Published: Jul 05, 2022

Automated Quantification of Multiple Cell Types in Fluorescently Labeled Whole Mouse Brain Sections Using QuPath

Jo-Maree Courtney, Gary P. Morris, Elise M. Cleary, David W. Howells and Brad A. Sutherland*

Tasmanian School of Medicine, College of Health and Medicine, University of Tasmania, Hobart, Tasmania, Australia
*For correspondence: brad.sutherland@utas.edu.au

Abstract

The quantification of labeled cells in tissue sections is crucial to the advancement of biological knowledge. Traditionally, this was a tedious process, requiring hours of careful manual counting in small portions of a larger tissue section. To overcome this, many automated methods for cell analysis have been developed. Recent advances in whole slide scanning technologies have provided the means to image cells in entire tissue sections. However, common automated analysis tools do not have the capacity to deal with the large image files produced. Herein, we present a protocol for the quantification of two fluorescently labeled cell populations, namely pericytes and microglia, in whole brain tissue sections. This protocol uses custom-made scripts within the open source software QuPath to provide a framework for the careful optimization and validation of automated cell detection parameters. Images obtained from a whole-slide scanner are first loaded into a QuPath project. Manual counts are performed on small sample regions to optimize cell detection parameters prior to automated quantification of cells across entire brain regions. Even though we have quantified pericytes and microglia, any fluorescently labeled cell with clear labeling in and around the nucleus can be analyzed using these methods. This protocol provides a user-friendly and cost-effective framework for the automated analysis of whole tissue sections.

Keywords: Image analysis, Cell counting, QuPath, Brain, Pericyte, Microglia, Slide scanning microscope

This protocol was validated in: eNeuro (2021), DOI: 10.1523/ENEURO.0177-21.2021

Cite as: Courtney, J. M. et al. (2022). Automated Quantification of Multiple Cell Types in Fluorescently Labeled Whole Mouse Brain Sections Using QuPath. Bio-protocol 12(13): e4459. DOI: 10.21769/BioProtoc.4459.

Background

Since the invention of the microscope, quantification of specific cell types in tissue sections has played a crucial role in biological discovery. Advances in cell labeling, including immunohistochemistry, in situ hybridization, and the use of genetically encoded fluorescent labels, have enabled cells to be identified and distinguished with increasing specificity. However, the quantification of labeled cells has largely remained a tedious manual or semi-automated process. The time-consuming nature of this process means it has only been practical to quantify small sample areas of tissue, leaving the majority of tissue uncounted. Even when an entire brain can be imaged, computer-based semi-automated image analysis tools available in programs such as ImageJ are not able to handle the large files produced; therefore, analysis has largely remained limited to small sub-regions. In a complex and heterogeneous organ, such as the brain, these restricted processes can lead to experimenter bias, due to the subjective nature of manual counting and sampling bias, if a selected subset of tissue is not an accurate representation of a whole region.

Recent advances in computational image analysis and whole slide scanning technology have made it possible to quantify every single cell in a whole tissue section on a regular desktop computer. It is now practicable to quantify and further analyze the characteristics of hundreds of thousands of cells in a few minutes—a process that would take days or weeks to do manually. However, this process is not perfect. Software programs that examine digitized images to determine what is, and is not, a cell, are only as good as their underlying algorithms (which are hard-coded by the engineers of the software) and the validity of any user-determined input parameters. It remains important to carefully assess the validity of automated detection algorithms manually, as small differences in tissue processing, staining, and imaging can have significant impacts on the performance of automated algorithms when they are applied across whole projects. Despite this, many studies using automated cell counting approaches do not present evidence of optimization, or justification for the selection of specific parameters.

Here, we present a semi-automated method for counting and classifying two fluorescently labeled cell types—pericytes and microglia—in whole mouse brain tissue sections using the open source software QuPath (Bankhead *et al.*, 2017) as used in Courtney *et al.* (2021). Our method uses extensive optimization to objectively determine the specific automated cell detection parameters required for each different brain region.

Although this method is written specifically for quantifying pericytes labeled with the fluorophore DsRed and microglia labeled with the fluorophore GFP in mouse brain tissue, it could easily be adapted to quantify any type of cell in any type of tissue, so long as the stain is localized within or immediately around the nucleus. In addition, we present the optimization of just one specific parameter important for automated cell detection, the fluorescence intensity threshold. However, we provide the framework for adding optimization steps for additional cell detection parameters, if desired.

Collectively, the method presented here offers a user-friendly and cost-effective framework for the automated quantification of cell numbers in whole tissue sections, without the need for extensive and time-consuming manual counting.

Equipment

1. VS120 Slide Scanner (Olympus) or any other microscope capable of scanning and digitizing whole slides
2. Computer: 64-bit Windows, Linux, or Mac with minimum 16 GB RAM and a fast multicore processor (*e.g.*, Intel Core i7)
   Note: This protocol will likely work with a less powerful computer or with lower RAM, but analyses will be slow and may encounter memory errors. For more details, please refer to the QuPath online documentation (https://qupath.readthedocs.io/).

