



Introduction to Bioimage Analysis using QuPath

Antoine A. Ruzette & Simon F. Nørrelykke

Image Analysis Collaboratory



Workshop plan

1. Introduction to digital image analysis
2. Installing QuPath and your first project
3. GUI layout and toolbars
4. Introducing objects: annotations and detections
5. Saving, sharing and receiving QuPath projects
6. Nuclei detection and measurements (incl. StarDist)
7. Cell classification
8. Automating tissue annotations (pixel classifier)
9. Advance topic: scripting and workflows

Acknowledgments

- **Pete Bankhead et al.**
 - QuPath and its amazing documentation
- **Peter Sobolewski**
 - *Introduction to QuPath* workshop at the The Jackson Laboratory
- **Nina Kozlova**
 - Whole-slide image used in this workshop

Self-introductions

1. My **name** is *Antoine*
2. My **position** is as an *Associate in Systems Biology*
3. My **lab** is *the Image Analysis Collaboratory and the Megason Lab*
4. I have **microscopy images** of *cancer tissues, embryos, brains ...*
5. A **fun fact** about me is *I created a microbrewery at 19*

Self-introductions

1. **Motivate** the use of algorithms in image analysis
2. **Introduce** some image-analysis nomenclature
3. **Learn** to use QuPath effectively and reproducibly

Reasons to learn image processing

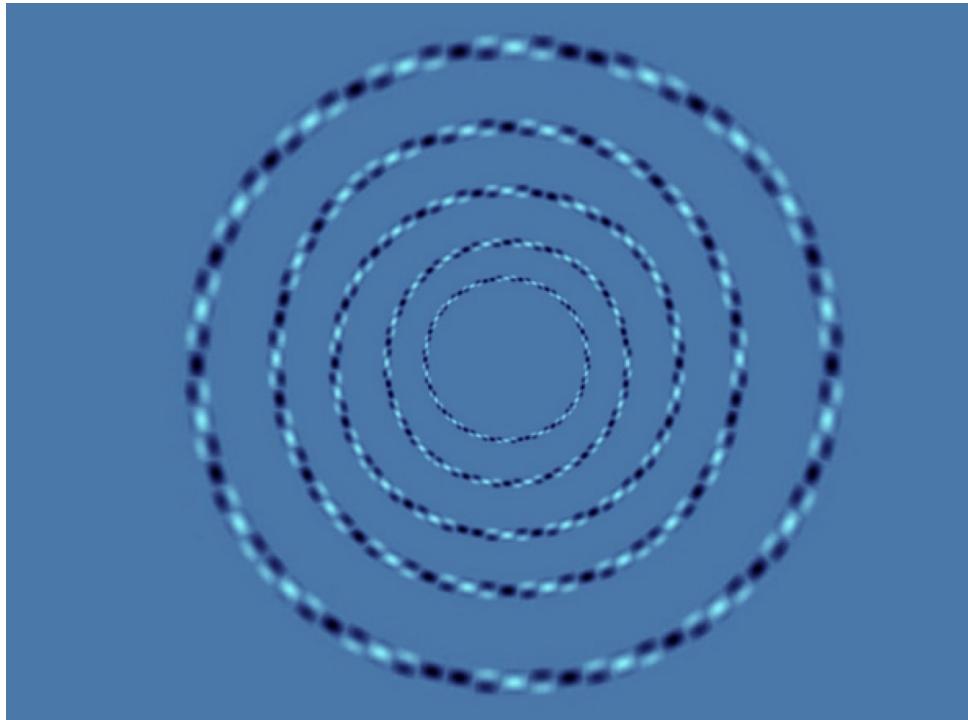
- Make pretty pictures (processing)
 - publications, talks, websites, ...
- Get numbers out of pictures (analysis)
 - cell sizes, vessel lengths, GFP expression level, ...
- Make experiment possible (automation)
 - whole-genome screen: millions of images
- Objectivity and Reproducibility
 - in science, it's your duty!

Reasons not to learn image processing

none

Why should we analyze images with computers at all?

Color perception and pattern recognition is individual – science less so

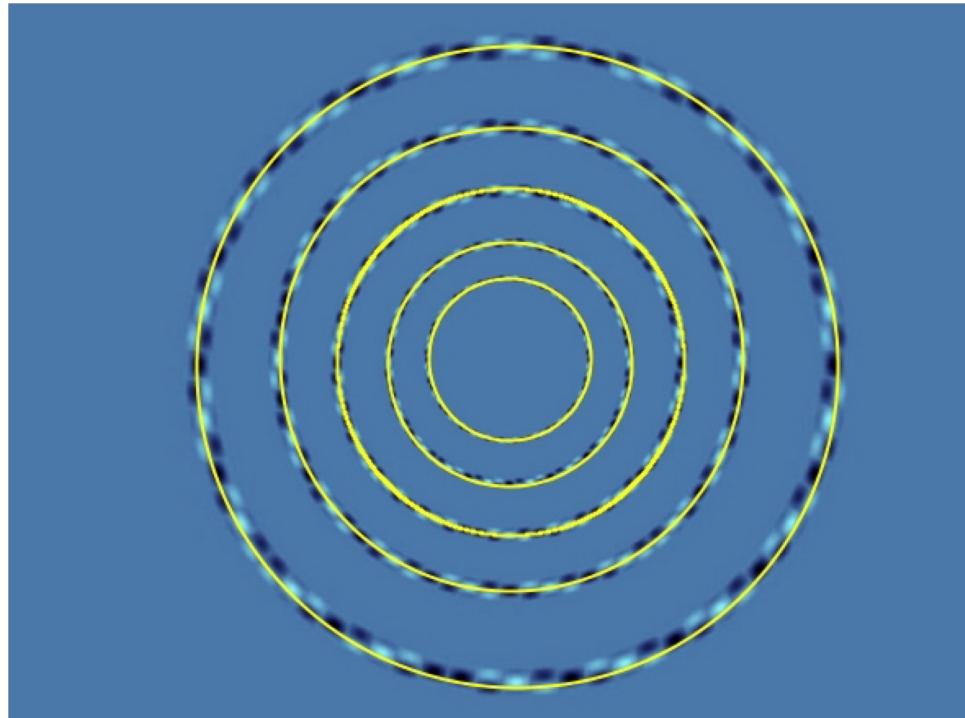


<https://www.moillusions.com/perfect-circles-optical-illusion/>

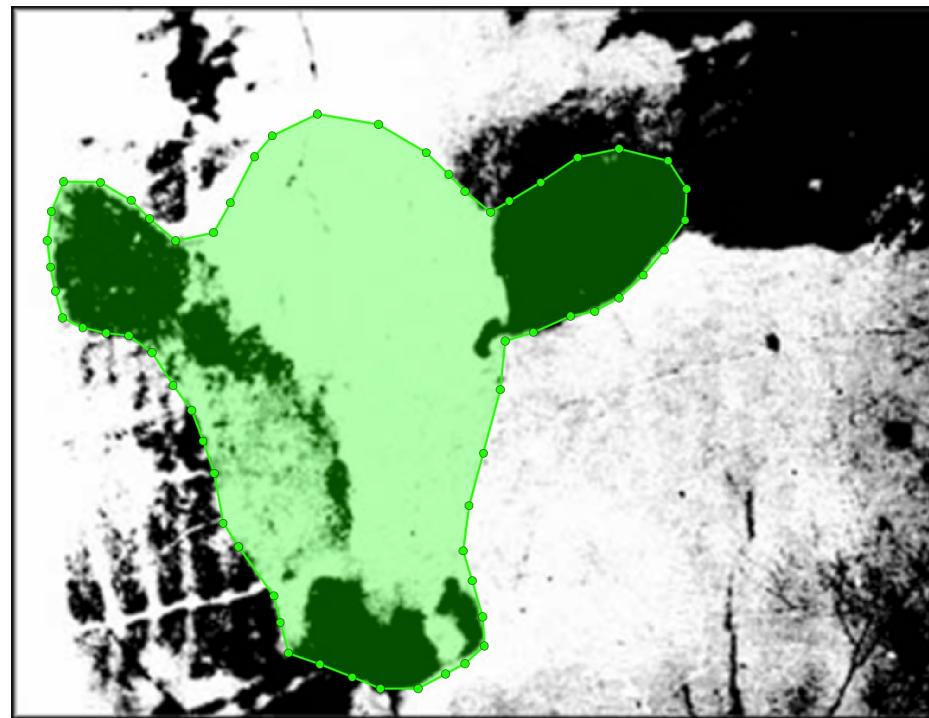


<http://www.brainbashers.com>

Color perception and pattern recognition is individual – science less so



<https://www.moillusions.com/perfect-circles-optical-illusion/>



<http://www.brainbashers.com>

In other words,

“Each human brain is a very complex neural network trained on different data – predictions will vary”

