Atlas-based imaging data analysis tool for quantitative mouse brain histology

AIDAhisto

Department of Neurology
University Hospital Cologne

v1.2
September 2022

- **Current contributors:** Aref Kalantari, Markus Aswendt
- **Former contributors:** Niklas Pallast, Leon Scharwächter, Annika Vohn

1 Introduction

AIDAhisto provides accurate and fast results for cell nuclei as well as immunohistochemical stainings of neurons, astrocytes and immune cells in the mouse brain with respect to the associated regions of the Allen Brain Reference Atlas (ARA). The transformation between the atlas as a source and the brain slice as a target image was conducted by a landmark based registration.

2 Download & Install

1. Download & Install **Python 3.6** or higher using Anaconda and enter the following command to install all necessary Python packages
   `pip install numpy==1.14.3 argparse==1.4.0 scipy==1.1.0 matplotlib==3.0.3`

2. The code is also implemented in **Matlab** (tested with version R2018a). Note: Matlab processing is faster and visualization better compared to Python; otherwise there are no differences.

3. Download the zip-File `AIDAhisto-master` using the following Link. After opening the web page, the zip-file can be downloaded.

There are two main folders: Matlab and Python, which contain the related code.
Matlab: `AIDAhisto.m` and `run_example.m`
Python: `AIDAhisto.py` and `run_example.py`

3 Atlas Transformation

3.1 Brain

In this example we conduct the registration process using a simple landmark-based registration with ImageJ as described below. For more information follow the link: ImageJ Landmark Correspondences. Other tools allow even stronger transformations and thus facilitate the choice of the correct layer in Atlas. These tools are listed below, but are not described in detail BigStitcher

and QuickNII.
The transformed atlas slice (`wholebrain_atlas.tif`) is already included in the folder test images. In order to perform the atlas transformation, conduct the following steps:

1. Load ARA with highest resolution ($10\mu m$) from the folder `.../AIDAhisto-master/ARA/annotation_10.nii.gz` and open the image with ImageJ

2. At first a black picture appears because the color space is not scaled correctly. To adjust the color space follow these steps:

   - Select slide 855
   - Open the B&C Tool with Image → Adjust → Brightness/Color
   - Click Auto

3. Load microscopy image in ImageJ from the folder `.../AIDAhisto-master/testImages/wholebrain_slice.tif`

4. Select the matching slice in the ARA and make a substack with ImageJ: Image → Stacks → Tools → Make Substack

5. Choose the corresponding slice number of the ARA and save the substack.

6. Choose "Multi-point" in ImageJ and place 20-40 landmarks in the microscopy image and the corresponding positions in the ARA (Figure 1). Note: that step requires some experience in mouse brain anatomy to match in atlas landmarks with the microscopy - especially if like in this example the mouse brain tissue is strongly deformed due to a local brain lesion.

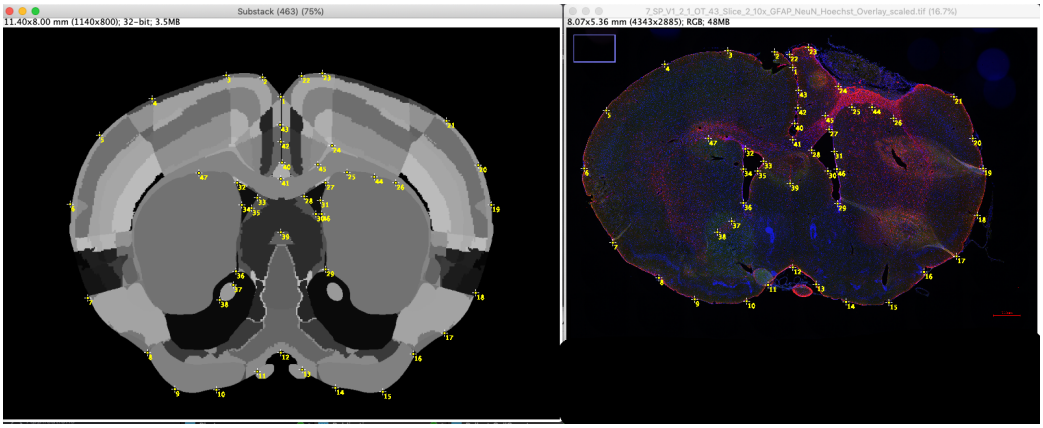


Figure 1: 30-50 landmarks in the microscopy image and the corresponding positions in the ARA

