

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

Niklas Pallast  
Department of Neurology  
University Hospital Cologne

2019

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.

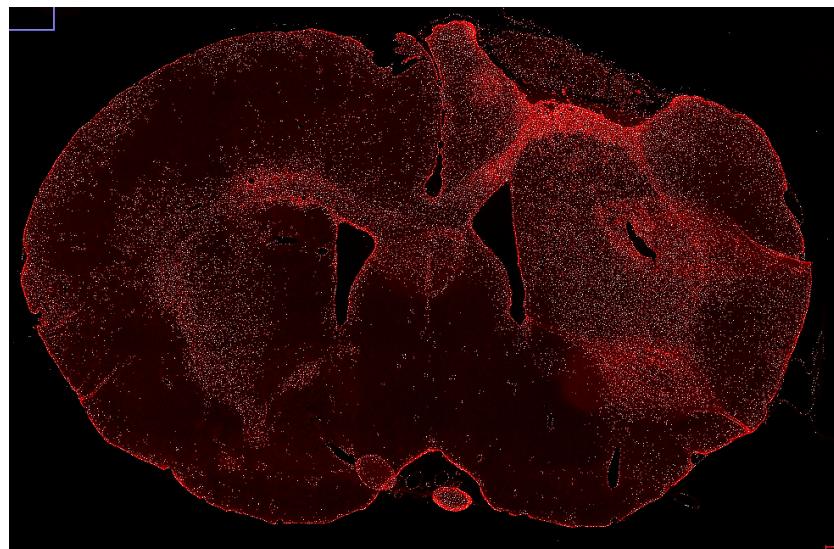


Figure 1: Example microscopy image (red channel): white spots indicate results of cell counting.

# 1 Download & Install

1. Download the zip-File `AIDAhisto-master` using the following [Link](#).  
After opening the web page, the zip-file can be downloaded (Figure 2).

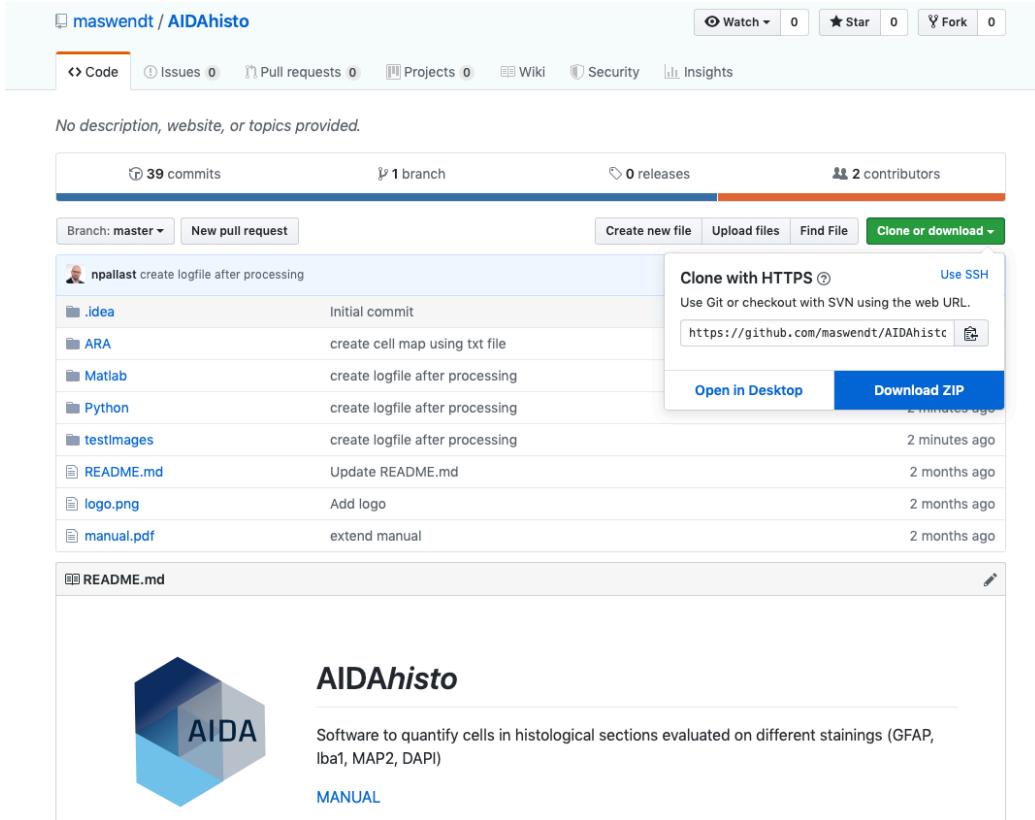


Figure 2: Click on Clone or Download → Download ZIP to download the zip-File.