Antoine

A typical image analysis workflow

- There are typically *five* steps in an image analysis
- Often a good idea to structure work along these lines before starting

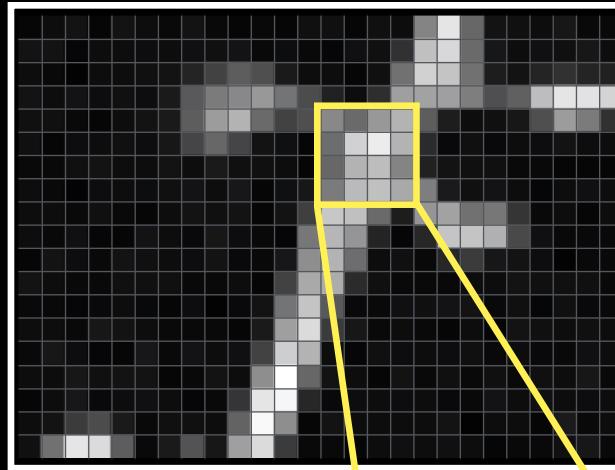


Think of this even ***before*** you acquire the images!

otherwise image analysis may become only a *post-mortem* on your experiment

What is an image?

A digital image is a matrix of numbers!



=

6	13	19	6	19	13	9	19	9	6	16	16	6	13	132	229	103	19	16	13	23	9	9			
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	23	13	
13	9	4	13	13	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	33	
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	212	
16	16	9	6	13	19	26	93	156	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6
13	9	4	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13	
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9
19	13	9	9	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	9		
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	16	16	19	19	9	16	6	16	9	9	
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	9	16	13		
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9
6	13	23	9	9	13	16	13	6	9	53	229	246	39	9	13	13	9	9	19	13	16	13	13	13	13
13	19	59	76	26	9	16	13	99	249	142	6	19	13	13	13	13	19	4	13	13	6	26	9	13	
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	6	9	9	16	23	9		

=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

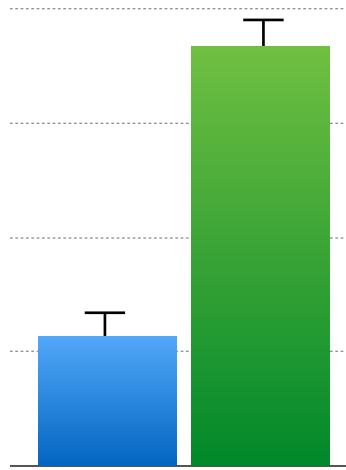
Pixel = Picture Element

Images in publications and presentations
should be used to **communicate** a finding...
not **be** the finding

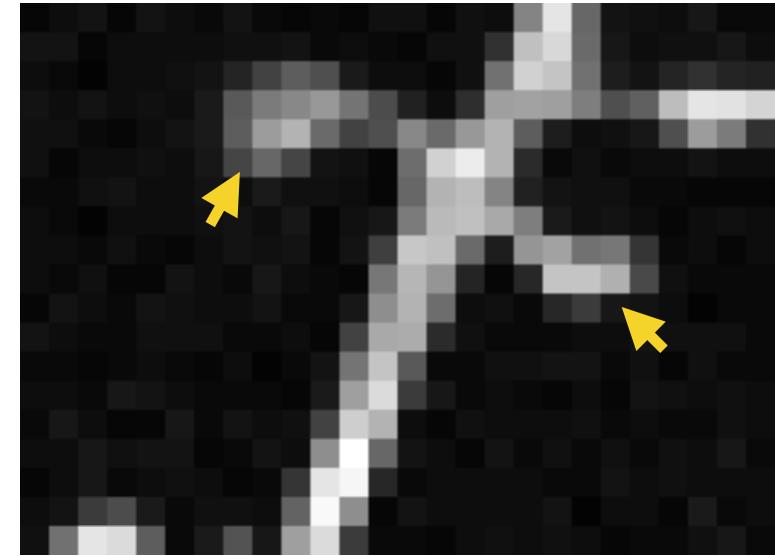
this is your **data**

6	13	19	6	19	13	9	19	9	6	9	6	16	16	6	16	13	132	229	103	19	16	13	23	9	9	
19	19	6	13	13	13	13	16	16	19	9	13	9	6	16	16	49	192	216	106	23	13	16	16	23	13	
13	9	4	13	13	16	16	19	36	66	93	79	26	13	13	6	16	113	209	196	113	29	19	36	49	36	33
19	13	19	13	16	13	26	89	123	136	152	116	76	33	13	46	159	162	159	126	79	96	189	229	226	212	
16	16	9	6	13	19	26	93	158	179	106	66	79	136	106	152	179	93	29	13	16	23	79	156	123	49	
16	6	13	13	16	13	23	69	103	69	19	16	6	109	209	236	179	43	9	16	9	13	13	19	13	13	
9	9	16	19	13	13	19	13	26	16	16	13	6	103	179	189	132	33	19	16	16	9	9	6	6	6	
13	9	4	13	13	13	16	19	13	23	6	16	23	123	186	192	169	126	26	16	19	13	6	13	16	13	
13	13	9	16	9	6	13	19	16	19	6	19	63	199	192	106	29	149	162	113	119	53	9	13	6	13	
13	9	16	6	6	19	13	9	23	13	9	6	119	182	149	36	6	39	196	196	176	73	16	9	9	9	
6	19	13	9	19	16	13	13	19	9	9	23	142	179	109	13	16	9	39	59	23	19	13	4	9	9	
19	13	9	9	16	16	16	9	9	13	6	66	169	172	43	16	9	9	9	13	13	19	16	16	16	9	
9	9	6	9	13	9	6	13	4	9	19	116	196	89	9	9	16	16	19	19	9	16	6	16	9	9	
13	13	9	23	19	13	9	9	9	6	26	159	219	59	23	9	13	9	6	13	6	19	16	13	16	13	
9	23	13	6	6	23	9	19	13	16	66	206	179	13	6	16	13	13	13	16	9	13	9	9	16	13	
13	13	23	16	19	19	6	9	19	13	142	255	103	19	13	6	19	9	16	9	16	9	16	13	23	9	
6	13	23	9	13	16	13	6	9	53	229	246	39	9	13	13	13	9	9	19	13	16	13	13	13	13	
13	19	59	76	26	9	16	16	13	99	249	142	6	19	13	13	13	19	4	13	13	6	26	9	13		
16	113	229	219	93	9	26	83	23	159	219	59	9	9	6	13	16	13	16	13	6	9	9	16	23	9	