7. Landmark registration with ImageJ: Plugins → Transform → Landmark Correspondences

8. Choose the substack of the ARA as "source image" and the microscopy image as template image in the transform menu

9. Choose: Transformation method: Moving Least Squares (non-linear), Transformation class: Affine, No interpolation (Figure 2).

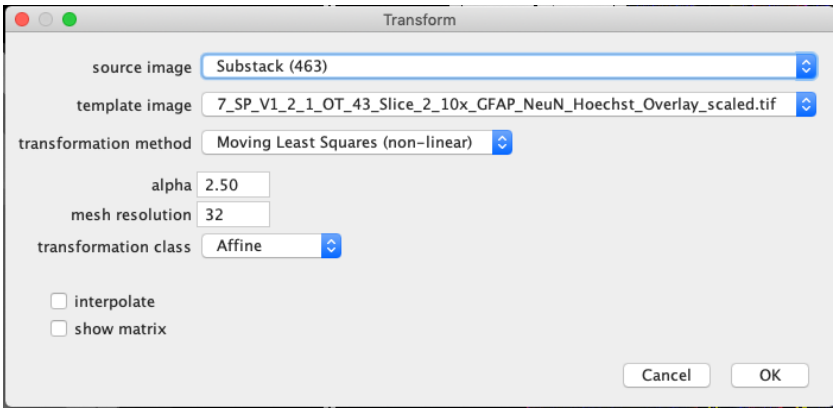


Figure 2: Choose the shown transformation parameters to register the atlas with the microscopy image

3.2 Spinal cord

For the landmark registration with our custom digital version of the Allen Mouse Spinal Cord Reference Atlas, the same workflow as for brain slices is applied. Relevant files:

1. Chose the matching section of the spinal cord atlas and load the microscopy image and the ARA intro ImageJ.

2. Choose "Multi-point" in ImageJ and place 20-40 landmarks in the microscopy image and the corresponding positions in the ARA (Figure 3). Note: that step requires some experience if the mouse spinal cord tissue is strongly deformed.

3. Landmark registration with ImageJ: Plugins → Transform → Landmark Correspondences

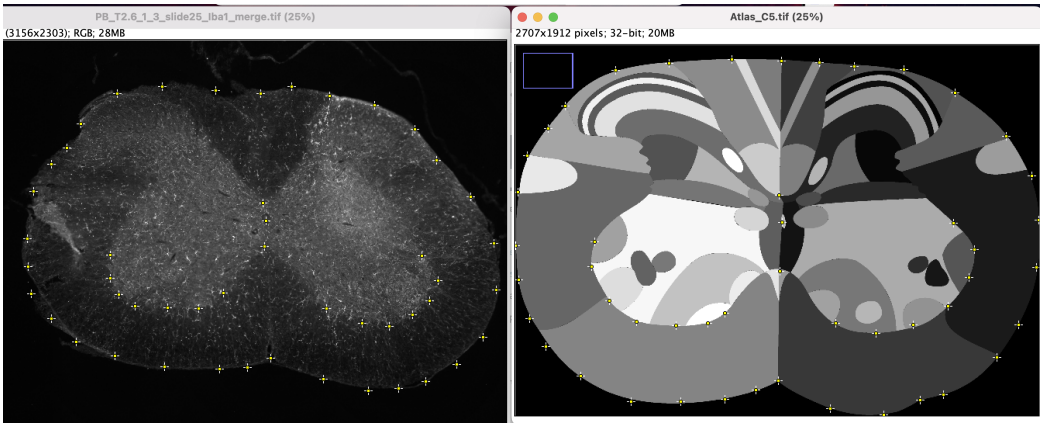


Figure 3: 30-50 landmarks in the microscopy image and the corresponding positions in the ARA

4. Choose the substack of the ARA as "source image" and the microscopy image as template image in the transform menu

5. Choose: Transformation method: Moving Least Squares (non-linear), Transformation class: Affine, No interpolation (Figure 4).

