Regional Brain Analysis (QUINT) Overview

MXH_12.23.2021

Purpose

The purpose of this workflow is to enable mouse brain segmentation, registration and quantification of regional signal. The simplest segmentation is done in ***QuPath*** *because this program handles whole slide images and has good segmentation algorithms. Registration is done using 3 programs:* ***QuickNII*** *(aligns to a 3D atlas, typically the 2017 CCFv3 Allen Brain Atlas),* ***Visualign*** *(allows for non-linear warp transformation of the atlas to match sections),* ***QMask*** *(masks each side of the brain to allow for bilateral assessment of brain regions). Segmentations and registrations are then brought together in* ***Nutil,*** *enabling the generation of quantitative measures for every region of the brain.*

Necessary Programs and Locations

1. **QuPath:** https://qupath.github.io/
2. **QuickNII:** https://www.nitrc.org/projects/quicknii (SELECT ABA Mouse Edition)
3. **Visualign:** https://www.nitrc.org/projects/visualign/
4. **Qmask:** \\pn.vai.org\projects_secondary\henderson\NOAH\QMask-Standalone
5. **Nutil:** https://www.nitrc.org/projects/nutil/

   *****QuickNII, Visualign, and Nutil can be a bit finicky, and we have found it is best to have these folders on the desktop.

Scanned Slides

To enable high throughput, slides are typically scanned on the Aperio or Axioscan microscopes. After slide scanning images will be found at https://aperio.vai.org/Login.php AND ON THE SERVER AT \\pn.vai.org\projects_primary\henderson\vari-core-generated-data\PBC-Aperio. Images have a slide ID designation instead of the external case id. Manually update the names of each slide by copying the external case id from the website and changing the matched slide ID in our server folder.

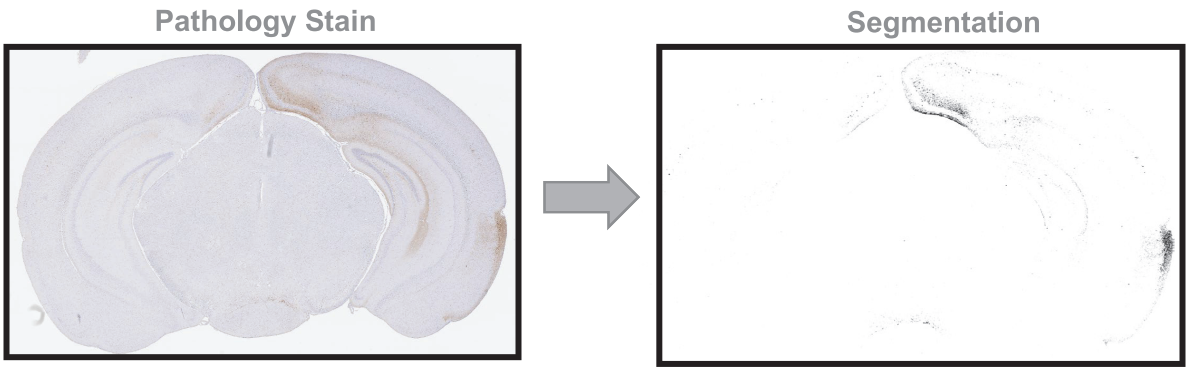
Folder Organization

The programs for registration and quantification rely on having exact file paths from which to call the data. Therefore, it is easiest to set these folders up from the beginning. We have the following typical layout for folders:

1. Project (e.g. Tau-MLi-2)
    a. Figure (e.g. Figure 23)
        i. Repeat (e.g. 1 or 2)
            1. Cohort/Block (e.g. TM1)
                a. QuPath (place for qupath project and exported
                b. QVN (for QuickNII, Visualign, Nutil)
                    i. Atlas
                    ii. Input
                    iii. Mask
                    iv. Output_Left
                    v. Output_Right
    b. Figure (next one, same internal folder organization)…

QuPath Visualization/Segmentation

QuPath is a visualization and segmentation platform optimized for whole slide images. The is the place where you can visualize all your slides, and create the output that will be used by all other programs. In addition, we use QuPath to segment pathology and erase artifacts.



1. Open **QuPath**. Click “Create Project.”
2. Select your QuPath folder for this project and upload images by pressing “Add Images.” Navigate to full slide scans and add slides for this project.
3. Upload series of images into new project on QuPath. Don’t worry about what the images are named or how they are ordered yet.
4. Adjust settings for images prior to import. Our typical settings are shown in the image to the right, although these may change (for example, for fluorescent images).

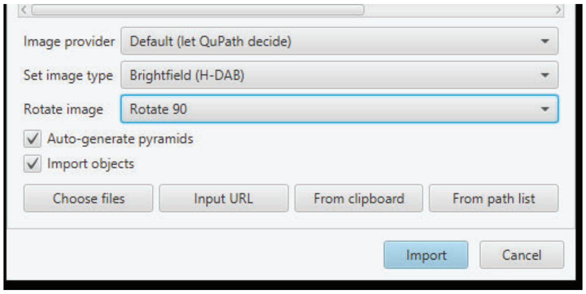


Image provider: Default (let QuPath decide)
Set image type: Brightfield (H-DAB)
Rotate image: Rotate 90
☑ Auto-generate pyramids
☑ Import objects

Image Export for Registration

5. Click the Rectangle Button. Draw a rectangle around desired brain.
6. Click File > Export images > Original pixels > Export format (PNG), Downsample factor (12.0)
7. Save in QVN folder with appropriate **Naming Designation**, and record this designation in the QUINT Section Log: Cohort abbreviation: Brain_s(Figure). So, for project NTG-G20 Inj 18, Brain A, for figure 23, we would give the image the name “NG18A_s023_”. In practice, if you have multiple sections around the same figure, you will need to name them uniquely so you don’t have multiple of the same image. For example, you may name sections “NG18A_s022_” and “NG18A_s024_”.
8. Move images to the **QVN** folder.

Color Deconvolution

9. This defines what **QuPath** will see as brown and what it will see as blue. Color settings should be kept constant for a project. Generally, settings should be similar as long as you are using the same chromogen and same microscope. However, we have found that the default settings in QuPath are not ideal so these settings will need to be changed in each project. This can be done with a simple analysis script. Information on Setting Stain Vectors in QuPath is included in the Appendix. Set the **color deconvolution** by going to the “Image” tab. Draw a rectangle over the desired color from pathology (be as specific as possible)

a. Click on the “Image” tab.

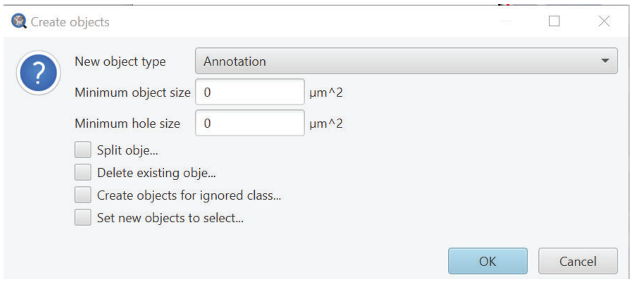
b. Draw a rectangle over a pure brown part of the image (9 pixels+).

c. Click on the “DAB” and it will ask you if you want to “Set stain vector from ROI” and click yes.

d. If you are doing multiple images, go to the “Workflow” tab and click on “Set color deconvolution stains” under it and click “create script” and run the script it has automatically created you for all images. You NEED to do this before setting your thresholder and running it for consistent results.

Segmentation

10. Name each annotation with abbreviated name designation found in the final column of the QUINT section log and described in **Step 7**.
    a. Right click on the rectangle annotation in the annotation tab and select “Set Properties”.
    b. Write the abbreviated name you would like your input file to be denoted as in the ‘Name’ box and select ‘OK’.
    c. Repeat for all desired annotations.
11. Load your classifier for this project: Classify > Pixel classification > Load pixel classifier.
12. Choose the correct classifier and “Any annotations.” Click “Create Objects.”
13. Pixel Classifier: click “All Annotations.” Click “OK.”
14. Create objects: leave settings as they are, and click “OK.”
    **This is a place where you could threshold by size or split object, if desired>*



New object type Annotation

Minimum object size 0 µm^2

Minimum hole size 0 µm^2

□ Split obje...

□ Delete existing obje...

□ Create objects for ignored class...

□ Set new objects to select...

15. Click “Annotations” tab. Navigate to Annotations tab on the left side of the program screen and click it. Find the Annotation (Geometry) you just generated. Right click the row and select “Unlock”.
16. Remove artifactual stain (see Appendix for examples; bubbles, edge effect, vasculature).
    a. Click on to the slide, zoom in, and select the brush tool .
    b. To erase, hold down the Alt key while erasing artifacts.
    Note: Zoom out and the brush is larger in what it erases, zoom in very close and the brush becomes small and precise.
17. After completing erase of artifact select “Automate” > Show script editor.
18. Script Editor: enter the script listed below into the box at line “1” and press Crtl + R to run. This will create your segmentation images that will be used in Nutil

