Apr 20, 2024

A Free Analysis Pipeline to Coregister 3D Lightsheet or Serial Sectioned Mouse Brains to the Allen Brain Atlas Using ABBA and Obtain Region Specific Cell Density Quantifications for 800+ Regions

DOI
dx.doi.org/10.17504/protocols.io.8epv5rz1ng1b/v1

Liam McLaughlin[1]

[1]Washington University in St. Louis

Human BioMolecular Atlas Program (HuBMAP) Method Development Community
Tech. support email: **Jeff.spraggins@vanderbilt.edu**

Liam

Liam McLaughlin
Washington University in St. Louis

DOI: dx.doi.org/10.17504/protocols.io.8epv5rz1ng1b/v1

Protocol Citation: Liam McLaughlin 2024. A Free Analysis Pipeline to Coregister 3D Lightsheet or Serial Sectioned Mouse Brains to the Allen Brain Atlas Using ABBA and Obtain Region Specific Cell Density Quantifications for 800+ Regions. **protocols.io** **https://dx.doi.org/10.17504/protocols.io.8epv5rz1ng1b/v1**

Protocol status: Working
We use this protocol and it's working

Created: April 20, 2024

Last Modified: April 20, 2024

Protocol Integer ID: 98553

Funders Acknowledgement:
NIH
Grant ID: R01AG078512

Disclaimer

DISCLAIMER – FOR INFORMATIONAL PURPOSES ONLY; USE AT YOUR OWN RISK

The protocol content here is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to **protocols.io** is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with **protocols.io**, can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

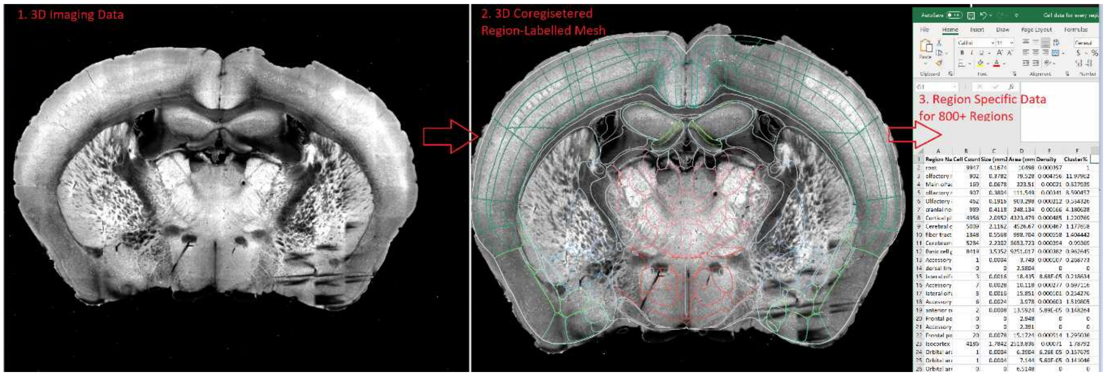
Abstract

Coregistration of data from imaged brains unto a brain atlas is one step to establish significance and legitimize differences in the densities/distributions of objects such as cells in the brain between separate channels or experimental cohorts. In this method, I provide a free, simple, and largely automated pipeline to coregister and analyze 3D mouse brain data using the tool ABBA, a free coregistration platform developed by Nicolas Chiaruttini, as well as some python scripts and ImageJ macros which I have developed. This protocol will go over preprocessing steps to prepare your image for import into ABBA as a serial, multi-channel .tiff image. It will provide instructions and tips on using ABBA to coregister 3D brains. Finally, it will explain a pipeline using python and the free imaging software 'ImageJ' to automatically obtain region specific information for all regions simultaneously, such as cell counts, cell densities, and a normalized, robust clustering factor that can safely be compared between heterogenous images, and provides a method for obtaining significance differences in regions between two cohorts regions based on the extracted information.

Introduction

1 Coregistration of data from imaged brains unto a brain atlas is one step to establish significance and legitimize differences in the densities/distributions of objects such as cells in the brain between separate channels or experimental cohorts. There are several services that provide such analysis for 3D brain images, however they are often costly and may occur outside of the laboratory preforming experiments. In this method, I provide a free, simple, and largely automated pipeline to coregister and analyze 3D mouse brain data using several pre- and post-processing scripts I developed, as well as the tool ABBA[1] , a free coregistration platform developed by Nicolas Chiaruttini.

The following method provides a free, fast, and accurate way to coregister mouse brains to the Allen Brain Atlas[2], automatically obtain region-specific data on 800+ regions, and establish significant differences between regions of separate mouse cohorts. These steps allow coregistration to be performed in-house, providing a great degree of control over this analysis within a lab. Once a user is comfortable with this pipeline, a 3D brain image can be converted into region-specific data in around two hours each. An example of this pipeline is shown in Fig. 1. below.


1. 3D Imaging Data
2. 3D Coregisetered Region-Labelled Mesh
3. Region Specific Data for 800+ Regions

	A	B	C	D	E	F
1	Region Na	Cell Count	Size (mm2	Area (mm	Density	Cluster%
2	root	9947	4.1674	10498	0.000097	1
3	olfactory r	902	0.3782	79.528	0.004756	11.97902
4	Main olfa	169	0.0678	323.51	0.00021	0.527935
5	olfactory r	907	0.3804	111.549	0.00341	8.590457
6	Olfactory	462	0.1916	903.298	0.000212	0.534326
7	cranial ne	989	0.4118	248.134	0.00166	4.180628
8	Cortical pl	4956	2.0952	4323.479	0.000485	1.220766
9	Cerebral c	5009	2.1162	4526.67	0.000467	1.177658
10	fiber tract	1348	0.5568	998.704	0.000558	1.404442
11	Cerebrum	5284	2.2302	5653.723	0.000394	0.90365
12	Basic cell g	8419	3.5352	9251.017	0.000382	0.962645
13	Accessory	1	0.0004	3.749	0.000107	0.266773
14	dorsal lim	0	0	2.5804	0	0
15	lateral olf	3	0.0016	18.435	8.68E-05	0.218634
16	Accessory	7	0.0028	10.118	0.000277	0.697116
17	lateral olf	3	0.0016	15.851	0.000101	0.254276
18	Accessory	6	0.0024	3.978	0.000603	1.519805
19	anterior o	2	0.0008	13.5924	5.89E-05	0.148264
20	Frontal po	0	0	2.948	0	0
21	Accessory	0	0	2.391	0	0
22	Frontal po	20	0.0078	15.1724	0.000514	1.295036
23	Isocortex	4195	1.7842	2513.836	0.00071	1.78792
24	Orbital ar	1	0.0004	6.3904	6.26E-05	0.157675
25	Orbital ar	1	0.0004	7.144	5.60E-05	0.141046
26	Orbital ar	0	0	6.5148	0	0

Fig 1. This pipeline provides a fast, free, and largely automated way to accurately coregister 3D brain data and obtain quick results for all 800+ regions.

Installation

2 The first step is to install ABBA[1] and other necessary tools. Detailed instructions for ABBA installation can be found at https://biop.github.io/ijp-imagetoatlas/installation.html. Using Windows, installation of all ABBA dependencies can be easily accomplished by running an installer, but otherwise, each dependency will need to be installed individually per the instructions in that link. ABBA can also be installed through Python using pip.

3 Pre- and post- processing through this pipeline additionally requires the image processing software FIJI/ImageJ[3] [https://imagej.net/software/fiji/downloads], which also must be installed for ABBA.

4 Next, to follow these pre- and post-processing steps, download the python scripts and ImageJ macros from my github:
https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git

5 Determining exact angular lineups, which will speed up coregistration in instances of misangled images, can be aided with the platform Imaris [https://www.oxinst.com/news/imaris-10-1-released], although the license is expensive, and this step can be safely skipped.

6 Additionally, Python should be installed should be installed via the installer downloadable at: **https://www.python.org/downloads/release/python-3123/**. When prompted by the python installer, make sure the option to enable Python as an environment variable is enabled to allow its scripts to be run in any directory.

7 Additionally, once python installed, please install the following modules by typing the following commands into the command terminal:

```
pip install pandas==2.2.0
pip install numpy
pip install openpyxl
pip install tifffile
pip install scipy
```

Imaging

8 Imaging techniques may widely vary, including 3D lightsheet images to serial 2-photon slices. However, be advised that the following tips will ease coregistration:

9 The brain should ideally be imaged perpendicular to either the coronal, sagittal, or horizontal axis. For optimal coregistration, the entirety of the brain in the intended coregistration plane should be imaged. As coronal coregistration is the easiest, optical or serial sections may want to encompass the entirety of the brain in the coronal plane if possible. Without the entire brain in the plane, the automated coregistration tools in ABBA will not function properly, permitting only manual coregistration.

10 Additionally, coregistration will be easiest with a channel that provides a strong autofluorescent background signal for the brain and associated subregions, for easy matching of fiducials between image and atlas.

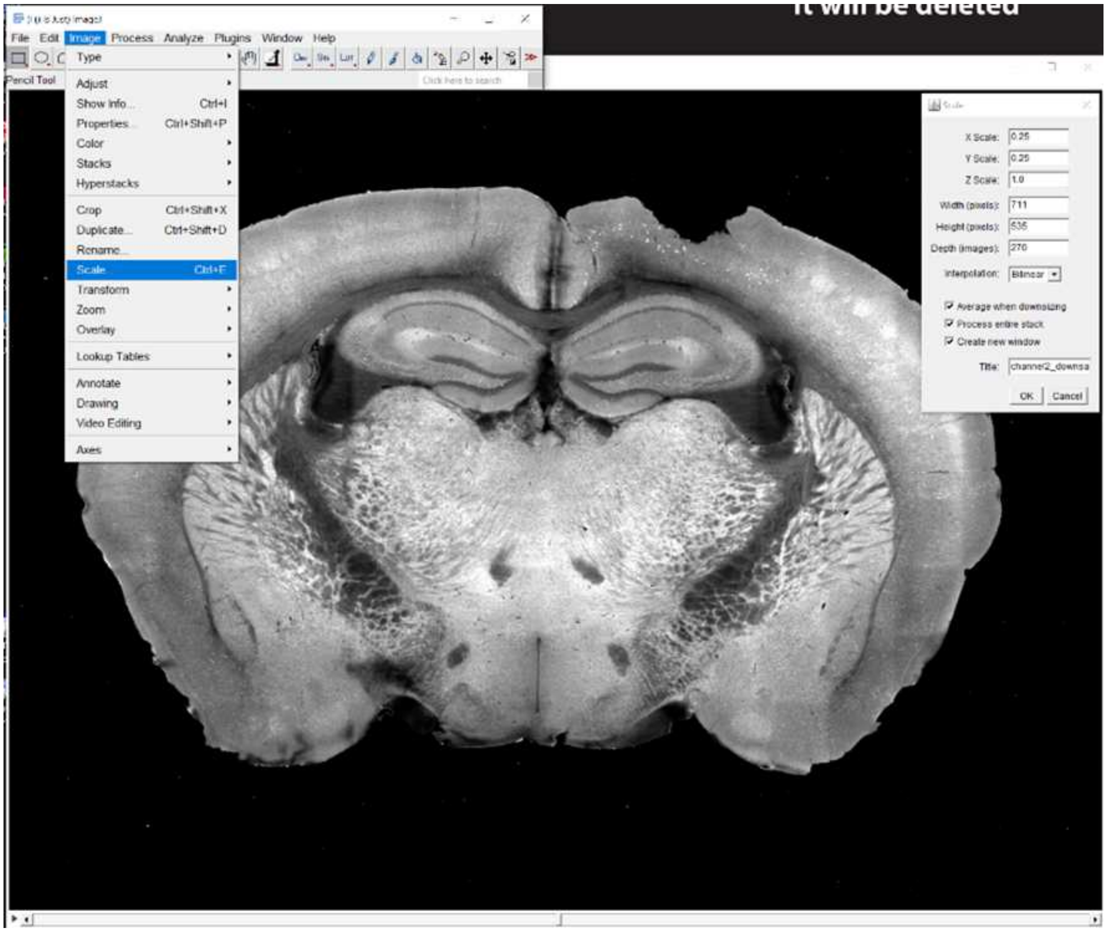
Preprocessing:

11 This pipeline is designed for stitched tiff stack images. If the image field is taken over multiple tiles, it can be stitched using a program such as the free imageJ plugin BigStitcher:

[https://imagej.net/plugins/bigstitcher/]. If the image is not in .tiff format, it can be converted using FIJI. Open the image stack in FIJI and select File->Save As->Tiff.

Downsampling and Reslicing:

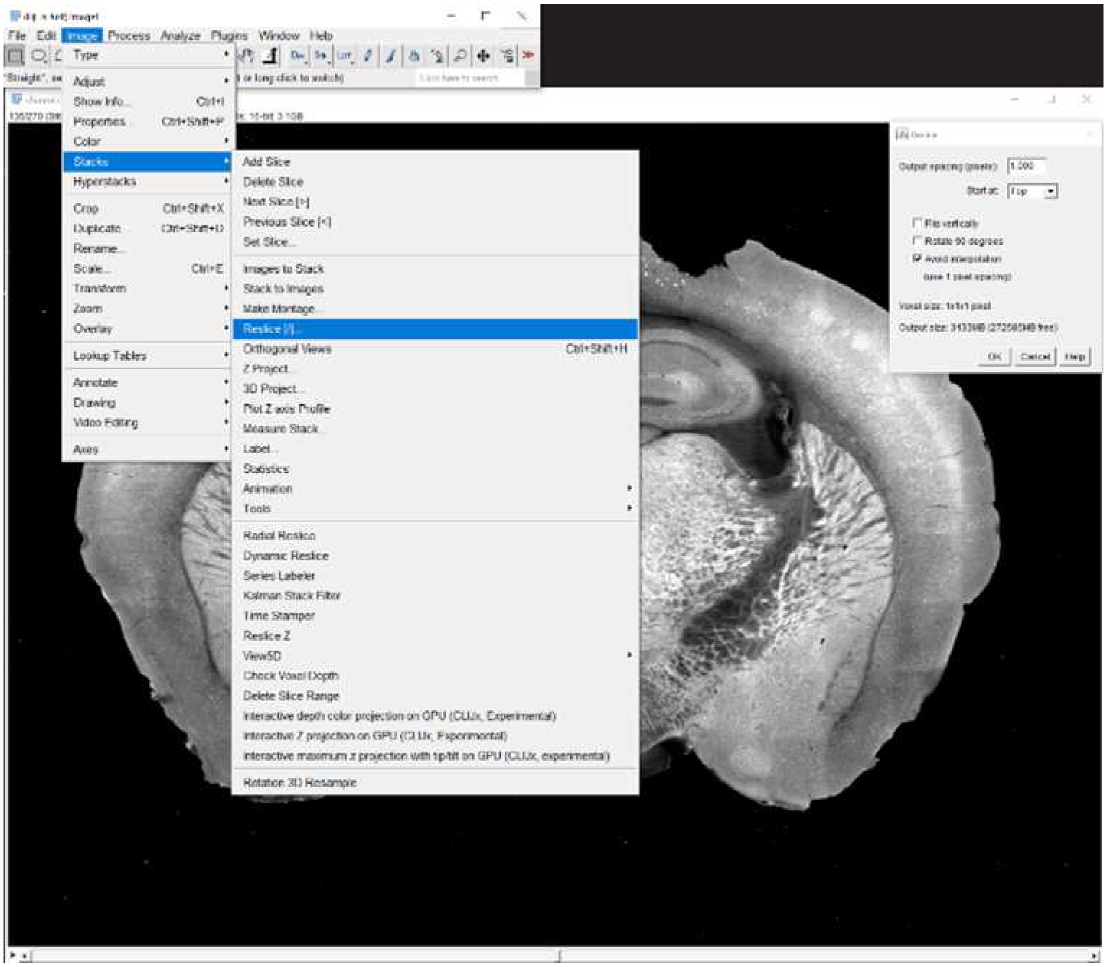
11.1 Resave each channel as a tiff stack and complete the following same preprocessing steps on each: Depending on the specifications of the computer being used for downsampling, it may be a good idea to downsample the mouse brain data prior to coregistration. This may be accomplished using FIJI. Open the tiff image in FIJI and select Image->Scale and designate downsampling parameters in the X and Y plane only (Fig. 3).



X Scale: 0.25
Y Scale: 0.25
Z Scale: 1.0
Width (pixels): 711
Height (pixels): 535
Depth (images): 270
Interpolation: Bilinear
☑ Average when downsizing
☑ Process entire stack
☑ Create new window
Title: channel2_downsa

Fig. 3. Downsample if necessary. The above image is being downsampled 4x in X and Y.