this is your **result**



this just helps to
communicate the result

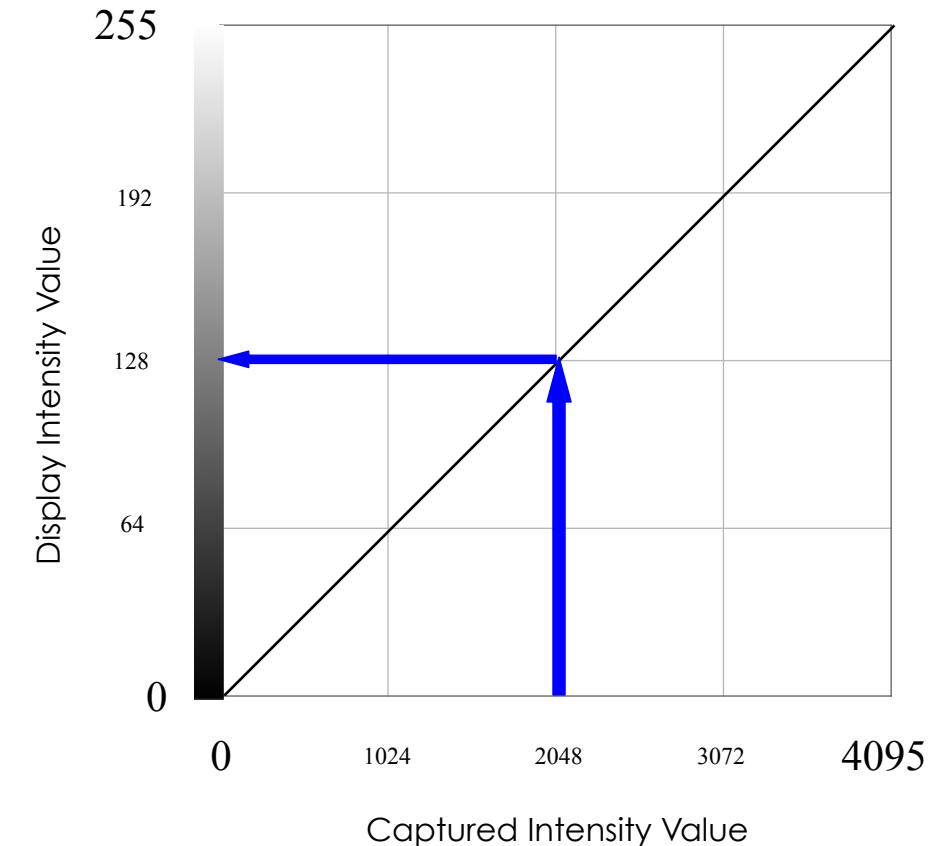
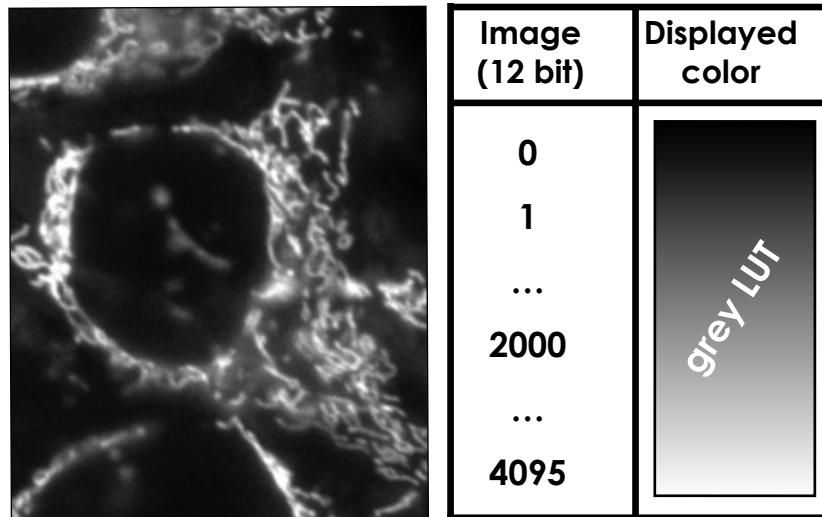
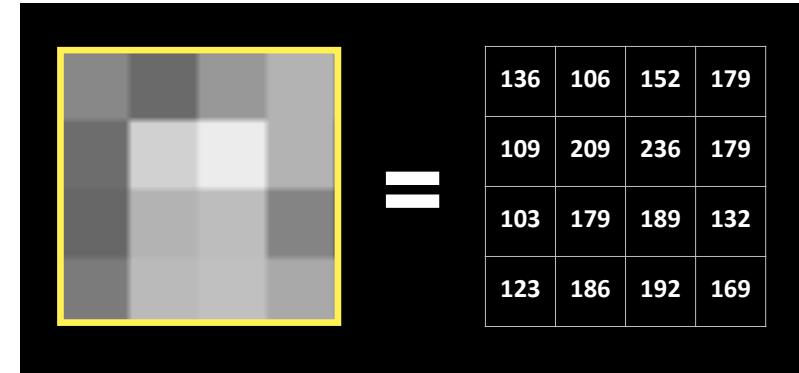


Display your images

Mapping Image Intensity to Monitor Intensity (LookUp Tables)

LUT = how the grey values are displayed

LUTs do not change the pixel values



Images and Colors

Lookup Tables (LUTs)

LUT = how the grey values are displayed

LUTs do not change the pixel values

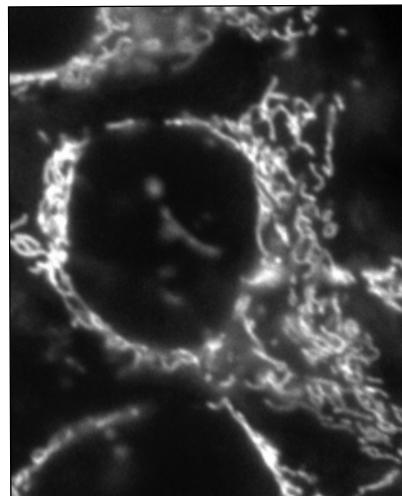


Image (8 bit)	Displayed color
0	
1	
...	
100	
...	
255	grey LUT

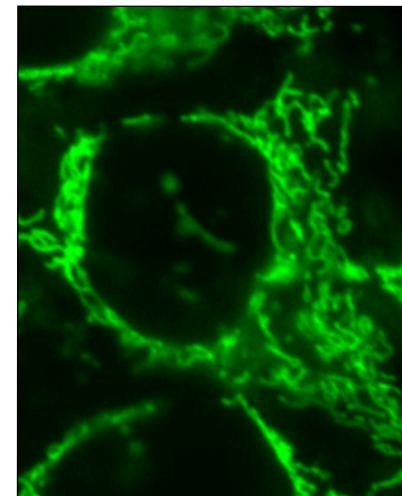
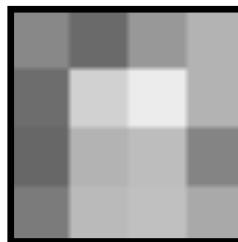
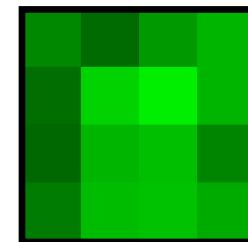


Image (8 bit)	Displayed color
0	
1	
...	
100	
...	
255	green LUT



=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169



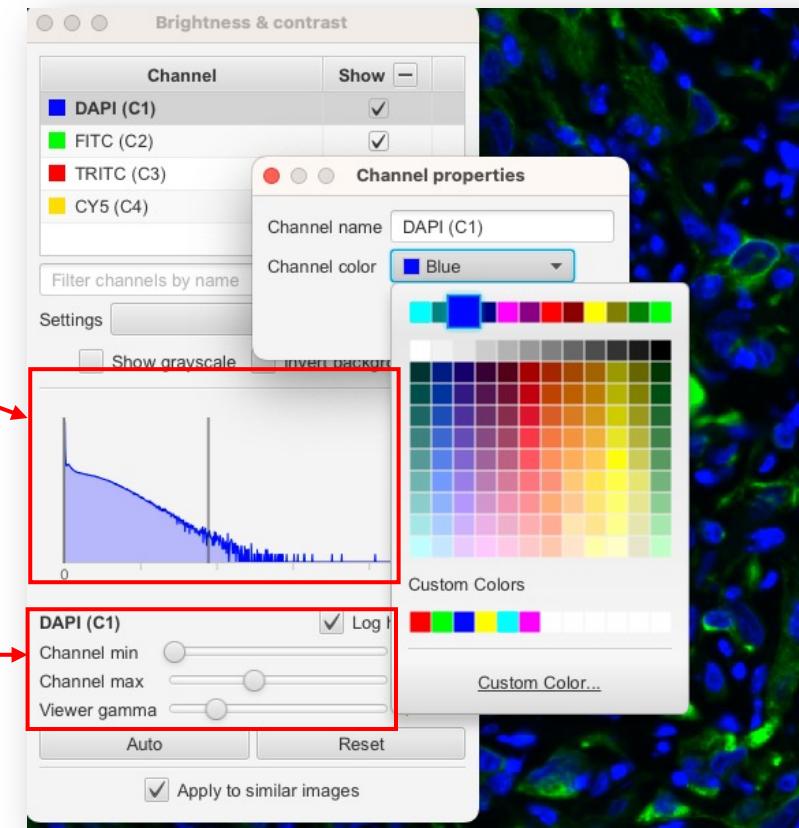
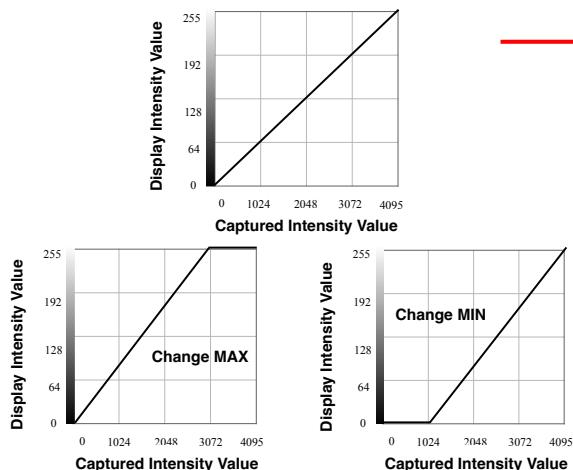
=

