User Manual for SliceMap: An Algorithm for Automated Brain Region Annotation

Michaël Barbier and Winnok H. De Vos

1 Plugin installation

To use SliceMap one can either copy the jar-file *SliceMap_-1.0-SNAPSHOT-jar-with-dependencies.jar* into the plugins folder of the ImageJ/FIJI application (typically this folder is …/Fiji.app/plugins/), or just start it (by e.g. double clicking it), this way it will open its own ImageJ instance to run in.

2 Plugin user interface

The SliceMap plugin has a very simple user interface (see Fig. 1) asking the user to provide an "input folder" location containing the reference library of slices, a "sample folder" location, and the location of the "output folder" where the output will be generated in. One can filter the sample files in the sample folder by requiring that the sample file name contains a certain substring. If Force regeneration of the stack is ticked off, the reference stack will be regenerated for each sample, this is only useful if the sample folder contains also reference slices (they will be excluded then). To work the plugin needs to be provided with reference slices/ROI's in a specific format and folder structure, see the input section and Fig. 2 for specifications. There is advanced setup button which allows to adjust some of the default options used by the algorithm. The advanced options are described in Table 1.

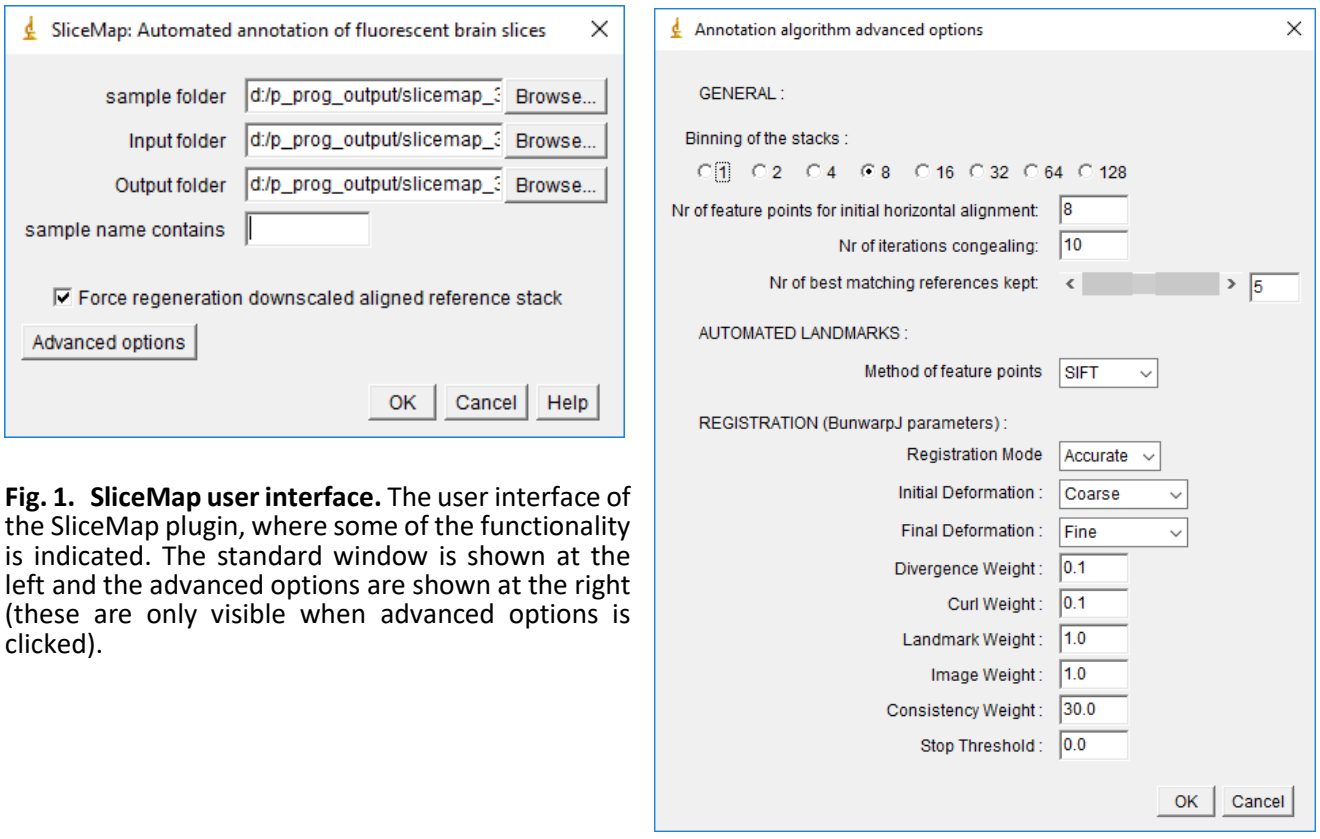

Fig. 1. SliceMap user interface. The user interface of the SliceMap plugin, where some of the functionality is indicated. The standard window is shown at the left and the advanced options are shown at the right (these are only visible when advanced options is clicked).