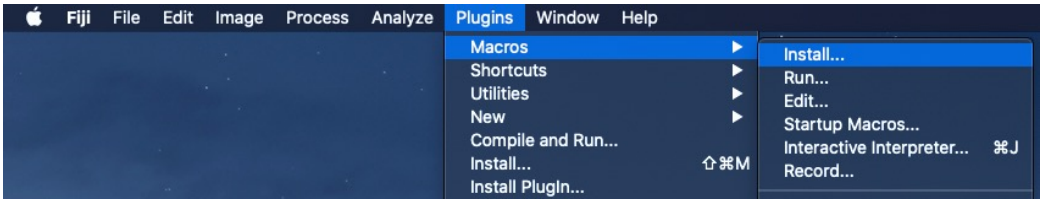


Figure 5: Installation

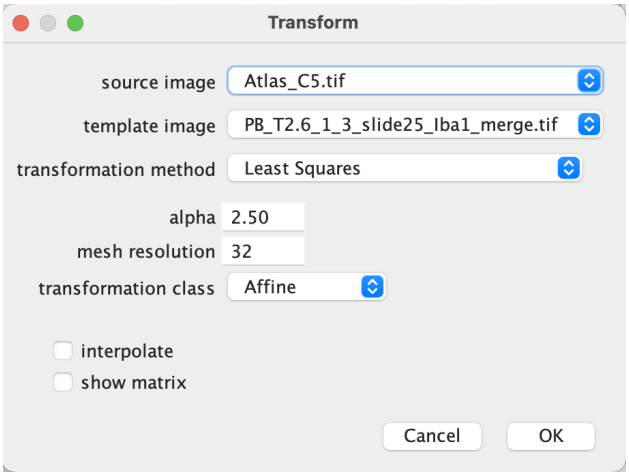


Figure 4: Choose the shown transformation parameters to register the atlas with the microscopy image

3.3 Export and import landmarks

To save and reuse the landmarks of the microscopy image and the ARA file a certain plugin can be used. This can be useful if adjustments need to be done later on.

1. Install the Macro as it is displayed in the image 5.

2. Choose the file **exportmultipointset.ijm.txt** if you want to export the landmarks from the image. Chose the file **importmultipointset.ijm.txt** if you want to reload previous landmarks.

3. After the installation of the import or export file landmarks can be saved or imported via Plugin, Macros. For exporting landmarks chose a file under which name the landmarks are saved as .csv file. The

easiest way is to save them under the same name of the image that the landmarks where placed on. Save landmarks for the microscopy image and Atlas image individually!

4. Importing landmarks follows the same path. Landmarks are imported by selecting the correct .csv file.

4 Count Cells

This is a step-by-step procedure to count all cells visible in the red channel of a whole brain slice as shown in Figure 1.