136	106	152	179
109	209	236	179
103	179	189	132
123	186	192	169

Display images: color, brightness & contrast

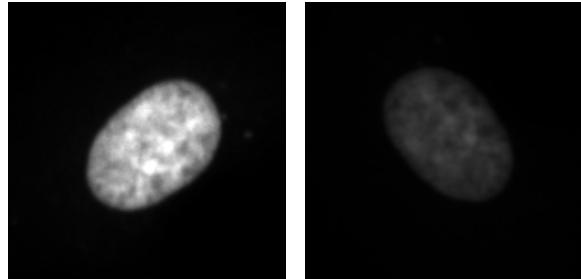
- If you are imaging a blue fluorophore, you are not forced to display it in blue!
- Pixel histogram represents the distribution of pixel values in the image
- LUT range

*You are **NOT** changing the pixels values, you are just changing how the image is displayed (unless you click on the "Apply" button).

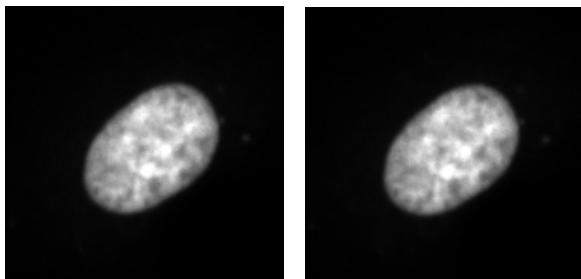


Display a file: Brightness & Contrast

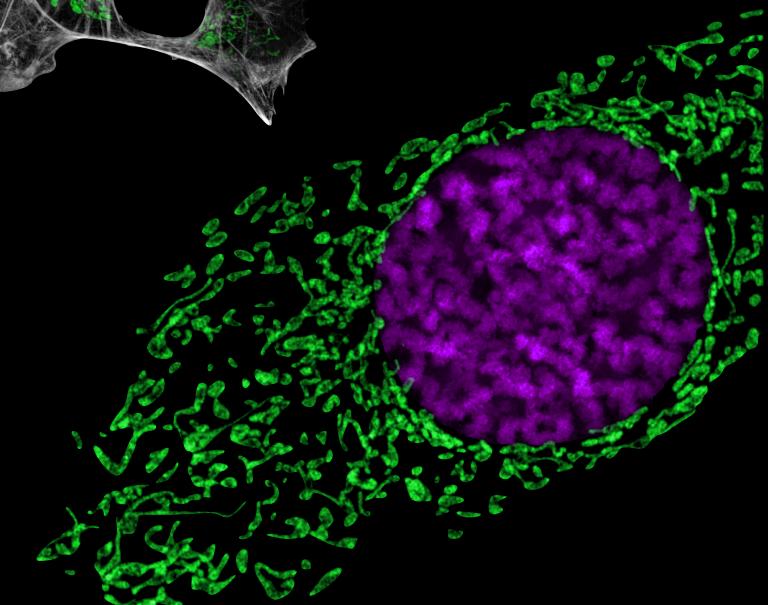
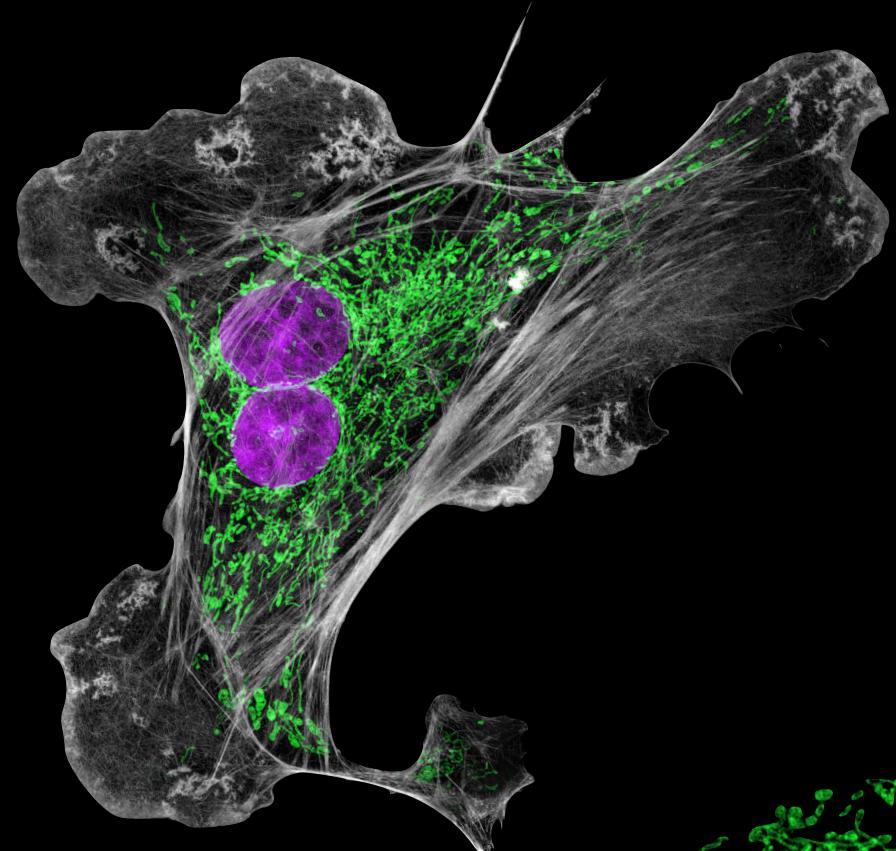
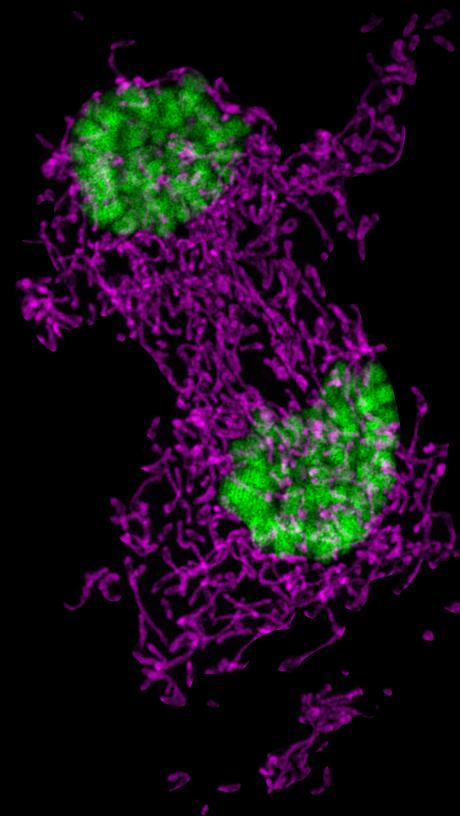
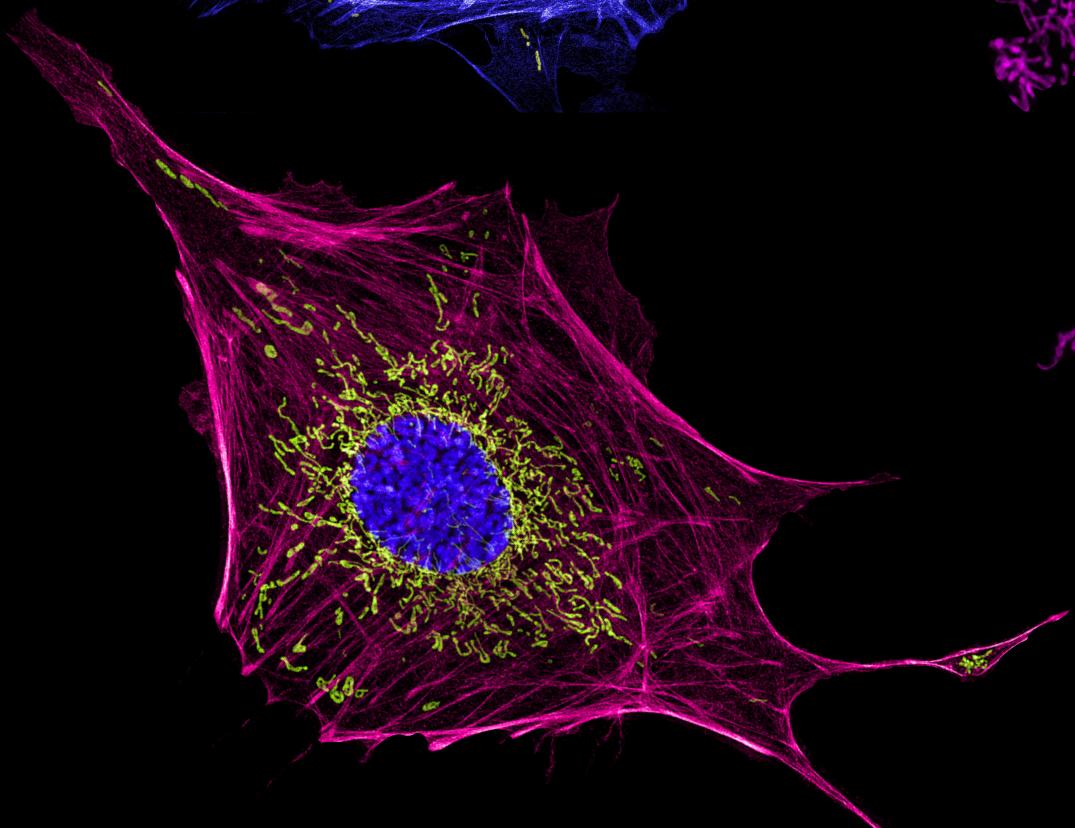
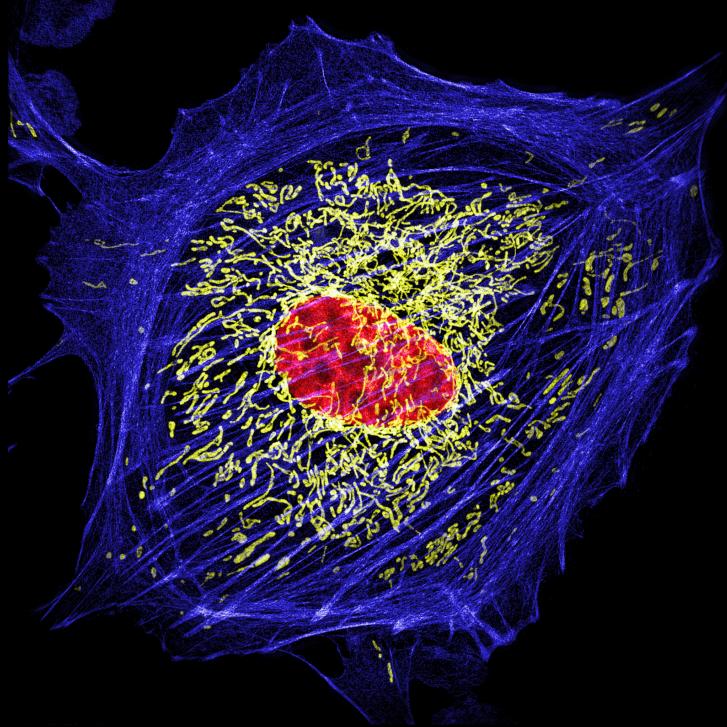
Which image has more fluorescence?



