

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

## AIDAhisto

Niklas Pallast and Leon Scharwächter  
Department of Neurology  
University Hospital Cologne

v1.1  
January 2021

## 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 (Figure 1).

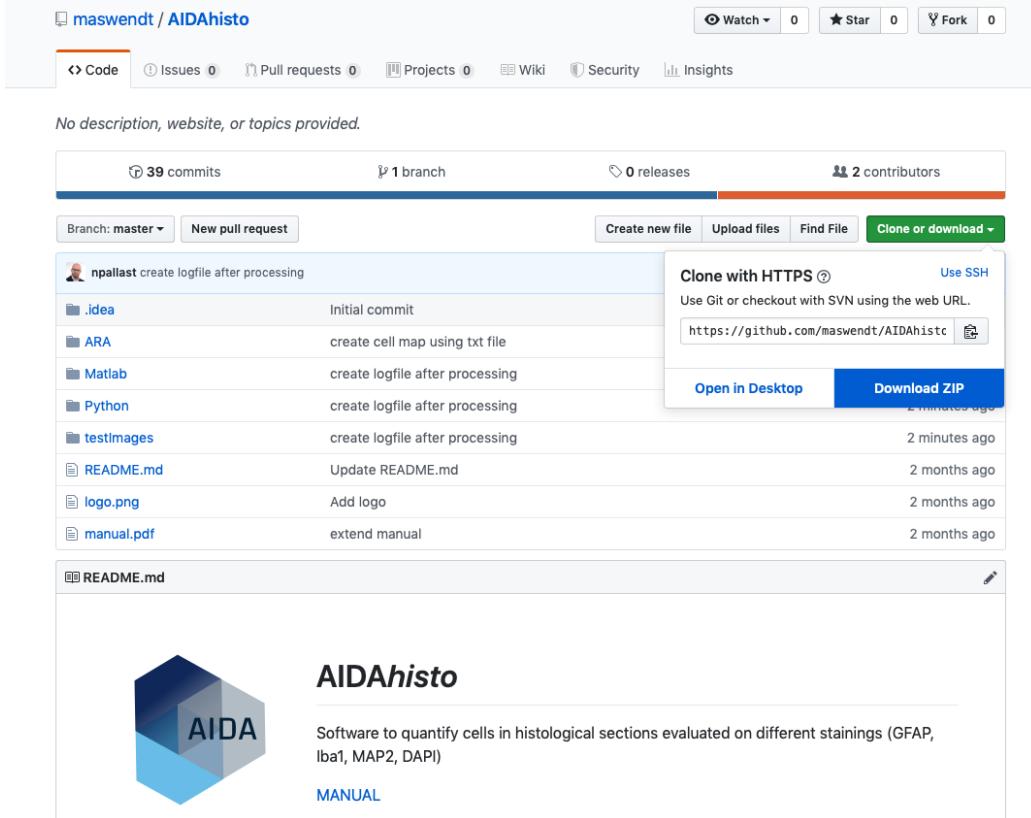


Figure 1: Click on Clone or Download → Download ZIP to download the zip-File.

In the following, the content of each folder is explained exactly:

**ARA** Contains the atlas in 10um and 50um with parental and original regions. Each atlas is also in a split version to distinguish the cell count between the left and right hemisphere. Furthermore, the acronyms are included that correspond to those of the Allen Brain Institute

**Matlab** Contains the Matlab code with the appropriate program call `AIDAhisto.m` and `run_example.m`

**Python** Contains the Python code with the appropriate program call `AIDAhisto.py` and `run_example.py`

**testImages** Contains sample images of different staining (GFAP, IBA1) with different resolutions. There is also an already transformed atlas in original version `wholebrain_atlas.tif` and split version `wholebrain_atlas_splitted.tif` which is overlaid with the brain slice `wholebrain_slice.tif`. The results after the processing are described in detail in the manual.pdf. Those who process these images are listed in the logfile `wholebrain_slice.tif_process.log`

### 3 Atlas Transformation

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](#)
- [QuickNII Tool](#)

The transformed atlas slice (`wholebrain_atlas.tif`) is already included in the folder `.../AIDAhist-master/testImages/`. 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 2). 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.