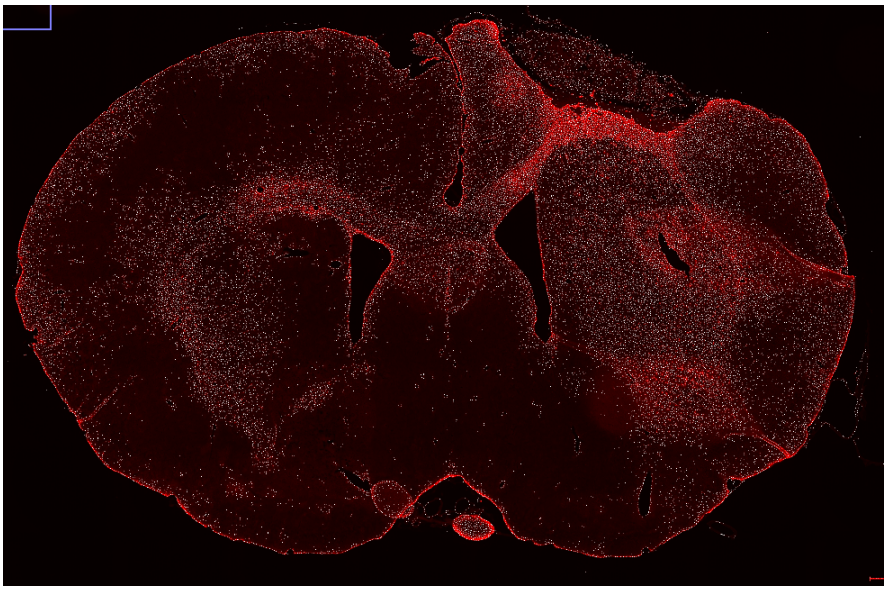


Figure 6: Example microscopy image (red channel): white spots indicate results of cell counting.

1. Count all cells in the given image with Matlab: Here, you have to open Matlab and set `.../AIDAhisto-master/Matlab` as "Current Folder" (Figure 7A). Type the same command in the "Command Window" (see Figure 7B) like shown in figure 8.

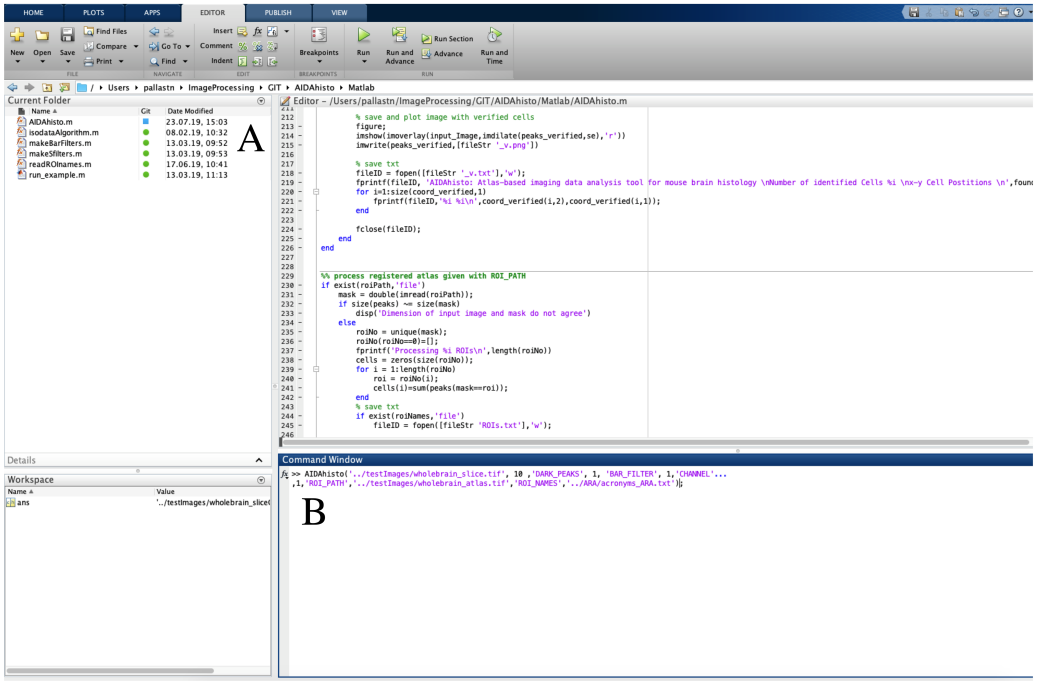


Figure 7: Set the `.../AIDAhisto-master/Matlab` as "Current Folder" (A) and type the command of figure 8 in "Command Window" (B)

