Sep 20, 2024 Version 2

🌐 QuPath Visualization/Segmentation V.2

Michael X. Henderson[1]

[1]Van Andel Institute

Michael X. Henderson: ORCHID: 0000-0001-9710-0726

| ASAP Collaborative Rese… | Team Biederer | 1 more workspace |

Michael Henderson
Van Andel Research Institute

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

Created: August 16, 2023

Last Modified: September 20, 2024

Protocol Integer ID: 108111

Keywords: ASAPCRN

Abstract

This protocol describes QuPath visualization/segmentation.

Attachments

812-2118.pdf
4.6MB

Guidelines

QuPath is a visualization and segmentation platform optimized for whole slide images. This 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.

Part of **SPRINGER NATURE**
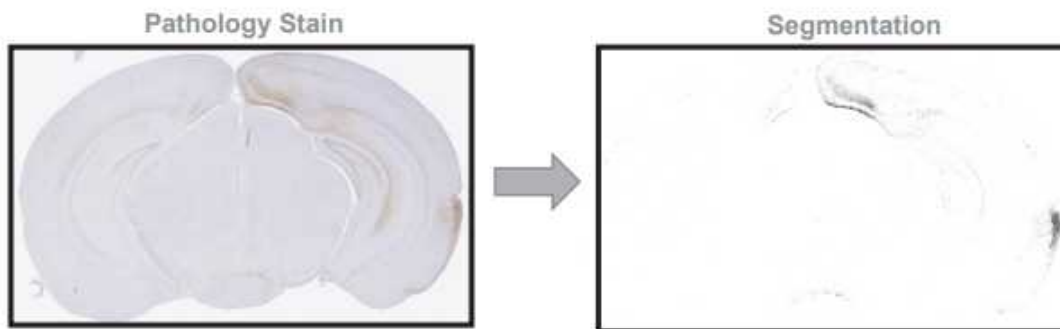
QuPath Visualization/Segmentation

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.
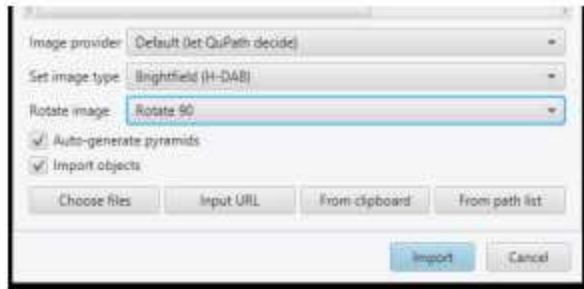
3   Upload series of images into new project on QuPath. Don't worry about what the images are named or how they are ordered yet.

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 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).

9.1   Click on the "Image" tab.

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

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

9.4   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.

10.1   Right click on the rectangle annotation in the annotation tab and select "Set Properties".

10.2   Write the abbreviated name you would like your input file to be denoted as in the 'Name' box and select 'OK'.

10.3 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."

> **Note**
>
> *This is a place where you could threshold by size or split object, if desired

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).

16.1 Click on to the slide, zoom in, and select the brush tool .

16.2 To erase, hold down the Alt key while erasing artifacts.

> **Note**
>
> **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().getMeta
data().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 corresponding to your classifier's resolution.

20   Plate the image and place into "**Input**" folder that will be used in final Nutil step.