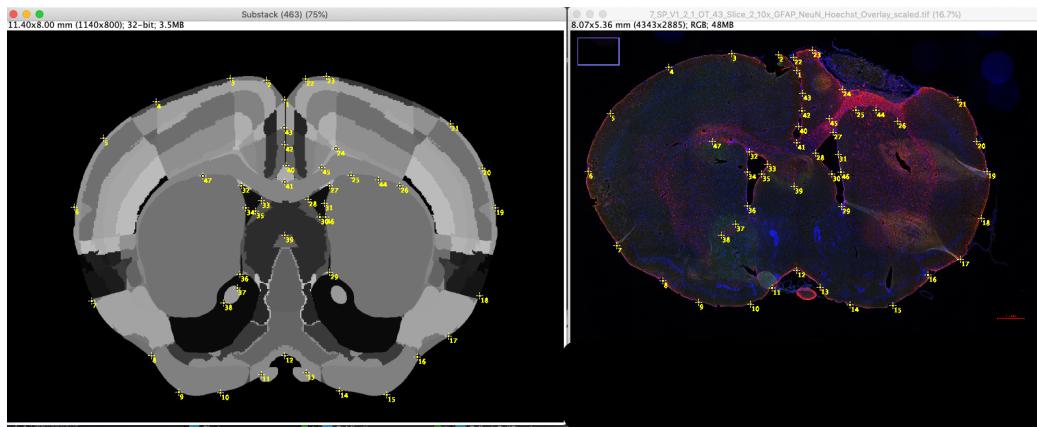


Figure 2: 30-50 landmarks in the microscopy image and the corresponding positions in the ARA

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 3).

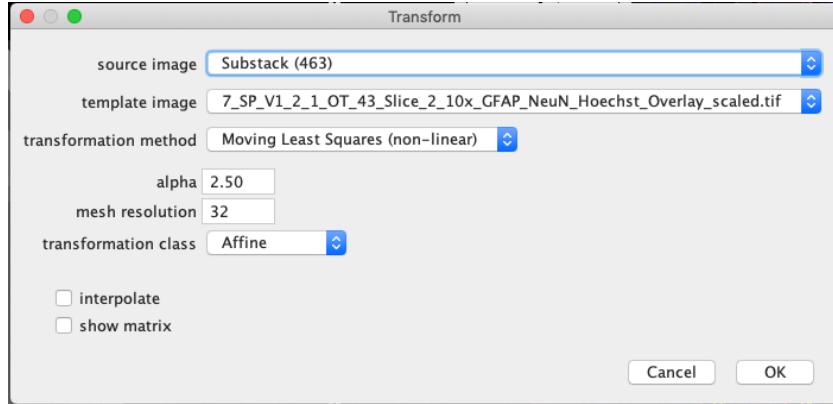


Figure 3: Choose the shown transformation parameters to register the atlas with the microscopy image

## 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.

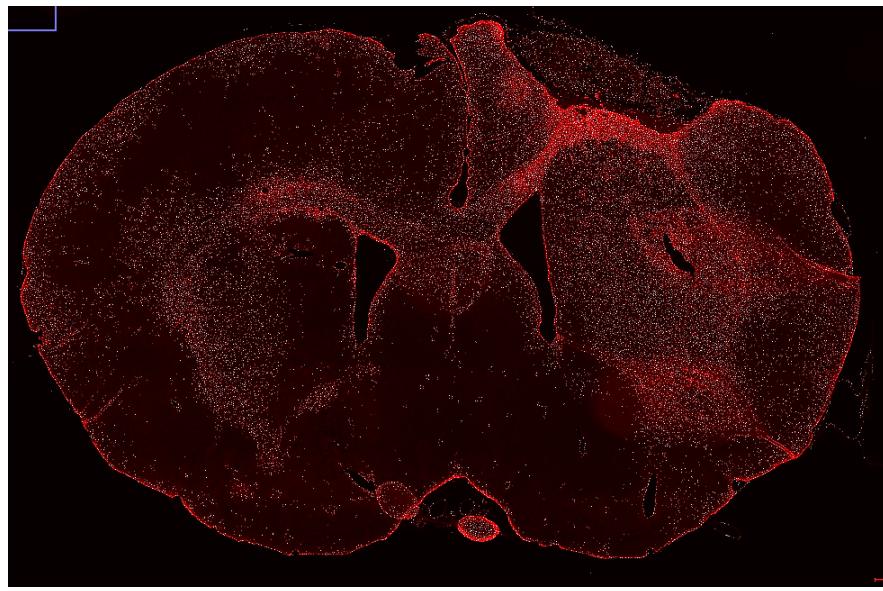


Figure 4: Example microscopy image (red channel): white spots indicate results of cell counting.

- Count all cells in the given image with Matlab: Here, you have to open Matlab and set `.../AIDAhisto-master/Matlab` as "Current Folder" (Figure 5A). Type the same command in the "Command Window" (see Figure 5B) like shown in figure 6.