Mean:	4803	4803
Display range:	188- 16828	188- 45514



Mean:	4803	4803
Display range:	188- 16828	188- 16828



Talley Lambert

Introduction to QuPath



Download the course materials



https://drive.google.com/drive/folders/1d75blbKTOtHpqw5olvx8ryktxSeeui9O?usp=drive_link

What is QuPath?

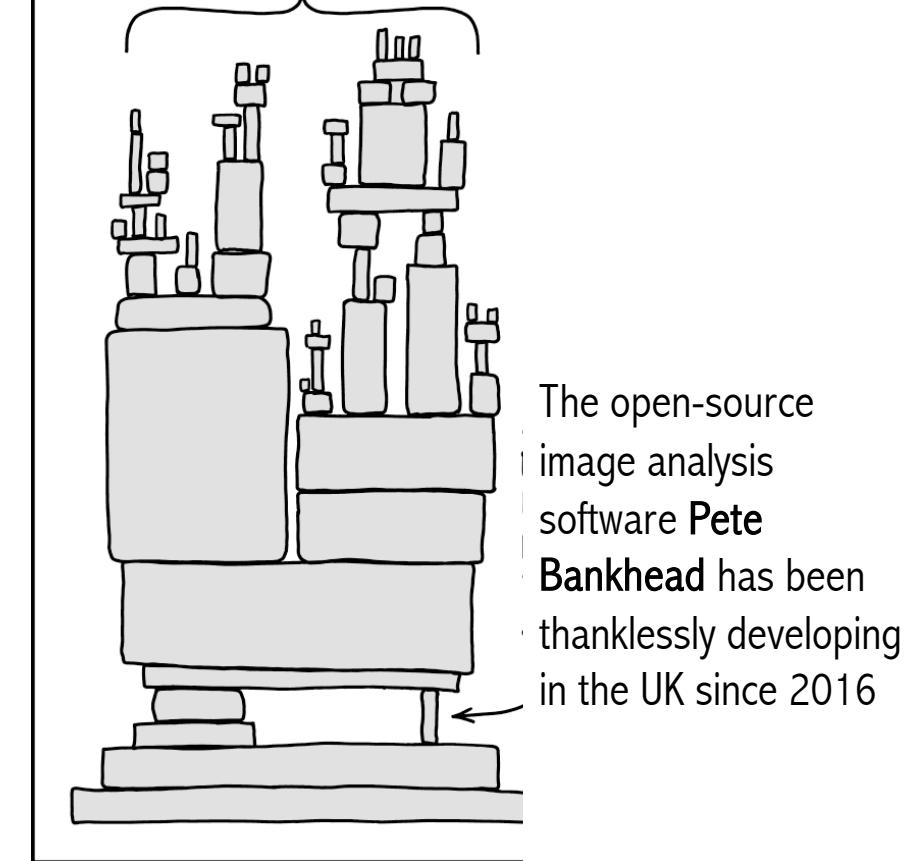
QuPath is an open-source software for bioimage analysis

- Developed and maintained by Pete Bankhead and his team at the University of Edinburgh

Key features:

1. Performant when working with very large 2D images, like those produced by slide scanner
2. Extremely well maintained

The world's most complex biomedical research



Adapted from <https://xkcd.com/2347/>

What is QuPath good at?

- It has a nice graphical user interface (GUI)
- It was designed to handle very large 2D images
- It supports common image analysis tasks: segmentation, annotation, feature analysis, and classification
- It support extensive visualization options
- It integrates with many other existing tools (Stardist, ImageJ, ...)
- It support scripting (Groovy, akin to Java)

What is QuPath not good at?

- Limited to 2D images
 - Only supports the visualization of single planes
- Does not support all file format (e.g. zarr/NGFF)



Installing Qupath

```
-----INSTALL.SH-----  
#!/bin/bash  
  
pip install "$1" &  
easy_install "$1" &  
brew install "$1" &  
npm install "$1" &  
yum install "$1" & dnf install "$1" &  
docker run "$1" &  
pkg install "$1" &  
apt-get install "$1" &  
sudo apt-get install "$1" &  
steamcmd +app_update "$1" validate &  
git clone https://github.com/"$1"/"$1" &  
cd "$1";./configure;make;make install &  
curl "$1" | bash &
```