11.2 ABBA allows coregistration along the sagittal, coronal, and horizontal axis. However, as fiducials are easiest to match in the coronal plane, resampling the image in the coronal axis will make the coregistration the easiest. Assuming it is perpendicular to one of the three major brain axis, resampling can easily be accomplished in FIJI. In FIJI, select Image->Stacks->Reslice (Fig. 4) and select the resample direction to transform the image into the coronal axis. Before coregistration, the image should begin with the olfactory bulbs at the start of the Z-stack (plane 1) and end with the brainstem at the end of the Z-stack. Additionally, the brain should be positioned the cortex on top. In the instance the brain is not aligned this way, it can be transformed in FIJI using the tools under Image->Transform (Fig. 5).



Reslice

Output spacing (pixels): 1.000

Start at: Top

□ Flip vertically

□ Rotate 90 degrees

☑ Avoid interpolation (use 1 pixel spacing)

Voxel size: 1x1x1 pixel

Output size: 3433MB (27258MB free)

Fig. 4. Reslice the brain in a new axis to make it coronal, if necessary. The above image is already positioned correctly, but if I applied the parameters on the right, it would reslice it starting from the "top" of the image, which would result in a horizontal axis slice.

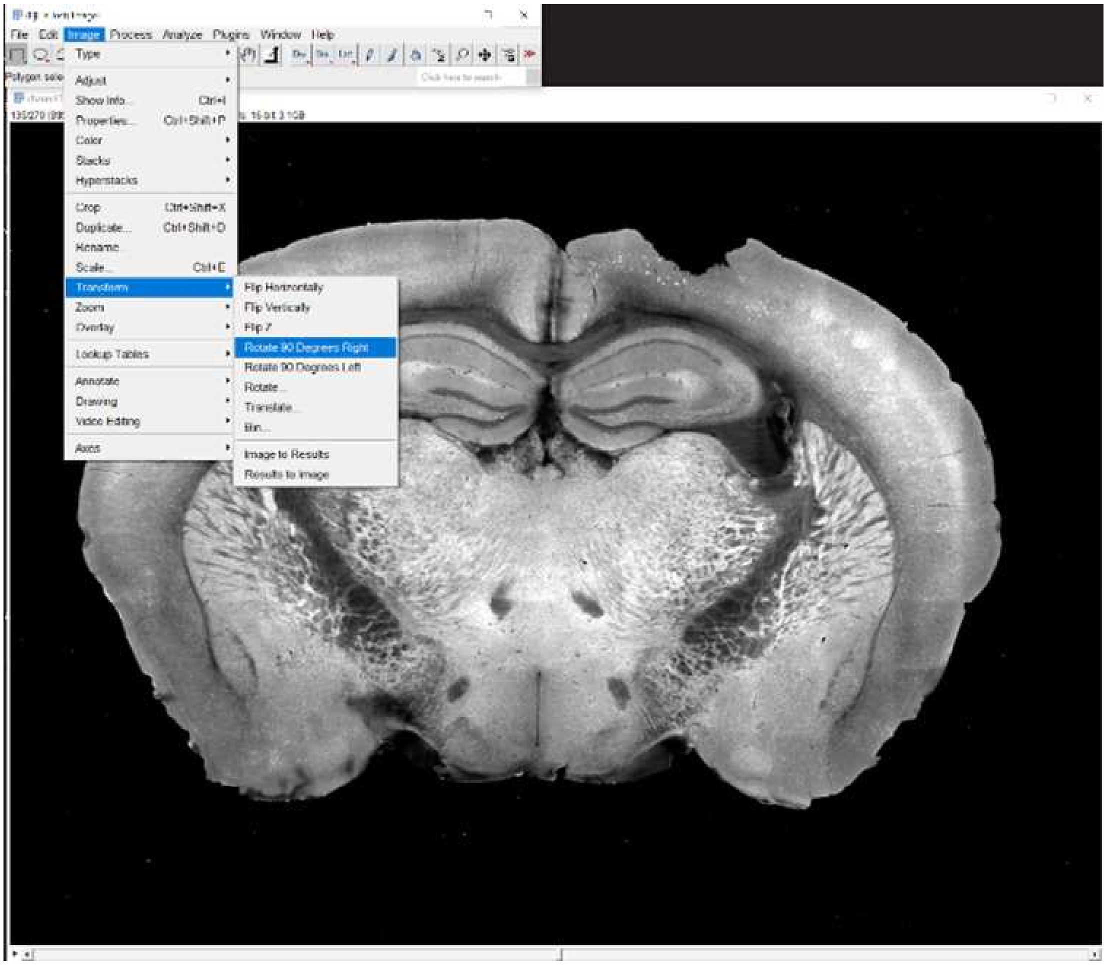
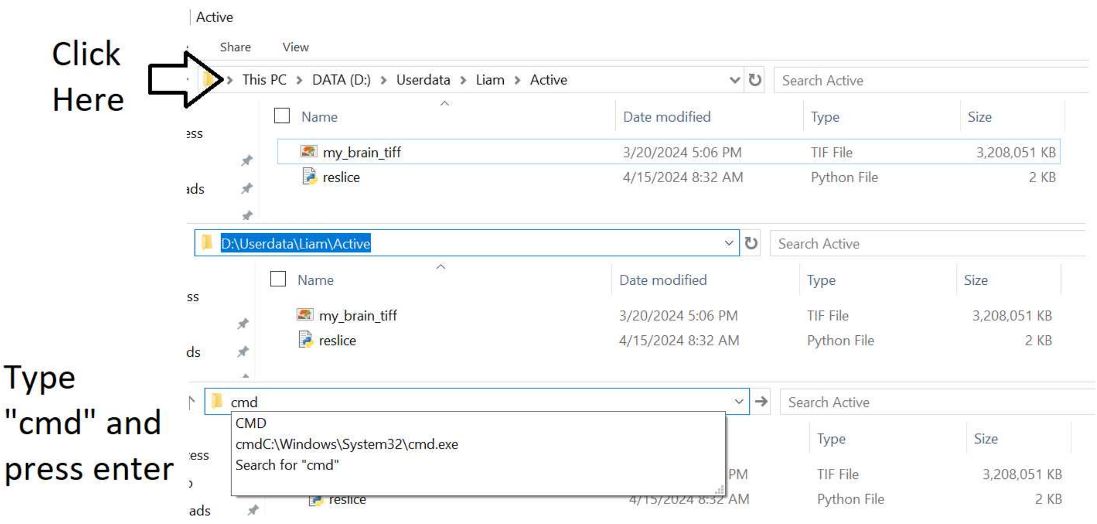


Fig. 5. Transform the brain to position it correctly, if necessary. The above image is already positioned correctly, but if it was not, I would use this window to rotate it into the correct position. Additionally, "Flip Z" may be used to position the olfactory bulbs near slice 1 if the brain is backwards.

11.3 Because ABBA is limited in the number of planes it will process in parallel, it may be a good idea to reslice the brain along the coronal axis if there are an abundance of slices, such as in a lightsheet image. This may be accomplished using my reslice.py python script available at https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git. Run the script from the command terminal in the same directory as the image to reslice (Figs 6, 7). Input the image’s name in the console and select a reslice factor to generate an imaged resliced by that factor in the Z-plane, aiming to end up with a few hundred slices if possible, as too many will slow processing and too few will undersample the data. Ultimately, after downsampling and reslicing, tiff-images with channel files that were under 5 GB in size worked well for ABBA, while larger images may pose processing challenges.



This PC > DATA (D:) > Userdata > Liam > Active

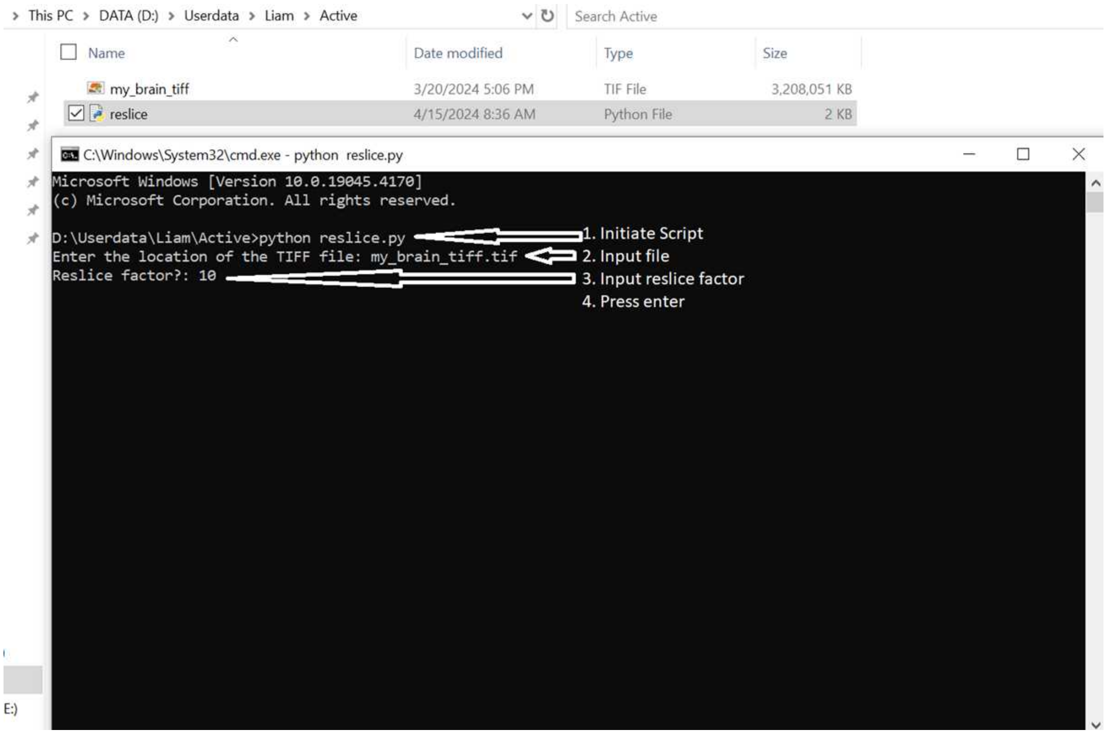
Name	Date modified	Type	Size
my_brain_tiff	3/20/2024 5:06 PM	TIF File	3,208,051 KB
reslice	4/15/2024 8:32 AM	Python File	2 KB

D:\Userdata\Liam\Active

cmd

CMD
cmdC:\Windows\System32\cmd.exe
Search for "cmd"

Fig. 6: How to easily open the command terminal in a folder to run scripts (windows). Click the filepath bar in the windows explorer, type "cmd", and press enter.



This PC > DATA (D:) > Userdata > Liam > Active

Name	Date modified	Type	Size
my_brain_tiff	3/20/2024 5:06 PM	TIF File	3,208,051 KB
reslice	4/15/2024 8:36 AM	Python File	2 KB

C:\Windows\System32\cmd.exe - python reslice.py

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\Userdata\Liam\Active>python reslice.py
Enter the location of the TIFF file: my_brain_tiff.tif
Reslice factor?: 10
```

1. Initiate Script
2. Input file
3. Input reslice factor
4. Press enter

Fig. 7: How to run the reslice script (Windows). Making sure the script and brain image are in the same folder as the command terminal, type "python reslice.py" to start the script. The "python" command tells windows to apply the python compiler on whatever file is called. Input the name of the file, including the ".tif" extension. Specify a reslice factor and press enter to run the script.

Creating Qupath Import

12 Transferring your image into ABBA requires first converting it into a new project in QuPath. Prior to transforming the image into a QuPath project, it should be turned into a set of serial images in a directory rather than a single tiff stack file. Open all image channels in FIJI and merge them into one using Image->Color->Merge Channel (*Fig. 8)*.

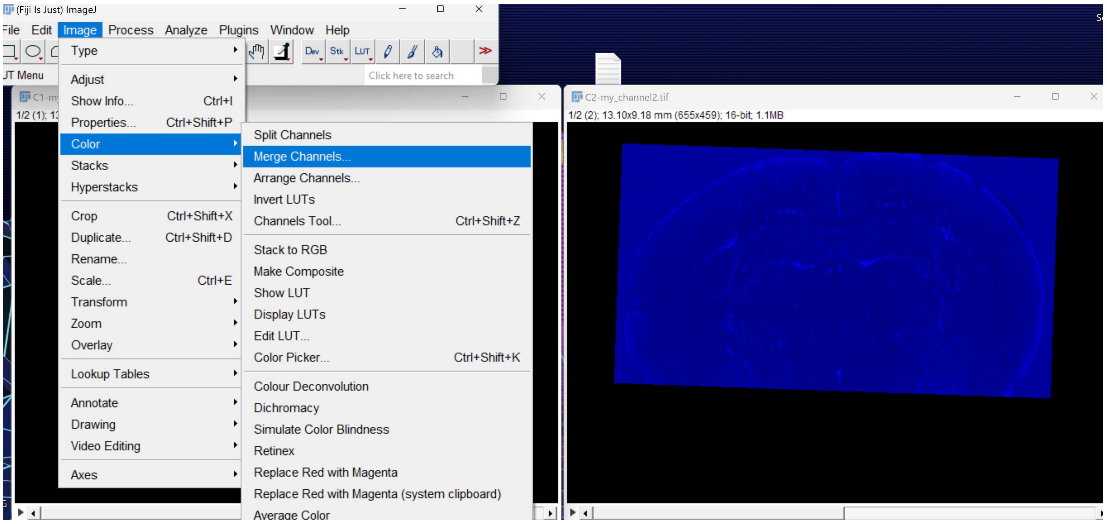
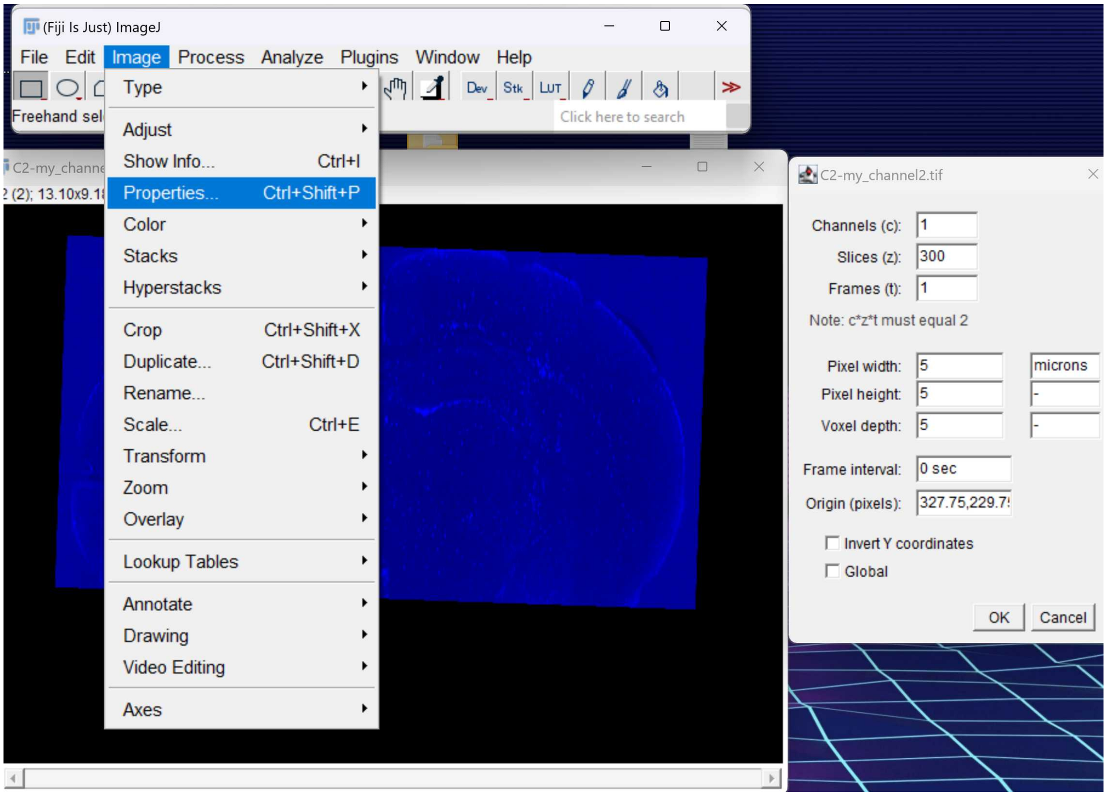


Fig 8: Until this point, channels have been processed as separate files. Now, merge them back into one image.

12.1 To ensure the brain image is imported into ABBA in its proper size, voxel-sizes (the true micro-or-milimeter size of the 3D pixel) may be verified at this point using Image->Properties (*Fig. 9)*. Please input the proper voxel sizes of the brain in the current XY plane into the image, which may have changed if the axis were shifted. Ensuring proper Z-axis size is not necessary. In addition, since the image will be warped onto the Atlas, correct voxel size is not actually needed as the image can be freely resized during coregistration, but this step will make initial alignment more straightforward.



Channels (c): 1
Slices (z): 300
Frames (t): 1
Note: c*z*t must equal 2

Pixel width: 5 microns
Pixel height: 5 -
Voxel depth: 5 -

Frame interval: 0 sec
Origin (pixels): 327.75,229.7!

□ Invert Y coordinates
□ Global

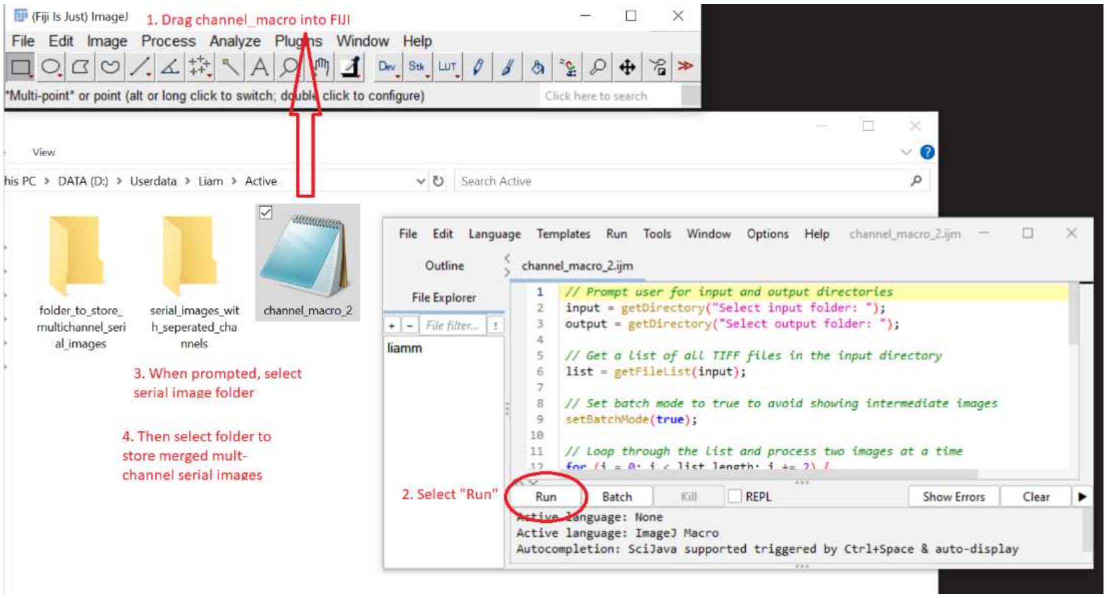
OK Cancel

Fig. 9: Adjust the voxel sizes of the image, if necessary. Since the image will be resized during coregistration, these values do not actually matter but may make things more straightforward. Keep in mind that if the brain image was resliced into a coronal axis, the Z-axis which typically has different voxel sizes than X and Y will have been shifted into either X or Y. This may make the image appear stretched out, but if proper voxel sizes are designated, it will look normal when imported into ABBA.

12.2 Save this image as a sequence in a new folder using FIJI by selecting File->Save As->Image Sequence, and choosing Tiff as the file format.

12.3 If the image is only a single channel, please skip to step 12.7. Otherwise, the saved image sequence will not combine the channels from the same slice, which should be done before importing into ABBA if multiple channels are to be coregistered. To transform the image sequence containing multiple channels as separate images into an image sequence containing all channels in one image per section, first create a new directory.

12.4 Open my channels_macro available at available at https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git in FIJI and select run (Fig. 10). There are several channels_macros, each which correspond to the number of channels they will assemble into one slice. See step 12.6 for more explanation. Note the first directory containing the channels to be merged should only contain the data that FIJI saved by the steps described above, or the channel macro may fail.



(Fiji Is Just) ImageJ
File Edit Image Process Analyze Plugins Window Help
Multi-point or point (alt or long click to switch; double click to configure)
Click here to search
View
his PC > DATA (D:) > Userdata > Liam > Active
Search Active
folder_to_store_multichannel_serial_images
serial_images_with_seperated_channels
channel_macro_2
File Edit Language Templates Run Tools Window Options Help channel_macro_2.ijm
Outline
File Explorer
File filter...
liamm
channel_macro_2.ijm
Run
Batch
Kill
□ REPL
Show Errors
Clear

1. Drag channel_macro into FIJI

2. Select "Run"

3. When prompted, select serial image folder

4. Then select folder to store merged mult-channel serial images