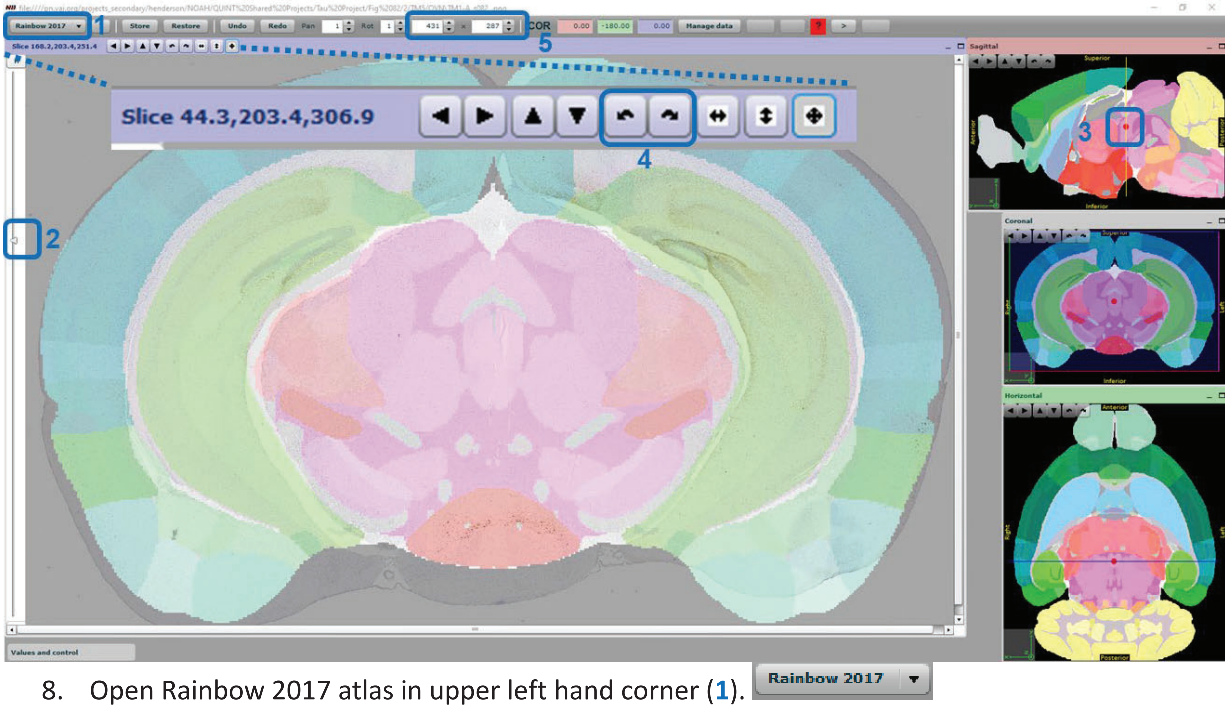
Code to Run:

```
import qupath.lib.images.servers.LabeledImageServer
def imageData = getCurrentImageData()
// Define output path (relative to project)
def name = GeneralTools.getNameWithoutExtension(imageData.getServer().getMetadata().getName())
def pathOutput = buildFilePath(PROJECT_BASE_DIR, 'export', name)
mkdirs(pathOutput)
// Export at full resolution
double downsample = 1.0
// Create an ImageServer where the pixels are derived from annotations
def labelServer = new LabeledImageServer.Builder(imageData)
 .backgroundLabel(0, ColorTools.WHITE) // Specify background label (usually 0 or 255)
 .downsample(downsample) // Choose server resolution; this should match the resolution at which tiles are exported
 .addLabel('Positive', 1) // Choose output labels (the order matters!)
 .multichannelOutput(false) // If true, each label refers to the channel of a multichannel binary image (required for multiclass probability)
 .build()
// Export each region
int i = 0
for (annotation in getAnnotationObjects()) {
   name = annotation.getName()
 if (annotation.getROI().getRoiName() == "Rectangle") {
 def region = RegionRequest.createInstance(
 labelServer.getPath(), downsample, annotation.getROI())
 i++
 def outputPath = buildFilePath(pathOutput, name + '.png')
 writeImageRegion(labelServer, region, outputPath)
}}
```

19. Find exported images in your folder scheme under **QuPath\export**. The segmentation image should be a black and white image with objects in black. The images should be the exact same dimensions of the earlier region of interest image you created as a PNG with a downsize factor of 12.
20. Plate the image and place into "**Input**" folder that will be used in final Nutil step.

QuickNII Brain Atlas Registration

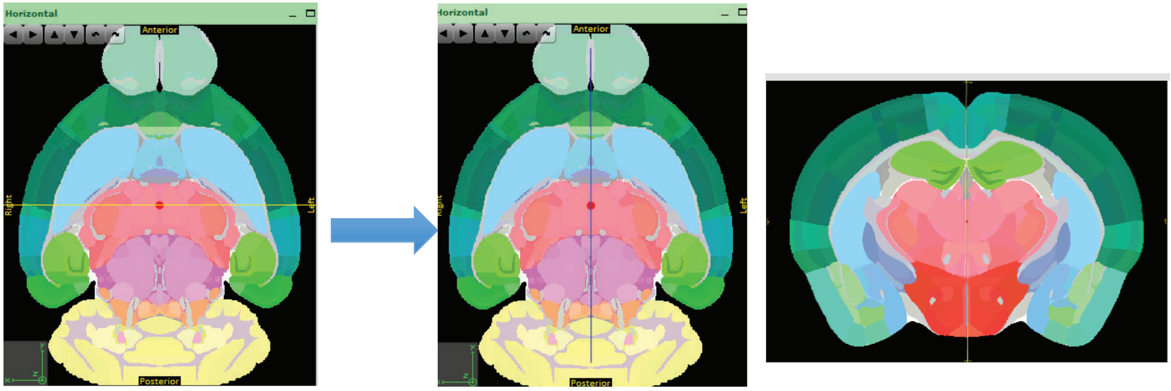
1. Open the **QuickNII** program folder.
2. Open **Filebuilder**.
3. Navigate to the **QVN** folder with the brain image exports from QuPath.
4. Click all images to be registered, and click “Open”.
   **It is useful to add a shortcut of the QVN folder to your desktop for simpler navigation.*
5. Press “SAVE XML”. Navigate to the **QVN** folder and save as "**Filebuilder XML**". Close FileBuilder.
6. Open the application **QuickNII**. Press “Manage Data” > “Load” and load the XML file that was just generated in step 11.
7. Double click on the first image, and it will show up on the Atlas in QuickNII.



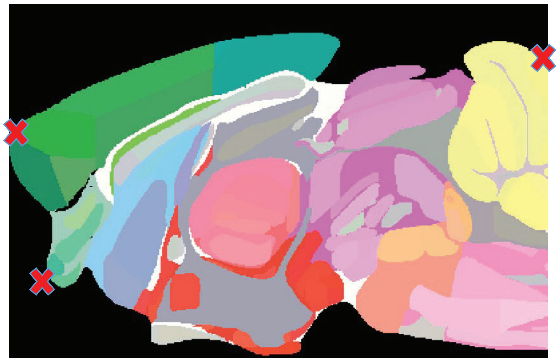
8. Open Rainbow 2017 atlas in upper left hand corner (1).

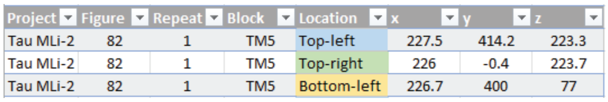
9. You can see the overlay better by dragging the transparency bar (2)
10. For **first** section, find anteroposterior position. To do this, drag the sagittal red dot (3) to the correct rostro-caudal position. Click “**Store**” to save the position.
11. Repeat for the **last** section. This will bring all other sections to the approximately correct position.
12. Adjust each individual section to the appropriate place in the atlas by adjusting. These may need to be done iteratively until the correct plane of section is identified.
    **Alignment will not be perfect, only the plane of section, but the better job you do here, the easier Visualign transformations will be.*
    a. **Rotation:** clockwise or counter-clockwise (4).
    b. **Brain Size:** in the x and y direction (5).
    c. **Rostro-caudal position:** sagittal view.
    d. **Left-Right plane:** Pull Horizontal view bar.
    e. **Front-back plan:** Pull Sagittal view bar.
    ****Note:** All of the Atlas needs to remain in the view or it will be lost for analysis.*
13. Click “**Store**” before moving off section or it will not save!

14. Navigate to next section by double clicking on it in the Manage Data window, or by clicking the < and > arrows in the upper right. Edit all sections as noted in **step 12**.
15. Click "Manage Data".
16. Click "Export Propagation" and save this XML file as "**QuickN XML.xml**" within the **QVN** folder.
    **Is not automatically recognized as a .xml file, hence the need to add ".xml" to the end of the name.*
    *Click "Export Slices" if you want atlas images, but this is not necessary for the workflow and will clutter your folder.
17. Click "Save JSON" and label it "**QuickN JSON**". This JSON file is used for **Visualign**.
18. For QMask, go back to the **first** section.
19. Adjust the horizontal plane (Right) in the atlas to match hemispheric split. Ensure proper bisecting by confirming with in the coronal plane.



20. Open **QMask Coordinates** Excel File.
21. Hover the cursor over the brain viewing window and record x-y-z coordinates (shown in the top left of the window) for three parts of the brain: Top-Left, Top-Right, Bottom-Left. Record coordinates in the **QMask Coordinates** file.

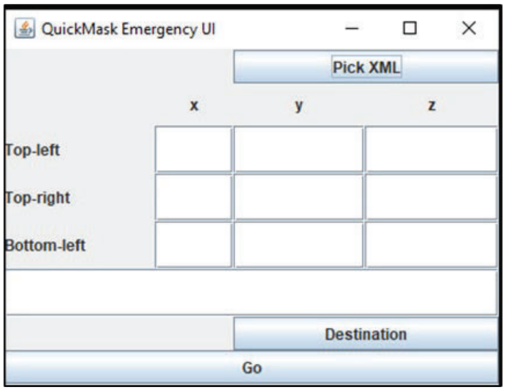




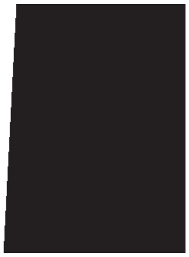
Project	Figure	Repeat	Block	Location	x	y	z
Tau MLi-2	82	1	TM5	Top-left	227.5	414.2	223.3
Tau MLi-2	82	1	TM5	Top-right	226	-0.4	223.7
Tau MLi-2	82	1	TM5	Bottom-left	226.7	400	77

QMask Hemispheric Separation

1. Open the Qmask tool and click “pick XML” and open the **QuickN XML** file generated in QuickNII
2. Enter the appropriate coordinates from the **QMask Coordinates file**.
3. Click “Destination” and navigate to the **QVN\Mask** folder. Click “Go”.