2. 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=1.2.1 matplotlib=3.0.3`
3. The code was also implemented in Matlab (tested with version R2018a).  
Note: Matlab processing is faster and visualization better compared to Python; otherwise there are no differences.

## 2 Atlas Transformation

In the provided example files in the folder `.../AIDAhist-master/testImages/wholebrain_atlas.` the transformed atlas slice is already included. 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`
2. Load microscopy image in ImageJ from the folder  
`.../AIDAhisto-master/testImages/wholebrain_slice.tif`
3. Select the matching slice in the ARA and make a substack with ImageJ:  
Image → Stacks → Tools → Make Substack
4. Choose the corresponding slice number of the ARA and save the substack.
5. 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 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.

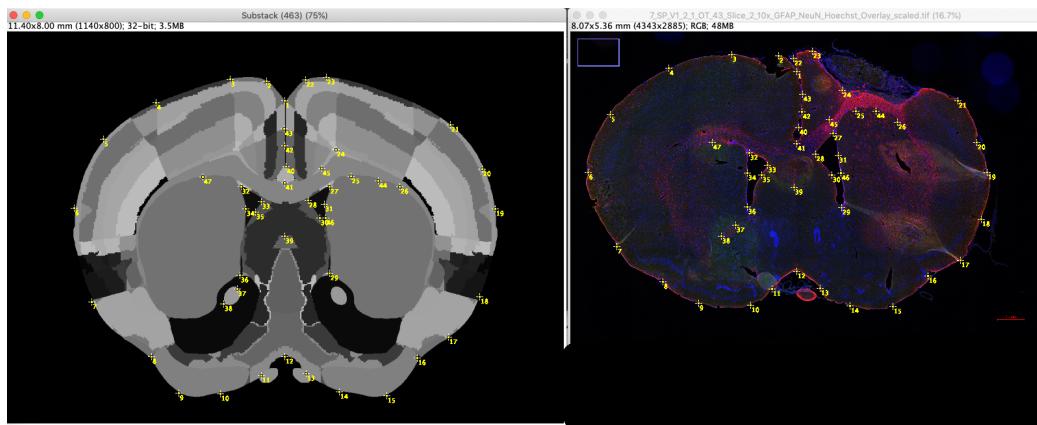


Figure 3: 30-50 landmarks in the microscopy image and the corresponding positions in the ARA

6. Landmark registration with ImageJ: Plugins → Transform → Landmark Correspondences
7. Choose the substack of the ARA as "source image" and the microscopy image as template image in the transform menu
8. Choose: Transformation method: Moving Least Squares (non-linear), Transformation class: Affine, No interpolation (Figure 4).

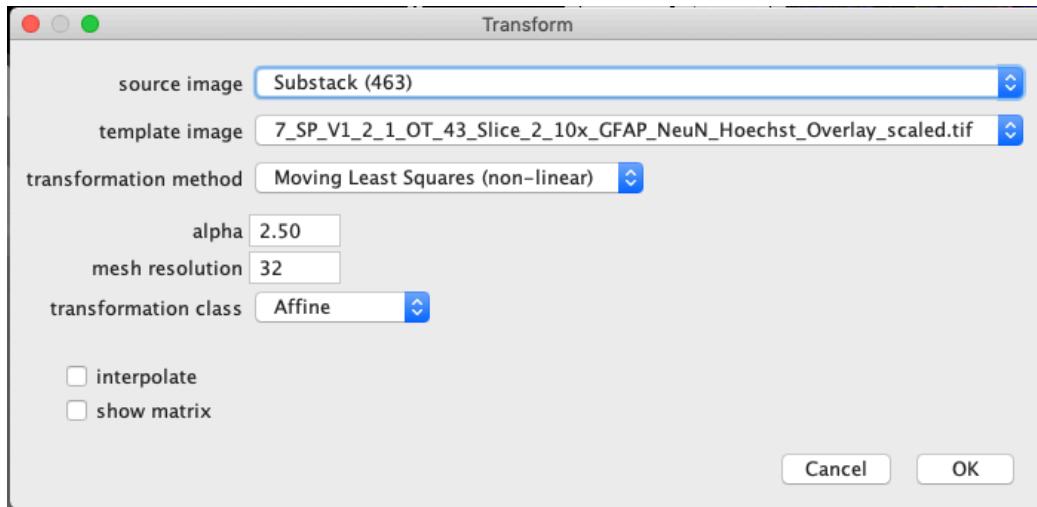


Figure 4: Choose the shown transformation parameters to register the atlas with the microscopy image

### 3 Count Cells

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 5A). Type the same command in the "Command Window" (see Figure 5B) like shown in figure 6.