```
// Prompt user for input and output directories
input = getDirectory("Select input folder: ");
output = getDirectory("Select output folder: ");

// Get a list of all TIFF files in the input directory
list = getFileList(input);

// Set batch mode to true to avoid showing intermediate images
setBatchMode(true);

// Loop through the list and process two images at a time
for (i = 0; i < list.length; i += 2) {
```

```
Active language: None
Active language: ImageJ Macro
Autocompletion: SciJava supported triggered by Ctrl+Space & auto-display
```

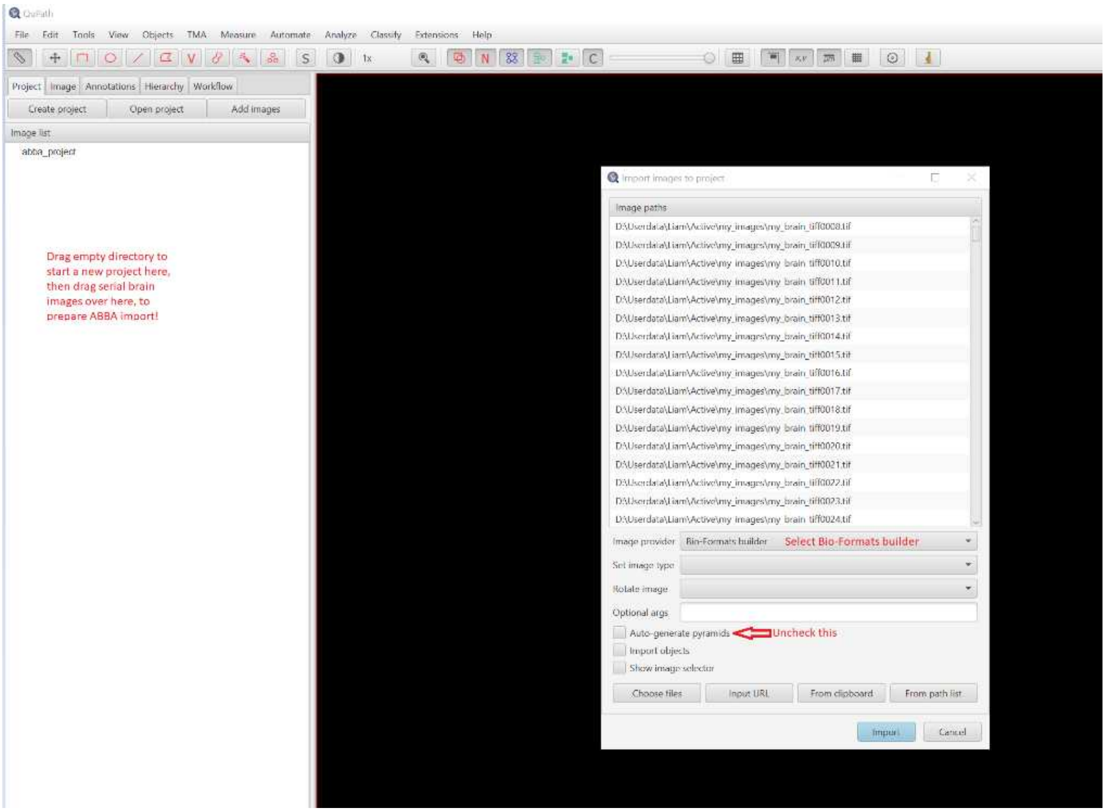
Fig. 10. To run macros in FIJI, simply drag them into FIJI and press run. To merge the serial, split-channel images into serial, multi-channel images, use the corresponding channel_macro. In the above example, I am merging two channels with channel_macro_2.

12.5 When prompted, find the directory containing your serial images, and select the new directory to save the output. The channels macro will iterate through the first folder, combine the channels from the same slice together, and save the multi-channel slices in the output folder.

12.6 *Tip: Note that this is intended to only work for sequences saved using the pipeline described above. In addition, there are multiple versions of the channels macro, each designed for a certain number of channels. Use “channel_macro_2” for two channel images, “channel_macro_3” for three channel images, “channel_macro_4” for four channel images, etc. Using the wrong channel macro will either fail or cause separate slices to merge as a multi channel image.*

12.7 At this point, the image is ready to be opened as a QuPath project and exported to ABBA. First, create an empty directory and open QuPath. Drag and drop the empty directory into the project toolbar on the left side of QuPath (or use the “create project” ribbon).

12.8 Next, select all the serial, multi-channel slices of the brain image and drag them into the left area of QuPath under “Image list”. When prompted, change the “Image Provider” option from “Default (Let QuPath decide)” to “Bio-Formats builder”, and uncheck “Auto-generate pyramids” (Fig. 11).


QuPath
File Edit Tools View Objects TMA Measure Automate Analyze Classify Extensions Help
Project Image Annotations Hierarchy Workflow
Create project Open project Add images
Image list
abba_project
Import images to project
Choose files
Input URL
From clipboard
From path list
Import
Cancel

Drag empty directory to start a new project here, then drag serial brain images over here, to prepare ABBA import!

Image paths