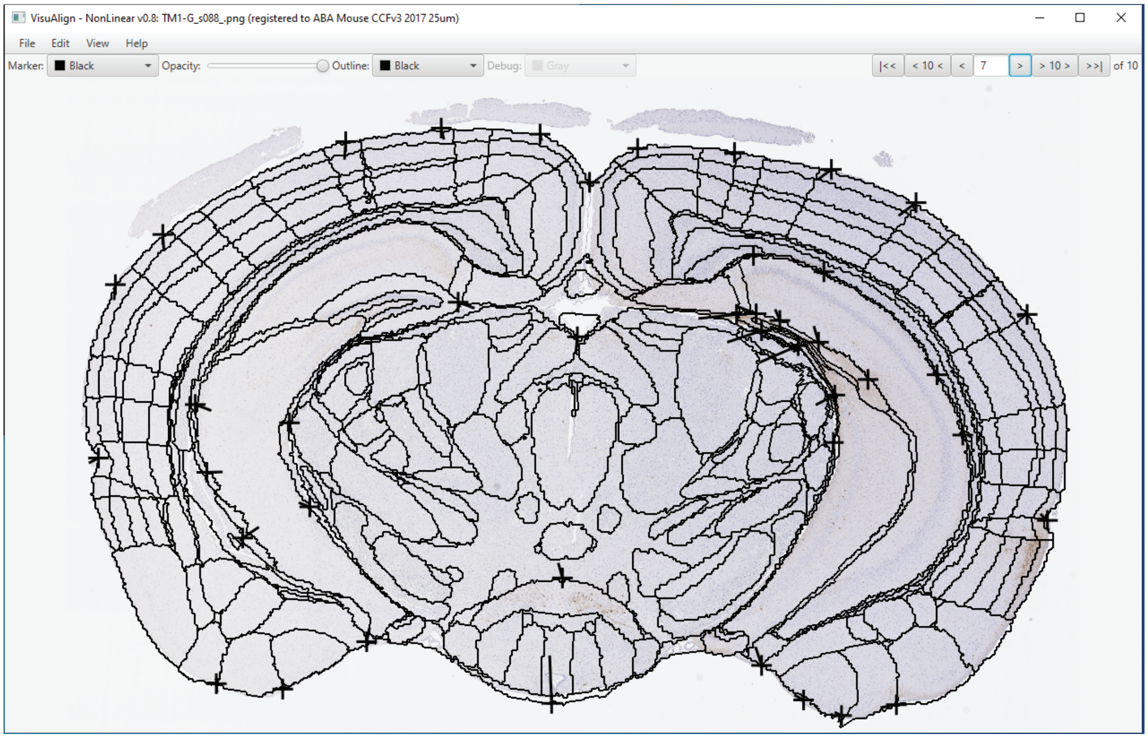
The mask output should be a black and White PNG that looks something like this:



4. Rename the mask files using the appropriate naming convention. Example: NG18I_s061_mask

Visualign Transformation

1. Open **Visualign** program from folder on the desktop.
2. File > Open > select "**QuickN JSON**" you created in QuickNII.
3. Drag the opacity bar all the way to the right to where it says outline, and you will have all outlined regions displayed. This is the easiest format for transformation.
4. Align all regions properly
    a. Hover over an atlas region.
    b. Click the **SPACE bar**.
    c. Drag the cross symbol to the correct location.
    d. Crosses can be moved at any point. They can also be deleted by hovering over them and clicking **Delete** on the keyboard.
    e. Tip: start with outer alignments before moving onto inner.
5. Click the < and > arrows at the top right to navigate between sections. There is no need to "Store" these as in QuickNII. See a finished section below.



VisuAlign - NonLinear v0.8: TM1-G_s088_.png (registered to ABA Mouse CCFv3 2017 25um)

Marker: Black

Opacity:

Outline: Black

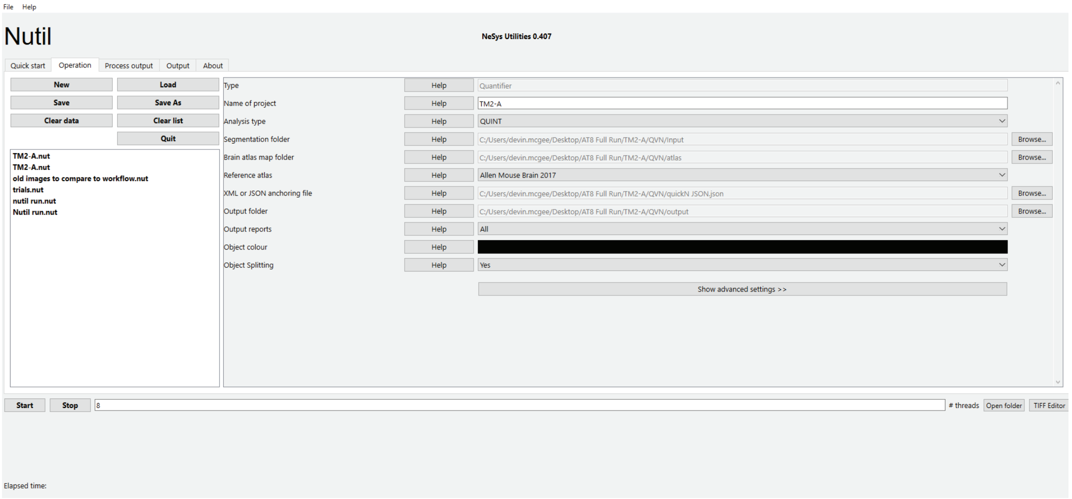
Debug: Gray

7 of 10

6. Once completed, click **File > Save As >** save "**Visualign JSON**" in the QVN folder. This can be opened again to continue aligning later.
7. To convert into a flat file, click **File > Export**, navigate to the **Atlas** folder and click "Select Folder". This will export the files needed for **Nuti**l.

Nutil Data Quantification

1. Open Nutil and find operation tab

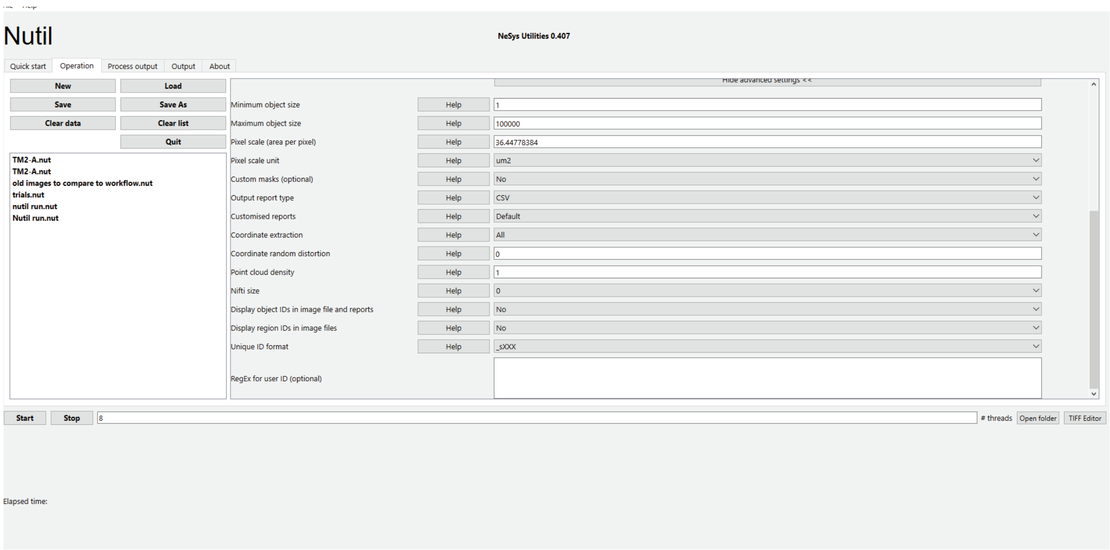


TM2-A.nut
TM2-A.nut
old images to compare to workflow.nut
trials.nut
nutil run.nut
Nutil run.nut

Type	Quantifier
Name of project	TM2-A
Analysis type	QUINT
Segmentation folder	C:/Users/devin.mcgee/Desktop/AT8 Full Run/TM2-A/QVN/Input
Brain atlas map folder	C:/Users/devin.mcgee/Desktop/AT8 Full Run/TM2-A/QVN/atlas
Reference atlas	Allen Mouse Brain 2017
XML or JSON anchoring file	C:/Users/devin.mcgee/Desktop/AT8 Full Run/TM2-A/QVN/quickN JSON.json
Output folder	C:/Users/devin.mcgee/Desktop/AT8 Full Run/TM2-A/QVN/output
Output reports	All
Object colour	■
Object Splitting	Yes

8

2. Find the 3 folders you created at the beginning of this workflow and populated during the workflow and populate them into their corresponding list above. (segmentation is your input folder, brain atlas map folder is your atlas folder, output folder is output and should be empty still XML or JSON anchoring file is the QuickN JSON file that you created earlier – you can also use visualign json file it doesn't matter the anchor is set in quickNII and nonlinear adjustments are made in reference to it. Object color Is black because that's what the staining of our segmentations look like if it were a different color we would need to look for another object color.



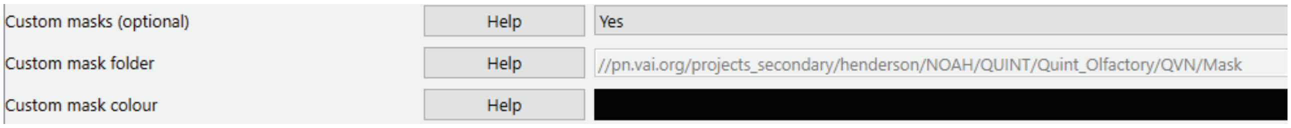
TM2-A.nut
TM2-A.nut
old images to compare to workflow.nut
trials.nut
nutil run.nut
Nutil run.nut

Minimum object size	1
Maximum object size	100000
Pixel scale (area per pixel)	36.44778384
Pixel scale unit	um2
Custom masks (optional)	No
Output report type	CSV
Customised reports	Default
Coordinate extraction	All
Coordinate random distortion	0
Point cloud density	1
Nifti size	0
Display object IDs in image file and reports	No
Display region IDs in image files	No
Unique ID format	_sXXX
RegEx for user ID (optional)	

8

3. Navigate to advanced settings In the row labeled Global Pixel Scale/Unit, set it to 36.44778384 um^2. (6.0 is new value for van andel scans) The NEW images that are scanned here at Van Andel are resolved at 0.5031 microns per pixel, so if we downsized by 12 we get 6.0372 then this number is squared to account for um^2. the final value is 36.44778384.