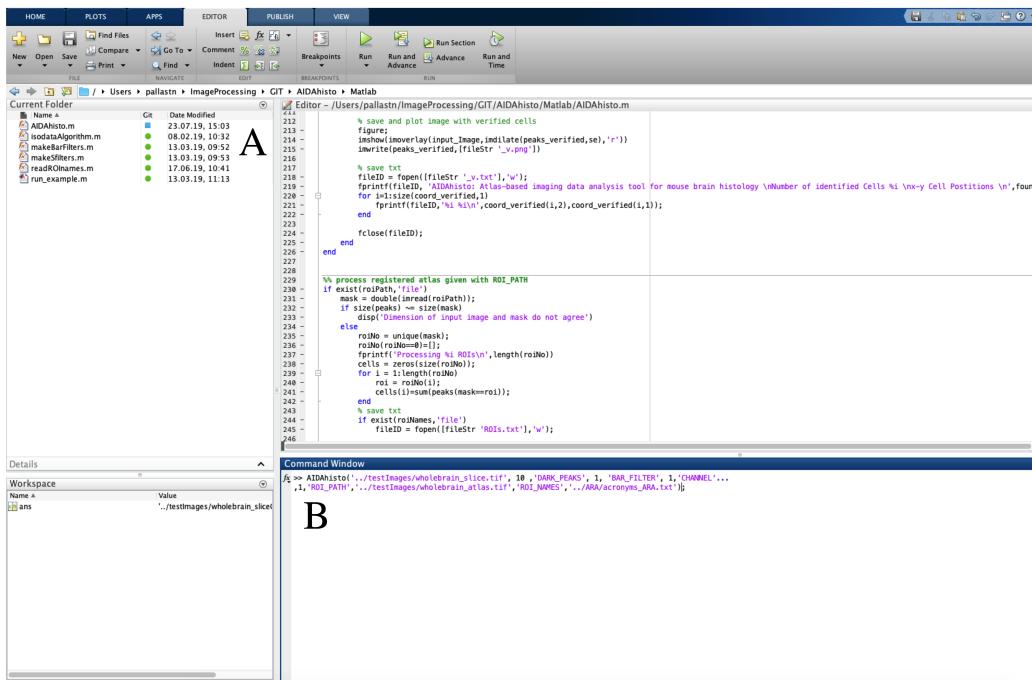


Figure 5: Set the `.../AIDAhisto-master/Matlab` as "Current Folder" (A) and type the command of figure 6 in "Command Window" (B)

```

Command Window
fx >> AIDAhisto('..../testImages/wholebrain_slice.tif', 10 , 'DARK_PEAKS', 1, 'BAR_FILTER', 1,'CHANNEL',...
    ,1, 'ROI_PATH','..../testImages/wholebrain_atlas.tif','ROI_NAMES','..../ARA/acronyms_ARA.txt');

```

Figure 6: Type the shown command in the "Command Window"

- 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 7)

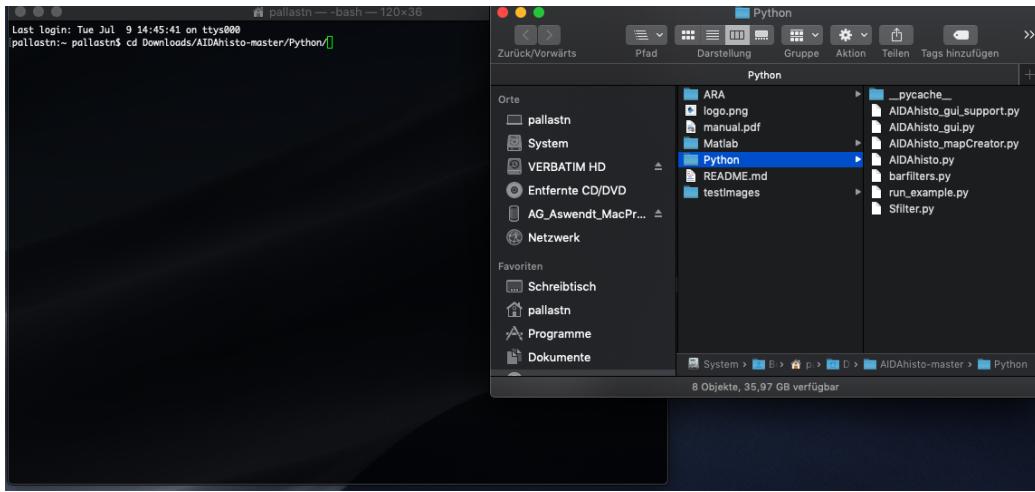


Figure 7: Set the Python folder as your current folder using the command `cd`

3. Type (not copy) the following command in the command window.

```
python AIDAhistro.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 `AIDAhistro-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.

## 5 Results

After successful completion of the previous steps you will end up with two text files (see Fig. 8) 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.