Software

1. QuPath 0.3.2 open source software (https://qupath.github.io/)
2. Custom QuPath scripts available at: https://github.com/jo-maree/BioProtocol-2022-Scripts
3. Notepad or any other simple text editor for making changes to scripts and classifiers

4. Microsoft Excel or similar
5. GraphPad Prism (GraphPad Software https://www.graphpad.com/) or similar

Procedure

A. Stain and scan tissue sections

The specific protocol for staining and scanning sections will depend on the starting material, individual laboratory practices, and slide scanning equipment, so will not be covered here. For details on our tissue processing and imaging, please refer to Courtney *et al.* (2021). Tissue should be stained with a clear nuclear stain [*e.g.*, DAPI (4',6-diamidino-2-phenylindole)], which is used to identify nuclei of cells. Cells should be clearly labeled with appropriate fluorophores. In our case, the fluorescent tags were genetically encoded, but a well-optimized immunohistochemical stain should give comparable results.

QuPath uses BioFormats (Linkert *et al.*, 2010) to handle the import of files from most slide scanning platforms. BioFormats currently supports over 150 different file formats including .vsi, .svs, .czi, and .tiff. Images should be scanned at high enough resolution to enable clear visual identification of nuclei and other cellular structures—we used the 40× objective of the VS120 Slide Scanner to provide images with a resolution of 160.3nm/pixel; however, with clear staining, this analysis should be possible with lower resolution (*e.g.*, 322nm/pixel from a 20× objective). Images imported into QuPath should contain a single plane (our images were taken in a single plane, but this protocol would also work with a projection from a z-stack image), and each brain section should be saved as an individual file rather than an entire slide with multiple sections.

Note: QuPath can import both z-stack images and single images with multiple sections, so this kind of analysis is possible in such situations; however, the custom scripts written for this analysis will need to be adapted to cope with these scenarios, and describing how to do so is outside the scope of this protocol.

B. Create a project and prepare images

The first step to any analysis in QuPath is to create a project. This allows the saving of scripts and classifiers that can be used across multiple images. Note that the project will never contain actual image files, just the data pertaining to them and links to the original images. The project folder does not have to be stored in the same place as the images, but if they are separated (*e.g.*, images on a server and the project on an external hard drive), it may be difficult to reinstate the links at a later time; therefore, it is recommended to at least keep them on the same drive.

1. Create a project and add images

a. In your file management system, create a new folder (directory) and give it an appropriate name. This folder will house the QuPath project file and all associated data.

b. Start QuPath and choose 'Create project'.

c. Navigate to the folder you just created, double-click to open it, and then click Select Folder.

d. Choose 'Add Images' to open the Import Images dialog box (Figure 1A) and select options as follows:

   i. Choose the files to be analyzed by dragging and dropping them into the box or selecting one of the four buttons below (*e.g.*, Choose Files to select from your file manager).

   ii. Image provider: leave as Default (let QuPath decide).

   iii. Set image type: Fluorescence.

   iv. Rotate image: Depending on the orientation of your scanned images, you may want to rotate them by 90°, 180°, or 270°, or leave with no rotation.

   v. Optional args: Leave this blank.

   vi. Auto-generate pyramids: checked.

   vii. Import objects: unchecked.

e. Click the Import button to link the images to your QuPath project.

f. Open your file manager, and, in the folder you created in step B1b, there should now be two folders, 'classifiers' and 'data'. Add new folders called 'scripts' and 'exports' to sit alongside them. Your folder should now look like Figure 1B.

g. Copy all the script (.groovy) files associated with this protocol into the scripts folder (Figure 1C).

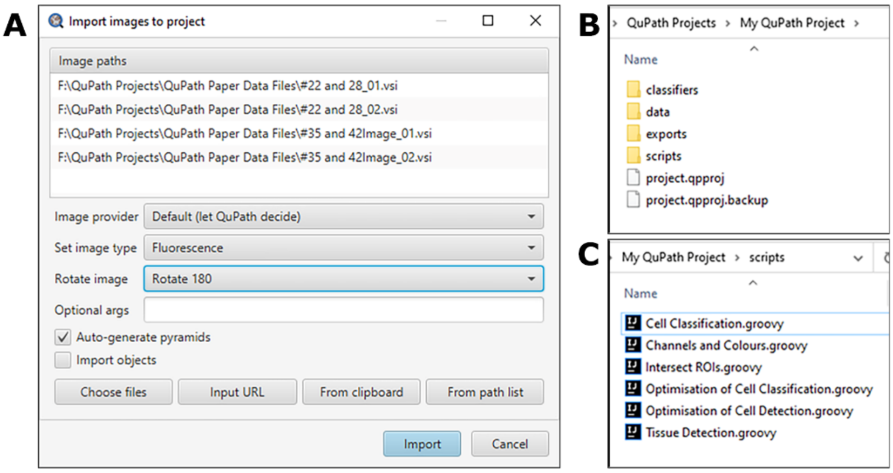


Image paths
F:\QuPath Projects\QuPath Paper Data Files\#22 and 28_01.vsi
F:\QuPath Projects\QuPath Paper Data Files\#22 and 28_02.vsi
F:\QuPath Projects\QuPath Paper Data Files\#35 and 42Image_01.vsi
F:\QuPath Projects\QuPath Paper Data Files\#35 and 42Image_02.vsi

