
User Manual

SliceMap: An Algorithm for Automated Brain Region Annotation

Barbier M¹, Bottelbergs A², Nuydens R², Ebneith A² and De Vos WH^{1,3,*}

¹Laboratory of Cell Biology & Histology, Department of Veterinary Sciences, University of Antwerp, Universiteitsplein 1, 2610 Wilrijk, Belgium, ²Department of Neuroscience, Janssen Research and Development, Turnhoutseweg 30, 2340 Beerse, Belgium, ³Department of Molecular Biotechnology, Ghent University, Coupure Links 653, 9000 Ghent, Belgium.

*To whom correspondence should be addressed.

1 Introduction

Whole brain tissue slices are commonly used in neurobiological research for analyzing pathological features in an anatomically defined manner. However, since many pathologies arise in and spread from specific regions of the brain, region-based analysis is mandatory. SliceMap is a FIJI (ImageJ) plugin for automated brain region annotation of fluorescent brain slices. The plugin uses a reference library of pre-annotated brain slices (the brain region templates) to annotate brain regions of unknown samples. To perform the region annotation, SliceMap registers the reference slices to the sample slice and uses the resulting image transformations to morph the template regions towards the anatomical brain regions of the sample. The resulting brain regions are saved as FIJI/ImageJ ROI's (Regions Of Interest) as a single ZIP-file for each sample slice.

In brief the used method can be described as follows:

- First, a group-wise registration of all reference slices together with one of the samples (using the (slightly adapted) congealing algorithm (Learned-Miller, 2006)) is performed. At this point all the reference slices are roughly matching the sample slice (sample slice can be transformed as well)
- Next, all these roughly registered references are compared with the sample by calculating the Pearson correlation. A subset of the N references which match best (N can be given in the advanced options) is kept.
- Then, the reference slices from the above subset are elastically registered to the sample, for this we use using the BunwarpJ plugin (Arganda-Carreras, et al., 2006; Sorzano, et al., 2005) of FIJI (ImageJ). This BunwarpJ allows registration enhanced by landmarks, which we use. For each of these references there is now a transformation between the original reference and the sample.
- The regions of the references are then transformed (using the latter transformations) to the sample. This gives for every region as many candidate regions as there are references in the subset.
- The transformed regions (region candidates) are then fused to a single region by interpolation.

2 Quickstart

2.1 Installation

To install SliceMap, the following steps can be followed:

1. Download & install FIJI (ImageJ) which is an open-source image processing software environment that can be found at www.fiji.sc.
2. Download SliceMap itself from: <https://github.com/mbarbie1/SliceMap>, as SliceMap_-1.0-SNAPSHOT-jar-with-dependencies.jar
3. To install the SliceMap plugin, 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/). The plugin can then be called from Plugins > SliceMap > SliceMap.

OR, as an alternative installation option one can use SliceMap directly, without the installation of FIJI, just by starting (by e.g. double clicking it). This way, it will open its own ImageJ instance to run in, and a choice menu pops up which lets you choose the tool.

The example dataset with 60 downsampled brain slices and brain region annotations can be downloaded from: <https://github.com/mbarbie1/SliceMap/tree/master/dataset>

2.2 Running SliceMap

The SliceMap plugin has a very simple user interface (see Fig. 1) asking the user to provide:

- a **Sample folder** location containing the sample slices which need to be annotated,
- an **Input folder** location containing the reference library of slices (see Fig. 2),
- an **Output folder** location where the output will be generated in.
- One can filter the sample files in the sample folder by requiring that the **sample name contains** a certain substring.
- checked If **Force regeneration downsampled aligned reference stack** is checked, a reference stack will be regenerated for each sample (excluding the sample image). This is only useful for leave-one-out cross-validation. By default, this option is unchecked and one single reference stack will be made for the entire sample image set.
- **Downscale factor of the slices:** The downscaling factor for reference and sample images serves to speed up the SliceMap procedure, the value should give rise to downsampled images with widths/heights of 200 → 1000 pixels (for the example references a reasonable value is 8).
- The **advanced options** button allows to adjust some of the default parameters used by the algorithm. The advanced options are briefly described in Table 1, together with the default value. Here we give some insight on how to pick good parameter values for some of them:
 - GENERAL
 - **The number of feature points for the 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. The value depends on how many texture is available in your samples and whether it is always consistent. For example, in our example dataset one can see that there are two prominent feature-rich regions (the hippocampus and the cerebellum), since the both regions contain multiple prominent “corner”-points, it is useful to have more than 1 or 2 detected points, while too many (hundreds) detected corner-points would allow less prominent corners (noise) from other regions to be included, therefore the default value is in between (8).