```

Number of detected cells: 31053
cell positions (xy):
68 2441
73 2436
106 2419
129 2421
131 2189
135 2292
136 2195
138 2211
139 2163
140 2239
144 2225
144 2250
145 2219
146 2259
147 2459
148 2240
148 2423
149 2266
150 2245
150 2153
151 2444
152 2431
153 2442
156 2243
157 2459
158 2511
160 2236

```

(a) wholebrain\_slice\_ch1\_cC.txt

```

Number of detected cells in 94 given ROIs
2 SSp-m6b 74
19 V1 19
36 GU1 152
72 ADP 4
81 VL 111
117 och 129
120 ADP1 152
129 V3 8
148 GU4 49
163 ATp2/3 98
180 OTp3/3 191
187 GU5 34
211 ACAd2/3 213
226 LP0 16
252 Lp1 71
258 LSr 871
263 AVP 8
272 AVPV 3
296 ACAv2/3 229
297 NW 43
314 ATp6a 17
320 MoP1 394
342 SI 74
344 ATp5 37
351 BST 172
450 SSp-u1 236
452 MEPO 6
477 STR 473

```

(b) wholebrain\_slice\_ch1\_cCROIs.txt

Figure 8: 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.

- **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

With the implementation in Python, a generic user interface has been implemented to support the arrival of AIDAhisto. To start the GUI set /Python as your current folder in the command window of your System and open the GUI (see 9) by typing `python AIDAhisto_gui.py`.

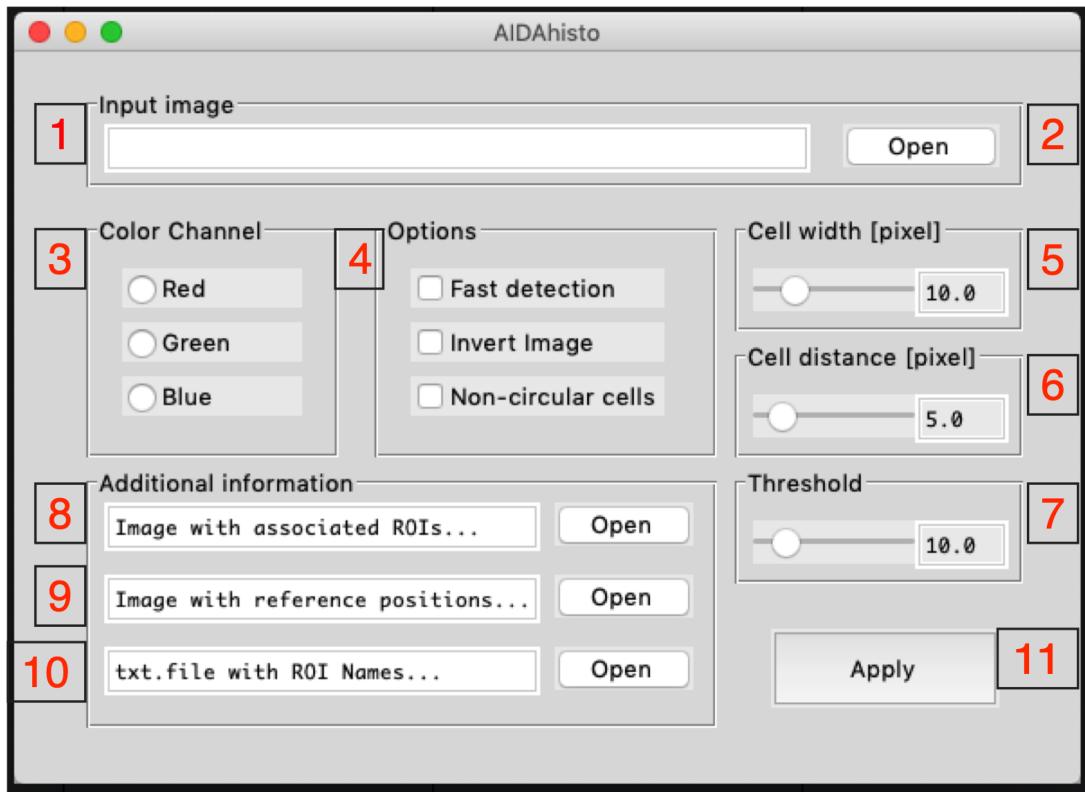


Figure 9: Description of the user interface to process with all provided input parameters.

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 9:

1. Path the file of the input image with the postfix .jpg, png, tiff
2. Press button to open file dialog and select a image file
3. Choose the color channel that should be evaluated. 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 are dark and the background is bright, the image should be inverted by choosing *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. The minimum cell distance is pre-set but can also be adapted.
7. The automatically calculated threshold value can be adjusted and weighted here.
8. If regions are superimposed with the image, the region image can be selected here.
9. You can enter the image of a previous investigation here and take it as a reference.
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/10.1101/124> 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](#).