Image provider: Default (let QuPath decide)
Set image type: Fluorescence
Rotate image: Rotate 180
Optional args:
☑ Auto-generate pyramids
□ Import objects

QuPath Projects > My QuPath Project >
classifiers
data
exports
scripts
project.qpproj
project.qpproj.backup

My QuPath Project > scripts
Cell Classification.groovy
Channels and Colours.groovy
Intersect ROIs.groovy
Optimisation of Cell Classification.groovy
Optimisation of Cell Detection.groovy
Tissue Detection.groovy

Figure 1. Import of images and example folder structure.
A) Import Images dialog box showing the import of four fluorescent images that need to be rotated 180 degrees. B) Set up of QuPath project folder structure. C) Groovy script files in the scripts folder.

2. Set up channel colors, names, and classes

QuPath automatically colors channels in the order red, green, blue, yellow, cyan, and magenta, but, depending on the file format and scanning options, this will likely be incorrect for your images. This can be corrected for all images at once using the following script.

a. Find the Channels and Colours script in QuPath in the menu Automate > Project Scripts and open it in the Script Editor (Figure 2A).

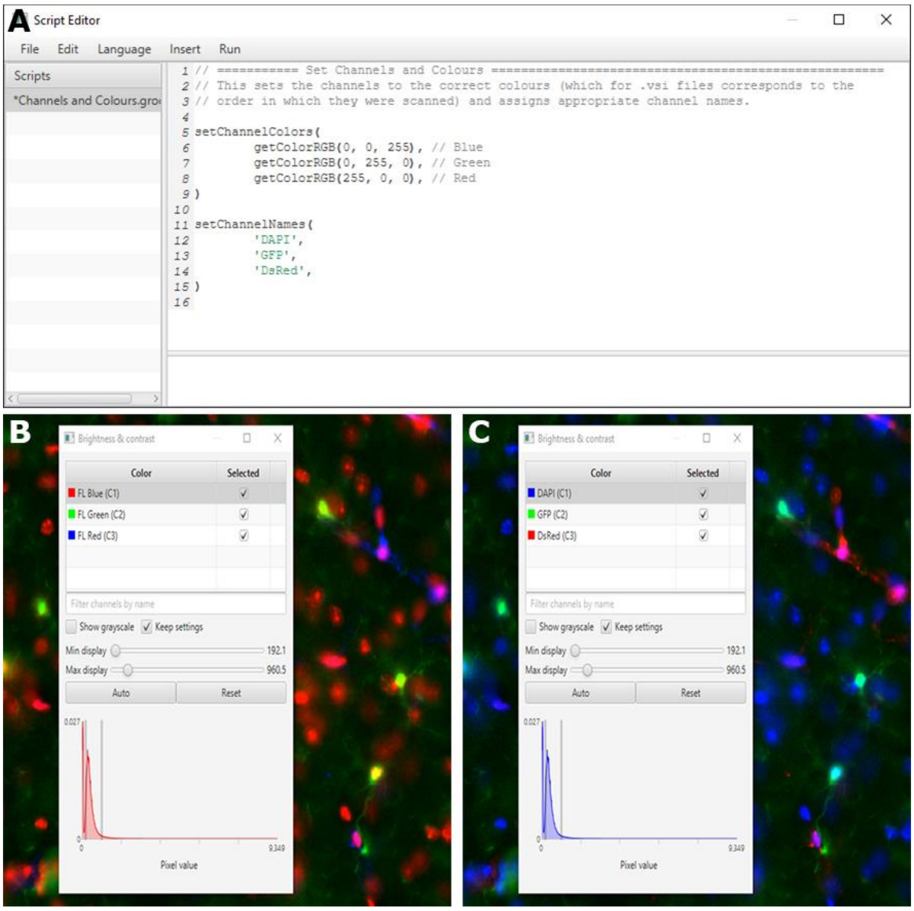
b. Our image channels were scanned in the order Blue > Green > Red, so the script is ordered in this way. To check your own settings, open the Brightness & contrast panel in QuPath to see the current order (Figure 2B)—you will need to open an image to do this. Adjust the order of the getColorRGB lines if needed (Figure 2A). Colors are set using RGB values (*i.e.*, 255, 0, 0, = pure red).

c. Channel names are set in the same order as the colors—change the names or the order as appropriate for your project.
Note: These channel names are referred to in other scripts, so if you use different ones here, you must remember to change them elsewhere.

d. In the Run menu, choose 'Run for Project' to apply this script to all images. The colors and channel names will change and will be reflected in the images (Figure 2C).
Note: If you have an image open while running the script, you will need to File > Reload data (select OK) to see the changes in the open image.

e. Below the Class list, open the More Options list (⋮) and select 'Populate from image channels'. When asked whether to keep existing available classes, select 'No'. The Class List should now reflect the channel names.

f. Open the More Options list again and create two new classes named 'Tissue' and 'Vessels'.



```
// =========== Set Channels and Colours ===================================================
// This sets the channels to the correct colours (which for .vsi files corresponds to the
// order in which they were scanned) and assigns appropriate channel names.

setChannelColors(
        getColorRGB(0, 0, 255), // Blue
        getColorRGB(0, 255, 0), // Green
        getColorRGB(255, 0, 0), // Red
)

setChannelNames(
        'DAPI',
        'GFP',
        'DsRed',
)

```

Figure 2. The QuPath Script Editor and Changing Colours and Channels.
A) QuPath's Script Editor with the Channels and Colours script open. B) Brightness & contrast panel when images are imported showing that the colors do not match the channel names provided by the VS120 Slide Scanner. C) Brightness & contrast panel after running the Channels and Colours script showing appropriate colors and channel names.