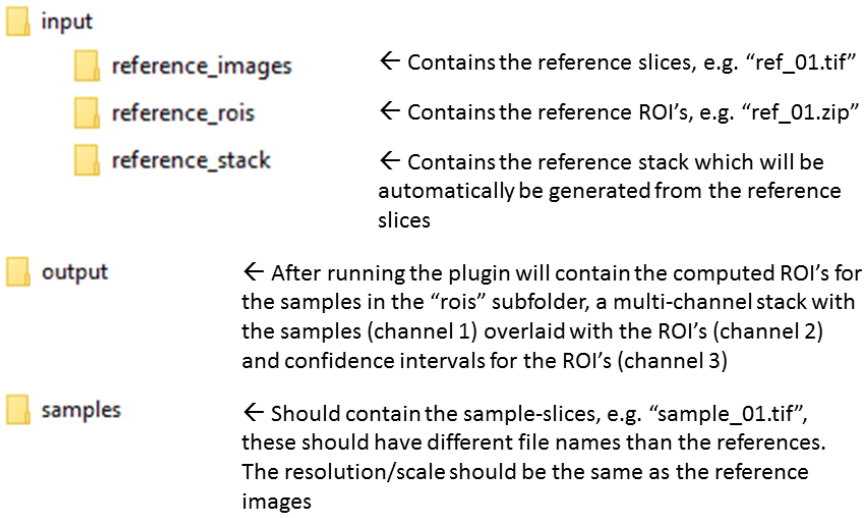

Fig. 2. Folder structure. The folder structure of SliceMap, with descriptions of the contents.

3 Input

To run the SliceMap plugin a reference library of reference slice images with corresponding regions is needed. The reference images should be stored in the user-defined input folder under the subfolder "reference_images", and the reference ROI's under the subfolder "reference_rois", and the file names should be corresponding (except for the file format extension). The format of the reference slices should be one that FIJI can handle (e.g. tif, png, jpeg, etc.) and the regions should be ImageJ ROI's saved as zip-files (originating e.g. from the RoiManager plugin in ImageJ). The regions and the reference slices should have the same scale/resolution.

4 Output

SliceMap generates multiple outputs:

- **The new ROI's** for every sample, both fitted to the rescaled small size of the downscaled stack used for the computation, as well as to the scale of the original sample image. The ROI's are ImageJ ROI's saved as zip-files (in the "output/roi" subfolder) which can be opened in the ImageJ RoiManager.
- A **multi-channel image stack shown in Fig. 3**. containing for each sample
    - The downscaled sample (channel 1),
    - Overlaid with the regions (channel 2),
    - Overlaid with the confidence intervals (channel 3)
- A **summary** log
    - The performance metrics for the references used are averaged, and the durations are the total time durations for the sample. The meanings of the parameters and performance metrics are listed in Table 2.
- Debug & Log files for every sample