4. Also, under advance settings change custom masking to yes and select the color of the hemisphere of interest i.e. white to view hemisphere masked in white mask. Also select the folder location of the mask. Repeat the color selection with black to view the other hemisphere masked by black



Custom masks (optional)	Help	Yes
Custom mask folder	Help	//pn.vai.org/projects_secondary/henderson/NOAH/QUINT/Quint_Olfactory/QVN/Mask
Custom mask colour	Help	■

QUINT Workflow Appendix

QuPath

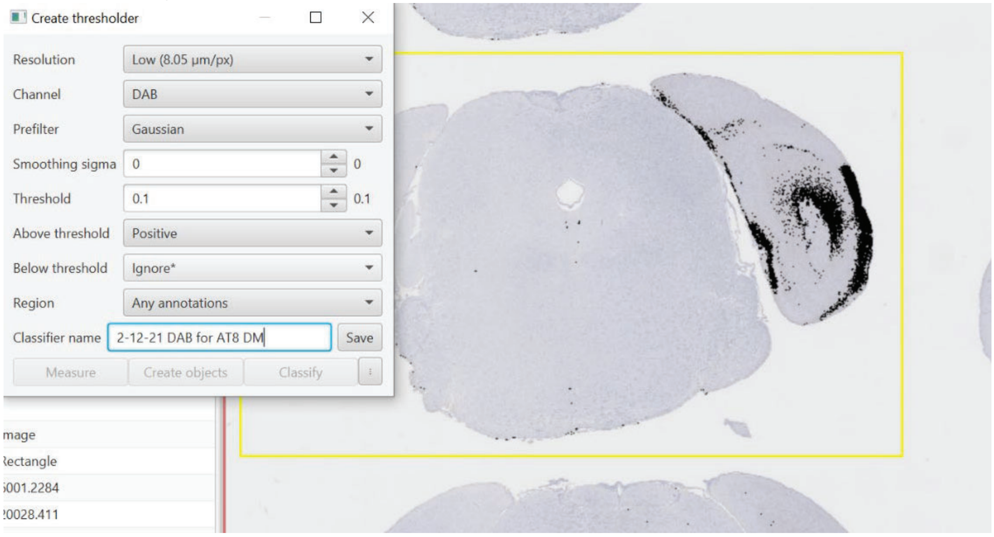
Setting Stain Vectors

1. Draw a small rectangle over a characteristic DAB or hematoxylin stain.
2. On Image tab, double-click the Stain you would like to set. Click Yes on the prompt.
3. Give the new settings a unique name (Project ID). Click OK.
4. *Note: You could also use “Estimate Stain Vectors.”
5. This has only saved the parameters for this particular image. To change all images in a project, go to the Workflow tab. This script can also be added to an analysis script and run at the same time.
6. “Set color deconvolution stains” should be the last Command. Click on Command, verify that the parameters are correct, and click “Create script.”
7. In the Script Editor, click Run > Run for Project.
8. Move all images from Available to Selected, and click “OK.”
   Modified setting from Aperio Scanner:
   Hematoxylin: 0.667, 0.658, 0.349
   DAB:0.324, 0.547, 0.772
   DAB:0.485, 0.616, 0.620

Pixel Classifier Generation

To create a new pixel classifier for segmentation:

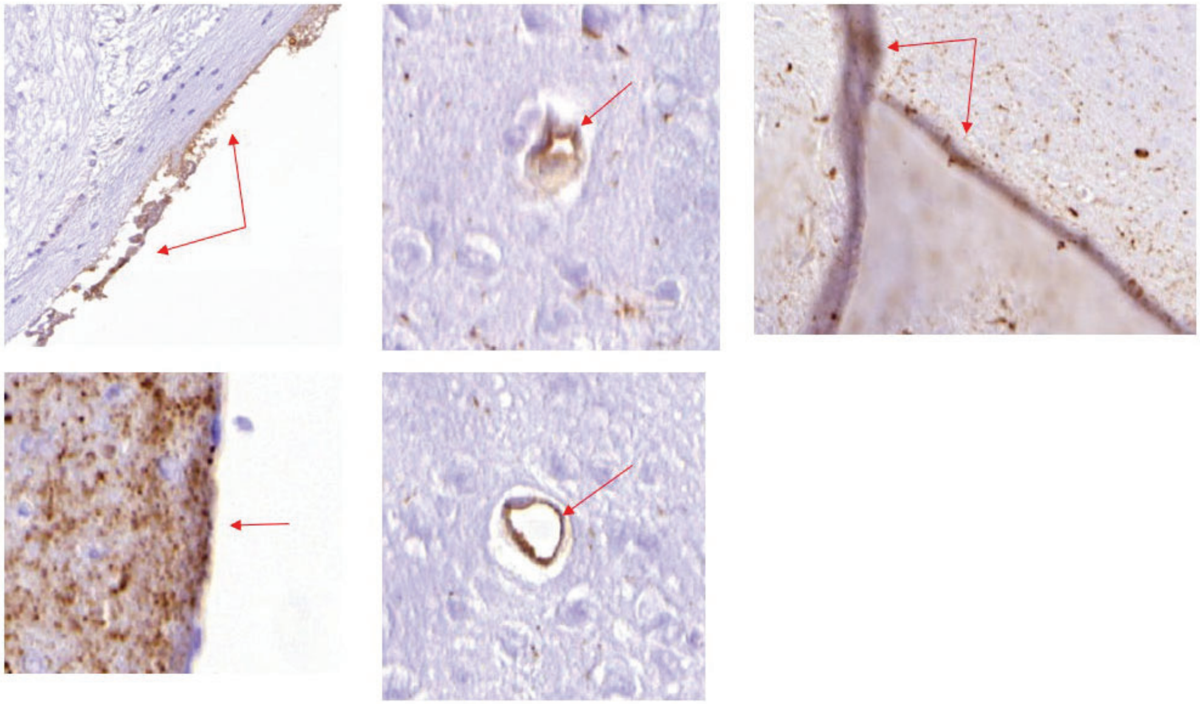
1. It is easiest if the output segmentation from QuPath is black. Click on the “Annotations” tab, double click on “Positive” and change the color to black. Click “OK”.
2. Use the rectangle tool to box an are you can use to test the classifier.
3. Classify > Pixel classification > Create thresholder
4. Example settings are below, but can be optimized by project. Generally higher resolution is better, but slower to run and will create larger files. We generally use “Very high (1.01 um/px)” resolution. Give the classifier a name and click “Save”. The classifier will be in your project folder under classifiers and can be moved to another qupath project folder to share with another set of images for consistency.



Create thresholder

Resolution: Low (8.05 µm/px)
Channel: DAB
Prefilter: Gaussian
Smoothing sigma: 0
Threshold: 0.1
Above threshold: Positive
Below threshold: Ignore*
Region: Any annotations
Classifier name: 2-12-21 DAB for AT8 DM

Examples of Staining Artifact:



DATA ANALYSIS SECTION

CODE FOR FIGURE AVERAGE:

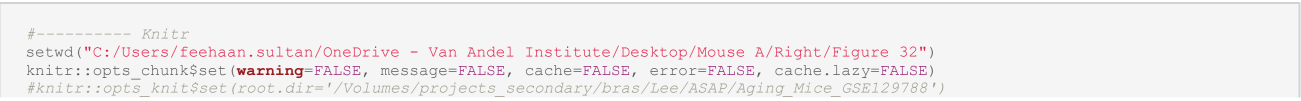
Make a folder with another folder inside called: nutil_outputs. Inside of nutil outputs, put your data in it. All your Nutil data transported in as tab delimited. (Must open the nutil data in excel and then save it as tab delimited data).

For example see: \\pn.vai.org\projects\henderson\FEEHAAN\GBA Pilot Data Analysis DONE\Individual Mouse Data\Mouse A\Right\Figure 32\nutil_outputs

You can pull out the RMD and put it next to your nutil_outputs and pull it out in R studio and run that code. You can also knit straight to HTML by using the HTML in that folder and clicking the knit button after running the code. It will also give you an excel sheet of your data at the end.

If you choose to copy and paste the code, you can run both together at the same time.

Packages: (If you do not have them, must download them first)



```
#---------- Knitr
setwd("C:/Users/feehaan.sultan/OneDrive - Van Andel Institute/Desktop/Mouse A/Right/Figure 32")
knitr::opts_chunk$set(warning=FALSE, message=FALSE, cache=FALSE, error=FALSE, cache.lazy=FALSE)
#knitr::opts_knit$set(root.dir='/Volumes/projects_secondary/bras/Lee/ASAP/Aging_Mice_GSE129788')
```

```
#options(scipen=999)
#options(stringsAsFactors = FALSE)
# Chunk Options
# include = FALSE, runs code, prevents code and results from appearing
# echo = FALSE, runs code, prevents code from appearing but not the results
# eval = FALSE, does not run code
# message = FALSE prevents messages that are generated by code from appearing
# warning = FALSE prevents warnings that are generated by code from appearing
# fig.cap = "..." adds a caption to graphical results.

#---------- Packages
#if (!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
#BiocManager::install("fgsea")
#install.packages('metap')
suppressPackageStartupMessages({
  library(data.table)
  library(dplyr)
  library(ggplot2)
  library(knitr)
  library(readr)
  library(reshape2)
  library(viridis)
})
sessionInfo()
```

Nutil Code:

```
#---------- Set Folder
file_path <- "./nutil_outputs"

#---------- Whole_Brain_Average
file_names <- list.files(path = file_path)
for(i in seq(length(file_names))){
  if(i == 1){
    Whole_Brain <- read_delim(paste(file_path,file_names[i],sep="/"),
                              delim = "\t",
                              escape_double = FALSE,
                              trim_ws = TRUE) %>%
                     select(`Region Name`, `Region area`, `Object area`) %>%
                     mutate(`Object Percentage` = `Object area` / `Region area` * 100)
  } else {
    Whole_Brain <- rbind(Whole_Brain,
                         read_delim(paste(file_path,file_names[i],sep="/"),
                                    delim = "\t",
                                    escape_double = FALSE,
                                    trim_ws = TRUE) %>%
                           select(`Region Name`, `Region area`, `Object area`) %>%
                           mutate(`Object Percentage` = `Object area` / `Region area` * 100))
  }
}
Whole_Brain_Avg <- Whole_Brain %>%
  group_by(`Region Name`) %>%
  summarise_each(funs(mean, sd))

#---------- Plot
Whole_Brain_Avg %>% ggplot(aes(x=`Region area_mean`,y=`Object area_mean`)) +
  geom_point() +
  geom_smooth(method=lm) +
  ggtitle("Whole Brain Average") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(plot.title = element_text(hjust=0.5, face="bold"))
```