C. Determine optimal parameters

1. **Duplicate project**
   a. Close the Project (File > Project… > Close Project).
   b. In your file manager, navigate to the folder that contains the project and duplicate the entire folder. Now you have one project for optimization and one for analysis. Rename the folders appropriately.
   c. In QuPath, open the optimization project.
2. **Create small annotations for manual counting**
   Determining the optimal parameters for automated cell detection begins with manually counting the DAPI stained nuclei, DsRed labeled pericytes, and GFP labeled microglia in small annotations. These counts are compared to those from a range of automated detection parameters—in this case, differing intensity thresholds. To ensure the chosen thresholds are appropriate for each individual region, we placed test annotations in every brain region of interest and analyzed these regions independently (Figure 3). In brain regions with marked heterogeneity (*e.g.*, the cortex and hippocampus), two test regions were placed to reflect different characteristics of the region. The size of the test regions was set such that each contained ~100 nuclei. If your project contains a large number of images, you may choose to only optimize using a subset of images, in which case the other images may be deleted from the optimization project.
   a. Open the first image.

b. Objects > Annotations… > Specify annotation.

c. Check 'Use µm' then specify an annotation that is 300 µm wide, 200 µm high, and Name = 'Upper Cortex' (Figure 3A). Click Add annotation.
Note: These are the specifications and names of the small annotations that we used, but these can be changed to the size and names relevant to your project.

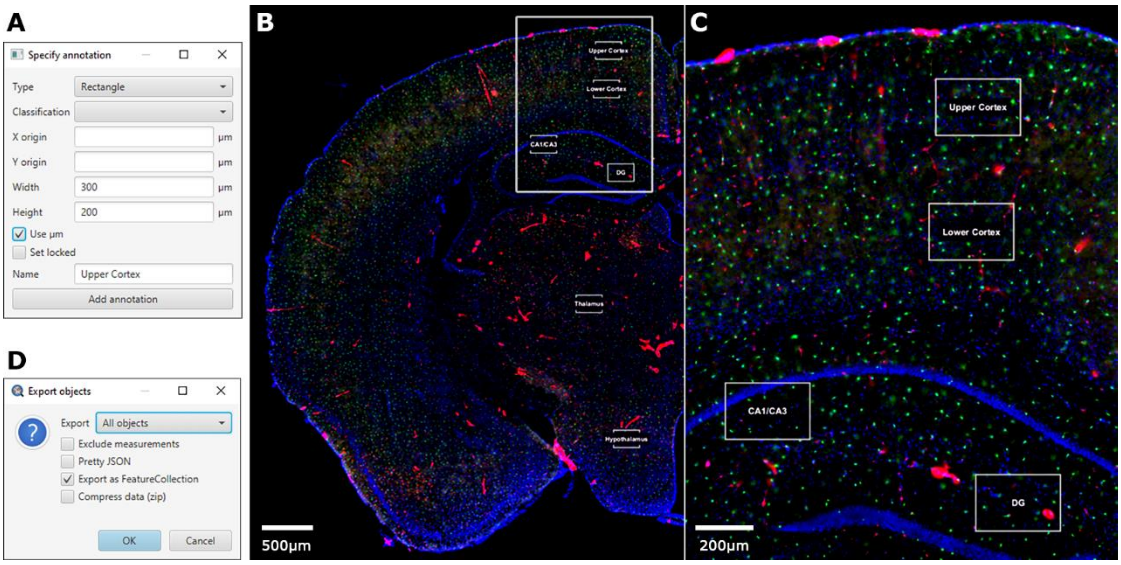
d. The Specify Annotation box will remain open, so continue to add another five annotations with the names 'Lower Cortex', 'DG', 'CA1/CA3', 'Thalamus', and 'Hypothalamus' (Figure 3B, C).

e. Close the Specify Annotation box. If needed, check View > Show Names.

f. Move each annotation to the appropriate position on the image. Try to avoid positioning the annotations on holes in the tissue or large blood vessels. Take care when moving annotations that you do not accidentally resize them—make sure you click in the center to move, not near the edges.

g. In the File menu, select Object data… > Export as GeoJSON. Select All objects and leave default options selected (Figure 3D). This saves a record of the annotations to the project folder as a .geojson file.

h. For each of the remaining images, drag and drop the .geojson file into the image to copy the annotations, then adjust their positions as appropriate.