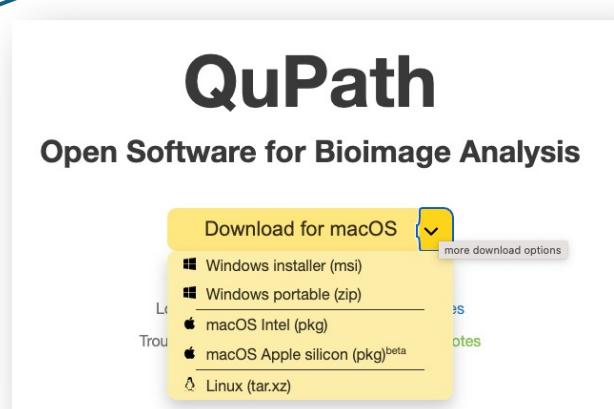
<https://xkcd.com/1654/>

Download QuPath

1. Go to <https://qupath.github.io/>
2. Download the installer for the latest version



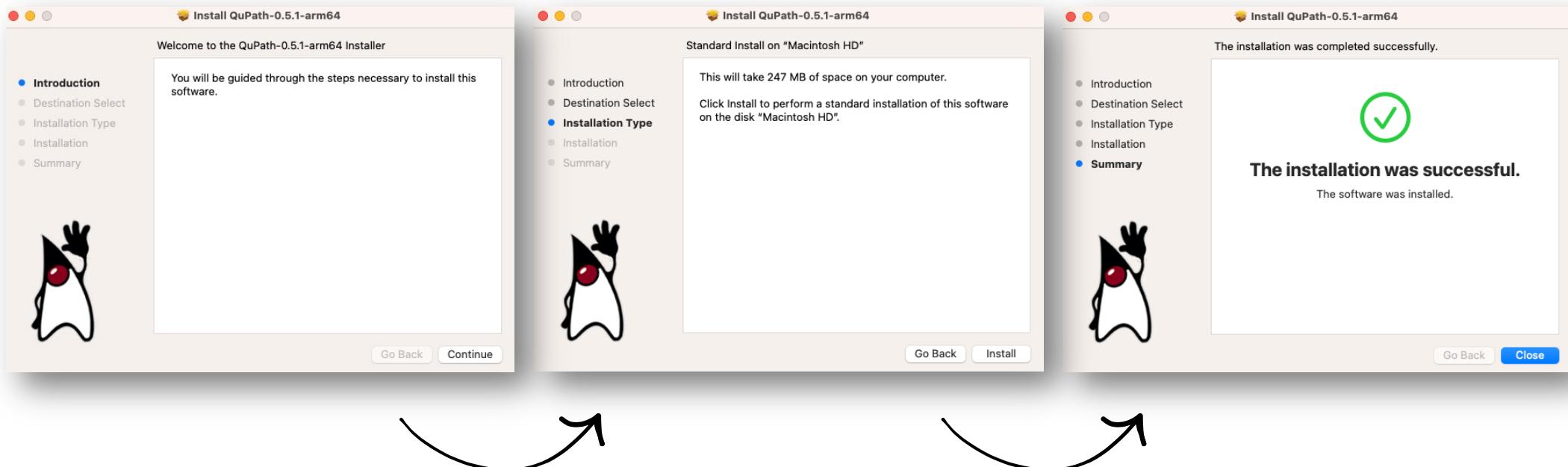
3. Choose your OS



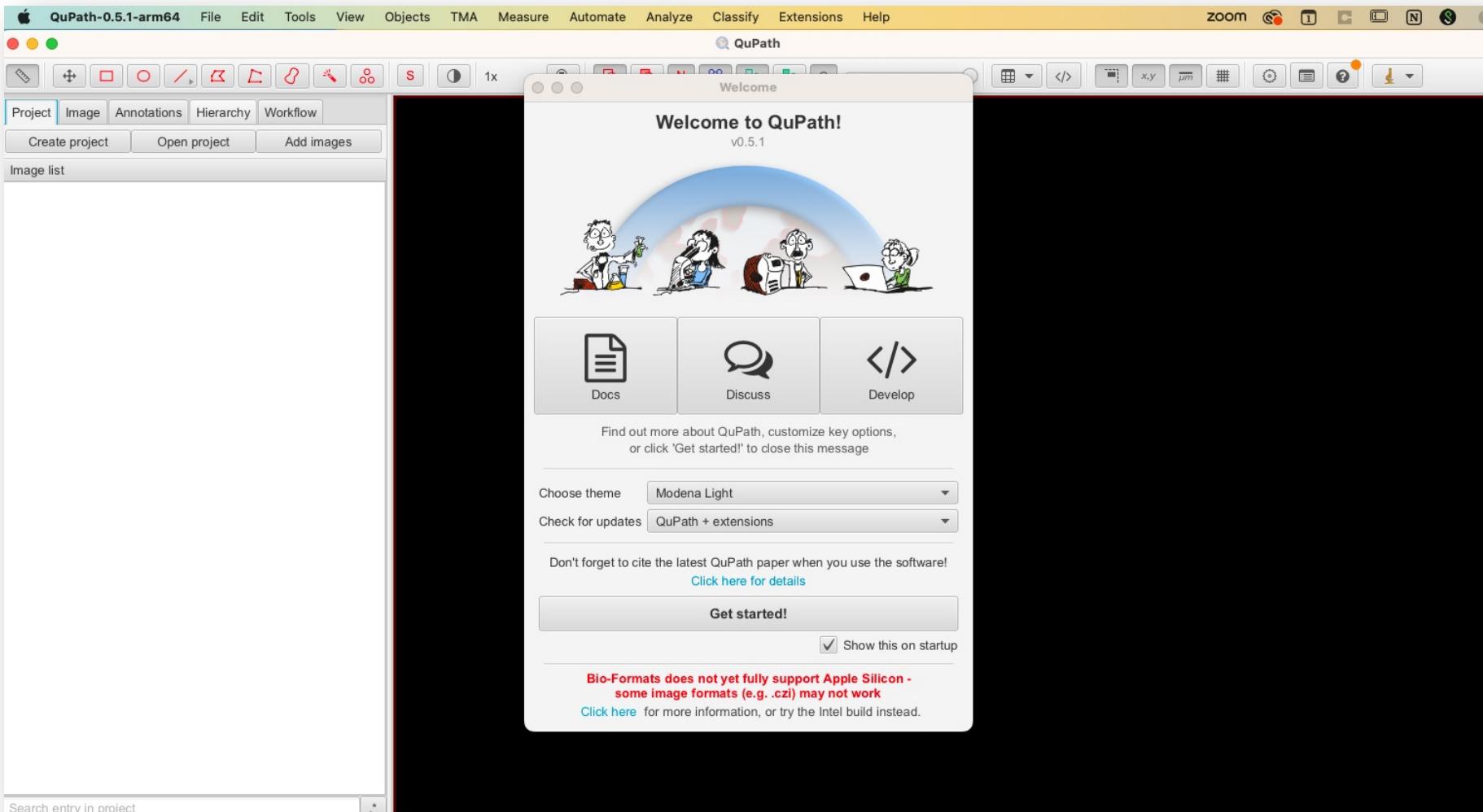
4. Right-click on the installer file > Open > Confirm Open

Download QuPath

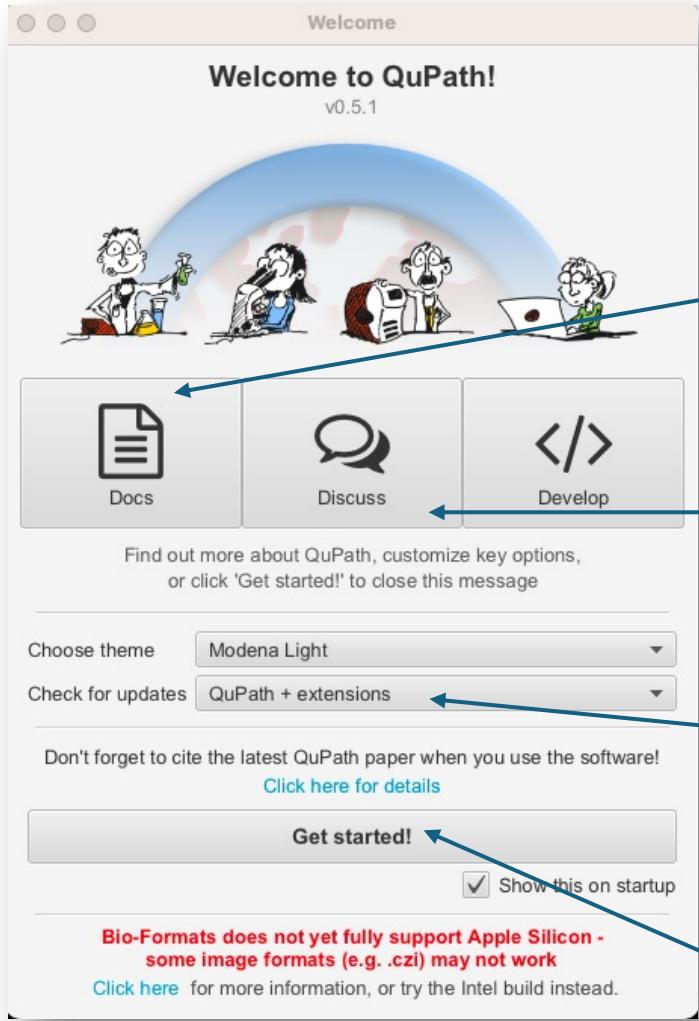
Follow the steps of the installer



Open the QuPath application



Welcome to QuPath!



