynapseLocator combines several image processing steps into a single graphical user interface written in Matlab (Matlab version 2018b, Windows 7 and Windows 10 operating system). In order to run the software ImageJ and elastix must be installed (see below).

Prepare ImageJ/Fiji

Install Fiji (https://fiji.sc/), note that Fiji should not be installed to "Program Files". Choose rather “C:\fiji.app” to make it system-wide available or chose a directory somewhere in your user space (e.g., "C:\Users\[your name]\Fiji.app"). Download code for DeconvolutionLab2 (http://bigwww.epfl.ch/deconvolution/deconvolutionlab2/) and place the jar file into the plug-ins folder of Fiji. Start Fiji and run the “Update...” command from the Help drop down menu. Click on “Manage update sites” and add update sites “ImageScience” and “3D ImageJ Suite”. Click Apply changes, restart Fiji.

Prepare elastix

Install elastix transformation software (http://elastix.isi.uu.nl/). Download the latest version (elastix-5.0.0) from github (https://github.com/SuperElastix/elastix/releases/tag/5.0.0) (versions 4.8.0 and 4.9.0 were also tested). Unpack the zipped file and move all files into new folder “~\ SynapseLocator\elastix\elastixProgram”.

Prepare Matlab

Add the “SynapseLocator” directory with all its subfolders to the Matlab search path. Test images are also included (see sub-directory “Data”). Use Matlab (“Open as text”) to open the parameter file “synapseLocatorParams.txt” from the SynapseLocator directory and edit line 19 “IJ_exe, C:\ Fiji.app\ImageJ-win64.exe” to update the path setting for your installed Fiji executable.

Prepare Input

To check the software functionalities, load the example data (“~\Data”, stack1.tif and stack2.tif) which contain typical experimental data (G0 holds the “green” data from time point 1, R0 the corresponding “red” channel data and so forth). Alternatively, load your own input data (check for the required format, square images, interleaved two channel data). It is recommended to start with small image stacks (256×256 pixels and a height of approximately 5-10x the spot diameter) to limit the computation time. If necessary, crop larger image stacks using ImageJ and search for a region that contains a number of spots in various environments (e.g., isolated spots and spots that have neighbouring spots at a close distance). If necessary, region masking (i.e. soma) should also be done with ImageJ. Save stacks to a separate analysis directory.

It is advised to process images including a deconvolution step prior to registration and spot finding and this is part of SynapseLocator's workflow. However, preprocessing steps can be tested independently by starting ImageJ macro "preprocess.ijm" (see SynapseLocator directory "~/fiji/IJ1Macros") from ImageJ's macro editor. Preprocessing sequentially applies median filter/Gaussian filter, bandpass filter, deconvolution (DeconvolutionLab2), and background subtraction. Parameters for the individual steps, PSF size, and voxel size are set interactively. Default values apply to the example data set but can be adjusted by changing the respective fields in the "synapseLocatorParams.csv" file. Strictly avoid settings that distort the image. Adjustment of the PSF parameters and the voxel size according to the resolution of your microscope may be needed.

Using Synapse Locator

Start SynapseLocator GUI from Matlab command window (command "SynapseLocator"). The program reads parameters from the enclosed "synapseLocatorParams.txt" file and starts with default settings and loads a model for spot detection.

Click the "Params" button (upper right) to access the parameter settings. An extended list of parameters is accessible through the "Expert" button but do not modify "Expert Settings" at this stage (see below). Keep default settings from the "Image (Pre)Processing" panel (note that changes made to parameters in this panel must precede image loading!). Click "Load Data" and select image stacks from time-point #1 and #2 ("Data" directory, tif files "stack1.tif" and "stack2.tif"). Accept the suggested output folder. (Click "No" in the "Load feature set" pop-up, see "Using SynapseLocator with previously registered images"). Adjust parameters in the "Preprocess params" pop-up to match the current experiment. In particular, check PSF and voxel size (default settings work for the example data set). SynapseLocator starts loading images and runs preprocessing. This may take a few minutes depending on PC features. The status indicator in the GUI header displays information on the current processing step. When finished, an image from G0 (="leading channel", time-point #1, at the approximate stack center section, see "z Level" field) is displayed.

Inspect images by changing between channels and time-points using buttons from the “Image” panel (i.e. G0, G1, R0, R1). Use Matlab’s Data Cursor button (upper left corner of GUI) to activate the Datatip function (i.e. little cross icon) and inspect the image intensity in image G0 and G1 around spot positions. Adjust the threshold settings (parameter fields, “Intensity Threshold” panel); suggested values (derived from histogram thresholding using Otsu's method) might be too low. Choose values above background pixel values. Options in “Image Similarity” and “Marker Density” allow to set the registration parameters so that they match the appearance of the submitted image stacks (i.e. image similarity judged by visual inspection and sparsity of labelling). The “Iterations N” panel allows to switch between a quick exploratory run and more elaborated registration. Check “quick” for a first trial.