- **Number of iterations congealing:** The default number of iterations is taken fixed and quite low since the elastic registration is still executed afterwards.
- AUTOMATED LANDMARKS: There are different methods to obtain automated landmarks, but the underlying is the following: The methods try to match specific points of interest from the sample image onto the reference image. To do so, the points need to fulfil two conditions: (1) the features calculated for the individual points should be roughly similar (*i.e.*, matching in feature space) and (2) there should be a geometrical transformation that roughly maps the combined points onto each other in 2D space. The parameters of this geometrical transformation are optimized so as to maximize the number of matching points. The geometrical transformation itself is mostly one of the following: only translations, rigid transformations, similarity transformations, affine transformations, In this case, we fixed it to a similarity transformation (which allows translation + rotation + scaling). The methods describe the type of feature points which are used for the points of interest.
- REGISTRATION (Bunwarpl parameters): There is an excellent description of these parameters on the following website: <http://imagej.net/BUnwarpl>.

In order to work properly, the plugin needs to be provided with reference slices/ROI's in a specific format (slice images in PNG or TIF format and the ROI's as ImageJ ROI's saved as ZIP-file) and folder structure, see the input section and Fig. 2 for specifications.

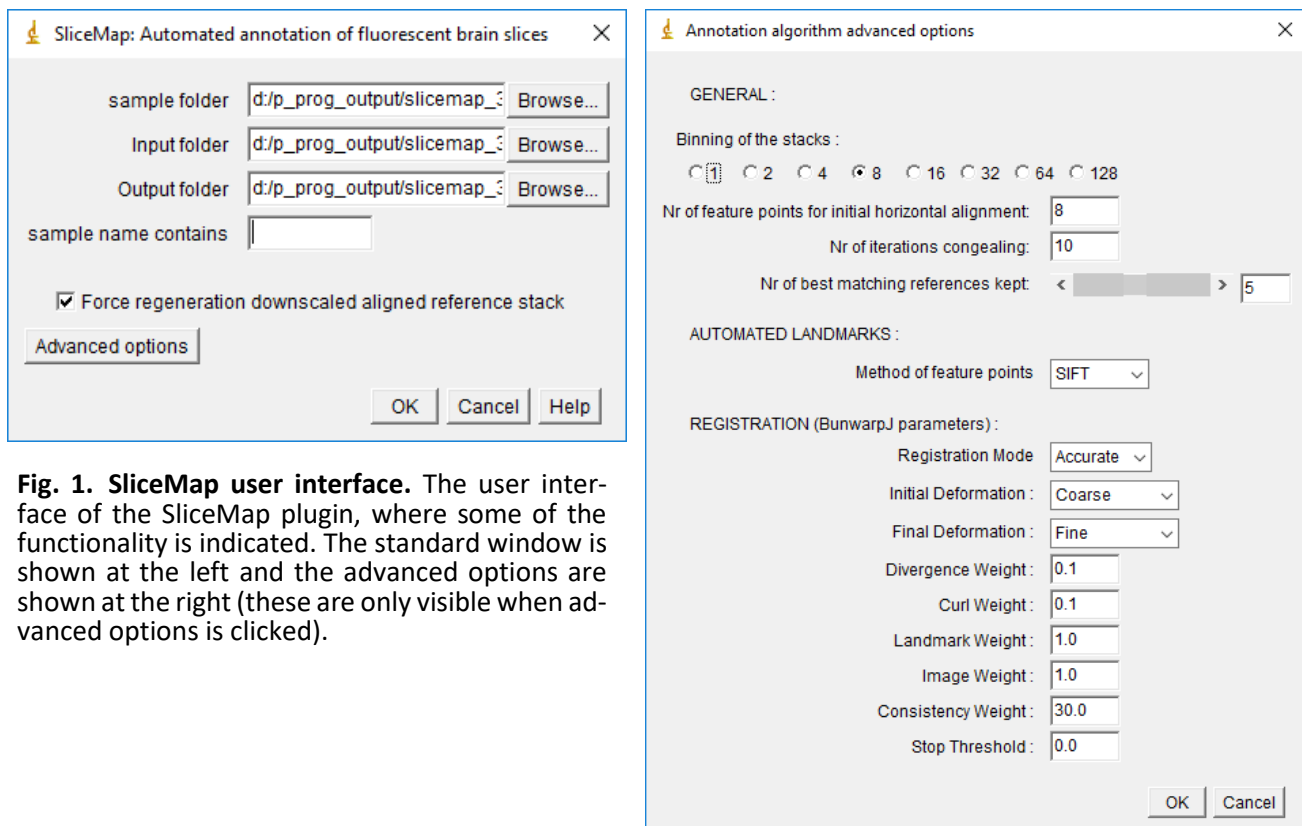


Fig. 1. SliceMap user interface. The user interface of the SliceMap plugin, where some of the functionality is indicated. The standard window is shown at the left and the advanced options are shown at the right (these are only visible when advanced options is clicked).

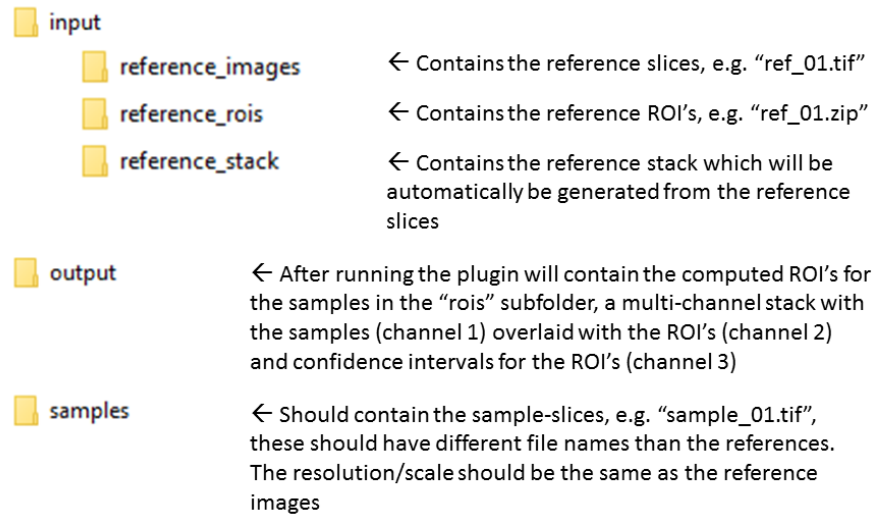


Fig. 2. Folder structure. The folder structure of SliceMap, with descriptions of the contents.