- Log files located in subfolder debug: “registration_[sample_id].csv” files (tables) with performance metrics for the coarse (congealing) and elastic registration each reference of the reference subset used for the final annotation.
- Image stack: alignedReferencesAndSampleMirrors_[sample_id].tif located in subfolder debug/congealing, a 32-bit image stack containing the coarse aligned images [references + sample + 3 sample mirror] after congealing.
- Composite image stack: elasticRegistration_[sample_id].tif located in subfolder debug/elastic, ImageJ multi-channel composite image stack containing for each reference of the subset:
  - Channel 1 / red: the coarse aligned reference
  - channel 2 / green: the elastically registered reference
  - channel 3 / blue: the sample
  - channel 4 / gray: the original unregistered reference
- Annotation ROI’s (ImageJ ROI’s saved as zip-files) after the coarse alignment without performing elastic registration located in subfolder debug/congealing with prefixes:
  - _roi_: ROI’s on original scale but with width and height the same
  - _roiCrop_: ROI’s cropped to original sample size
  - _roiSample_: ROI’s masked by a segmentation of the total sample
  - _roiSmall_: ROI’s at the scale of the downscaled output stack

Parameter	Description	Default
General parameters:		
Binning of the stack	Binning factor of the references and samples, the annotation algorithm will use the binned images further on to speed up calculations, the output ROI's will be on the original size though.	8
Number of feature points for initial horizontal alignment	The number of automated landmarks which will be used during pre-aligning the stack horizontally, the feature points are Harris corner points	8
Number of best matching references kept	The elastic registration is only applied to a subset of references, the ones matching best with the sample. This is the number of references in the subset	5
Number of iterations for congealing	The number of iterations in the congealing algorithm used during the coarse alignment	10
Parameters for the automated landmarks detection algorithm used (Harris, SIFT):		
Method of feature points	Automated landmarks for BunwarpJ: the method used to generate the points: SIFT, Harris, …	8
Parameters for the BunwarpJ elastic registration plugin:		
param_bunwarpj_accuracy_mode	BunwarpJ: accuracy mode	Accurate
param_bunwarpj_min_scale_deformation	BunwarpJ: minimal scale deformation	Coarse
param_bunwarpj_max_scale_deformation	BunwarpJ: maximal scale deformation	Fine
param_bunwarpj_divWeight	BunwarpJ: divergence weight	0.1
param_bunwarpj_curlWeight	BunwarpJ: curl weight	0.1
param_bunwarpj_landmarkWeight	BunwarpJ: landmarks weight	1
param_bunwarpj_imageWeight	BunwarpJ: image weight	1
param_bunwarpj_consistencyWeight	BunwarpJ: consistency weight	30
param_bunwarpj_stopThreshold	BunwarpJ: threshold of similarity for the registration	0.01

Table 1. Description of the advanced options of the user interface. A short explanation is given for each parameter which can be set. For the BunwarpJ parameters a more detailed explanation can be found at http://imagej.net/BUnwarpJ.

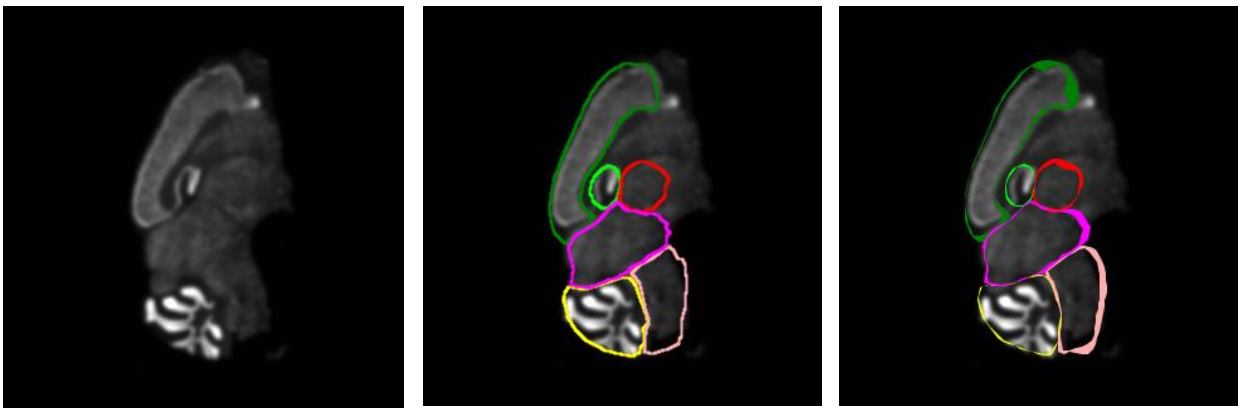

Fig. 3. Output image stack annotation overview. A multi-channel image stack containing the downscaled samples, the samples with the annotated regions, and the samples with the confidence intervals for each of the regions.