Type: Rectangle
Classification:
X origin: µm
Y origin: µm
Width: 300 µm
Height: 200 µm
☑ Use µm
☐ Set locked
Name: Upper Cortex
Add annotation

Export: All objects
☐ Exclude measurements
☐ Pretty JSON
☑ Export as FeatureCollection
☐ Compress data (zip)

Figure 3. Placing and exporting annotations for optimization.
A) The Specify Annotation dialog box with the requirements to specify the first counting annotation. B) Left hemisphere of a brain section showing the positioning of the six counting annotations. C) Detail of area enclosed by white box in B. D) Settings for the export of objects.

3. Manually count cells

Before starting the process of manually counting cells, it is worth spending time examining the images and determining how you will decide which nuclei and which cells should be counted and marked as positive. Having specific criteria for making decisions before starting will ensure your counting is consistent.

a. Open the first image.

b. Adjust the zoom and Channels view to clearly see nuclei—usually, it is best to turn off all channels except DAPI and set to greyscale for clarity.

c. Navigate to the first 300 µm × 200 µm box.

d. Select the Points annotation tool—the Counting Window will open (Figure 4A).

e. Click Add three times to create three new points annotations. Double click them each, in turn, to rename them DAPI, GFP, and DsRed and change their colors to blue, green, and red, respectively.

f. Select the DAPI annotation and start placing points in the center of each nucleus (Figure 4B).

g. When you have identified all nuclei, change the Channel view so you can see the GFP expressing microglia.

h. Select the GFP annotation and check each annotated nucleus. If it is positive for GFP expression, add a green point alongside, or overlapping, the blue one (Figure 4C).
Note: Do not label a cell as GFP-positive if there is no nucleus marked. This prevents the accidental counting of fluorescent spots that are not actually cells and also ensures that sampling is consistently restricted to those cells for which the nucleus is in the focal plane. You may find it helpful to keep both the Counting and the Brightness & Contrast Windows open and adjust the view and points as necessary; however, do ensure that any adjustments to the brightness and contrast do not bias results.

i. Repeat for the DsRed annotation to label the DsRed-positive pericytes.

j. With the Hierarchy tab visible, choose Object > Annotations…> Resolve Hierarchy. This should insert the three points annotations into the parent Rectangle annotation.

k. Repeat steps C3e–C3j for each of the small rectangular counting rectangles.
Note: You will end up with multiple annotations called DAPI, GFP, and DsRed in the Counting Window—using the Hierarchy tab and ensuring you insert into hierarchy after completing each set makes it easier to keep track. You can select specific annotations in the Hierarchy tab, and they will be selected in the Counting Window.

l. Repeat this process for all other images.

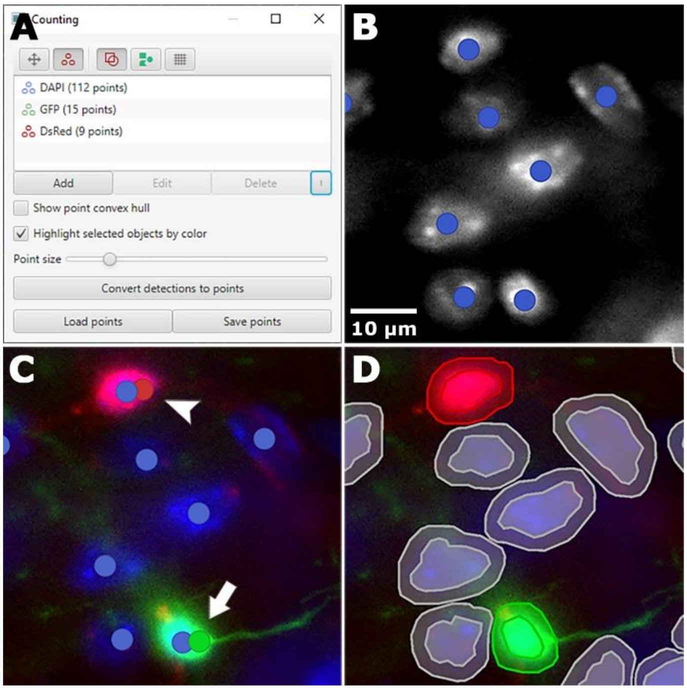


Figure 4. Example of manual count annotations and cell detections.
A) The Counting Window. B) DAPI-stained nuclei shown in greyscale and marked with blue point annotations. C) GFP (arrow) and DsRed (arrowhead) positive cells marked with green and red points annotations, respectively. D) The same cells following automated cell detection and classification.

4. **Export manual count data**

a. Ensure all images are saved, and then choose Measure > Export Measurements.

b. Add all images to the selected column. Click Choose and select the exports folder within the optimization project structure and enter an appropriate file name (*e.g.*, manual counts). For Export type, select Annotations, and for Separator, select Tab (tsv).

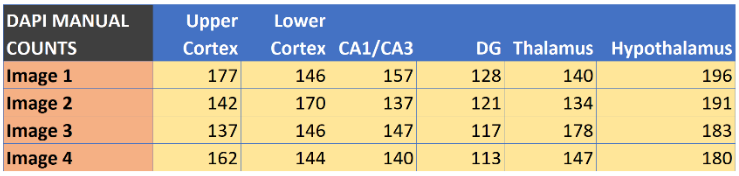
c. Click on Populate to populate the Columns to Include list. From this list, select: Image, Name, Parent, Num points.

d. Click Export.

e. Open the file with Excel.

f. Arrange the manual count data into three tables—one for each channel (DAPI, GFP, DsRed)—as in Figure 5.