The whole process comprising registration and spot finding can be started by clicking “Run”. However, first time users should run the two steps separately (use the “single/two step process” toggle just beside the “Run” button(s) to switch between combined and consecutive two-step processing) to speed up potential troubleshooting. For now, click on “RUN elastix”. SynapseLocator opens a Windows console in which elastix registration is started. The calculated transformation field is then used to register images from the second time-point (G1/R1) to the first time-point. Watch the status indicator to see when registration is finished. Check the “Image Match” panel to evaluate the success of the registration. Reported is the Pearson correlation coefficient considering voxels with intensities above user-set intensity thresholds (green channel, pre and post registration). The absolute numbers vary with image quality, but in general, an increase by 10 percentage points indicates a successful registration. Visually check for successful registration and continue with spot localization.

The standard spot model (“SL_genericSpotModel”, in “Spot Detection” panel) is a good point to start. Adjust pre/post threshold values for the green channel signal (“Spot Threshold” panel). Taking values from the “Registration Threshold” panel is good starting point but shifting threshold to somewhat higher values may speed up calculation while fully maintaining spot detection ability. Again, visually inspect typical spot areas, choose values well above background (the localization step can be rerun easily with modified thresholds), and click on “RUN locator”. Processing steps can be followed at the SynapseLocator progress bar. When finished, SynapseLocator displays the number of detected spots and assigned labels in the main GUI area (“Locator” panel). To inspect the spots in the main GUI click on Reset (“R”, GUI upper left corner) and toggle visibility of identified spot locations (clicking “Spot” panel header). Use “Spot Center” and “Spot Voxels” buttons to switch between showing ROIs finally assigned as spots (center position given) and voxels contributing to define the spot region, respectively. Note that for better representation the “Center” option presents identified spots as light green circles at their center z-position and as smaller dark green circles in adjacent sections. Display assigned labels (click “Label” panel header) and switch between default and custom threshold settings for spot R channel analysis. The “custom threshold” option allows to control assignment of labels through the “Synapse Locator Summary Table” GUI (activate “Live Plot” to automatically update “Spot Intensity Histogram” and “Spot Intensity Overview”). The table is initiated with “Label Mode” set to “All” such that information for all detected spots is displayed. You may wish to compare the values from the “..._match” columns that report the Pearson correlation coefficient for voxel intensities from respective channels. A graphical representation is available in the “Spot Intensity Overview” figure (lower left plot). Spots with SynTagMA-labelled neuronal activity are expected to have both high g0g1_match (=excellent registration) and high g0r1_match (=intensity distribution of red channel at time-point #2 follows intensity distribution of green channel at time-point #1). “spotMatch_probs” reports in how far the red intensity at time-point #2 matches a model generated from the spots green intensities at time-point #1 (right part in figure). Spot intensity ratios (R/G pre, R/G post, and (R1-R0)/G0) are plotted in figure “Spot Intensity Overview”. Switching between “processed”, “med filtered”, and “raw” modi in the “Data process type” panel allows for fast comparison of processing effects. Switching between “All”, “default”, and “custom” in the “Label Mode” panel activates the respective filtering modus (“Polish” and “Filter by...” panels). The “Polish” settings marks spots with respective features so that they are excluded from further analysis. The “no edges” button and the “no twins” button marks spots laying at the image stack border and spots with center positions too close. Toggling between “default” and “custom” sets upper thresholds for either G0 or R0 intensity. The “default” setting uses a simple outlier detection logic: Spots with intensities above third quartile plus 1.5 the interquantile range value are considered outlier. The “custom” mode reads upper thresholds for either G0 or R0 intensity (check histogram plot) from the GUI. Note that all spots are shown in the “Spot Intensity” plots and symbolized as circles. Open symbols represent excluded spots, light blue symbols represent considered spots, and spots with both blue and red match the respective filter settings. With “default” (“Label Mode” panel) setting the dR/G0 threshold from the main SynapseLocator GUI is used as filter. Switching to “custom” allows to use several spot features for filtering. Check the effect of modifying threshold values in the “Spot Intensity” plots (“Live Plot” activated).

The activated “custom threshold” modus (“Label” panel in SynapseLocator main GUI also activated) allows to zoom in on a spot by a simple mouse click in the respective row of the summary table.

In general, we recommend to inspect pre/post activation intensity ratio, (R1-R0)/(G0), spot size and match quality.

In case, registration or spot localization does not lead to convincing results, changes made in either “Registration Params” or “Spot Detection” panel allow to re-run the respective process with updated settings using the registered image stacks. Changes in the image pre-processing step require reloading of images.