```
D:\Userdata\Liam\Active\my_images\my_brain_tiff0008.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0009.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0010.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0011.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0012.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0013.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0014.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0015.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0016.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0017.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0018.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0019.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0020.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0021.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0022.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0023.tif
D:\Userdata\Liam\Active\my_images\my_brain_tiff0024.tif
```

Image provider: Bio-Formats builder — Select Bio-Formats builder

Set image type:

Rotate image:

Optional args:

☐ Auto-generate pyramids ⇐ Uncheck this

☐ Import objects

☐ Show image selector

Fig. 11. Drag the serial, multi-channel slices into QuPath and open it with the above settings.

Coregistering Using ABBA

13 ***An in depth, step-by-step tutorial to use ABBA can be found at https://www.youtube.com/watch?v=sERGONVw4zE. This tutorial was created by the platform's developer, Nicolas Chiaruttini, and should be referenced for more detailed instructions.***

Setup

14 Start ABBA by searching for “ABBA” in FIJI and pressing “start”. When prompted, select “Adult Mouse Brain – Allen Brain Atlas V3p1” and the proper slicing plane, such as “coronal”.

14.1 In ABBA, open your image by selecting [Import -> Import Current QuPath Project] and finding the QPPROJ file associated with your QuPath project within the directory created for the QuPath project.

14.2 When the images are loaded in, separate them by selecting “Change Overlay Mode”. To spread the slices out, manually move the first in the stack to the left and the last in the stack to the right, then select “distribute spacing”.

14.3 It may be the case that a few slices were imported out of order. Check to make sure that the slices are in serial ascending order from left to right. If they are not, the slices can be selected and moved to their proper location.

14.4 Next, increase number of displayed atlas slices to aid coregistration using the slider next to "Displayed slicing [atlas steps]. A display factor such as "8" is a good starting point.

14.5 Change the brightness on the Atlas and your image until both are easily visible.

Aligning Atlas in Z

15 Finding proper alignment in the Z plane is the most important step to ensure good coregistration. In ABBA, this is accomplished by designating "key slices" where the imaged brain's region fiducials match the atlas's. Brain slices can be freely dragged to match their locations in the Z plane with the atlases. Ctrl+a can be used to drag all slices at once. However, any slices that are "designated" as key slices will remain rooted in place.

15.1 Ensuring Proper Angular Alignment

15.2 To start, find where the hippocampus first appears in the coronally-sliced brain and drag the entire brain until the hippocampus-slice matches where the hippocampus first appears in the Atlas. Select the hippocampus-slice and choose "set as Key Slice" (Fig. 12). Key slices may also be removed by selecting a slice and choosing "Remove Key Slice".

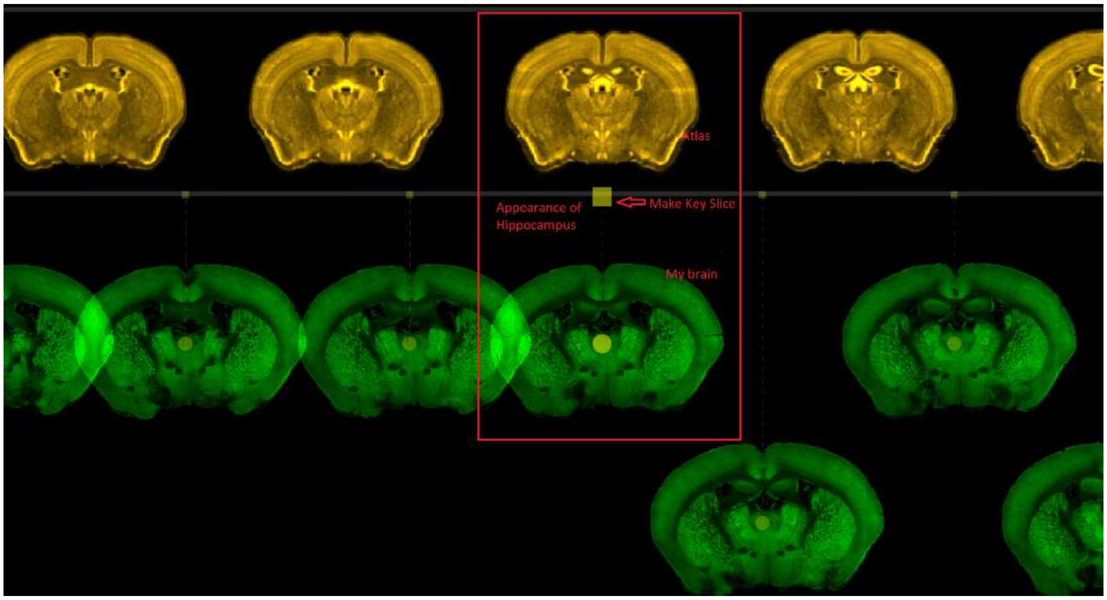


Fig. 12. Example alignment of hippocampus appearance fidiciual. The atlas (yellow) and the brain image (green) have been matched for the slice where the hippocampus first appears coronally. Having found this match, I will mark this as a key slice and move onto matching other fiducials as keys, allowing ABBA to interpolate a Z-alignment.

Ensuring Proper Angular Alignment

16 If the brain image was not taken perfectly-perpendicular to one of the three brain axis, the angle of the atlas will need to be changed to permit proper alignment. If the brain image is already perpendicular and not angled relative to the atlas, skip to step 17.

16.1 A perfect angle-adjustment can be obtained with the platform Imaris. If you do not have Imaris, skip to step 16.5. Download the Allen Brain Atlas v3 from [https://scalablebrainatlas.incf.org/mouse/ABA_v3] and convert both it and your brain into .ims images using the Imaris file converter.

16.2 Open your image in Imaris and import the Allen Brain Atlas over it by selecting File -> Load Image.

16.3 The atlas can be freely rotated and translated using its frame that is generated when it is opened in Imaris. In the image frame corresponding to the Allen Brain Atlas, input 90° as the rotation factor and rotate and translate the Atlas until it is approximately aligned with your brain.

16.4 In the instance there is an angular mismatch, input 1° as the Atlas Rotation Factor and rotate it 1° at a time until it is angularly aligned with the brain image. Write down how many degrees it was moved in the X and Y planes.

16.5 In ABBA, input the number of X/Y rotation degrees with the rotation sliders beneath “Atlas Slicing” to rotate the Atlas into angular alignment with the brain image.

16.6 In the instance that Imaris is not available, rotation in X can be determined using symmetrical mismatches between your brain at the atlas. For example, if the right side of the brain has its hippocampus appear before the left side, the Allen Brain Atlas should be rotated to match this.

16.7 In Y, rotation can be determined by moving the atlas at until the base and top of the brain in fiducial regions appear the same in the brain image and the Atlas.

16.8 For example, if the base of the brain image where the hippocampus appears is significantly shorter than the atlas, rotate the atlas forward in Y until these regions appear similar. Note that perfect angle is more difficult to obtain this way and may need to be readjusted throughout alignment, which will also require reassigning key slices.

Continuing Z Align

17 Once approximate angular alignment is obtained, continue to create key-slices at points where the brain image fiducials match the atlas. Note that angular alignment may need to be reevaluated throughout, which will require moving key slices.

17.1 Other useful fiducials beyond the appearance of the hippocampus, in order in Z, are not limited to but include:

1. The appearance of the Olfactory Bulb.
2. The appearance of the cortex (Frontal Pole)
3. The appearance of the ventricles.
4. The appearance of the hippocampus
5. The point where the hippocampus begins to descend in the cortex.
6. The point the hippocampus fully descends and connects with the lower cortex.
7. The disappearance of the bright core of the hippocampus.
8. The full disappearance of the hippocampus.
9. The disappearance of the cortex.
10. The appearance of the cerebellum.
11. Various fidicuals throughout the cerebellum (Cerebellar layers appear in the center before enlarging like rings and disappearing. These layers can be matched).
12. The disappearance of the cerebellum.
13. Various other fiducials throughout, such as the descent of the hypothalamus, etc.

17.2 After fiducials have been aligned, select “Distribute Spacing” to allow ABBA to interpolate the rest of the Z-alignment. “Distribute Spacing” can be utilized throughout alignment to assess the quality of the Z-alignment.

17.3 *Tip: It may be the case that not all fiducials in Z can be aligned perfectly. For example, if the brain image has a rather large cortex that overlaps with the appearing cerebellum, whereas in the atlas, it disappears, there may be a few frames of slight mismatch. If this cannot be avoided, the more important region should be prioritized. Certain regions such as the hippocampus may take* *priority due to how they act as a major central anchor point for coregistration, but other regions can be instead prioritized if they are important for the experiment. In most cases, however, this will not be a major issue.*

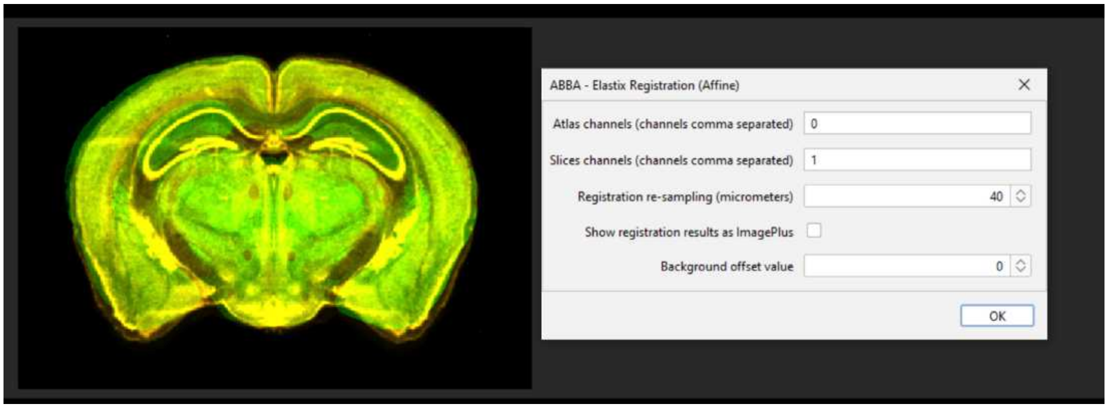
Aligning the Atlas in XY

18 Once Z-alignment has been completed, move on to XY alignment by selecting the “Review” mode (Z-alignment can be returned to by selecting the “Positioning” mode).