Note: The layout and coloring shown in Figure 5 is not essential but will assist with the lookup process described in step C6d.



DAPI MANUAL COUNTS	Upper Cortex	Lower Cortex	CA1/CA3	DG	Thalamus	Hypothalamus
Image 1	177	146	157	128	140	196
Image 2	142	170	137	121	134	191
Image 3	137	146	147	117	178	183
Image 4	162	144	140	113	147	180

Figure 5. Table of manual counts of DAPI-positive nuclei from small annotations within multiple brain regions.

5. **Test parameters for cell detection**

Note: QuPath's Cell Detection algorithm offers a number of different parameters that can be changed to optimize cell detection. We have found that the most important parameter to optimize is the intensity threshold and, for our tissue, leaving other parameters at their default levels gives good results. Therefore, this protocol and its associated scripts only optimizes the threshold parameter. You may find that you need to further optimize other parameters, including background radius (the size of the rolling ball used to subtract background staining; it may be useful to optimize if there is a high level of background staining) and sigma (a measure of the level of smoothing which is applied; may need to be raised if nuclear staining is uneven or lowered if nuclei are often very close together). If this is the case, then the scripts associated with Courtney et al. (2021) offer the ability to optimize for sigma and background radius, as well as threshold, and could be further adapted for other parameters.

a. Find the Optimisation of Cell Detection script under Automate > Project Scripts.

b. Select Run > Run.

c. You will be presented with a series of input boxes to specify how many different threshold values you want to test for cell detection in the DAPI channel, as well as what the starting (lowest) test value should be and the amount to increment for the remaining values (*e.g.*, if you want to test thresholds of 150, 200, and 250 you should enter '3', '150', and '50', respectively).

d. When the process is complete, you will see a pop-up notification, and the data file will be saved in a folder called Optimisation Results in the project folder.

6. **Analyze cell detection parameters**

To determine the optimal cell detection parameters, the automated counts need to be compared to the manual counts using the formula:

$$\% \, Difference = \left(\frac{AutomatedCount - ManualCount}{ManualCount}\right) * 100$$

a. Open the DAPI Results.csv file from the Optimisation Results folder with Excel.

b. Delete all columns except those headed Image, Annotation, Threshold, and Cells. These should now be columns A, B, C, and D, respectively.

c. Copy the DAPI Manual Count table created in step C4f into the spreadsheet a few columns to the right of the data (*e.g.*, starting in cell I1).

d. Add the heading Manual to the first cell in column E, and enter the relevant manual count numbers for each image and annotation using the DAPI Manual Count table as a reference.

Note: As the number of rows is the product of images, regions, and thresholds (and sigmas and radii if you have tested them), the rows can number in the hundreds or thousands. Entering manual count numbers by hand is both tedious and prone to errors, so we suggest you use one of Excel's various *lookup functions to aid this process. There are multiple ways to achieve this, and the one you choose will likely depend on your level of familiarity with Excel, but one way, using INDEX-MATCH, is detailed here:*

i. Ensure that the image and annotation names in the DAPI Manual Count table from step C6c match those in the data table exactly.

ii. Select the cells containing the manual counts (yellow in Figure 5) and name this range "Counts" in the Name Box.

iii. Select the cells containing the image names (orange in Figure 5) and name this range "ImageNames" in the Name Box.

iv. Select the cells containing the regions (blue in Figure 5) and name this range "Regions" in the Name Box.

v. In cell E2 enter the formula =INDEX(Counts, MATCH(A2, ImageNames, 0), MATCH(B2, Regions, 0))

vi. Copy this formula down the entire column.

e. Create a new column (F) formatted as Percentage and calculate the percentage difference using the formula =(D2-E2)/E2. Copy this formula down the entire column.

f. For each region, use GraphPad Prism or similar software to graph the percentage difference at each threshold. Identify which threshold most reliably gives a percentage difference close to 0. This will be the optimal DAPI threshold for this region (Figure 6A).

g. For additional confirmation that the chosen threshold is appropriate, the correlation between manual counts and automated counts in each image for that threshold can be calculated (Figure 6B).

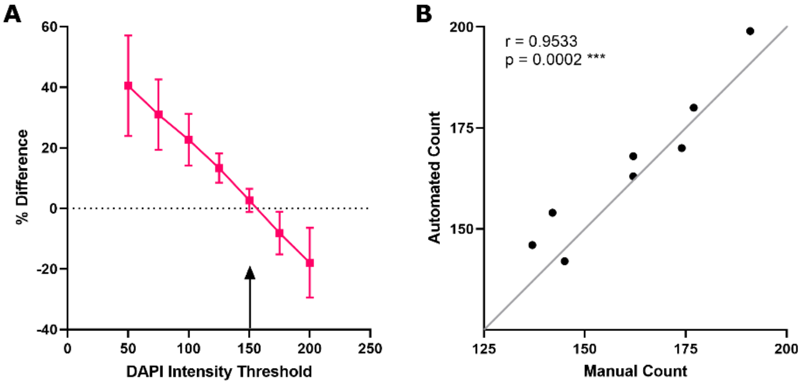


Figure 6. An example of graphs to determine the optimal DAPI threshold.
A) The mean (with standard deviation) percentage difference between manual and automated counts in one brain region is plotted against the fluorescence thresholds tested. The point at which the mean is closest to 0 (arrow) represents the optimal threshold. B) Correlation plot of manual against automated counts for the optimal threshold in a single brain region in multiple images. A Pearson Correlation Coefficient (r) approaching 1 suggests that the chosen threshold is appropriate.