Saving SynapseLocator results

The “Save Results” button from SynapseLocator main GUI saves all data from the current analysis. The program creates a directory “SynapseLocator_{date}_{time}” into which the resulting data are saved. A sub-directory “elastix” is created and contains log files from the registration step. Sub-directory “featureData” holds the features necessary to build a model and locate spots, these data can be kept and reloaded, if the location process should be repeated. Clicking “Save Results” writes results from current analysis state to spreadsheet and saves images (tif format) including spot location and label assignment. Images from image pre-processing are saved at individual step (unprocessed saved as “~_raw.tif”, just median filtered saved as “~_mf.tif”, after applying all filtering saved as “~_proc.tif”). The spreadsheet data holds all relevant data from spot detection (identified spot positions, channel intensities, intensity ratios, shape similarities, spot dimensions). No filtering step is included.

The “Save” button from Synapse Locator Summary Table GUI saves table data and plots using current parameters.

Build a Model

A good spot model is crucial for automatic spot detection. If the default spot model does not provide satisfactory results, a customized model can be easily obtained after very few iterations thanks to the implemented Weka machine learning algorithm (https://www.cs.waikato.ac.nz/~ml/index.html). It classifies voxels as being to either “Spot” or “No Spot” class based on user-defined positions defined as the respective classes. To do so, zoom into a region that contains representative spots. On “Add Class ROIs” panel, click on “Add ROI to” button. Draw a small ROI at the spot center and click the “SPOT” button. This adds the location to the spot list. Further add 3-4 spots. Now add ROIs for positions (again 3-4) between spots (or clearly outside spots) by adding those as “NO SPOT”. Train the machine (click on “Train Classifier” in “ML” panel). A text summary of model performance is printed in the Matlab console (percentage value in line “Correctly Classified Instances...” should approach 100). Try the new model by clicking on “RUN locator” and compare performance to previous results. Remove current locations from the list (right mouse-click) if they seem to deteriorate the model performance. Add further ROIs at positions where spot localization was not satisfactory. Train model again and check performance. Finally save your model (stay with default directory in the SynapseLocator tree).

The model building step can be performed after registration but also prior to registration.

Using SynapseLocator with previously registered images

The spot location process can be run independently of the registration if registered image stacks are available from an earlier run. The relevant files (“G0R0_SLready” and “G1R1_SLready”) are created together with result files from the “Save Results” command. To load registered images, the “Load transformed” button from the “Expert Settings” panel has to be activated. This forces SynapseLocator to skip input processing and to directly enter the spot location procedure. In this context, when asked for “Load feature set” an already calculated set of features may be loaded to reduce computation time. As the image processing step is skipped with this option, the summary table expects to get already processed data and treats all voxel intensities as coming from processed input images.

Expert Settings

The “Image Processing Expert” panel allows to set the leading channel for the spot finding (defaults to “G”). By activating “Load registered images” the initial image processing steps are skipped and a previously registered set of image stacks is loaded. With “Transform Raw Data” registration and spot finding is performed on processed data to achieve best results, but intensities are also reported from reference (= time-point#1) and transformed (= registered time-point#2) raw data. This option is activated by default and may be switched off eventually for a series of data based on processing success. “Composite tifs output” arranges input, output, detected spots, and detected signals (“dR/G0” threshold) in a multichannel tif file.

In cases where imaging has been performed with comparatively large z-steps the regions of interest may become rather flat. The registration and detection steps may benefit from rescaling the image stacks by a factor of 2 along z (“ups” option).

Under certain experimental conditions, it may happen that the initial green spot intensity (G0) is converted completely to red (R1) intensity impeding use of G1 for image registration and spot detection. Data with these constrains may be analysed using the “sum2” option in which a G0 and R1 and treated as relevant intensity channels for registration and detection.

Requirements

Matlab

Matlab “Image Processing Toolbox” license

ImageJ/Fiji

elastix

optional Matlab “Statistics and Machine Learning Toolbox” license

Licensing

This software is published under the GNU GENERAL PUBLIC LICENSE Version 3 license. For further information, read the LICENSE file.

Please note that SynapseLocator makes use of freely available software and code snippets from other sources which we hereby acknowledge. From Matlab’s File Exchange we used wekalab (https://de.mathworks.com/matlabcentral/fileexchange/58675-wekalab-bridging-weka-and-matlab), saveastiff (https://de.mathworks.com/matlabcentral/fileexchange/35684-multipage-tiff-stack), matlab_elastix (https://de.mathworks.com/matlabcentral/fileexchange/52982-matlab-elastix), KronProd (https://de.mathworks.com/matlabcentral/fileexchange/25969-efficient-object-oriented-kronecker-product-manipulation). For image stack loading we used ScanImage Tiff Reader (https://scanimage.gitlab.io/ScanImageTiffReaderDocs/index.html). Image handling tools were accessed through ImageJ/Fiji (https://fiji.sc/). For image deconvolution a plug-in was loaded from DeconvolutionLab2 (http://bigwww.epfl.ch/deconvolution/deconvolutionlab2/). For image segmentation we made use of Weka machine learning tools (https://www.cs.waikato.ac.nz/~ml/index.html) available as ImageJ plugin.