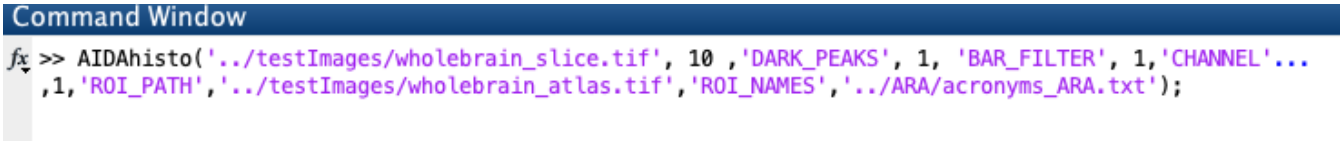


Figure 8: Type the shown command in the "Command Window"

Developer Tip:

In practice the input addresses of the function `AIDAhisto()` are usually very long, therefore using the function in the command window can get very tedious and a possible source of errors. Alternatively you can use a script called `AIDAhisto_easy.m` also available in the same `Matlab` folder. As shown in Figure 9 in the section of `%% User Input` you can set the addresses in a cleaner and more flexible way and by clicking the Run button the function will be executed. Note that for the optional argument `REF_PATH` or other parameters in the `%% User Input` section

you can either put in the corresponding address or replace it by an empty string as can be seen in Figure 9. Inforamtion about the different input options is available in the script and also observable in 9.

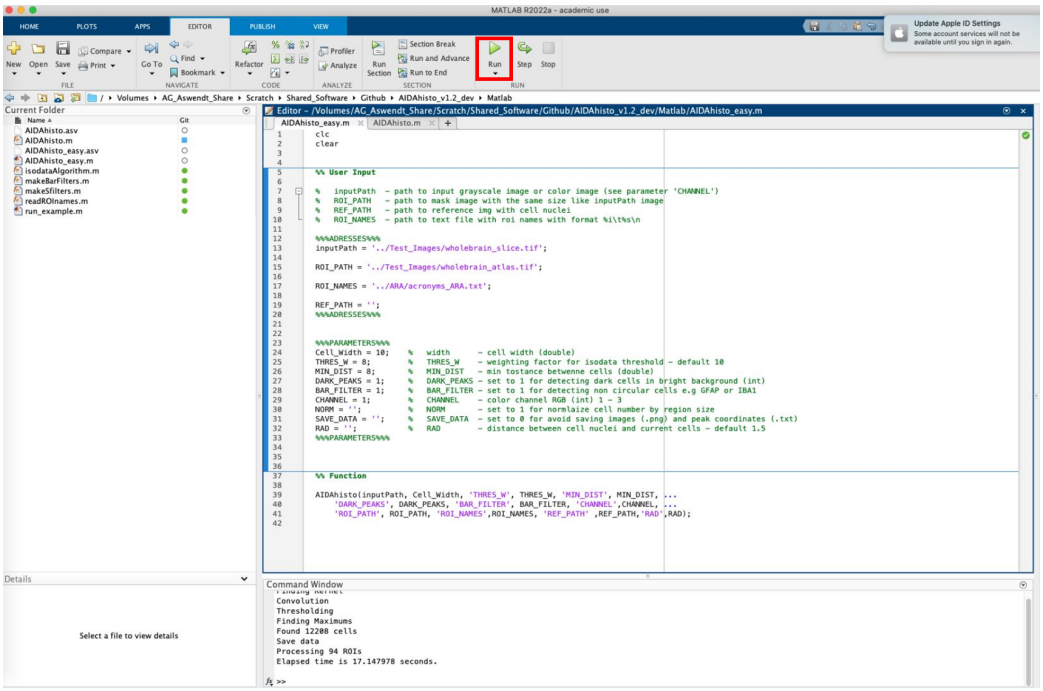


Figure 9: Set the `.../AIDAhisto-master/Matlab` as "Current Folder". Work with the addresses in a cleaner and more flexible way. By clicking the Run button the function will be executed. Note that for the optional argument `REF_PATH` or other parameters in the `%% User Input` section you can either put in the corresponding address or replace it by an empty string. Explanations of each input is written in the script and can be seen in the image.