7. **Test parameters for cell classification**

a. Find the Optimisation of Cell Detection script under Automate > Project Scripts.

b. Before running the script, you will need to adjust two lines of code as follows:

i. Line 16:
def regions = ['Upper Cortex', 'Lower Cortex', 'DG', 'CA1/CA3', 'Thalamus', 'Hypothalamus']
Adjust the region names to match your annotations (Note: this is why correct and consistent spelling and capitalization are crucial).

ii. Line 17:
def DAPIthresholds = [150, 150, 75, 75, 150, 150]
Adjust the DAPI thresholds to match the ones determined previously as optimal for each region.

Ensure the order is the same as in Line 16.

c. Select Run > Run.

d. When prompted, choose to optimize the GFP channel and set the required number, start and increment for thresholds.

e. The results will be saved in the Optimisation Results folder.

f. Repeat steps C7c–C7e for the DsRed channel.

g. Close the Optimization Project.

8. **Analyze cell classification parameters**

a. Open the GFP Results.csv file from the Optimisation Results folder with Excel.

b. Copy the GFP Manual Count table created in step C4f into the spreadsheet.

c. Create a new column and enter the relevant manual count numbers for each image and region. The same lookup strategy described in step C6d can be used here.

d. Create a new column formatted as Percentage and calculate the percentage difference:
(Cell Count – Manual Count)/Manual Count

e. For each region, use GraphPad Prism or similar software to graph the percentage difference at each threshold. Identify which threshold most reliably gives a percentage difference close to 0. This will be the optimal GFP threshold for this region.

f. Repeat steps C8a–C8e for DsRed.

D. Detect tissue and define regions of interest

1. **Define regions of interest**

a. Open the Analysis Project in QuPath. Open the first image.

b. Select the Brush tool.
Note: The diameter of the brush tool scales with image magnification—this setting and the starting diameter can be changed in Edit > Preferences > Drawing Tools. This tool allows you to click and drag to "paint" a region of interest. Regions can be further defined by clicking inside and pushing the boundaries out, or Alt-clicking outside and pushing the boundaries in.

c. Using the Allen Brain Atlas as a guide, draw regions corresponding to the cortex, hippocampus, thalamus, and hypothalamus on both left and right hemispheres.
Note: The regions may overlap the edges of the brain, and any holes in the tissue as these will be removed later (Figure 7A).

d. Merge the pairs of left and right hemispheres into a single annotation for each region.

e. Name each region (Cortex, Hippocampus, Thalamus, and Hypothalamus—be careful to use this exact spelling and capitalization) by right-clicking and choosing Set Properties.

f. Once the four regions have been defined in the first image, save the annotations using File > Object data… > Export as GeoJSON. Select All objects and leave default options selected. This saves a record of the annotations to the project folder as a .geojson file.

g. For each of the remaining images, drag and drop the .geojson file into the image to copy the annotations, then use the Brush tool to make minor adjustments to the annotations as needed.

2. **Detect tissue and remove vessels**

Note: The parameters for tissue detection used in our scripts may need to be adjusted for different tissues and scans. We recommend testing the script on a single image before running for all images, particularly as this script can take some time to run.

a. Open the script Tissue Detection.groovy (Automate > Project Scripts > Tissue Detection).

b. Run for Project with all images moved to the Selected list.

3. **Intersect regions of interest with tissue**

a. Open the script Intersect ROIs.groovy (Automate > Project Scripts > Intersect ROIs).

b. Run for Project with all images moved to the Selected list. Regions should now fit the edges of the tissue and have large DsRed positive vessels removed (Figure 7B).

4. **Visual check**

Visually check the tissue detection for each image and, if needed, use the brush tool to remove any areas with staining or scanning artifacts (*e.g.*, a fold in the tissue or a region that is out of focus; see Figure 7C–F for examples) or large vessels that were missed by the automated process and would interfere with the optimized cell detection/classification.