18.1 *Important: Please note that certain other FIJI plugins may prevent the automated transforms from working correctly. If this is encountered, please download a new version of FIJI and only install the ABBA update site using Help->Update…->Manage Update Sites-> Search for the update site “PTBIOP” and select it -> Apply and Close -> Apply Changes. Use this version of FIJI when running ABBA.*

18.2 First, find an approximate alignment by navigating to a central slice, using ctrl+a to select all slices, and selecting Align-> “ABBA - Interactive Transform”. Manually rotate, translate, and resize the brain slice in the X/Y dimensions until it is approximately aligned with the atlas.

18.3 At this point, the current slice may appear well-aligned, however the alignment will not be consistent throughout all slices. However, now that the brain is overall in near-alignment, an automated affine transform may be used to create a strong, initial aligned state for all planes. Select all slices with ctrl+a and use align -> “ABBA – Elastix Registration (affine)”, inputting the desired parameters. The affine transform will attempt to align all slices to the atlas, which may take several minutes (Fig. 13).



ABBA - Elastix Registration (Affine)

Atlas channels (channels comma separated) 0

Slices channels (channels comma separated) 1

Registration re-sampling (micrometers) 40

Show registration results as ImagePlus □

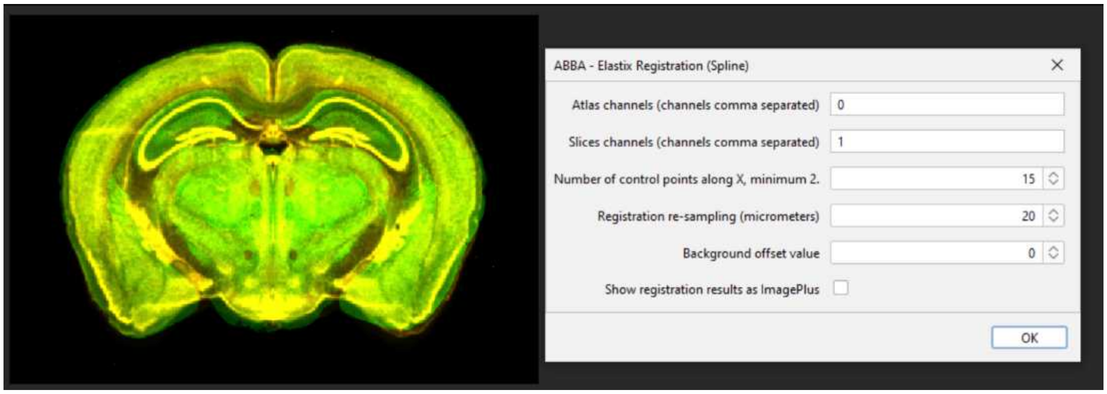
Background offset value 0

Fig. 13. Example settings for an automated affine transformation. An affine transform will translate, rotate, and warp the brain onto the atlas and is essentially an automated alternative to the interactive transform. However, it will not perfectly handle heterogenously shaped regions, for which a spline transform (step 18.5) may be necessary.

18.4 After completion, look through all slices to verify a decent initial alignment has been completed, comparing the before and after alignment by flipping between the previous and current registration. In certain cases, the affine alignment may not have worked well. If this occurs, remove the affine transform by selecting, then right-clicking the slices with the wrong registrations and selecting “remove last registration. For such slices, instead use the interactive transform option for a manual affine alignment.

18.5 Additionally, the brain imaged being clipped out of frame when originally imaged will make automated alignment in general fail, meaning these slices will likely require manual alignment alone. If lone manual alignment is utilized, one fast way to complete it is to select several similarly-sized, nearby slices and preform an interactive transform. Oftentimes, an interactive transform will work well for neighboring slices even when they were not the reference point. Use this method for several brain slices at once until the entire brain is approximately aligned.

18.6 After the affine transform, a closer-alignment may be achieved using a spline transform. Select all slices and use Align -> “ABBA - Elastix Registration (Spline)”, inputting the desired parameters (Fig. 14).



ABBA - Elastix Registration (Spline)

Atlas channels (channels comma separated) 0

Slices channels (channels comma separated) 1

Number of control points along X, minimum 2. 15

Registration re-sampling (micrometers) 20

Background offset value 0

Show registration results as ImagePlus □

Fig. 14. Example settings for an automated spline transform. A spline will differentially warp different parts of the image, allowing heterogenous regions to be stretched into the correct places.

18.7 In many cases, the spline is sufficient to create a near-perfect alignment. However, like before, look through all slices after the spline and remove any that resulted in a worse registration.

18.8 Alternatively, failed splines can be manually edited by selecting Align-> “ABBA - Edit Last Registration”, although this may be a time-consuming process. Additionally, splines may be manually generated from scratch using Align -> “ABBA - BigWarp Registration”, although this is also time consuming and usually unnecessary.

18.9 It is often the case that a decent coregistration usable for analysis can be generated regardless if the spline is successful, however a successful spline will lead to a near-perfect coregistration.

18.10 This manual alignment -> Affine -> Spline transform, with corrections throughout, is a general all-purpose method to accurately coregister the brain, however the manual, affine, and spline transforms may be repeated as needed.

18.11 *Reminder: As stated in step 18.1, certain other FIJI plugins may result in the automated transforms not completing. Please download a new version of FIJI with just the ABBA update site if this is the case.*

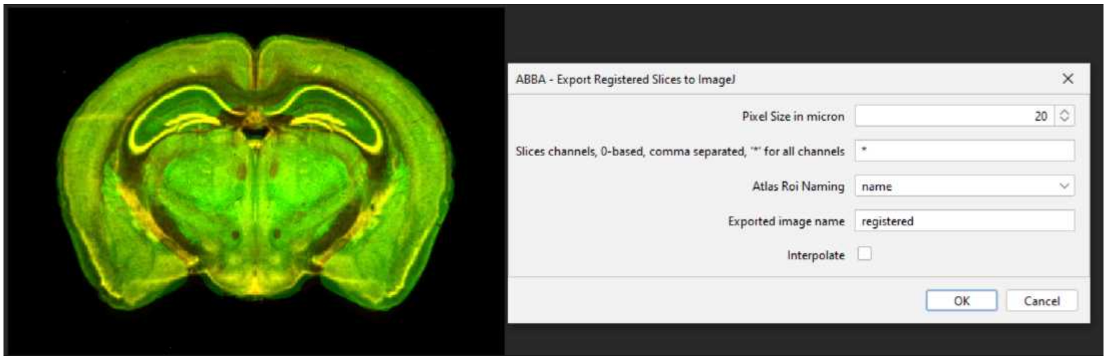
18.12 Finally, the current alignment in ABBA may be saved and reloaded if coregistration cannot be completed in a single sitting using File->Save Sate (Or Load State).

Exporting the Coregistered Brain

19 At this point, the coregistered brain can be exported (Fig. 15). There are various export options, however to use the below analysis pipeline, select the desired slices to export (or ctrl+a to export all slices) and select Export -> “ABBA - Export Registered Slices to ImageJ”-> and choose the desired naming format under Atlas ROI Naming, such as “name” for full names.

19.1 Input * under “Slices channels” to export all channels in one multi-channel image.

19.2 Finally, designate the Pixel Size in microns. The default value of 20 microns per pixel is generally sufficient. The coregistered brain will be exported to FIJI, which may take around ten minutes. Once the export is finished, please save it as a tiff in FIJI.



ABBA - Export Registered Slices to ImageJ

Pixel Size in micron: 20

Slices channels, 0-based, comma separated, '*' for all channels: *

Atlas Roi Naming: name

Exported image name: registered

Interpolate □

Fig. 15. Example settings for exporting the coregisetered brain. Please note the are several options for exporting the coregistered brain, but this pipeline will be for the settings specifically shown here.

Post Coregistration Analysis

20 **Interacting with the Coregistered Brain in FIJI**

The brain image will now have been warped onto the Atlas. This exported image will have 20 µm voxel-sizes in both X and Y (Assuming the default pixel size parameters were not changed), while the Z-voxel size will be inconsistent and not utilizable as a measurement, although that will not effect this analysis pipeline. The regions will be stored in the image as an overlay. The regions may be interacted with using Image->Overlay. Use “List Elements” to list all regions. Double clicking a region in the Elements window will bring you to said region. Use “To ROI Manager” to convert the Overlay into ROIs that may be used for quantification or transferring the overlay onto segmented images.

Segmenting the Brain for Cells or Other Objects of Interest with Machine Learning

21 This section describes how to create a binary mask for cells in the coregistered brain data, which will be used for region-specific quantification. A binary mask reduces an image to only black pixels representing background or white pixels representing signal which enables computational analysis. This pipeline is intended for cells, but theoretically other stained structures such as axons could be segmented and analyzed with these steps as well.

21.1 All analysis steps beyond this point must be completed on separate channels individually. In the coregistered brain, separate channels in FIJI using Image->Color->Split Channels (Fig. 16), then saving each as a tiff.

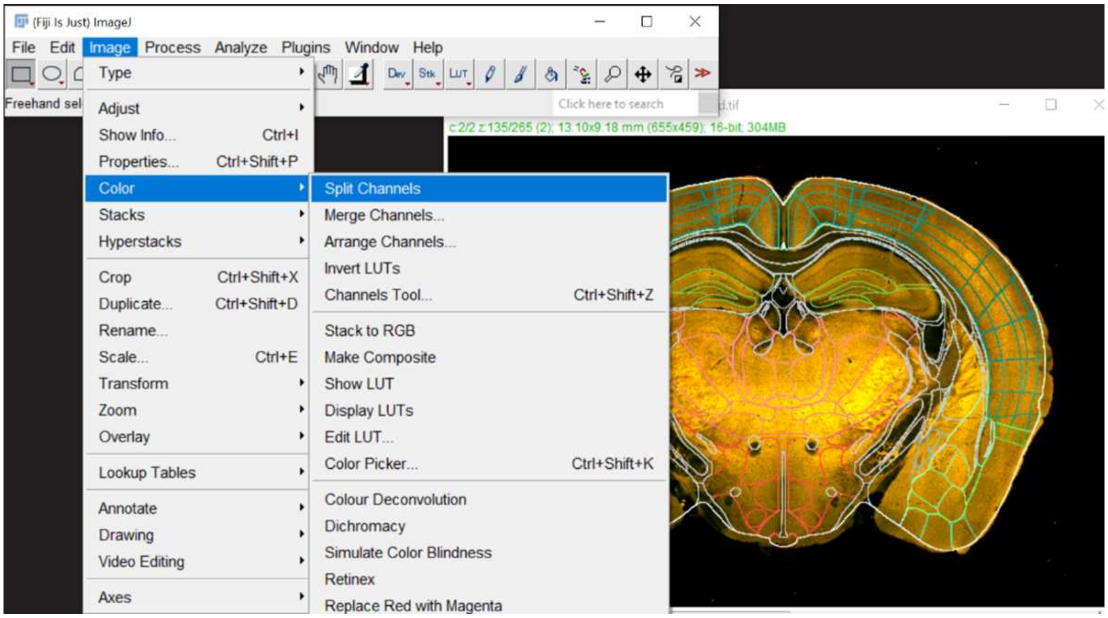


Fig. 16. Once again, split the channels of the coregistered brain for individual processing.

21.2 To obtain cell count information for all regions, we will use the FIJI plugin labkit[4], an intuitive ML-aided segmentation tool. Labkit will come pre-installed with all current FIJI downloads. *A detailed tutorial for labkit may be found here: https://www.youtube.com/watch?*

v=KopmsnC8GWI. ***Please also reference the following steps if this tutorial is utilized, as these steps contain information necessary to this specific analysis pipeline:***

21.3 To train a segmentation model, open the channel for segmentation in FIJI and select Plugins->Labkit->Open Current Image with labkit (Fig. 17). For some reason, the first time this is called, the overlayed image will take a long time to open in labkit, however this can be circumvented by simply repeating this step, which will immediately open the channel in a new labkit window.