Useful resources:

Documentation:

<https://qupath.readthedocs.io/en/0.5/>

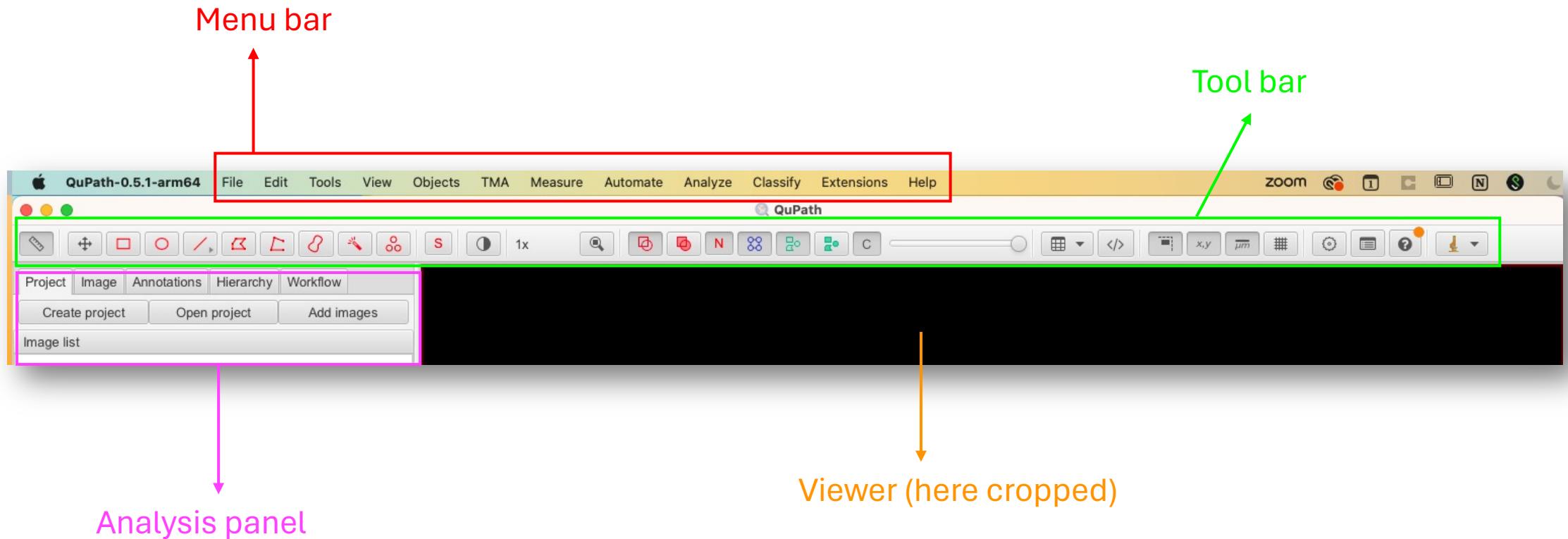
QuPath-specific place in ‘The Forum’:

<https://forum.image.sc/tag/qupath>

Updater

For now, let’s get started

Graphic User Interface (GUI) – intro

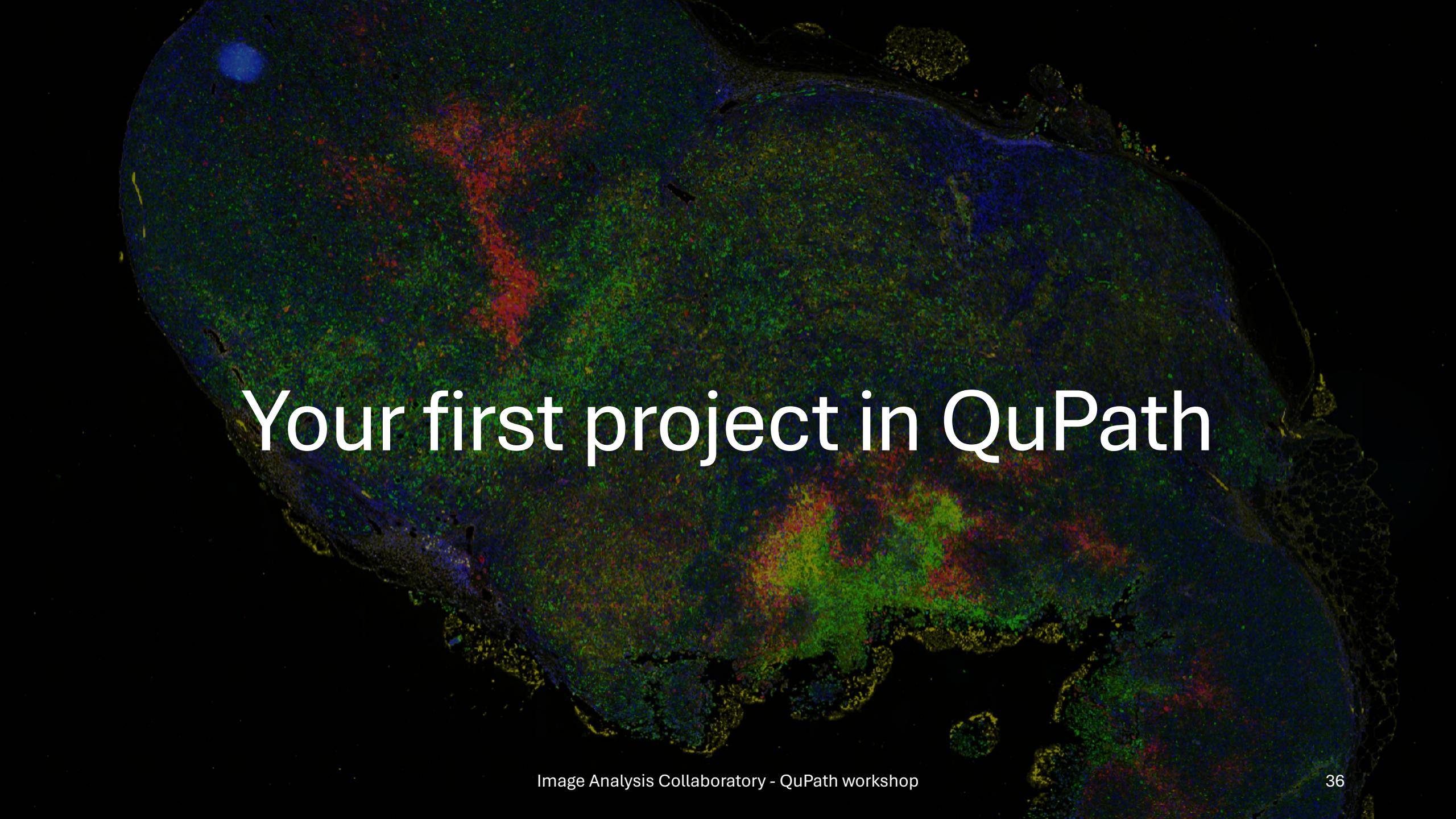


Getting help

- In-app documentation: Help menu



- QuPath documentation: <https://qupath.github.io/>
- The Forum: <https://forum.image.sc/>  image.sc
- During this workshop, ask questions to your neighbors, the TA's and me!