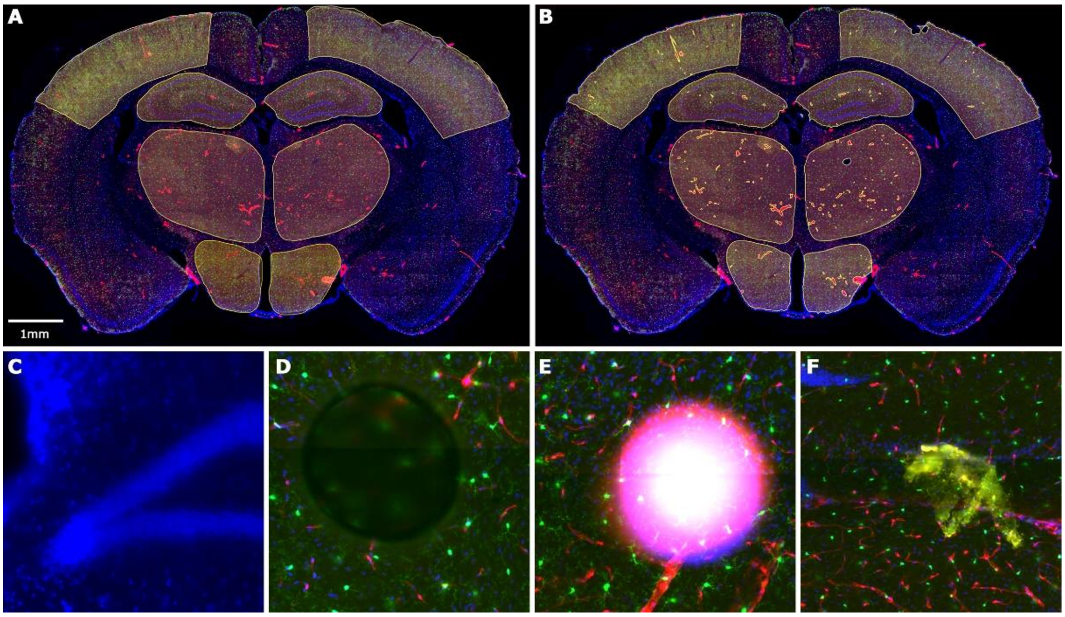


Figure 7. Region Annotations and Tissue Detection.
A) Brain region annotations following definition with the Brush Tool. B) Brain region annotations following tissue detection and intersection of ROIs. C–F) Lower panels show examples of artifacts that should be removed from the region annotations, including: C) an area where the automatic focusing has failed in the DAPI channel; D) a bubble in the mounting medium; E) a bright fluorescent patch of unknown origin; and F) a piece of debris.

E. Apply optimal parameters

1. **Create classifiers**
   As different regions are likely to require different detection thresholds, you will need to create a separate classifier file incorporating the optimal DsRed and GFP thresholds for each region.
   a. Within the classifiers directory of the Analysis project, create a new folder called 'object_classifiers'.
   b. Open the file DsRed-GFP Cortex.json with Notepad (or similar).
   c. Change the Threshold values to those determined in step C8 above and save the file with 'Cortex' replaced with the relevant region name in the object_classifiers directory.
   d. Repeat for each region.
2. **Insert optimal parameters into the analysis script**
   a. With the Analysis project open in QuPath, Automate > Project Scripts > Cell Classification
   b. Adjust lines 16 and 17 of the code as in step C7b above.
   c. Adjust line 18 of the code to correctly reference the classifiers you created in step E1.
3. **Run analysis for project**
   a. In line 20 of the code, make sure the regionNum = 1.
   b. Run for Project with all images moved to the Selected list.
   Following automated cell detection and classification, detected cells should appear as in Figure 4D.
   c. Export Annotation data for all images:
      i. Measure > Export measurements.
      ii. Move all images to the Selected pane.
      iii. Choose a save location (we suggest creating an 'outputs' folder within the Project directory) and

a filename with the appropriate region name.

iv. Set Export to Annotations and Separator to .csv.

v. Click Populate, then tick all columns to be included and click Export.

d. Open the .csv file in Excel and remove the rows for all annotations except the one you currently have chosen.

e. Repeat steps E3a–E3d for each region, changing the regionNum to 2, then 3, *etc.*

Note: Each region needs to be analyzed separately, and the data from that region needs to be saved and extracted after each analysis to make sure the appropriate cell detection parameters are used.

Data analysis

The specifics of data analysis will depend on the questions being asked. For details of our analysis, refer to Courtney *et al.* (2021) Figure 5.

Acknowledgments

This work was supported by the National Health and Medical Research Council for Australia grants APP1137776 and APP1163384 (BAS) and the University of Tasmania (DWH).
This protocol is based on that described in Courtney *et al.* (2021).

Competing interests

The authors declare no competing financial or non-financial interests.

Ethics

All animal procedures were approved by the Animal Ethics Committee, University of Tasmania (A0018608) and conformed with the Australian NHMRC Code of Practice for the Care and Use of Animals for Scientific Purposes – 2013 (8th Edition).

References

Bankhead, P., Loughrey, M. B., Fernandez, J. A., Dombrowski, Y., McArt, D. G., Dunne, P. D., McQuaid, S., Gray, R. T., Murray, L. J., Coleman, H. G., *et al.* (2017). QuPath: Open source software for digital pathology image analysis. *Sci Rep* 7(1): 16878.

Courtney, J. M., Morris, G. P., Cleary, E. M., Howells, D. W. and Sutherland, B. A. (2021). An automated approach to improve the quantification of pericytes and microglia in whole mouse brain sections. *eNeuro* 8(6): ENEURO.0177-21.2021.

Linkert, M., Rueden, C. T., Allan, C., Burel, J. M., Moore, W., Patterson, A., Loranger, B., Moore, J., Neves, C., Macdonald, D., *et al.* (2010). Metadata matters: access to image data in the real world. *J Cell Biol* 189(5): 777-782.