2. Count all cells in the given image with Python: Open the `Command Prompt` if you use a Windows System or `Terminal` if you use a Macintosh System. Set `AIDAhisto-master/Python` as your current folder using the command `cd` (see figure 10)

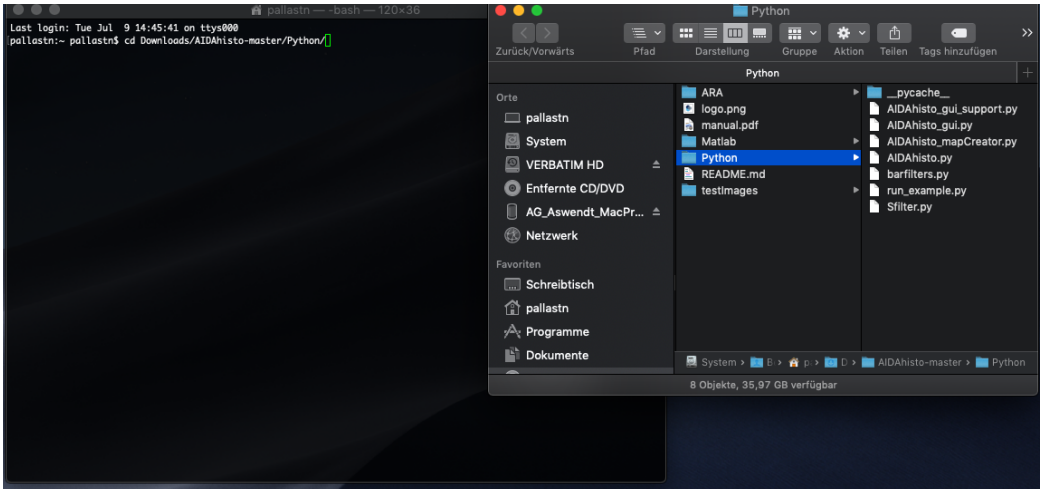


Figure 10: Set the `Python` folder as your current folder using the command `cd`

3. Type (not copy) the following command in the command window.
   ```
   python AIDAhisto.py ../testImages/wholebrain_slice.tif
   -f -w 9 -d -c 0 -a ../testImages/wholebrain_atlas.tif
   -l ../ARA/acronyms_ARA.txt
   ```

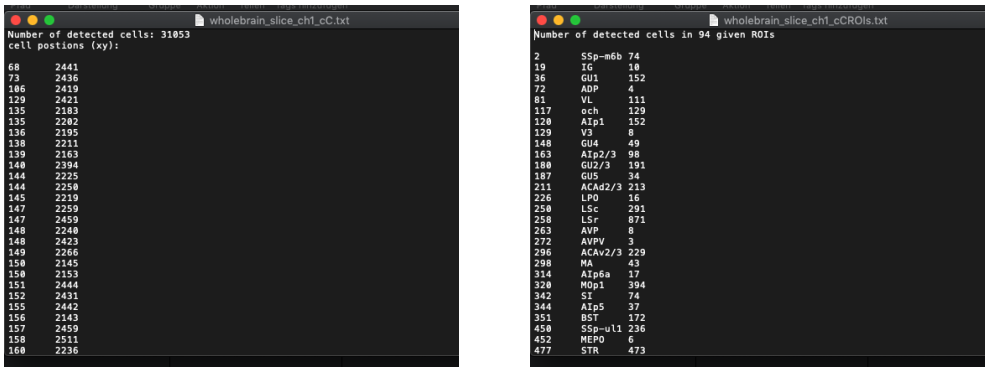
4. The results are stored in the folder `AIDAhisto-master/testImages` and can be easily overlaid in ImageJ like described here. Note: the counted cells will be represented by a single white pixel, which might be difficult to see in the overlay. Try to use the dilate function in ImageJ on a binarized version of the counting result.