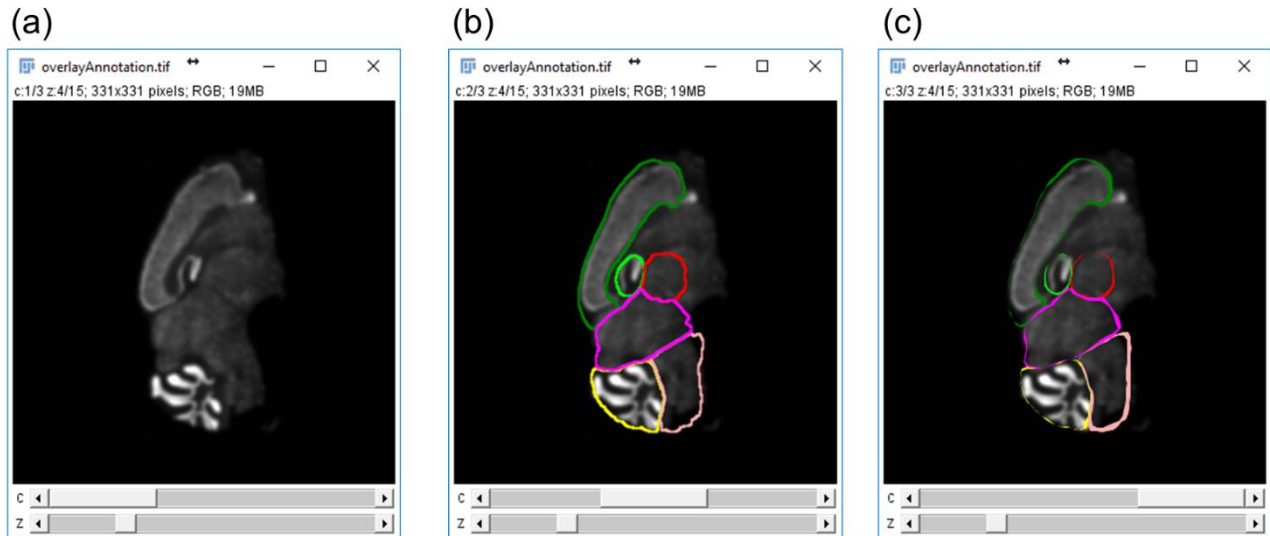


Fig. 3. Output image stack annotation overview. A multi-channel image stack containing (a) the downsampled samples, (b) the samples with the annotated regions, and (c) the samples with the confidence intervals for each of the regions.

2.3 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 similar (except for the file format extension, e.g. for slice image "ref-01.tif" the corresponding ROI's should be defined in "ref-01.zip"). The format of the reference slices should be either TIF or PNG 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.

2.4 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 hyperstack shown in Fig. 3.** containing for each sample
 - The downscaled sample (channel 1),
 - the median color-coded regions (channel 2),
 - color-coded confidence intervals (median \pm stdev) (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 meaning 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

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 Bunwarpl parameters a more detailed explanation can be found at <http://imagej.net/BUnwarpl>.

Parameter	Description	Default
General parameters:		
Binning of the (congealing) stack	“Extra” binning factor of the references and samples during the congealing step to further speed up this registration step.	2
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 Bunwarpl: the method used to generate the points: SIFT, Harris, ...	8
Parameters for the Bunwarpl elastic registration plugin:		
Registration mode	Bunwarpl: accuracy mode	Accurate
Initial Deformation	Bunwarpl: minimal scale deformation	Coarse
Final Deformation	Bunwarpl: maximal scale deformation	Fine
Divergence Weight	Bunwarpl: divergence weight	0.1
Curl Weight	Bunwarpl: curl weight	0.1
Landmark Weight	Bunwarpl: landmarks weight	1
Image Weight	Bunwarpl: image weight	1
Consistency Weight	Bunwarpl: consistency weight	30
Stop Threshold	Bunwarpl: threshold of similarity for the registration	0.01

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

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 Bunwarpl elastic registration plugin:	
param_bunwarpl_accuracy_mode	Bunwarpl: accuracy mode
param_bunwarpl_img_sub-samp_fact	Bunwarpl: image subsampling factor
param_bunwarpl_min_scale_deformation	Bunwarpl: minimal scale deformation
param_bunwarpl_max_scale_deformation	Bunwarpl: maximal scale deformation
param_bunwarpl_divWeight	Bunwarpl: divergence weight
param_bunwarpl_curlWeight	Bunwarpl: curl weight
param_bunwarpl_landmarkWeight	Bunwarpl: landmarks weight
param_bunwarpl_imageWeight	Bunwarpl: image weight
param_bunwarpl_consistency-Weight	Bunwarpl: consistency weight
param_bunwarpl_stopThreshold	Bunwarpl: stopthreshold
Parameters for the automated landmarks detection algorithm used (Harris, SIFT):	
param_landmarks_harris_alpha	Landmarks for Bunwarpl: Harris corner points: alpha parameter
param_landmarks_harris_th	Landmarks for Bunwarpl: Harris corner points: threshold
param_landmarks_harris_dmin	Landmarks for Bunwarpl: Harris corner points: minimal distance between point
param_landmarks_harris_nPoints	Landmarks for Bunwarpl: Harris corner points: number of points used for alignment
param_landmarks_harris_sigma	Landmarks for Bunwarpl: 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