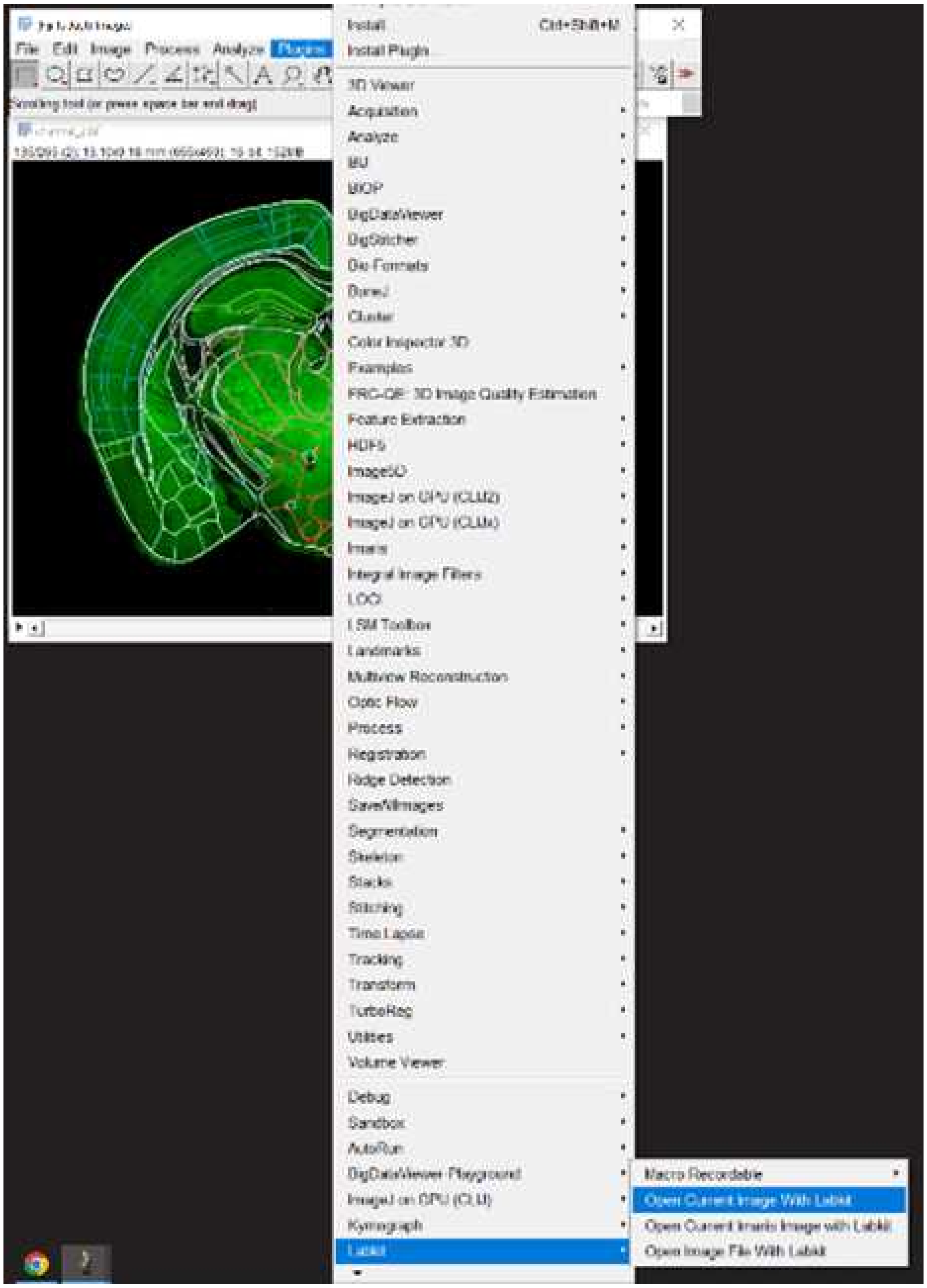


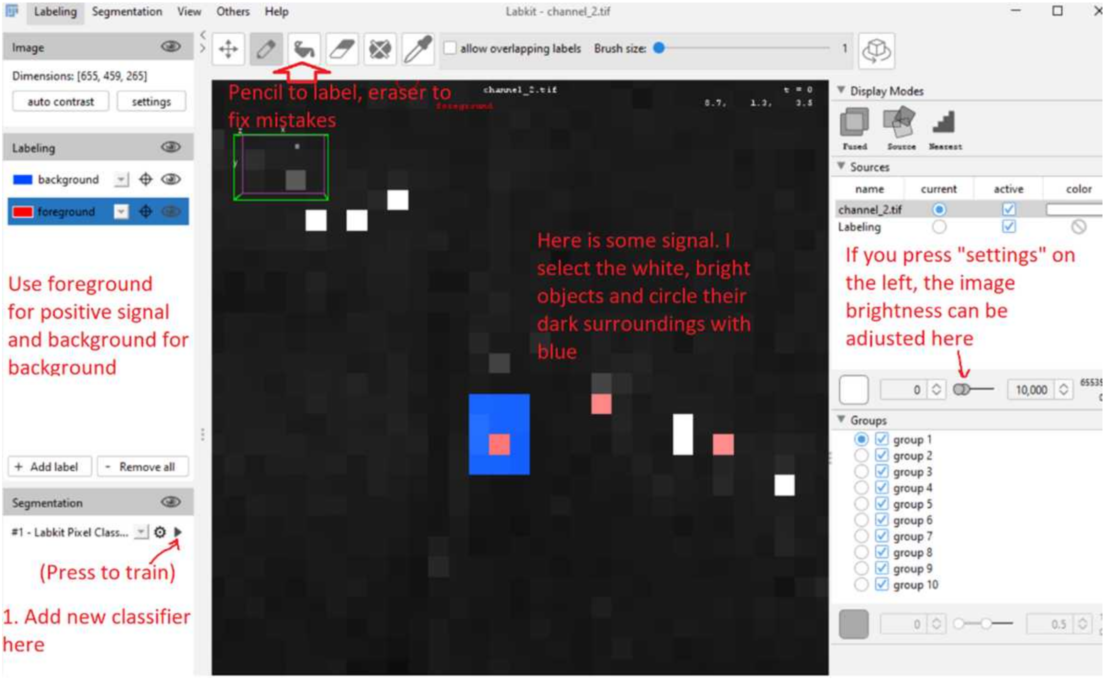
Fig. 17. How to open the image with labkit. Please note you may need to do this twice, as the first time may take very long due to the overlay. If labkit's opening command was executed twice, the original command may parse and open a second window of labkit some ten minutes later. This other window can be closed if this happens.

21.4 In Labkit, select “Labkit Pixel Classification” to begin training a new model. Use the pencil tool to label regions of positive signal with the red “positive” label, and regions of background with the blue “background” label.

21.5 Once a few examples have been labelled, train the model by pressing the arrow button under the Segmentation window. After a few seconds of training, labkit will begin attempting to segment the image based on the training parameters.

21.6 Use the pencil tool to correct any errors it makes while segmenting, such as applying the blue "background" label unto wrongly labelled signal, or the red "positive" label to correct wrongly missed positive signal, before retraining the model. Repeat this process until the segmentation seems satisfactory.

21.7 *Tip: While training in labkit, precise labelling should be used for better results. For example, if my cells look like white dots, I may select a white dot with red but circle its immediate surrounding pixels with blue, so the model knows to only select white dots and not nearby background. However, in areas with clumps of overlapping cells, I will highlight several interconnected white pixels with red so the model knows to select larger, fused signal if available. In the case of autofluorescent background that may appear similar to signal, I will trace them with blue in a way unique to only that autofluorescent region which cannot be construed with positive signal. For example, if the midline of the brain is highly autofluorescent, I will trace it all the way down with blue line in one stroke, so the model knows not to select very long, bright lines. If the edges of the brain are autofluorescent, I will trace the edge and the immediate background together with one stroke, so the model knows not to select bright regions that are immediately adjacent to very dark ones. An ideal model will select most positive signal and little background, prioritizing avoiding major false positives over selecting all signal.*



Pencil to label, eraser to fix mistakes

Use foreground for positive signal and background for background

(Press to train)

1. Add new classifier here

Here is some signal. I select the white, bright objects and circle their dark surroundings with blue

If you press "settings" on the left, the image brightness can be adjusted here

Fig. 18. Various instructions on using labkit to segment. Labkit offers a more-specific and still fast segmentation alternative to simply applying a mathematical algorithm or threshold, with machine-learning segmentation being generally superior.

21.8 Once the segmentation seems sufficient, select Segmentation -> “Calculate Entire Segmentation Result” to have labkit segment the entire image, which usually takes around a minute for images under 1 GB (Fig. 19). Scroll through the segmented image and correct any mislabeled regions before retraining the model and repeating the entire segmentation. Repeat this step until the segmentation is largely accurate at selecting only the positive signal.

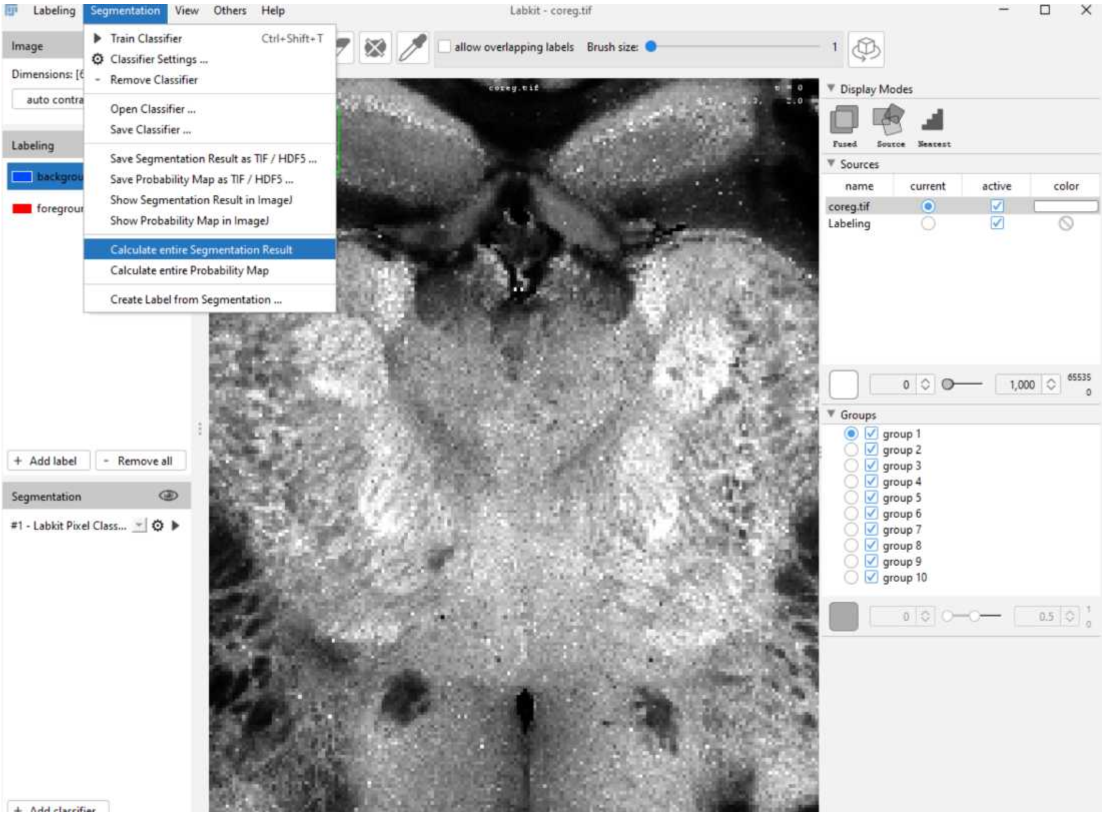


Fig. 19. Use "Calculate entire Segmentation Result" to have the model segment the entire image (may take a minute or longer if the file is large). Scroll through the segmented result and correct any errors, repeating this step until satisfactory.

21.9 Once the overall segmentation is satisfactory, select Segmentation -> "Show Segmentation Result in ImageJ" to open the mask as a new channel in FIJI. (Fig. 20).

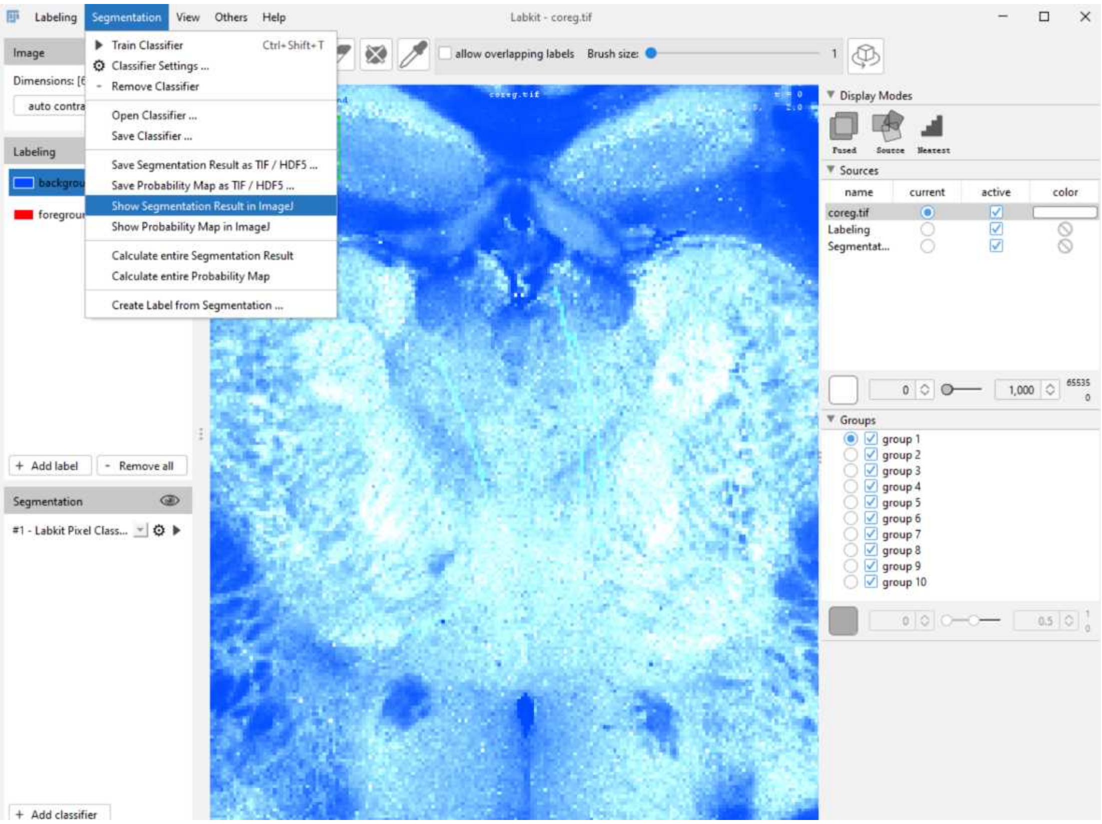
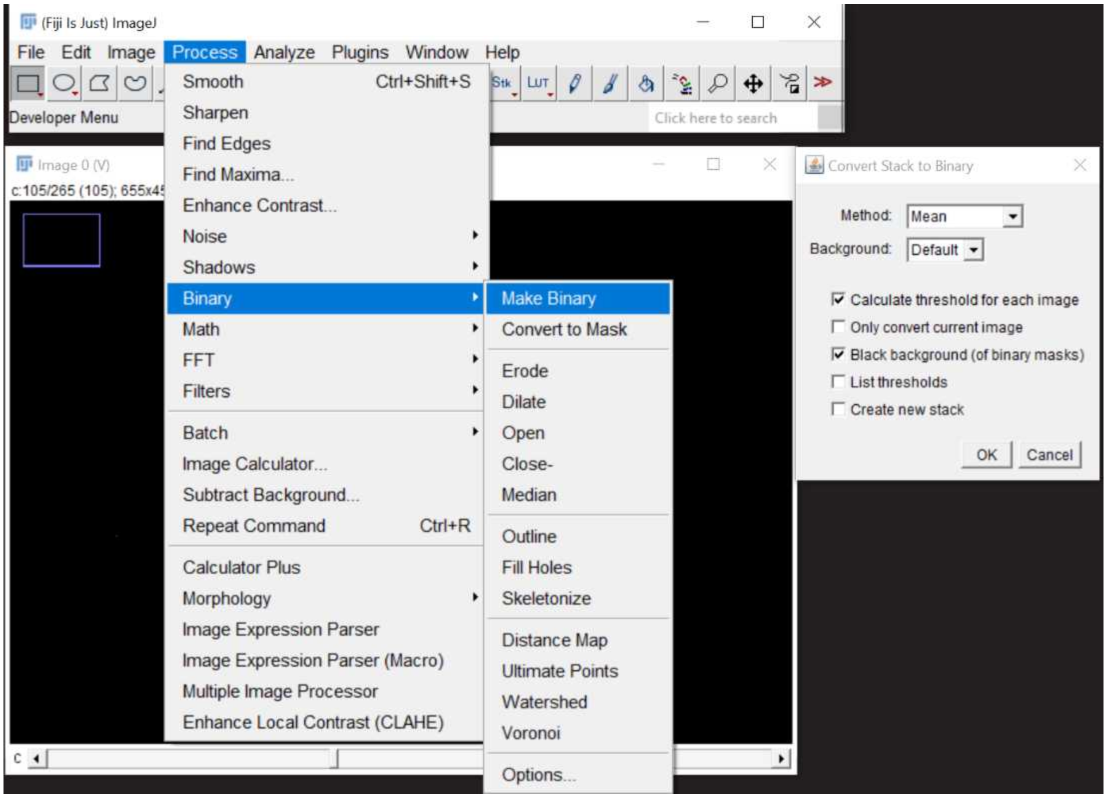


Fig. 20. Once the segmented image is largely accurate, export it to FIJI for a few final steps.

21.10 Before saving, first binarize the image by selecting Processing->Binary->”Make Binary”, using the “Mean” method (Fig. 21). This is a necessary step to ensure cell data can be calculated.



Convert Stack to Binary

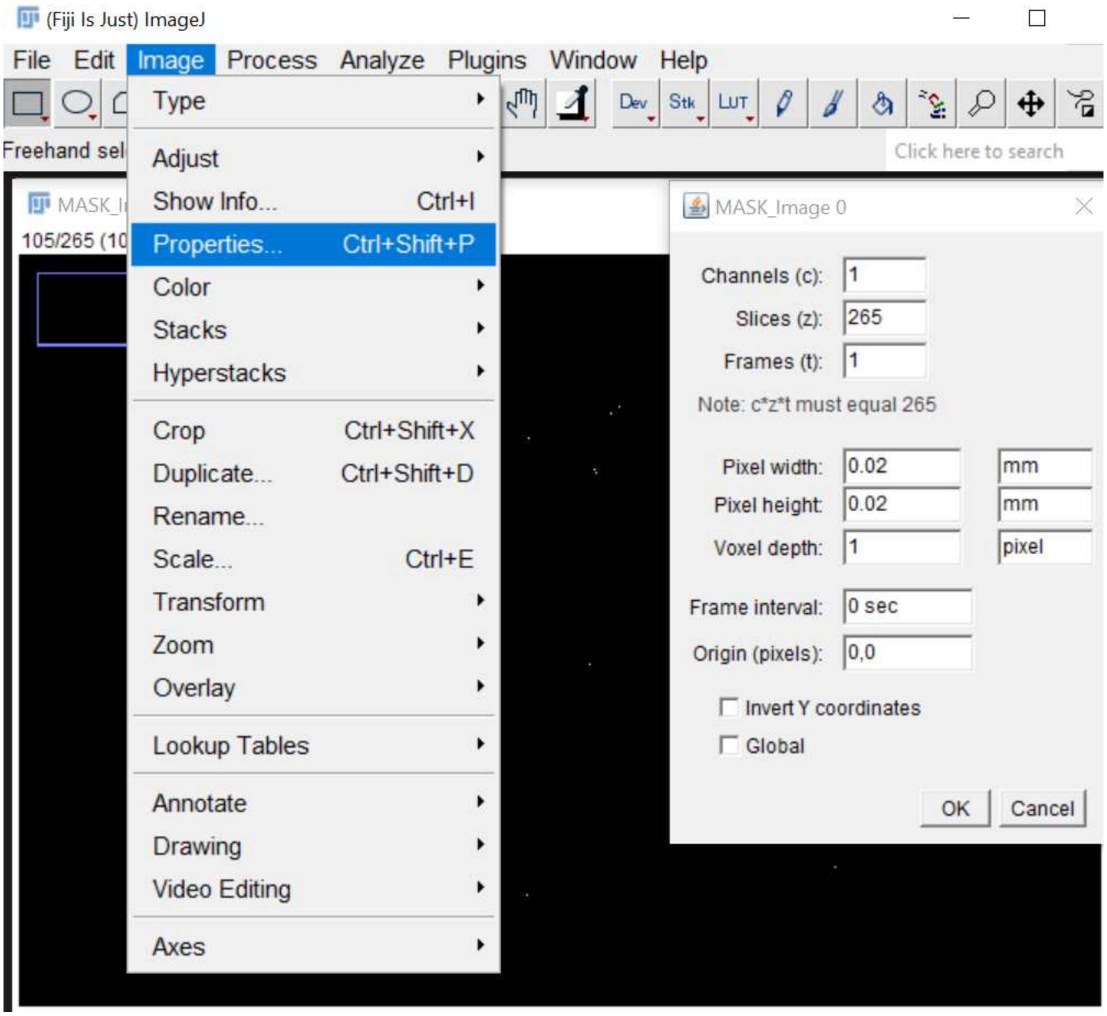
Method: Mean

Background: Default

☑ Calculate threshold for each image
□ Only convert current image
☑ Black background (of binary masks)
□ List thresholds
□ Create new stack

Fig. 21. Binarize the segmentation with the "Mean" method. This ensures the automated region analysis will work correctly.

Additionally, select Image->Properties and correct the voxel dimensions so that the X and Y dimensions contain the value 0.02 mm (mm will be added under “unit”; this is assuming the exported data from ABBA had the default 20 micron/pixel value. If this was changed, adjust the dimensions accordingly). Furthermore, labkit exports to ImageJ will often move their "slices" value into their "channels" value. This can also be corrected in Image-> Properties. If the "Channels (c)" is equivalent to the number of slices, cut this value and paste it in "Slices (z)". Next, replace "Channels (c)" with "1". Save the binarized, corrected image as a new segmentation mask (Fig. 22).



Channels (c): 1

Slices (z): 265

Frames (t): 1

Note: c*z*t must equal 265

Pixel width: 0.02 mm

Pixel height: 0.02 mm

Voxel depth: 1 pixel

Frame interval: 0 sec

Origin (pixels): 0,0

□ Invert Y coordinates

□ Global

OK Cancel

Fig. 22. Correct the voxel sizes. Width and height should be 0.02 mm each. Due to the interpolation in Z, the "Voxel Depth" will now be an inconsistent step size and cannot be used for analytical purposes. This value can be replaced with the step size of the original image (multiplied by the reslice factor used in step 11.3) for convenience. Additionally, correct the "Slices (z)". If the number of slices has become "1" due to switching values with the "Channels (c)" value, switch these values back.

21.11 This mask will be used to make quantifications. To align the mask with the coregistered overlay, open both the mask and the coregistered channel it came from together in FIJI. Select the coregistered channel and choose Image->Overlay->To ROI Manager (Fig. 23). Close the coregistered channel. Next, select the segmented mask and choose Image->Overlay->From ROI Manager (Fig. 24), which will import the overlay onto the segmentation. Save this segmented volume with the coregistration. This step is necessary, as the ROI overlay must be present in the segmentation for any analysis.

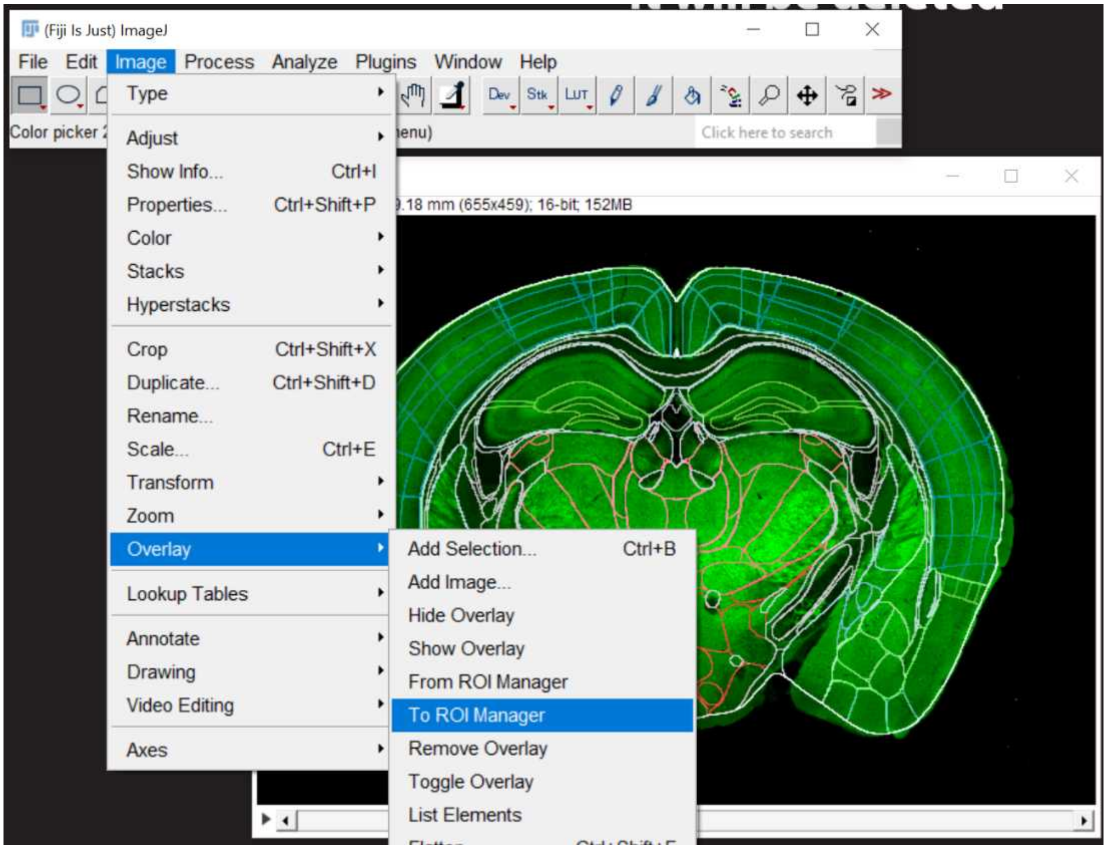


Fig. 23. To get the coregistration overlay on top of the segmentation, first move it off the coregistered brain and into the ROI manager.

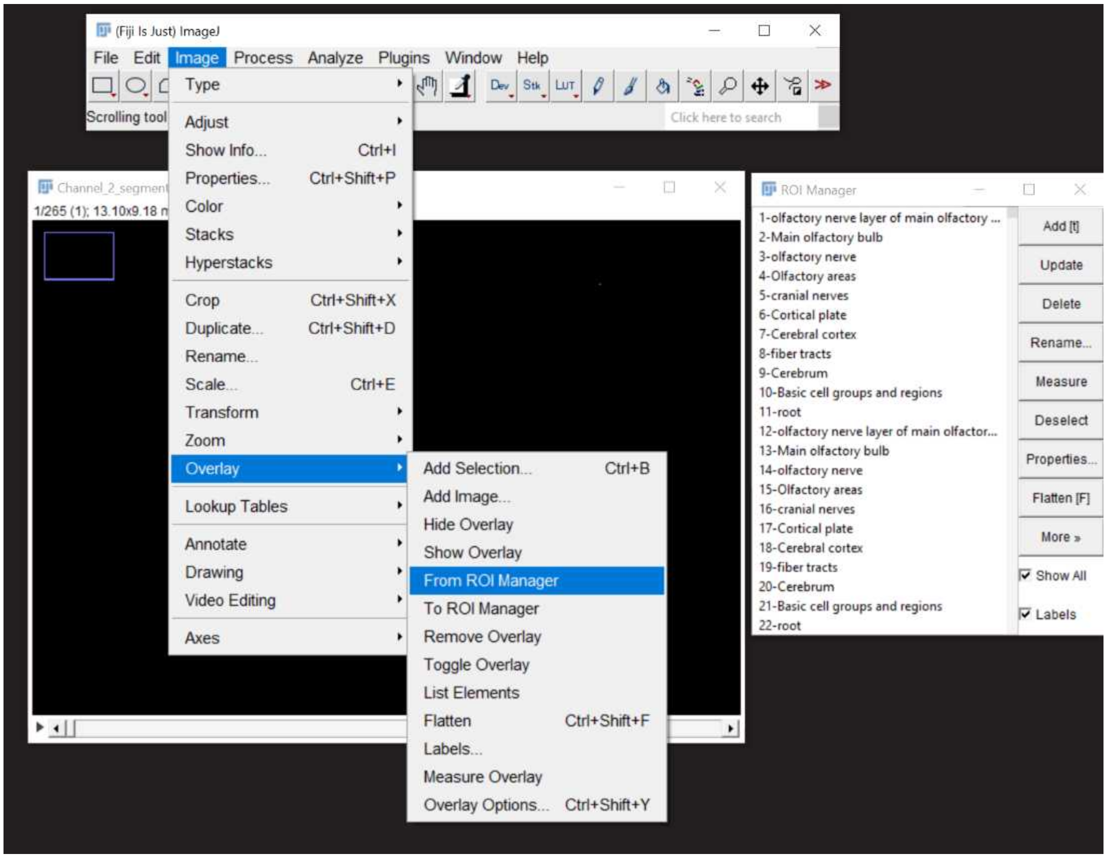


Fig. 24. Next, move the ROIs from the ROI manager onto the segmentation and save this overlayed segmentation. This file will be used for quantifications. The overlay must be present to identify regions.

Quantifying and Saving the Data from the Coregistered Segmentation as a .csv File

22 The following step is accomplished using my cells_macro available at my github: https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git. First, save the coregistered segmentation as an Image Sequence in a new folder using File->Save As->Image Sequence, specifying Tiff (Fig. 25).

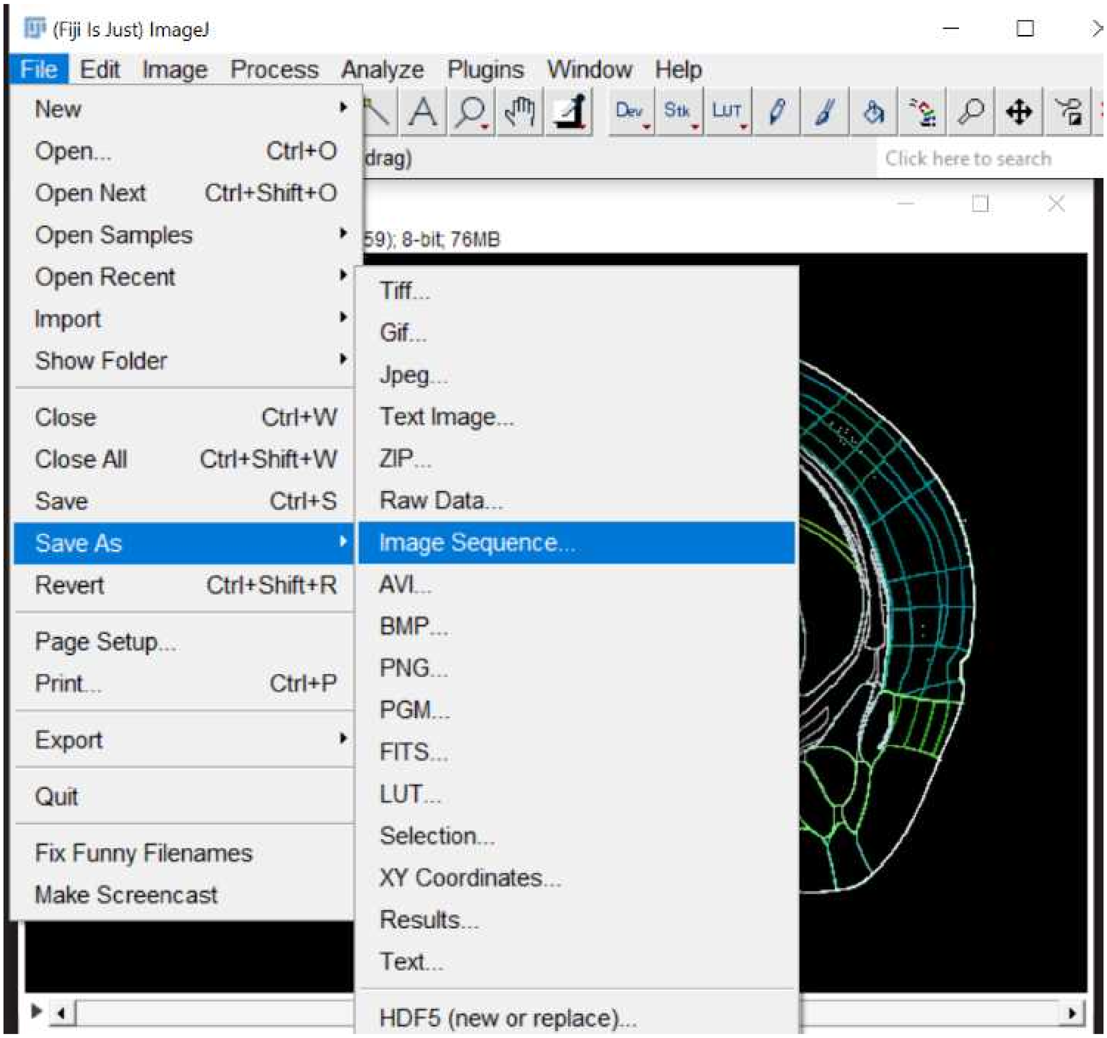
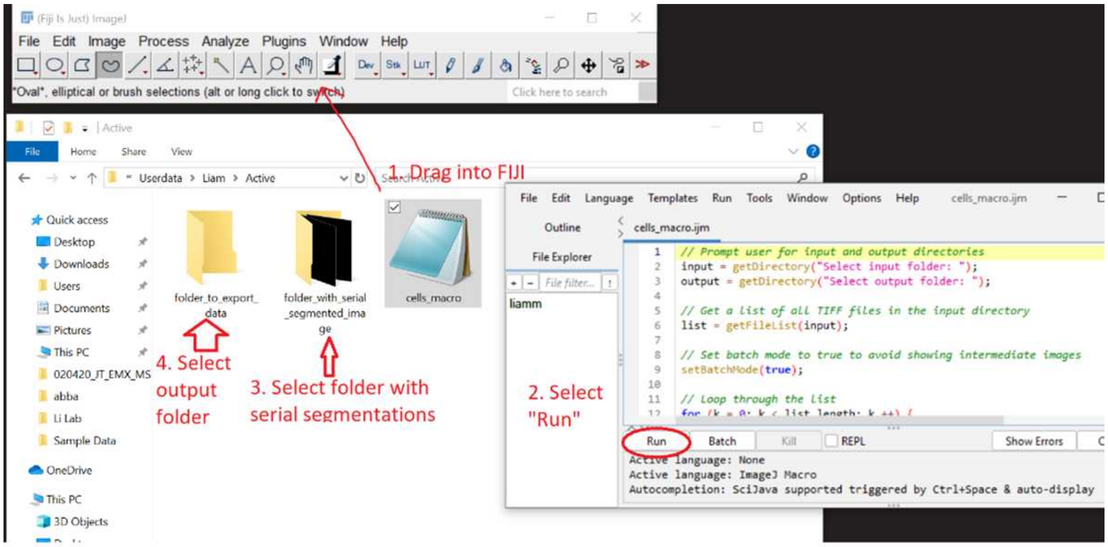


Fig. 25. The segmented, overlayed image must be saved in a new folder as an image sequence to enable automated analysis.

22.1 Create another empty folder and open the cells macro in FIJI. Select Run, then select the folder containing the coregistered segmentation Image Sequence, then the empty folder to save the data to. FIJI will then execute an analysis pipeline contained in the macro on all the images in the folder, saving the data to the empty folder (Fig. 26). This may take a few minutes.

Note: This script will not run if there are any slices missing overlays (signaled by the script crashing along with an error message about missing overlays), which can occur if slices from the imaged brain that were that existed beyond that start of the atlas's olfactory bulbs were exported from ABBA. To circumvent this, either only export slices from ABBA that are aligned to atlas slices, or simply delete any overlay-empty slices from the folder containing the coregistered slices.

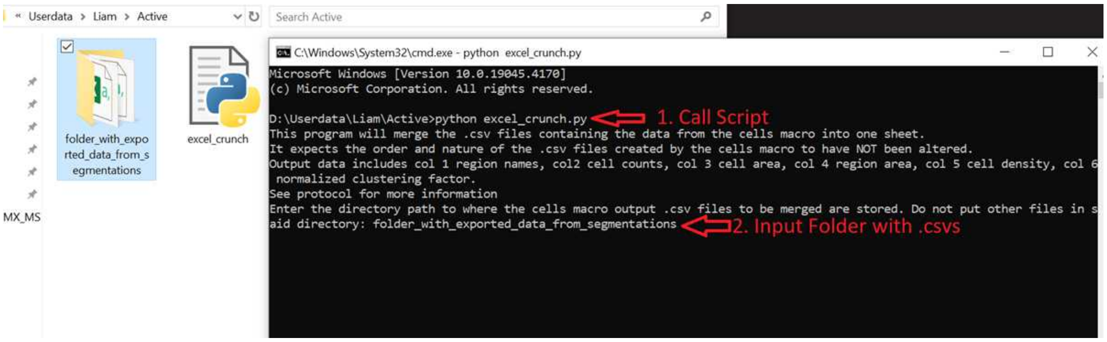