4.1 Spinal cord

Cell counting in Matlab for spinal cord images follows the same rules as whole brain slices. The command needs to be typed in the "Command Window" (figure 8B). The necessary input as well as the adjustable parameters are the same for both spinal cord and brain and require the microscopy image, the transformed atlas, the list with acronyms and (if wanted) a reference file with the position of cell nuclei (see figure 11).

```
>>AIDAhisto('.../testfile/spinal_cord.tif', 10, 'THRES_W', 8, 'CHANNEL', 1,
'MIN_DIST', 8, 'BAR_FILTER', 1, 'DARK_PEAKS', 1, 'SAVE_DATA', 1, 'ROI_PATH',
'.../testfile/spinal_cord_Atlas', 'ROI_NAMES', '...ARA/acronyms_ARA_SC.txt`,
'REF_PATH', `.../testfile/spinal_cord_DAPI_CH_1.png')
```

Figure 11: Command for spinal cord cell counting.



(a) wholebrain_slice_ch1_cC.txt (b) wholebrain_slice_ch1_cCROIs.txt

Figure 12: a) The first text file contains the number of all identified cells with x-y-positions in the input image. b) The second text file contains acronyms and pixel values of the given ARA with related cell number.

5 Results

After successful completion of the previous steps you will end up with two text files (see Fig. 12) and an image file:

- **wholebrain_slice_ch1_cC.tif:** A binary image which can be overlaid with the related red channel of the original image to visualise the result.
- **wholebrain_slice_ch1_cC.txt:** A text file which provides the number of identified cells and the x-y-coordinates of each cell position.
- **wholebrain_slice_ch1_cCROIs.txt** A text file with three columns. In the first column contains the pixel value of each region in the given atlas. The second column contains the acronyms of ARA. The third column contains the number of identified cells for each region.

6 Percentage of affected regions in Python

In Python, you can additionally examine how much regions are affected by a lesion using `AIDAhisto_maskSize.py`. This function returns the percentage of the affected regions within one brain slice both regarding the region size itself and the size of the whole brain slice due to a given lesion mask. The brain slice should be a transformed grey scale or color image (no microscopy image) where every grey / color value represents a unique region. The mask should consist of the same color range. Please specify both paths as string arguments (see example below). Large input images can be resized using the optional parameter -r behind the indication of the paths, where the value determines the new width in pixels. The output are two lists of atlas acronyms and the percentage affected area. The acronym numbers depend on the atlas used and can be found in the `ARA` folder.
Example: `python AIDAhisto_maskSize.py "path/to/brainSliceImage" "path/to/brainMaskImage" -r 3000`

Remember that resizing very large images can take several minutes.

7 Usage of AIDAhisto GUI with Python

In Python, a generic user interface has been implemented. To start the GUI, set `/Python` as your current folder in the command window and open the GUI (see 13):
`python AIDAhisto_gui.py.`