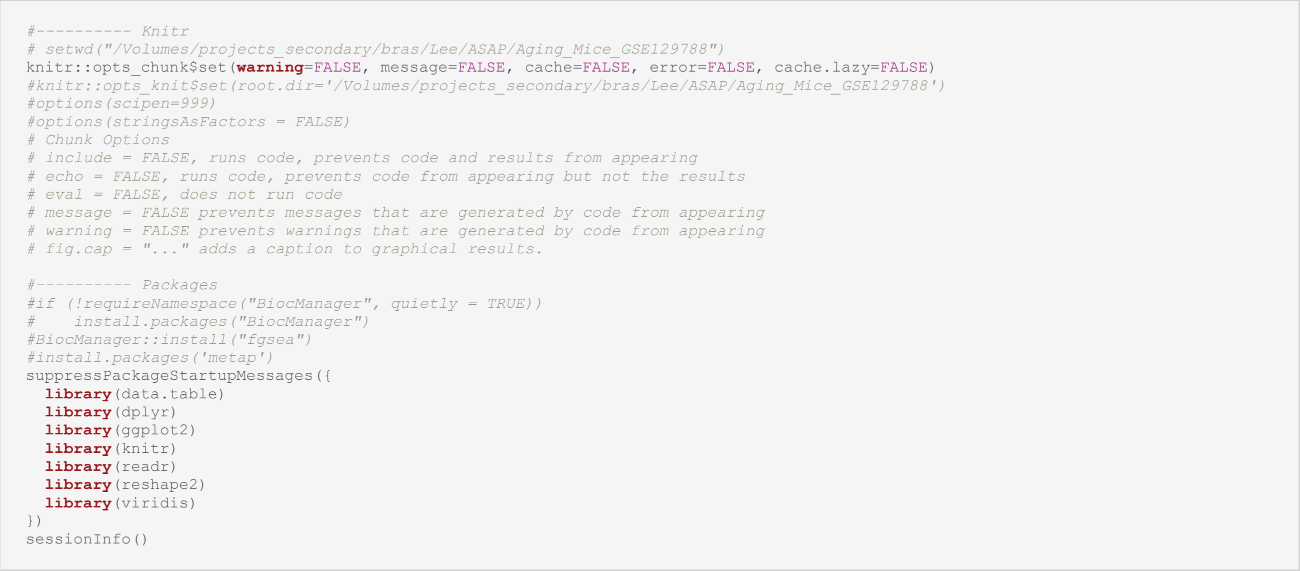
CODE FOR WHOLE BRAIN AVERAGE:

Make a folder with another folder inside called: nutil_outputs. Inside of nutil outputs, you can make separate folders with all your different mice within it. All your Nutil data transported in as tab delimited. (Must open the nutil data in excel and then save it as tab delimited data).

For example see: \\pn.vai.org\projects\henderson\FEEHAAN\GBA Pilot Data Analysis DONE\Left Mouse

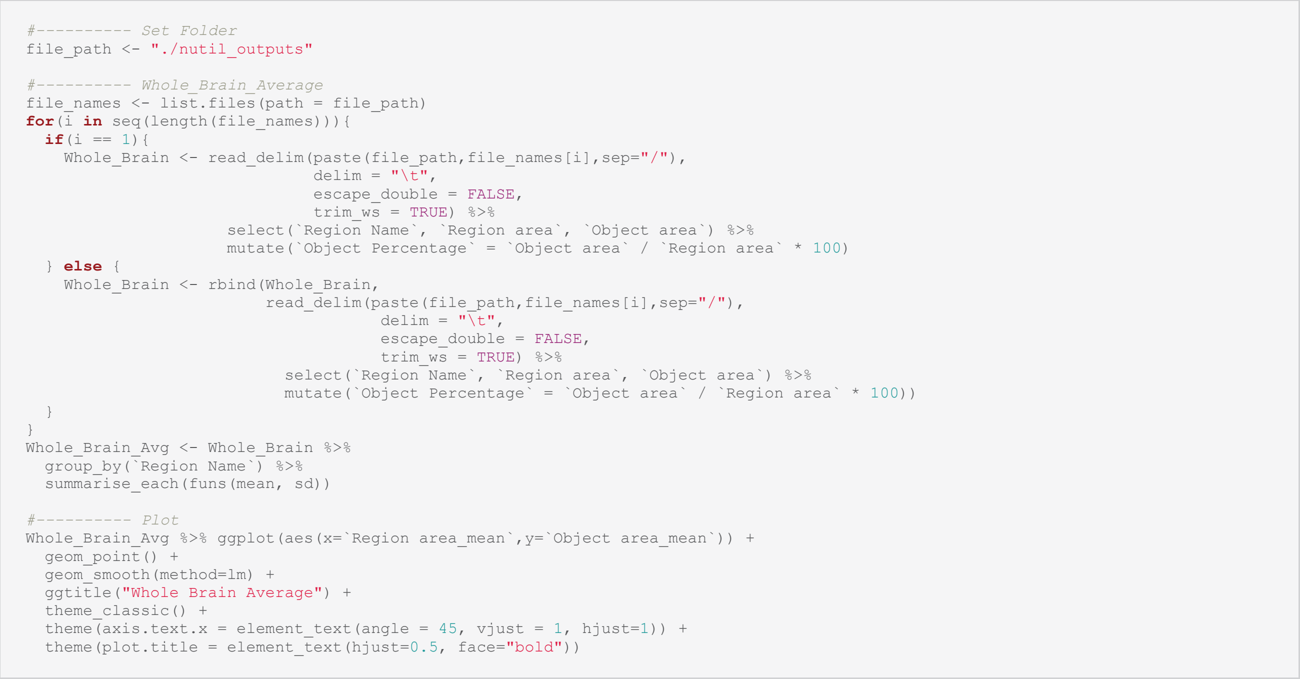
You can pull out the RMD and put it next to your nutil_outputs and pull it out in R studio and run that code. You can also knit straight to HTML by using the HTML in that folder and clicking the knit button after running the code. It will also give you an excel sheet of your data at the end.

Packages: (If you do not have them, you must download them first)



```
#---------- Knitr
# setwd("/Volumes/projects_secondary/bras/Lee/ASAP/Aging_Mice_GSE129788")
knitr::opts_chunk$set(warning=FALSE, message=FALSE, cache=FALSE, error=FALSE, cache.lazy=FALSE)
#knitr::opts_knit$set(root.dir='/Volumes/projects_secondary/bras/Lee/ASAP/Aging_Mice_GSE129788')
#options(scipen=999)
#options(stringsAsFactors = FALSE)
# Chunk Options
# include = FALSE, runs code, prevents code and results from appearing
# echo = FALSE, runs code, prevents code from appearing but not the results
# eval = FALSE, does not run code
# message = FALSE prevents messages that are generated by code from appearing
# warning = FALSE prevents warnings that are generated by code from appearing
# fig.cap = "..." adds a caption to graphical results.

#---------- Packages
#if (!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
#BiocManager::install("fgsea")
#install.packages('metap')
suppressPackageStartupMessages({
  library(data.table)
  library(dplyr)
  library(ggplot2)
  library(knitr)
  library(readr)
  library(reshape2)
  library(viridis)
})
sessionInfo()
```

Nutil Code:



```
#---------- Set Folder
file_path <- "./nutil_outputs"

#---------- Whole_Brain_Average
file_names <- list.files(path = file_path)
for(i in seq(length(file_names))){
  if(i == 1){
    Whole_Brain <- read_delim(paste(file_path,file_names[i],sep="/"),
                              delim = "\t",
                              escape_double = FALSE,
                              trim_ws = TRUE) %>%
                     select(`Region Name`, `Region area`, `Object area`) %>%
                     mutate(`Object Percentage` = `Object area` / `Region area` * 100)
  } else {
    Whole_Brain <- rbind(Whole_Brain,
                         read_delim(paste(file_path,file_names[i],sep="/"),
                                    delim = "\t",
                                    escape_double = FALSE,
                                    trim_ws = TRUE) %>%
                           select(`Region Name`, `Region area`, `Object area`) %>%
                           mutate(`Object Percentage` = `Object area` / `Region area` * 100))
  }
}
Whole_Brain_Avg <- Whole_Brain %>%
  group_by(`Region Name`) %>%
  summarise_each(funs(mean, sd))

#---------- Plot
Whole_Brain_Avg %>% ggplot(aes(x=`Region area_mean`,y=`Object area_mean`)) +
  geom_point() +
  geom_smooth(method=lm) +
  ggtitle("Whole Brain Average") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1)) +
  theme(plot.title = element_text(hjust=0.5, face="bold"))
```

QUINT Workflow for Fluorescence 02.02.2023

Purpose

The purpose of this workflow is to enable mouse brain segmentation, registration, and quantification of regional signals. The simplest segmentation is done in **QuPath** because this program handles whole slide images and has good segmentation algorithms. Registration is done using 3 programs: **QuickNII** (aligns to a 3D atlas, typically the 2017 CCFv3 Allen Brain Atlas), **VisuAlign** (allows for non-linear warp transformation of the atlas to match sections), **QMask** (masks each side of the brain to allow for bilateral assessment of brain regions). Segmentations and registrations are then brought together in **Nutil**, enabling the generation of quantitative measures for every region of the brain.

Necessary Programs and Locations

6. **QuPath:** https://qupath.github.io/
7. **QuickNII:** https://www.nitrc.org/projects/quicknii (SELECT ABA Mouse Edition)
8. **VisuAlign:** https://www.nitrc.org/projects/visualign/
9. **QMask:** \\pn.vai.org\projects_secondary\henderson\NOAH\QMask-Standalone
10. **Nutil:** https://www.nitrc.org/projects/nutil/
    *****QuickNII, VisuAlign, and Nutil can be a bit finicky, and we have found it is best to have these folders on the desktop.

Scanned Slides

To enable high throughput, slides are scanned on the Axioscan microscope. After slide scanning, images will be found at \\pn.vai.org\projects_primary\henderson\vari-core-generated-data\Axioscan\RNAScope. Move the slides into either the Human or Mouse folder and then into its corresponding stain and channel folder. If a folder does not exist, create one following the naming convention (i.e., Stain3_Slc17a7_Gad1_pS129).