3 Example dataset

An example dataset is provided to test the SliceMap workflow with 60 downscaled brain slices and brain region annotations can be found in: <https://github.com/mbarbie1/SliceMap/tree/master/dataset> One can move a few of the reference slices in a separate folder (e.g. samples) and use these as sample slices (when using SliceMap, point the **sample folder** location to that folder).

The brain slice images:

- Origin: P301S mouse brain, sagittal slices
- Label: stained with NeuN (neuronal nuclear marker)
- Images: Zeiss Axioscan, 20x magnification, 8 times binned post-acquisition

Brain region annotations:

- 6 regions are annotated:
 - BS – pons and medulla
 - CB – cerebellum
 - MB – mid-brain
 - CX – cortex
 - TH – thalamus
 - HP – hippocampus
- Annotated using the FIJI/ImageJ RoiManager tool

4 Creating and Correcting region annotations: Manual Annotation & Curation plugin

There are two times when manual interaction can be requested:

- SliceMap needs a reference library of pre-annotated images, with the regions saved as ImageJ ROI's. One way to create such reference images is by manually delineating the regions
- When SliceMap makes a mistake in the automated annotation of the regions, it is sometimes necessary to correct the faulty regions.

For these goals, we created a tool (based on the ROI Manager tool of ImageJ/FIJI) to assist the user in this process.