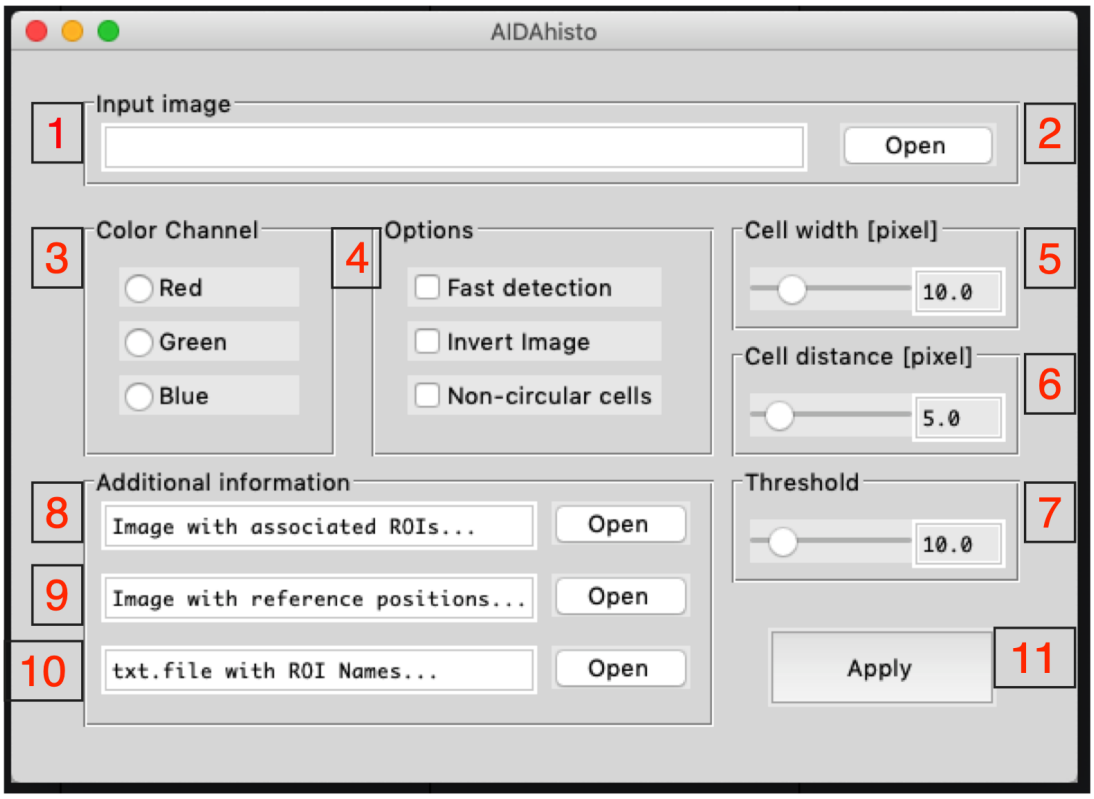


Figure 13: Description of the user interface to process with all provided input parameters.

You can only choose one image file and adapt the cell width (5) to run AIDAhisto, but we also provide some parameters to optimize the output and to meet all individual requirements of manifold investigations. Therefore, the following is a detailed explanation of the numbering in Figure 13:

1. Path of the input file `*.jpg/png/tiff`
2. Press button to open file dialog and select the image
3. Choose the color channel. If only one channel is present, the first channel will always be examined.
4. - For very large images, the process can be accelerated by choosing *Fast detection*

- If the cells appear dark and the background bright bright, the image should be inverted prior to analysis using *Invert image*.
- If the cells are not round and have a different shape choose *Non-circular cells*.

5. Choose the cell size in pixels.
6. Choose the minimum cell distance in pixels.
7. Adapt the automatically calculated threshold value if necessary.
8. Choose the transformed atlas image or any other superimposed image containing regions of interest.
9. Choose the result of the nuclei detection, which will be used as a reference to improve the counting of non-circular cells.
10. If certain names should be noted in the output file instead of the pixel value, the name of the respective pixel value can be entered here as a text file.

8 Allen Mouse Brain Atlas Database

In order to simplify the search for specific region numbers (e.g. 582 Caudoputamen) and list the related parental and child regions, we have summarized the ARA data available from (©2017 Allen Institute for Brain Science. Allen Mouse Brain Atlas (ccf3)) in an online database software, which we are using also for managing research data https://doi.org/10.1093/database/bay124 and https://github.com/maswendt/AIDAdb. To use the database, create an account here https://ninoxdb.de/de/templates/research and import via the "Import archive" function this file.