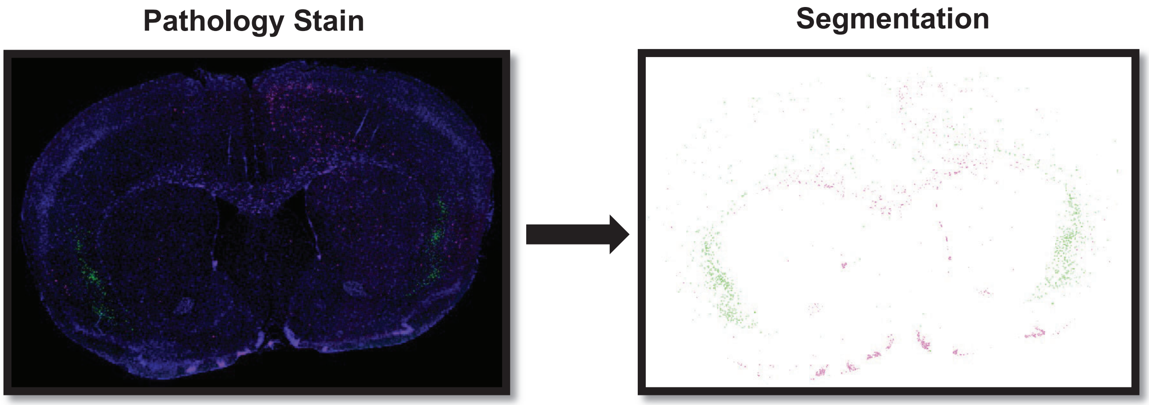
Folder Organization

The programs for registration and quantification rely on having exact file paths from which to call the data. Therefore, it is easiest to set these folders up from the beginning and using the following layout:

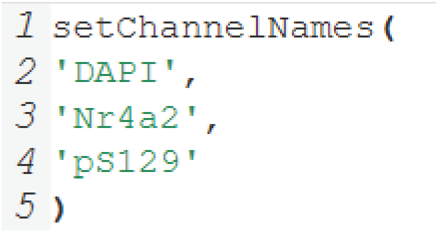
1. QUINT Workflow
   a. slide#stain#
      *This folder must be empty when you select it for your QuPath project in step #2 under the QuPath Visualization/Segmentation section. Once that is completed, you can then continue to create the following subfolders.
      i. QVN (for QuickNII, VisuAlign, Nutil)
         1. Atlas
         2. Input
         3. Mask
         4. Output_Left
         5. Output_Right

QuPath Visualization/Segmentation

QuPath is a visualization and segmentation platform optimized for whole slide images. This is the place where you can visualize all your slides and create the output that will be used for all other programs.



1. Open the **QuPath** application. Select ‘Create project’.
2. Select your slide/stain folder within your QUINT Workflow project folder (i.e., slide24stain19).
3. Select ‘Add images’ > ‘Choose files’. Navigate to the image file and select ‘Open’. Set image type to Fluorescence. Select ‘Import’.
4. Double click on the first image for it to appear in the viewing window.
5. Open the Brightness & contrast window (◑ d leave it open.
6. Select ‘Automate’ > ‘Show script editor’. Run the following script (change the channel names to the corresponding channels for your stain). This will change your channel names to the appropriate names.



```
setChannelNames(
'DAPI',
'Nr4a2',
'pS129'
)
```

7. Stay on the first image. In the script editor window select ‘Run’ > ‘Run for project’. Move all the images over to the selected column except for the first image. You cannot run a script for the whole project on an image that you have currently open in the viewer. Select ‘OK’. Close the script editor.
8. For each image, adjust the Min and Max display in the Brightness & contrast window for each channel as needed to make the scan easier to see. This only changes your visual, no actual properties of the scan.

9. Create annotation classifications for any new channel names you have previously not worked with. Select the ‘Annotations’ tab > three dots in the bottom right of the window > populate from image channels. Ensure that the class color matches the channel color.

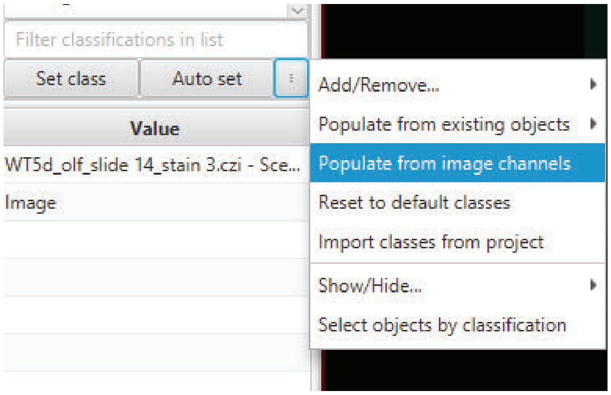

Filter classifications in list
Set class
Auto set
Value
WT5d_olf_slide 14_stain 3.czi - Sce...
Image
Add/Remove...
Populate from existing objects
Populate from image channels
Reset to default classes
Import classes from project
Show/Hide...
Select objects by classification

Image Export for Registration

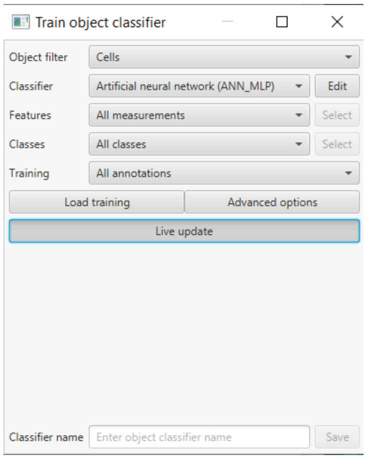
1. Select the rectangle button (□ nd select your ROI around the first brain.
2. Lock the rectangle by selecting the annotation (the rectangle will appear yellow). Right click inside the rectangle, select ‘Annotations’ > ‘Lock’. This will lock the rectangle, so it is not accidentally moved.
3. **MAKE SURE THE RECTANGLE IS SELECTED (YELLOW) AS YOU COMPLETE THIS STEP.** Select ‘File’ > ‘Export Images’ > ‘Rendered RBG (with overlays)’. Set the export format to (PNG). Set the downsample factor to 12.0. Select OK.
4. Save in the QVN folder with the appropriate naming designation (i.e., sl18st3sc1_s041, sl18st3sc2_s042_).
5. Repeat steps 1-4 for each image.

Cell Detection

1. Select ‘Analyze’ > ‘Cell Detection’ > ‘Cell Detection’ to open the cell detection window.
2. Adjust the Cell Detection parameters as needed. The most helpful parameters to adjust are Sigma, Minimum Area, Maximum Area, Threshold, and Cell Expansion.
    a. Minimum Area and Maximum Area will be dependent on whether you are analyzing mouse or human tissue.
    b. Keep all checkboxes checked “Split by Shape”, “Include Cell Nucleus”, “Smooth Boundaries”, and “Make Measurements”.
    c. A lower Sigma value will break up nuclei more. A higher Sigma value will combine nuclei more often.
    d. Once you are satisfied with the parameters, you can save the Cell Detection script to easily apply it to your other images. The same cell detection parameters must be applied to all images.
3. Remove cells from areas of damaged tissue or outside the brain.

Training Object Classifiers

1. Select the three dots to the right of ‘Auto Set’ > ‘Add/Remove…’ > ‘Add class’. Name the new class ‘Training’.
2. Choose 6-7 regions across all 8 images to use as training images. These regions should be ones that contain a variety of staining to properly train the classifier.
3. For each annotation rectangle created for the training classifier, set its class to ‘Training’ by right clicking in the annotation rectangle > ‘Set class’ > ‘Training’.
4. Once you have created all your training annotations, select ‘Classify’ > ‘Training images’ > ‘Create training image’. Set the classification of your combined training annotations to ‘Training’ and select ‘Rectangles only’. Set the image type of your combined training annotations to ‘Fluorescence’.
5. Create a duplicate of the combined training annotations for each different channel you have (excluding your nuclear stain). Rename the image to include the channel name. To duplicate, right click on the combined training annotation image and select ‘Duplicate image(s)’.
6. Draw a rectangle annotation around each training image. Lock the annotation. Run cell detection with the same parameters as your individual images.
7. Select ‘Classify’ > ‘Object classification’ > ‘Train object classifier’. Set the object filter to ‘Cells’, the classifier to ‘Artificial neural network (ANN_MLP)’, features to ‘Selected measurements’, classes to ‘All classes’, and training to ‘All annotations’. Select ‘Select’ to the right of ‘Selected measurements’ and select all measurements corresponding to the selected training channel. Leave the window open.
8. Open the points window (Select ‘Add’ two times to add two annotations. Set the first annotation to ‘Ignore*’ by right clicking on the annotation and select ‘Set class’ > ‘Ignore*’. Set the second annotation to the channel you are training.
9. In the Brightness & contrast window, turn off all channels except the channel you are training.
10. Train the classifier by using the points annotations to identify cells as either positive for the channel or ignore. Continue training the classifier until you are satisfied with the predicted output. Select live update to view how the points you select are affecting the classifier.
11. Give the classifier a name with the slide number, stain number, the channel, and the date (i.e., slide24stain19_Nr4a2_12.16.22). Select ‘Save’.
12. Repeat steps 7-11 for every training image.