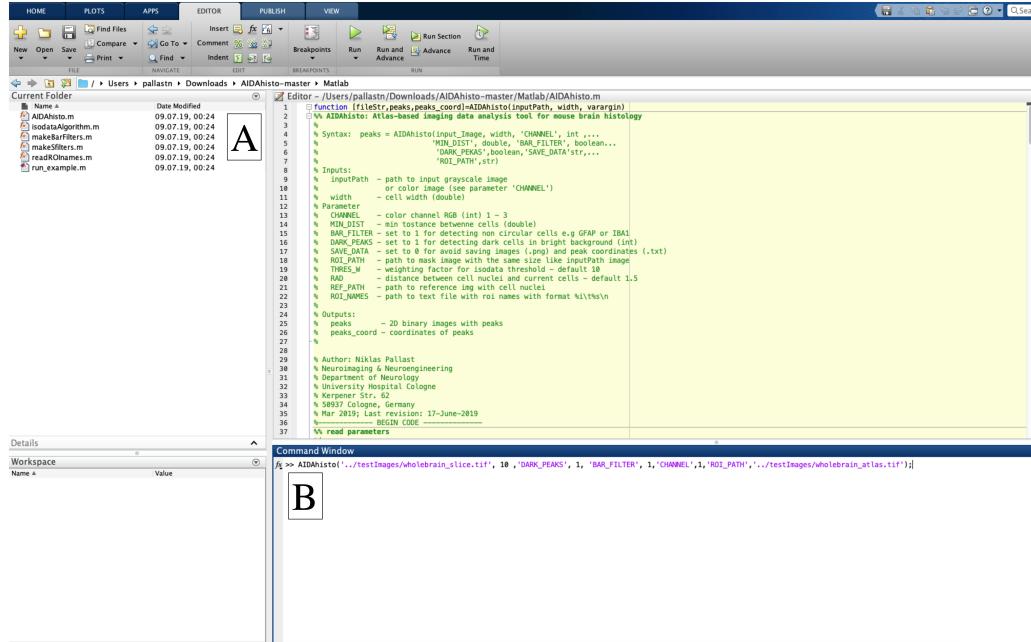


Figure 5: Set the `.../AIDAhisto-master/Matlab` as "Current Folder" (A) and type the command of figure 6 in "Command Window" (B)

```
f1 >> AIDAhisto('..../testImages/wholebrain_slice.tif', 10 , 'DARK_PEAKS', 1, 'BAR_FILTER', 1, 'CHANNEL',1,'ROI_PATH','..../testImages/wholebrain_atlas.tif');
```

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

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 7)

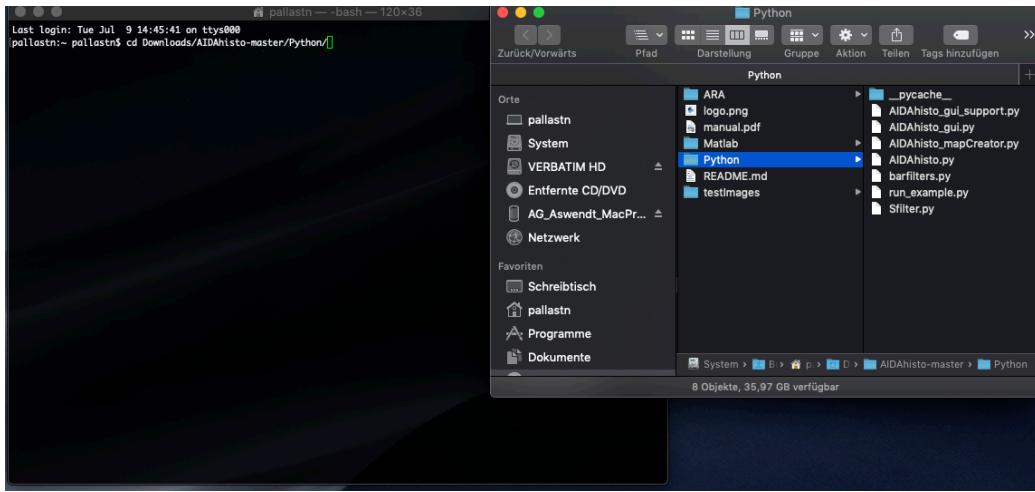


Figure 7: Set the Python folder as your current folder using the command `cd`

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.

## 4 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 2339
144 2225
144 2258
145 2219
146 2259
147 2459
148 2240
148 2423
149 2266
150 2145
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
15 18 19
36 GU1 152
72 ADP 4
81 VL 111
117 och 129
120 Atp1 152
129 V3 8
148 GU4 49
163 ATp2/3 98
180 GU5/3 191
187 GU5 34
211 ACAd2/3 213
226 LP0 16
251 LIP 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.