Column header	Description
param_sample_id	sample ID used
param_folder_sample	folder location of samples
param_prewarping_nLandmarks	Number of automated landmarks used during pre-aligning the stack horizontally
param_subset_nReferences	Number of references in the subset
param_binning	Binning factor of the references and samples
Parameters for the BunwarpJ elastic registration plugin:	
param_bunwarpj_accuracy_mode	BunwarpJ: accuracy mode
param_bunwarpj_img_sub-samp_fact	BunwarpJ_image subsampling factor
param_bunwarpj_min_scale_defor-mation	BunwarpJ: minimal scale deformation
param_bunwarpj_max_scale_defor-mation	BunwarpJ: maximal scale deformation
param_bunwarpj_divWeight	BunwarpJ: divergence weight
param_bunwarpj_curlWeight	BunwarpJ: curl weight
param_bunwarpj_landmarkWeight	BunwarpJ: landmarks weight
param_bunwarpj_imageWeight	BunwarpJ: image weight
param_bunwarpj_consistency-Weight	BunwarpJ: consistency weight
param_bunwarpj_stopThreshold	BunwarpJ: stopthreshold
Parameters for the automated landmarks detection algorithm used (Harris, SIFT):	
param_landmarks_harris_alpha	Landmarks for BunwarpJ: Harris corner points: alpha parameter
param_landmarks_harris_tH	Landmarks for BunwarpJ: Harris corner points: threshold
param_landmarks_harris_dmin	Landmarks for BunwarpJ: Harris corner points: minimal distance between point
param_landmarks_harris_nPoints	Landmarks for BunwarpJ: Harris corner points: number of points used for alignment
param_landmarks_harris_sigma	Landmarks for BunwarpJ: Harris corner points: sigma parameter
Performance metrics, sample and registered references from the subset are compared (mean of subset): Possible prefixes: **_cong_: compared with only coarse aligned references,** **_ori_ : compared with original unregistered references**	
error_(cong/ori)_pearson	Pearson correlation
error_(cong/ori)_pearson_max	Maximal Pearson correlation in scale space
error_(cong/ori)_scale_max	Scale of the maximal Pearson correlation in scale space
error_(cong/ori)_cc	Cross-correlation
error_(cong/ori)_ncc	Normalized cross-correlation
error_(cong/ori)_mse_roi	Mean square errors within ROI of the sample slice
error_(cong/ori)_mse	Mean square errors
error_(cong/ori)_rmse	RMSE Root Mean Square Errors
error_(cong/ori)_n_rmse	NRMSE Normalized Root Mean Square Errors
error_(cong/ori)_cv_rmse	Coefficient of Variation RMSE

Duration times of the various steps in the algorithm:	
time_run_sample	Total duration time for a single sample (in seconds)
time_refStack_generation	Duration of the reference stack regeneration (in seconds)
time_congealing_registration	Duration of the congealing alignment (in seconds)
time_congealing_annotation	Duration annotation of the region after congealing, i.e., coarse alignment (in seconds)
time_alignment_sorting	Duration of sorting the aligned reference images (in seconds)
time_elastic_registration	Duration elastic registration of the reference subset images (in seconds)
time_registration_error	Duration registration error metrics calculation (in seconds)
time_elastic_annotation	Duration elastic annotation (in seconds)
time_region_label_fusion	Duration label fusion procedure (in seconds)
time_confidence_interval	Duration calculation confidence interval (in seconds)
time_stamp_start	Timestamp of the timepoint when the annotation of the sample started
time_stamp_end	Timestamp of the timepoint when the annotation of the sample finished

Table 2. Description of the log output. A short explanation is given for each parameter which can be set. For the BunwarpJ parameters a more detailed explanation can be found at http://imagej.net/BUnwarpJ.

5 Example dataset

In the example folder one can find a number of reference images in the correct folder structure.