```
// Prompt user for input and output directories
input = getDirectory("Select input folder: ");
output = getDirectory("Select output folder: ");

// Get a list of all TIFF files in the input directory
list = getFileList(input);

// Set batch mode to true to avoid showing intermediate images
setBatchMode(true);

// Loop through the list
for (k = 0; k < list.length; k ++) {
```

```
Active language: None
Active language: ImageJ Macro
Autocompletion: SciJava supported triggered by Ctrl+Space & auto-display
```

Fig. 26. Run the cells macro to get automated results on all 800+ regions. First, select the folder with the segmented, serial overlayed slices. Next, select an empty folder to save the data in. This folder must be empty as the order of the output .csv files is relevant for further processing. Note this macro will fail on slices with no overlay (which may exist before the olfactory bulb). Please delete these before running the macro.

22.2 Due to the constraints of the FIJI macro language, this data will be separated into many different .csv files for each slice. However, the data in all the .csv files may be merged into one using the “excel_crunch” python script available at my github: https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git. Simply open the command prompt in the directory containing the folder that has the .csv files. In the command prompt, type “python excel_crunch.py”. When prompted, input the name of the .csv file folder. A combined excel file for the entire brain will then be created (Fig. 27).


Userdata > Liam > Active
Search Active
MX_MS
folder_with_exported_data_from_segmentations
excel_crunch
C:\Windows\System32\cmd.exe - python excel_crunch.py
1. Call Script
2. Input Folder with .csvs

```
Microsoft Windows [Version 10.0.19045.4170]
(c) Microsoft Corporation. All rights reserved.

D:\Userdata\Liam\Active>python excel_crunch.py
This program will merge the .csv files containing the data from the cells macro into one sheet.
It expects the order and nature of the .csv files created by the cells macro to have NOT been altered.
Output data includes col 1 region names, col2 cell counts, col 3 cell area, col 4 region area, col 5 cell density, col 6
 normalized clustering factor.
See protocol for more information
Enter the directory path to where the cells macro output .csv files to be merged are stored. Do not put other files in s
aid directory: folder_with_exported_data_from_segmentations
```

Fig. 27. Use the "excel_crunch.py" script to merge all the .csv files from the cells_macro into one file. Please note that this script expects only the .csv output from the cells_macro in the order it was created to exist in the input folder. If there are other files in said folder, or the .csv outputs of the cells_macro have been manipulated in any way, this script may not run correctly.

22.3 The combined excel file will contain 6 columns with the following information: The first column will contain the names of all the regions in the brain. The second column will contain an estimated cell count per region based on the FIJI "Analyze Particles" algorithm. This value may be unreliable on samples with lots of overlapping cells, as the algorithm itself rates each cell based on them being distinct binary objects in space. Therefore, this value should not be used beyond general appraisal in samples with lots of overlapping cells. The third column will contain the size of all segmented signal in said region, in square mm. This value represents a more reliable measurement than cell counts in the case of overlapping cells. The fourth column will contain the size of said region itself, in square mm. The fifth column will contain the density of positive signal in said region, obtained by dividing the third value by the fourth value.

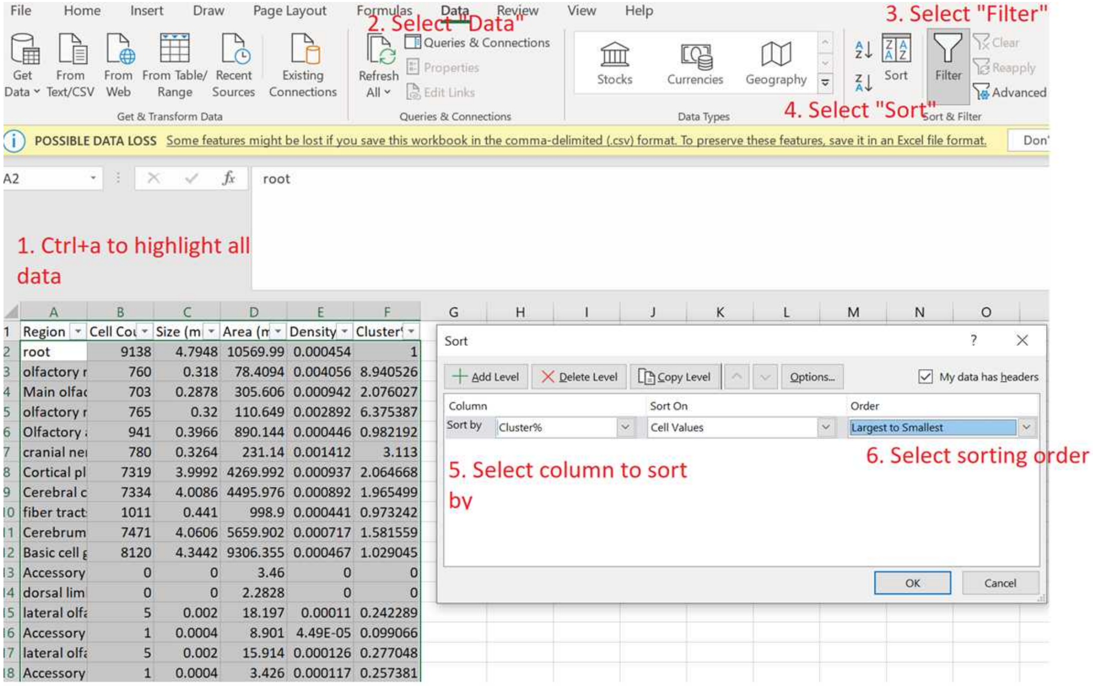
The sixth column is a normalized clustering factor, called Cluster%, that can be used for robust comparison between samples that circumvents potential confounding factors caused by differences in image quality or segmentation. The Cluster% value is obtained by dividing the cell density in a region by the cell density of its entire brain, providing a description of how many times more likely a cell is to cluster in that region as compared a random location in the brain. For example, a value of three for "hippocampus" means cells are three times as dense in the hippocampus then they are in the entire brain. This is a useful statistic for describing where cells generally are in the brain, and can be safely compared between different images, channels, and cohorts.

Note: I strongly recommend comparing data with the Cluster% value to see where cells are in the brain, as it is by far a more robust statistic than the others. Only compare other values such as "sizes" or "densities" to investigate if cell counts have globally increased or decreased.

Analyzing the Data and Establishing Significance

23 At this stage, the data for one channel detailing region-specific information will be ready for analysis. Before manipulating any of the .csv files, create a backup copy, as the original state of the data and the .csv format may be relevant for additional analytical steps.

24 The data within the .csv file can be easily organized with Microsoft Excel. Open the file in Excel. To sort regions (such as high to low density/cluster%, or by region size, etc), press ctrl+a to select all columns, then select Data->Toggle the "Filter" option on-> Sort. In the sort window, select the column to sort by and the order to sort the columns by (such as highest to lowest). This will sort all columns according to these specifications, allowing an easy way to browse the data based on magnitude (*Fig. 28)*. Within Excel, graphs and batch calculations within and between regions may be easily generated.



1. Ctrl+a to highlight all data

2. Select "Data"

3. Select "Filter"

4. Select "Sort"

5. Select column to sort by

6. Select sorting order

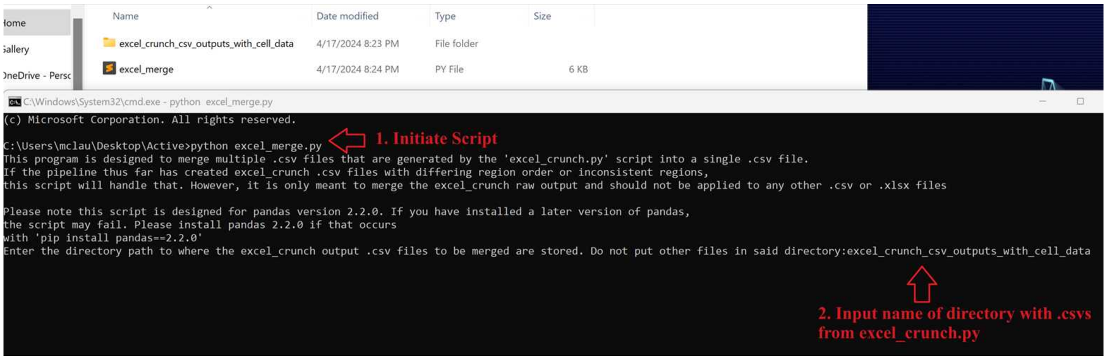
	A	B	C	D	E	F
1	Region	Cell Cou	Size (m	Area (m	Density	Cluster%
2	root	9138	4.7948	10569.99	0.000454	1
3	olfactory r	760	0.318	78.4094	0.004056	8.940526
4	Main olfac	703	0.2878	305.606	0.000942	2.076027
5	olfactory r	765	0.32	110.649	0.002892	6.375387
6	Olfactory a	941	0.3966	890.144	0.000446	0.982192
7	cranial ne	780	0.3264	231.14	0.001412	3.113
8	Cortical pl	7319	3.9992	4269.992	0.000937	2.064668
9	Cerebral c	7334	4.0086	4495.976	0.000892	1.965499
10	fiber tract	1011	0.441	998.9	0.000441	0.973242
11	Cerebrum	7471	4.0606	5659.902	0.000717	1.581559
12	Basic cell g	8120	4.3442	9306.355	0.000467	1.029045
13	Accessory	0	0	3.46	0	0
14	dorsal lim	0	0	2.2828	0	0
15	lateral olfa	5	0.002	18.197	0.00011	0.242289
16	Accessory	1	0.0004	8.901	4.49E-05	0.099066
17	lateral olfa	5	0.002	15.914	0.000126	0.277048
18	Accessory	1	0.0004	3.426	0.000117	0.257381

Column	Sort On	Order
Sort by Cluster%	Cell Values	Largest to Smallest

Fig. 28. Please use this method to sort all the data with reference to one column. This will allow regions to be sorted while all region-specific information is preserved.

24.1 In the case there are several datasets that should be merged into a single, large excel file, my python script “excel_merge”, available at https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git, will combine multiple .csv files that were created via the above steps into a single file. Take all .csv files which were combined from the "cells_macro" output by "excel_crunch" and place them into a new, empty folder. In the higher folder containing the folder with the .csvs, run excel_merge from the command terminal and identify the target folder (Fig. 29). A combined .csv file will be generated. Please note this script may not work in versions of pandas (the python module that works with excel documents) beyond 2.2.0.

As this script was specifically designed to work for this pipeline, only run it on multiple files generated as .csv output from the “excel_crunch” script. Importantly, excel_merge is designed to handle files that do not have their regions in the same order, or are missing regions. When combining all the files together, it preserves the necessary region-specific information across multiple samples, even when certain samples are missing regions or in a different order. This may be crucial in more heterogenous or angled brain images that cause the FIJI cells_macro to encounter regions in different orders while iterating through the slices, where information would not be properly transferred by simply copy and pasting all the data to one excel file.



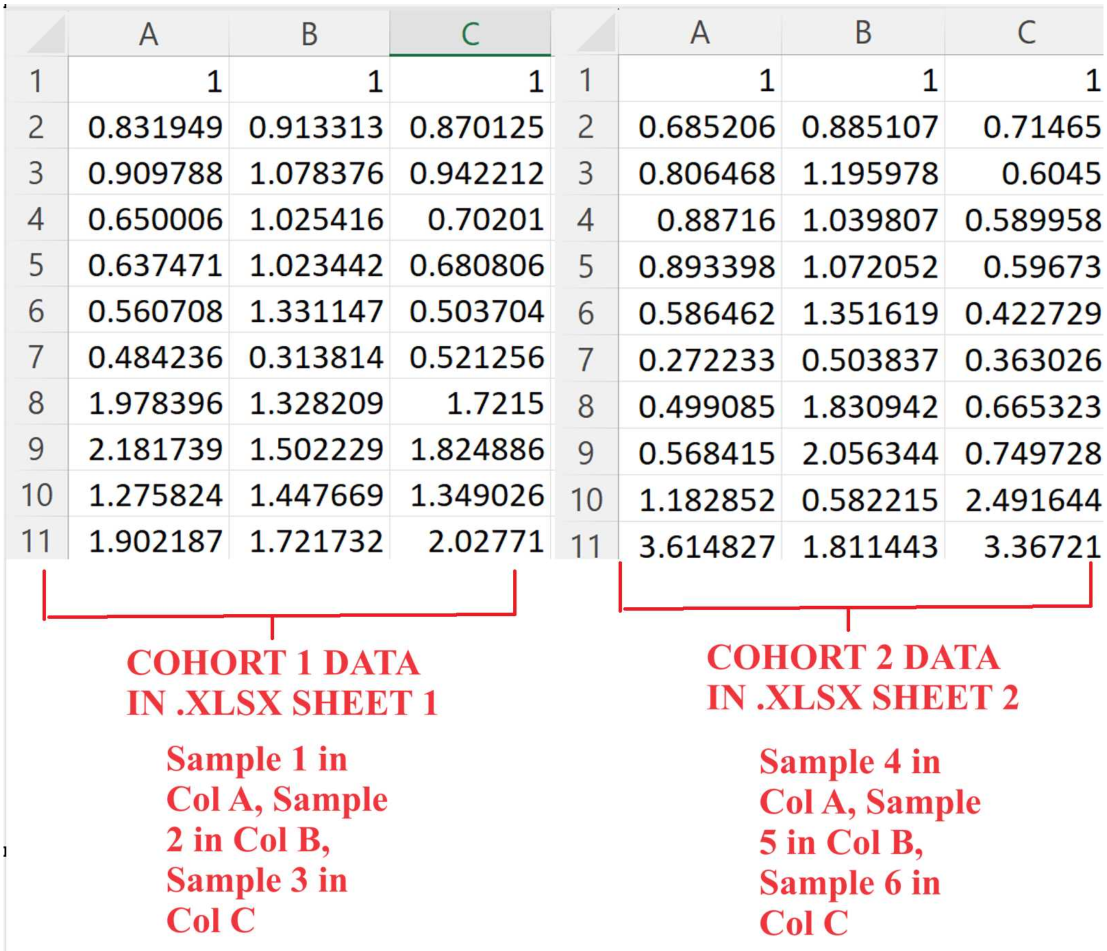
```
(c) Microsoft Corporation. All rights reserved.