Train object classifier

Object filter: Cells

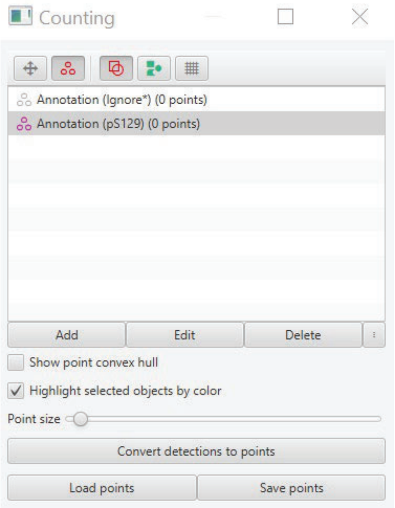
Classifier: Artificial neural network (ANN_MLP)

Features: All measurements

Classes: All classes

Training: All annotations

Classifier name: Enter object classifier name



Counting

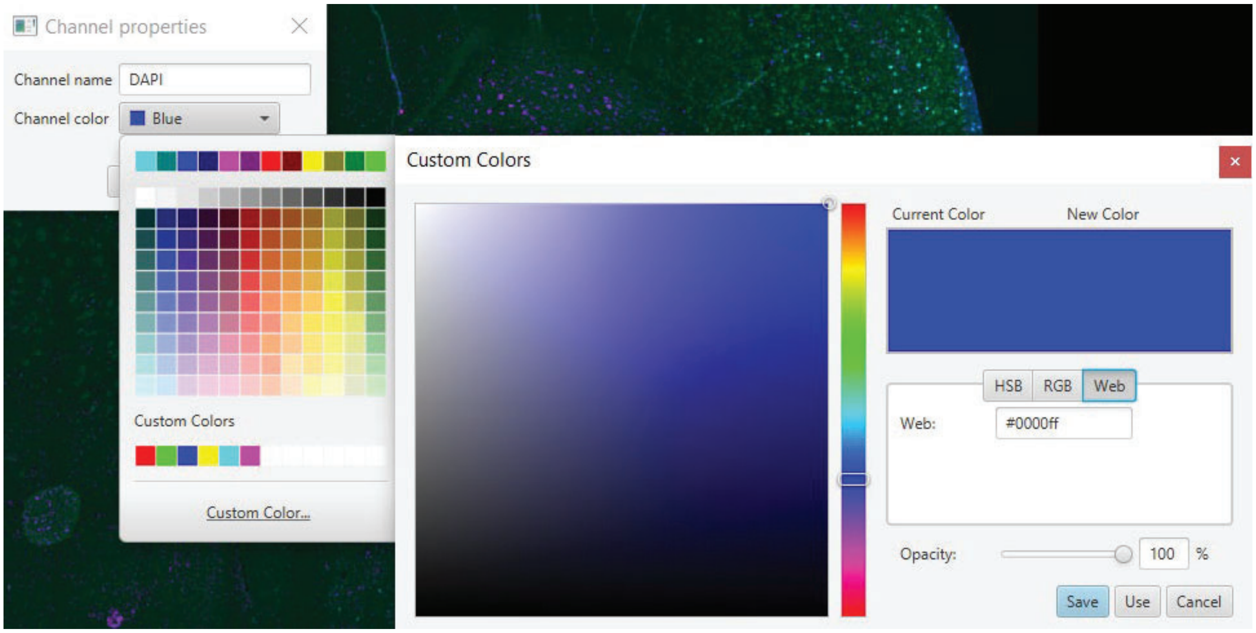
Annotation (Ignore*) (0 points)

Annotation (pS129) (0 points)

□ Show point convex hull

☑ Highlight selected objects by color

13. Select ‘Classify’ > ‘Object classification’ > ‘Create composite classifier’. Move the individual classifiers to the ‘selected’ side of the window in the order you want them to be applied. Enter a name for the combined classifier (i.e., sl18st19_composite_11.22.22) and select ‘Save & apply’.
14. For each whole brain image, select ‘Classify’ > ‘Object classification > ‘Load object classifier’, select the composite classifier you just created and ‘Apply classifier’.
15. To show sub-classes identified from your composite classifier (cells identified as positive for multiple channels) navigate to the ‘Annotations’ tab > three dots button in the bottom right of the window > ‘Populate from existing objects’ > ‘All classes (including sub-classes)’. Select ‘Yes’ when asked if you want to keep the existing available classes.
16. Write down the 6-digit web color code for each sub-class in your annotation window. You will need this information for Nutil. Double click the channels listed in the annotations tab to view the color code.



Channel name: DAPI

Channel color: Blue

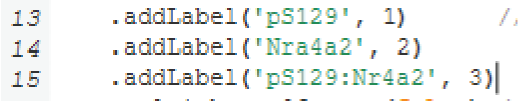
Web: #0000ff

Opacity: 100 %

Export Segmentation

1. Navigate to each scene image and rename the annotation that is off the entire brain image. Rename by right-clicking on the selected annotation image > ‘Annotations’ > ‘Set properties’. Rename the annotation using the following naming convention: ‘sl18st19sc1input_s041_’ ; ‘sl18st19sc2input_s042_’.
2. Navigate to one of your combined training images.
3. Select ‘Automate’ > ‘Show script editor’.

4. Enter the script in the blue box below at line 1. You can export multiple cell classifications together to simplify the downstream Nutil input. To do so, add more lines with “.addLabel(‘Name of classification’, #).



```
    .addLabel('pS129', 1)
    .addLabel('Nra4a2', 2)
    .addLabel('pS129:Nr4a2', 3)
```

5. Select ‘Run’ > ‘Run for project’. Move each scene image over to the ‘Selected’ window and select ‘OK’.

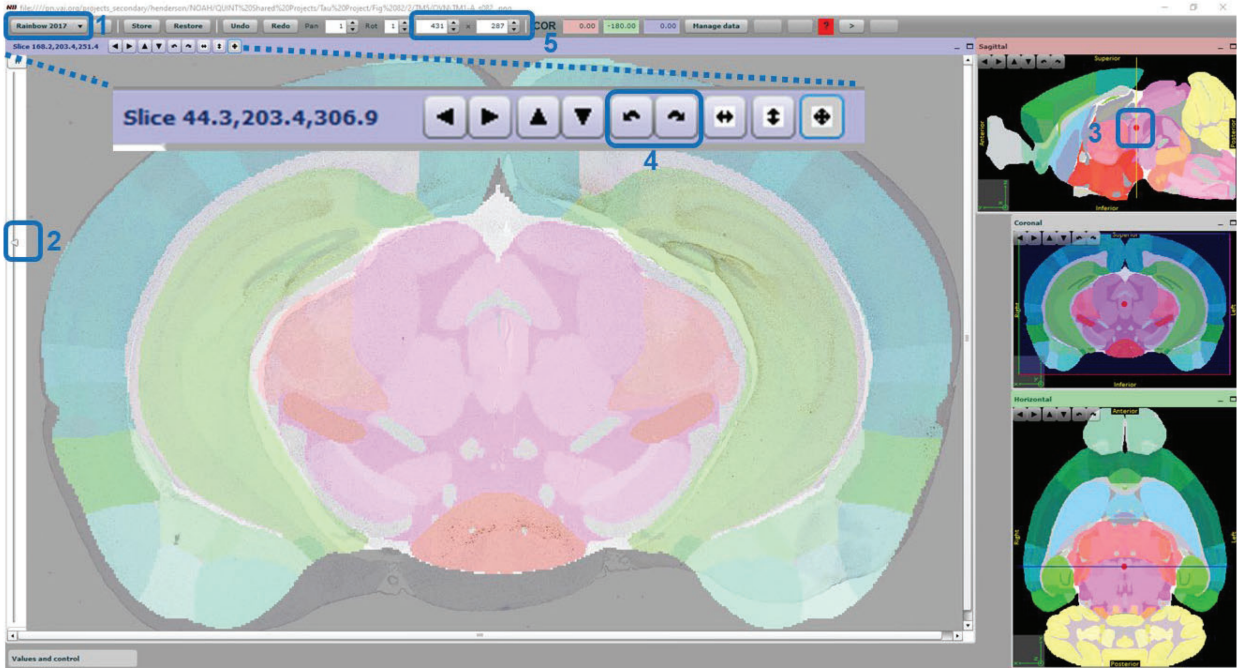
```
import qupath.lib.images.servers.LabeledImageServer

def imageData = getCurrentImageData()

// Define output path (relative to project)

def name = GeneralTools.getNameWithoutExtension(imageData.getServer().getMetadata().getName())

def pathOutput = buildFilePath(PROJECT_BASE_DIR, 'export', name)

mkdirs(pathOutput)

// Export at full resolution

double downsample = 1.0

// Create an ImageServer where the pixels are derived from annotations

def labelServer = new LabeledImageServer.Builder(imageData)

 .backgroundLabel(0, ColorTools.WHITE) // Specify background label (usually 0 or 255)

 .downsample(downsample) // Choose server resolution; this should match the resolution at which tiles are exported

 .addLabel('Slc17a7', 1) // Choose output labels (the order matters!)

.addLabel(‘Gad1’, 2)

 .multichannelOutput(false) // If true, each label refers to the channel of a multichannel binary image (required for multiclass
probability)

 .build()

// Export each region

int i = 0

for (annotation in getAnnotationObjects()) {

   name = annotation.getName()

 if (annotation.getROI().getRoiName() == "Rectangle") {

 def region = RegionRequest.createInstance(

 labelServer.getPath(), downsample, annotation.getROI())

 i++

 def outputPath = buildFilePath(pathOutput, name + '.png')

 writeImageRegion(labelServer, region, outputPath)

}}
```

6. Find the exported images in the export folder. The segmentation image should be white with objects in their WEB ID color. The images should be the exact same dimensions of the earlier region of interest image you created as a PNG with a downsize factor of 12.
7. Scenes with smaller annotation regions that you used to make your training images will have a second segmentation. Delete the segmentation image that corresponds to the smaller annotation.

8. Copy the images to the 'Input' folder in the QVN folder. These images will be used in the final step of Nutil. Close the QuPath application.

QuickNII Brain Atlas Registration

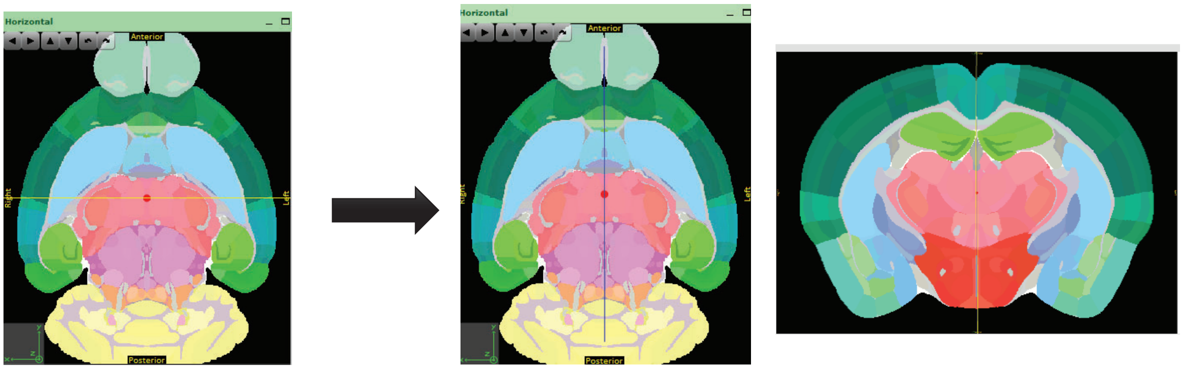
1. Open the **QuickNII** program folder.
2. Open **Filebuilder**.
3. Navigate to the **QVN** folder with the brain image exports from QuPath. These images are not the segmentation exports, but the original brain image exports. These images must all be surrounded by a yellow rectangle.
4. Select all the images to be registered and select 'Open'. *It is useful to add a shortcut of your QUINT Workflow folder to your desktop for simpler navigation.*
5. Select 'Save XML'. Navigate to the QVN folder and save as 'Filebuilder XML'. Make sure to save this file in the same folder as the brain image exports from QuPath.
6. Close Filebuilder.
7. Open the QuickNII application. Select 'Manage data' > 'Load' and select the XML file that was just generated in step 5.
8. Double click on the first image in Filebuilder to have it show up in the viewing window in the QuickNII application.