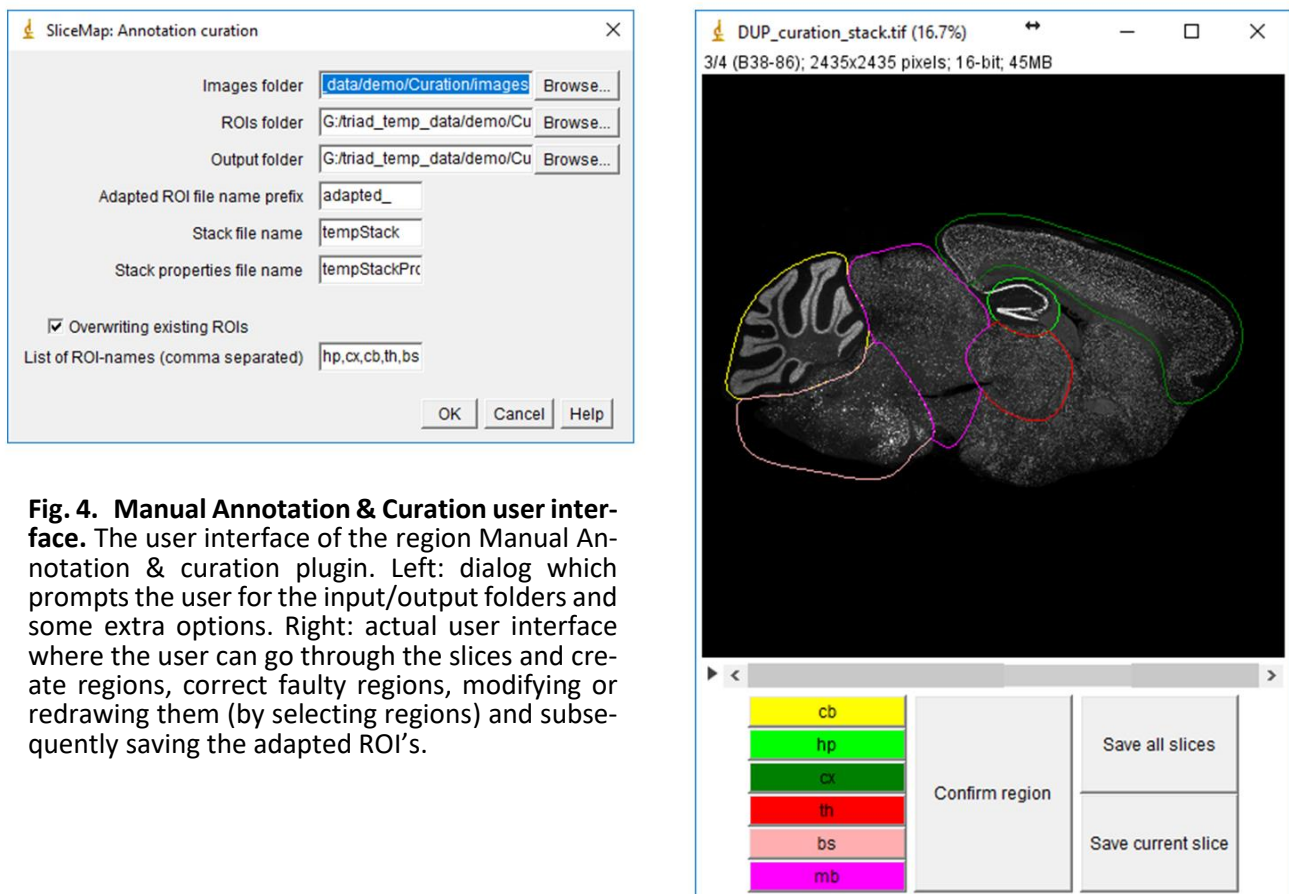


Fig. 4. Manual Annotation & Curation user interface. The user interface of the region Manual Annotation & curation plugin. Left: dialog which prompts the user for the input/output folders and some extra options. Right: actual user interface where the user can go through the slices and create regions, correct faulty regions, modifying or redrawing them (by selecting regions) and subsequently saving the adapted ROI's.

When started the Annotation curation plugin shows a dialog prompting the user to provide (see Fig. 4):

- a **Samples folder** location containing the sample slices which were annotated,
- a **ROI's folder** location containing the output ROI's from the annotation procedure,
- an **Output folder** location where the output will be generated in.
- **Adapted ROI file name prefix:** the output ROI's (saved as ImageJ ROI ZIP-files) can be given a prefix, as to distinguish from the original ROI files.

- When **Overwriting existing ROIs** is checked one can give the adapted ROI's the same names as the old ROI's
- The **List of ROI-names (comma separated)** field defines the regions which you want to correct (the names should be exactly the same as in the original ROI's).

When all parameters are filled in and pressing OK, an image stack window similar to the one shown in the right panel of Fig. 4 is shown. The image stack contains all the images in the **Samples folder** and shows the original ROI's found in the **ROI's folder**. To correct a ROI (region outline) one should perform the following steps:

- Go to the slice containing the region you want to adapt.
- Select the region by pressing the colored buttons with the region names located on the bottom of the window. When a ROI is selected its name will be shown in bold on the button, also, the region outline itself will become an ImageJ "selection" (and the outline will appear thicker)
- Now you can adapt the region as you would adapt an [ImageJ selection \(freehand selection\)](#). One way is to use the brush selection:
 - Hold *Shift* while selecting with the mouse a ROI which will be added to the original one.
 - Hold *Alt* while selecting with the mouse a ROI which will be subtracted from the original one.
- Once satisfied with the region, use the **Confirm region** button

When all regions are corrected save the new regions as ImageJ ROI ZIP-files by pressing the buttons **Save all slices** or **Save current slice**.

References

- Arganda-Carreras, I., *et al.* Consistent and elastic registration of histological sections using vector-spline regularization. *Lect Notes Comput Sc* 2006;4241:85-95.
- Learned-Miller, E.G. Data driven image models through continuous joint alignment. *IEEE Trans Pattern Anal Mach Intell* 2006;28(2):236-250.
- Sorzano, C.O.S., Thevenaz, P. and Unser, M. Elastic registration of biological images using vector-spline regularization. *Ieee T Bio-Med Eng* 2005;52(4):652-663.