C:\Users\mclau\Desktop\Active>python excel_merge.py
This program is designed to merge multiple .csv files that are generated by the 'excel_crunch.py' script into a single .csv file.
If the pipeline thus far has created excel_crunch .csv files with differing region order or inconsistent regions,
this script will handle that. However, it is only meant to merge the excel_crunch raw output and should not be applied to any other .csv or .xlsx files

Please note this script is designed for pandas version 2.2.0. If you have installed a later version of pandas,
the script may fail. Please install pandas 2.2.0 if that occurs
with 'pip install pandas==2.2.0'
Enter the directory path to where the excel_crunch output .csv files to be merged are stored. Do not put other files in said directory:excel_crunch_csv_outputs_with_cell_data
```

Fig. 29. How to merge multiple .csv files from different samples (generated by the cells_macro followed by excel_crunch) using excel_merge. Please note that there will be some deprecation errors thrown, which indicate this script may not run in future versions of pandas beyond 2.2.0. If the script will not run, please revert pandas to version 2.2.0.

24.2 The order of regions in the aggregate file generated by “excel_merge” represents the master order of regions in the study. From this point on, any manipulation to the order of rows within excel should be done on a new sheet, preserving the master order, as it may be relevant to future analysis. In the excel_merge output, data from all samples included will correspond to the same region within a row, allowing graphs and batch calculations between samples within excel.

24.3 To establish significance between two aggregated cohorts (which can be between two different channels within one group of mice or one channel between two different groups of mice, etc), the stat_test scripts at my github (https://github.com/mclaughlinliam3/ABBA_Coregistration_Analysis_Tools.git) can be used.

24.4 Organize each cohort into its own excel file by copy and pasting each row from the first cohort into one excel file, then copy and pasting each column for significance testing from the other cohort to a second excel file (Fig. 30). **These files should be in the .xlsx format.** For example, if I wanted to compare the Cluster% value across two cohorts of different ages to see if cells have changed their relative locations between the sample groups, I will make an excel file containing just the Cluster% values from the young mouse group and a second excel file containing the Cluster% values from an older mouse group. The values for a single group should be pasted side-by-side, without headers, in the same “master order” as established by the “excel_merge” script.



	A	B	C
1	1	1	1
2	0.831949	0.913313	0.870125
3	0.909788	1.078376	0.942212
4	0.650006	1.025416	0.70201
5	0.637471	1.023442	0.680806
6	0.560708	1.331147	0.503704
7	0.484236	0.313814	0.521256
8	1.978396	1.328209	1.7215
9	2.181739	1.502229	1.824886
10	1.275824	1.447669	1.349026
11	1.902187	1.721732	2.02771

	A	B	C
1	1	1	1
2	0.685206	0.885107	0.71465
3	0.806468	1.195978	0.6045
4	0.88716	1.039807	0.589958
5	0.893398	1.072052	0.59673
6	0.586462	1.351619	0.422729
7	0.272233	0.503837	0.363026
8	0.499085	1.830942	0.665323
9	0.568415	2.056344	0.749728
10	1.182852	0.582215	2.491644
11	3.614827	1.811443	3.36721

COHORT 1 DATA
IN .XLSX SHEET 1

Sample 1 in Col A, Sample 2 in Col B, Sample 3 in Col C

COHORT 2 DATA
IN .XLSX SHEET 2

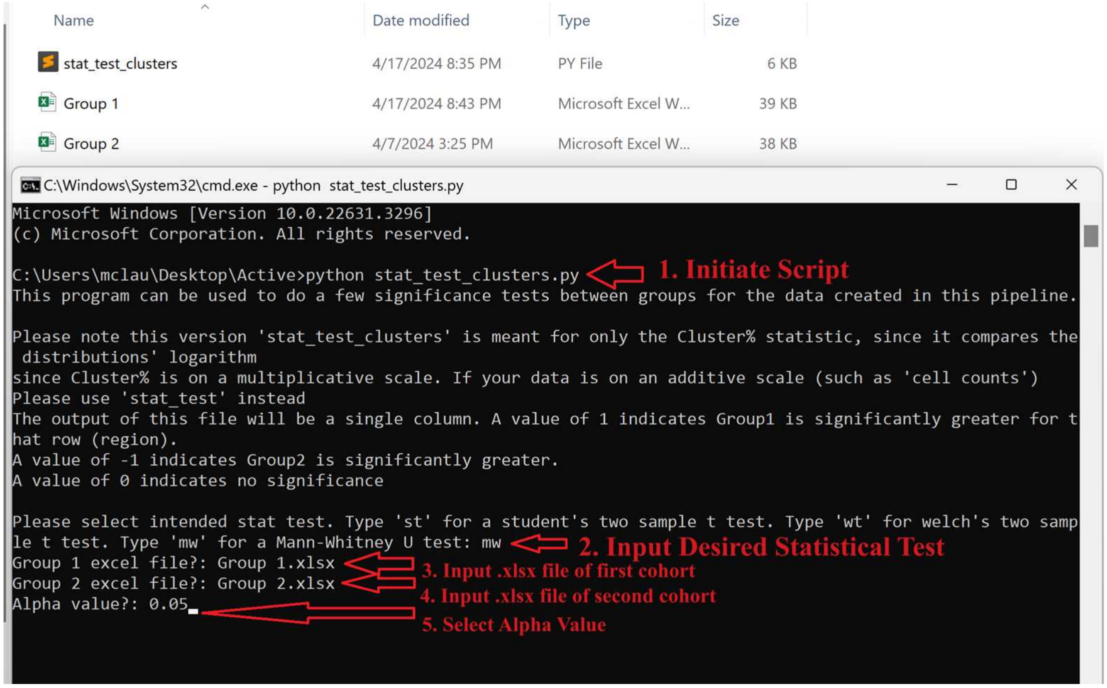
Sample 4 in Col A, Sample 5 in Col B, Sample 6 in Col C

Fig. 30. After multiple samples have been combined into one .csv file with "excel_merge.py", create two new .xlsx excel files for significance testing. Please copy the relevant data from one cohort into the first excel file. Do not use headers. Each column should represent its own sample. Please copy and paste this data ***in the exact order*** *as it exists in the combined .csv file created by "excel_merge.py" - the "master order". Do the same for a second cohort. These excel files are now ready for batch significance testing.*

24.5 Next, open the command prompt in the same folder containing the two excel files each containing data from separate cohorts and call the corresponding stat_test script. ***(Use stat_test_clusters.py when comparing the Cluster% statistic, which I recommend, and stat_test.py for the other values! See step 24.7 for more info).***

24.6 This script offers three tests for comparing two cohorts: A Mann-Whitney U test, a Student’s Two Sample t-test, and a Welch’s Two Sample t-test. Mann-Whitney U tends to preform superior to t at low sample sizes (n<10, for example) and makes no assumptions about the distribution’s shape. Welch’s t-test additionally makes no assumptions about the population variance and is generally more robust when Student’s t-test’s stringent assumptions of equivalent population variances cannot be met. Welch’s t-test is ideal for most instances of larger sample sizes, but Student’s t-test will offer slightly more power if its assumptions are met.[5]

Next, input the name of the Group1 excel file, then the Group2 excel file. Finally, input an alpha value, such as 0.05 (Fig. 31).



Name	Date modified	Type	Size
stat_test_clusters	4/17/2024 8:35 PM	PY File	6 KB
Group 1	4/17/2024 8:43 PM	Microsoft Excel W...	39 KB
Group 2	4/7/2024 3:25 PM	Microsoft Excel W...	38 KB

C:\Windows\System32\cmd.exe - python stat_test_clusters.py

```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mclau\Desktop\Active>python stat_test_clusters.py
This program can be used to do a few significance tests between groups for the data created in this pipeline.

Please note this version 'stat_test_clusters' is meant for only the Cluster% statistic, since it compares the
 distributions' logarithm
since Cluster% is on a multiplicative scale. If your data is on an additive scale (such as 'cell counts')
Please use 'stat_test' instead
The output of this file will be a single column. A value of 1 indicates Group1 is significantly greater for t
hat row (region).
A value of -1 indicates Group2 is significantly greater.
A value of 0 indicates no significance

Please select intended stat test. Type 'st' for a student's two sample t test. Type 'wt' for welch's two samp
le t test. Type 'mw' for a Mann-Whitney U test: mw
Group 1 excel file?: Group 1.xlsx
Group 2 excel file?: Group 2.xlsx
Alpha value?: 0.05
```

1. Initiate Script
2. Input Desired Statistical Test
3. Input .xlsx file of first cohort
4. Input .xlsx file of second cohort
5. Select Alpha Value

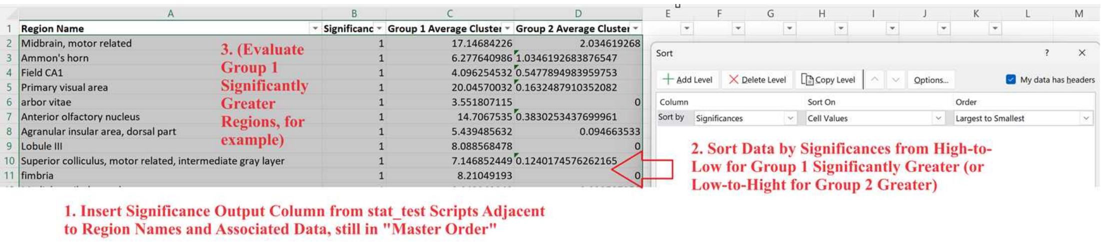
Fig. 31. Use the "stat_test_clusters.py" script to obtain signficances for all regions when comparing the Cluster% value. Use "stat_test.py" to obtain significances for all regions when comparing the other values.

24.7 *Note: **There are two versions of the "stat_test" script. The first, "stat_test_clusters" is designed specifically for the "Cluster%" value.** Unlike the default "stat_test", "stat_test_clusters" takes the log of values before comparing them. This is because Cluster% is a multiplicative distribution, where 0.33 represents an equivalent distance to 1 as 1 does to 3. Taking the log returns these values to an additive distribution, allowing direct statistical comparison. Please note that 0.1 is temporarily added to all values in "stat_test_clusters" prior to significance testing, since log(0) is undefined and requires the distribution to be shifted by some factor for analysis. Although adding a value such as 0.1 to a multiplicative distribution shifts values a non-standard amount after the log is taken, 0.1 is a small enough value as to make this shift negligible and is unlikely to affect significance results. However, this 0.1 value can be adjusted in the code as needed in the "list_logs" method. The default "stat_test" script does not utilize logs. Therefore, Cluster% should only be analyzed with "stat_test_clusters", while all other values should be analyzed with "stat_test".*

24.8 The test will output a new csv file with a single column containing values of 1, 0, or -1. A score of 1 represents a significant difference where Group1 was greater. A score of 0 represents no significant difference. A score of -1 represents a significant difference where Group 2 was greater.

24.9 To sort by significances, this column should be copy and pasted alongside a column containing the regions sorted into their “master order” (generated by the excel_merge script). In this sheet comparing significances, excel can sort the regions by the significances column from high to low with the same sorting steps described previously. This will shift all Group1 greater significance regions to the top of the excel file. Similarly, a low-to-high sort will shift all Group2 greater significance regions to the top of the excel file.

24.10 To ensure associated region-specific data is sorted as well, use the significance column to sort in a sheet containing the regions, significances, and any desired region-associated data starting from their “master order”. Like before, preform these manipulations in a new sheet so the master order is preserved. Doing this, all significant regions and associated data of interest can be easily sorted and exported into new excel sheets for specific calculations or graphical analysis (Fig. 32).



	Region Name	Significanc	Group 1 Average Cluster	Group 2 Average Cluster
2	Midbrain, motor related	1	17.14684226	2.034619268
3	Ammon's horn	1	6.277640986	1.0346192683876547
4	Field CA1	1	4.096254532	0.5477894983959753
5	Primary visual area	1	20.04570032	0.1632487910352082
6	arbor vitae	1	3.551807115	0
7	Anterior olfactory nucleus	1	14.7067535	0.3830253437699961
8	Agranular insular area, dorsal part	1	5.439485632	0.094663533
9	Lobule III	1	8.088568478	0
10	Superior colliculus, motor related, intermediate gray layer	1	7.146852449	0.1240174576262165
11	fimbria	1	8.21049193	0

Column	Sort On	Order
Sort by Significances	Cell Values	Largest to Smallest

3. (Evaluate Group 1 Significantly Greater Regions, for example)

2. Sort Data by Significances from High-to-Low for Group 1 Significantly Greater (or Low-to-Hight for Group 2 Greater)

1. Insert Significance Output Column from stat_test Scripts Adjacent to Region Names and Associated Data, still in "Master Order"

Fig. 32. Copy all significances into a new sheet in an excel file. Copy all regions, and any associated data, still in their "master order" that was generated by excel_merge, into the sheet adjacent to the significances column. Next, sort by the significances column high-to-low to sort Group1 significantly greater regions to the top, or low-to-high to sort Group2 significantly greater regions to the top. Again, these manipulations should happen in a new sheet, as the original "master order" should always be preserved.

24.11 An ideal use case of this analysis pipeline would be to use the Cluster% statistic to establish significant difference between where cells are in the brain. However, Cluster% cannot describe changes that are grossly the same between regions, such as a general reduction of cells in all regions of the brain, for which the density measurement may be preferred, although it may be confounded in the case of imaging and segmentation artifacts.

Acknowledgements

25 Coregistration was preformed with the ABBA platform, developed by Nicolas Chiaruttini and team. **https://biop.github.io/ijp-imagetoatlas/**

Data analysis were performed in part through the use of Washington University Center for Cellular Imaging (WUCCI) supported by Washington University School of Medicine, The Children’s

Discovery Institute of Washington University and St. Louis Children’s Hospital (CDI-CORE-2015-505 and CDI-CORE-2019-813) and the Foundation for Barnes-Jewish Hospital (3770 and 4642).

Example mouse brain datasets for the development of this pipeline were provided by Keran Yang, as well as by Dr. Tristan Li and Dvita Kapadia of Washington University School of Medicine.

References

26 1. N Chiaruttini, ABBA. **https://biop.github.io/ijp-imagetoatlas/**

2. Sunkin SM, Ng L, Lau C, Dolbeare T, Gilbert TL, Thompson CL, Hawrylycz M, Dang C. Allen Brain Atlas: an integrated spatio-temporal portal for exploring the central nervous system. Nucleic Acids Res. 2013 Jan;41(Database issue):D996-D1008. doi: 10.1093/nar/gks1042. Epub 2012 Nov 28. PMID: 23193282; PMCID: PMC3531093.

3. Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, Preibisch S, Rueden C, Saalfeld S, Schmid B, Tinevez JY, White DJ, Hartenstein V, Eliceiri K, Tomancak P, Cardona A. Fiji: an open-source platform for biological-image analysis. Nat Methods. 2012 Jun 28;9(7):676-82. doi: 10.1038/nmeth.2019. PMID: 22743772; PMCID: PMC3855844.

4. Arzt, M., Deschamps, J., Schmied, C., Pietzsch, T., Schmidt, D., Tomancak, P., … Jug, F. (2022). LABKIT: Labeling and Segmentation Toolkit for Big Image Data. Frontiers in Computer Science, 4. doi:10.3389/fcomp.2022.777728

5. West RM. Best practice in statistics: Use the Welch t-test when testing the difference between two groups. Annals of Clinical Biochemistry. 2021;58(4):267-269. doi:10.1177/0004563221992088