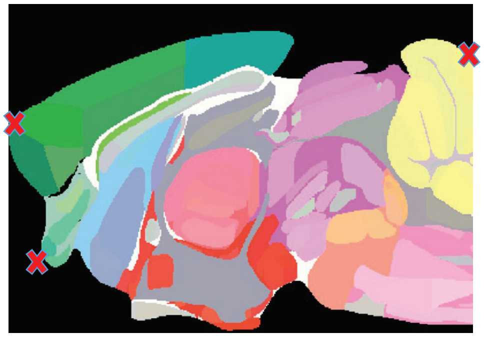
9. Select ‘Rainbow 2017’ from the drop-down menu in the upper left-hand corner of the toolbar (**1**).
10. You can adjust how you see the atlas overlay by dragging the vertical transparency bar on the left side of the screen (**2**).
11. For the first section, find the anteroposterior position. To do this, drag the sagittal red dot (3) to the correct rostro-caudal position. Select ‘Store’ to save the position.
12. Repeat step 11 for the last section. This will bring all other sections to the approximately correct position.
13. Adjust each individual section to the appropriate place in the atlas. Many adjustments may need to be made until the correct plane of section is identified. **NOTE:** All the atlas needs to remain in view, or it will be lost for analysis. Keep the atlas image smaller than the brain image. All of the Atlas needs to remain in the viewer or it will be lost for analysis. *Alignment will not be perfect, only the plane of section, but the better job you do here, the easier VisuAlign transformations will be.*
    a. **Rotation**: clockwise or counterclockwise (4).
    b. **Brain size**: in the x and y direction (5).
    c. **Rostro-caudal position**: adjust sagittal view.
    d. **Left-right plane**: adjust horizontal view.
    e. **Front-back plane**: adjust sagittal view.
14. Select ‘Store’ before moving off a section or it will not save!
15. Navigate to the next section by double clicking on the section in the Filebuilder, or by selecting the < and > arrows in the upper-right corner. Edit all sections as noted in step 13.
16. Select ‘Manage data’ > ‘Export Propagation’ and save this XML filed as “QuickN XML.xml” within the QVN folder. *It is not automatically recognized as a .xml file, hence the need to add “.xml” to the end of the name.*
17. Select ‘Save JSON’ and name it “QuickN JSON” within the QVN folder. This JSON file is used for VisuAlign. Make sure to save the JSON file in the same folder as the brain image exports from QuPath and the Filebuilder XML file.

QMask

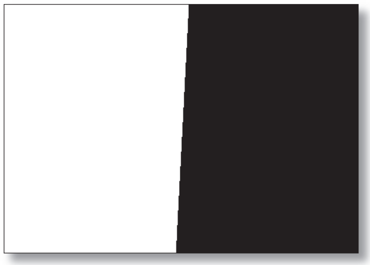
1. In the QuickNII application, go back to the first section. Show just the rainbow atlas image.
2. Adjust the horizontal plane to a hemispheric split (completely vertical). Ensure proper bisecting by confirming it within the coronal plane. You want to perfectly bisect the coronal plan. Change the angle if needed.

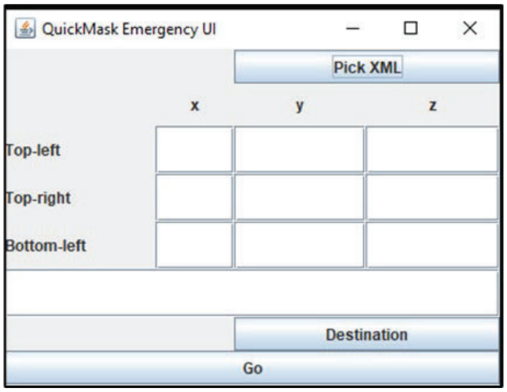


3. Hover the cursor over the brain viewing window and record the x-y-z coordinates (shown in the top left of the window) for the following three parts of the brain: top left, top right, and bottom left.



4. Open the QMask tool and select ‘Pick XML’ and open the QuickN XML file generated in QuickNII.
5. Enter the x-y-z coordinates.
6. Select ‘Destination’ and navigate to the Mask folder within your current project folder and select ‘Open’.
7. Select ‘Go’. The mask output should be a black and white PNG. Close QuickNII and the QMask tool.



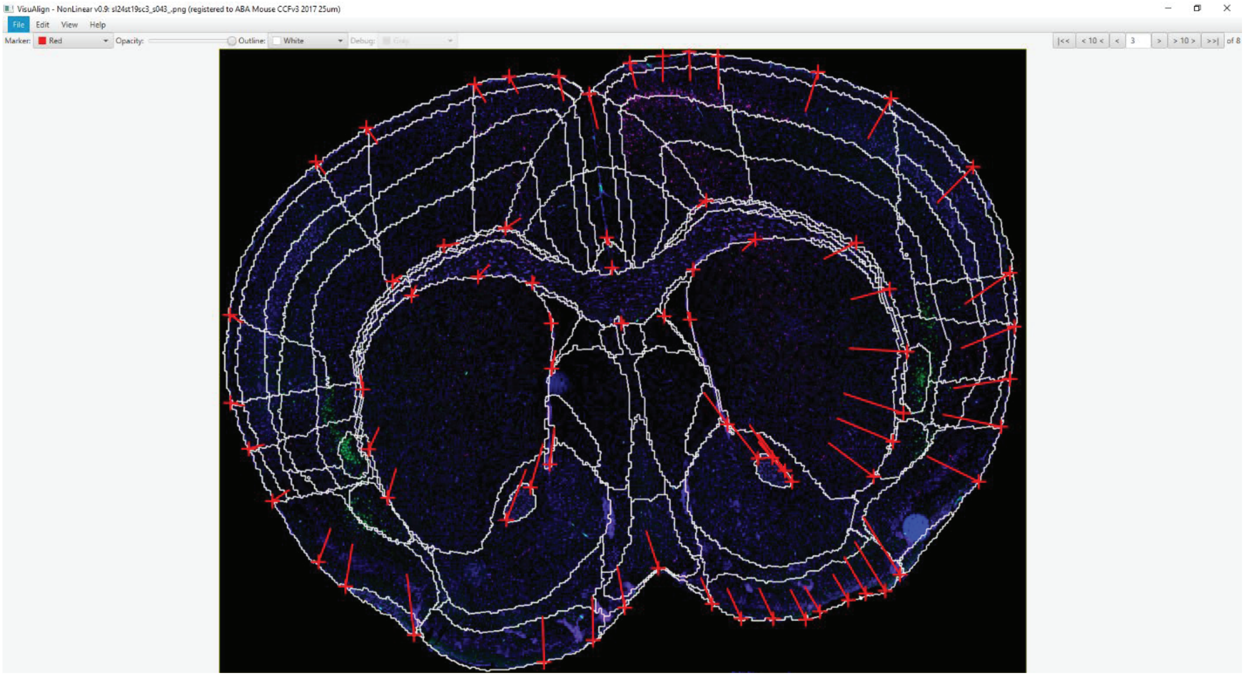


8. Ensure the mask files are named using the appropriate naming convention (i.e., sl18st3sc1_s041_mask).
9. You can check to see if your mask outputs accurately bisect for a hemispheric split by comparing the mask output to the output from VisuAlign.

VisuAlign

1. Open the **VisuAlign** application.
2. Select ‘File’ > ‘Open’ and select the QuickN JSON file you created in QuickNII.
3. Drag the opacity bar all the way to the right towards ‘Outline’. This will display an outline of all the regions. This is the easiest format for transformation. You can change the color of the outline and the marker for easier visualization.
4. Align all regions properly.
    a. Hover over an atlas region.

b. To create a marker, click the space bar.
c. Drag the maker symbol to the correct location.


VisuAlign - NonLinear v0.9: sl24st19sc3_s043_.png (registered to ABA Mouse CCFv3 2017 25um)

Marker: Red
Opacity:
Outline: White
Debug: Gray

d. Markers can be moved at any point. To delete a maker, hover over the maker and click delete on the keyboard.
e. Start with the outer alignments first then move inwards.

5. Select the < and > arrows at the top-right to navigate between sections. There is no need to save or store these as in QuickNII.
6. Once all images are complete, select ‘File’ > ‘Save as’ and save as ‘VisuAlign JSON’ in the QVN folder. This can be opened again to continue aligning later.
7. Select ‘File’ > ‘Export’ and navigate to the Atlas folder and select ‘Select Folder’. This will export the files needed for **Nutil**. These are also the images that you can compare to the Mask outputs to determine the accuracy of your hemispheric split.

Nutil

1. Open the Nutil application and navigate to the ‘Operation’ tab.
2. Select ‘New’. Select ‘Quantifier’ from the drop-down menu and select ‘Ok’.
3. Select ‘Save’ to save the overall Nutil project (i.e., “sl24st19.nut”).
4. Name each project using the following naming convention: ‘sl#st#_classifier_left/right’.
5. Set the ‘Segmentation folder’ to the ‘Input’ folder within the QVN folder.
6. Set the ‘Brain atlas map folder’ to the ‘Atlas’ folder within the QVN folder.
7. Select ‘Reference atlas’ to the ‘Allen Mouse Brain 2017’.
8. Set the ‘XML or JSON anchoring file’ to the ‘VisuAlign JSON.json’ file.
9. Set the ‘Output folder’ to the ‘Output_left’ or ‘Output_right’ folder within the QVN folder depending on which hemisphere you are running.