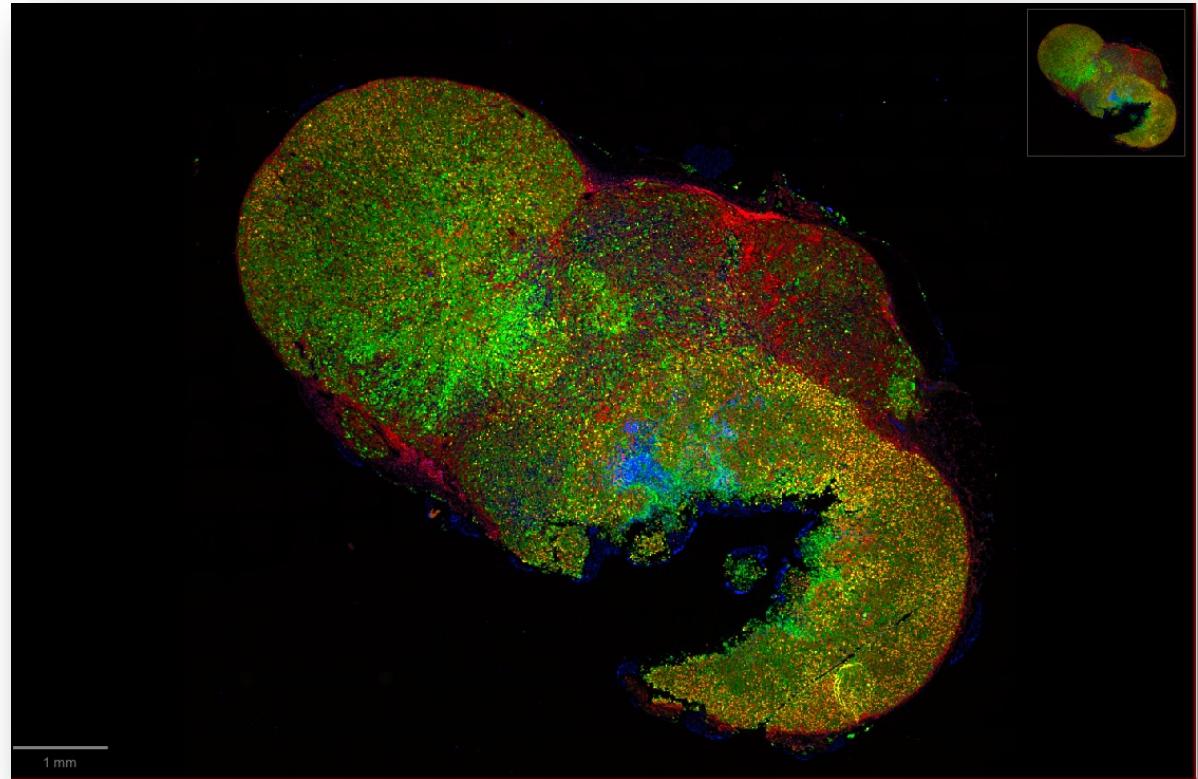
A fluorescence microscopy image showing a cross-section of tissue. The image is stained with multiple colors: green, red, blue, and yellow. The green signal appears throughout the field, while red is concentrated in a central region. Blue staining is visible at the top left and bottom right. Yellow staining is present along the outer edges and some internal structures. The overall image has a dark background.

Your first project in QuPath

Classification of proliferating cancer cells in solid tumors

- Whole-slide image
 - Already been stitched
- 4 channels
 - DAPI
 - Keratin (FITC)
 - Fibronectin (TRITC)
 - Ki67 (CY5)

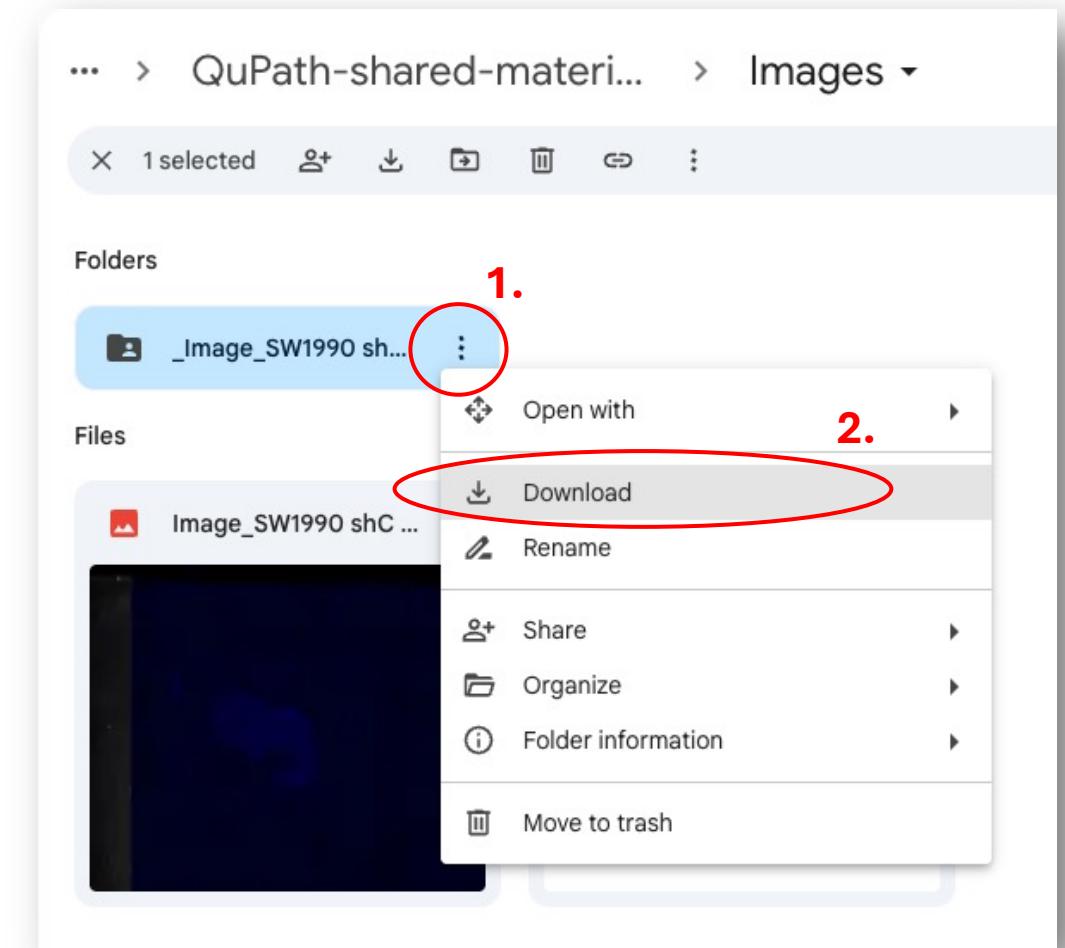
At the end of this course:
you will have classified
proliferating (Ki67) cancer
cells and reveal their spatial
distribution to regions with
high-fibronectin content



Courtesy of Nina Kozlova, PhD

Download the image from the shared folder

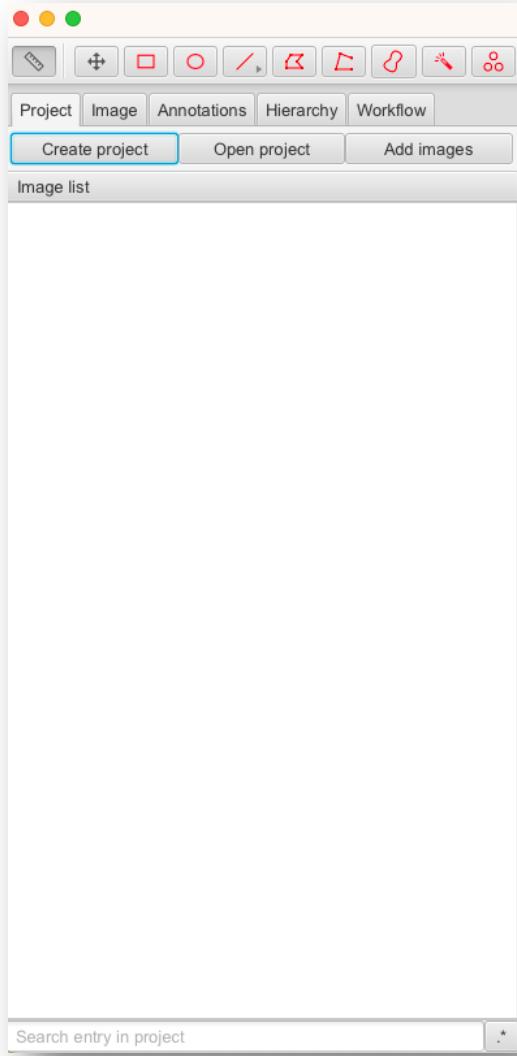
1. Download the whole folder from the Google Drive
 1. Image size: ~3GB; it will take a minute or two to download
2. Unzip it
3. Transfer the image in a new **Images** folder in your QuPath project folder



Key concept: QuPath project

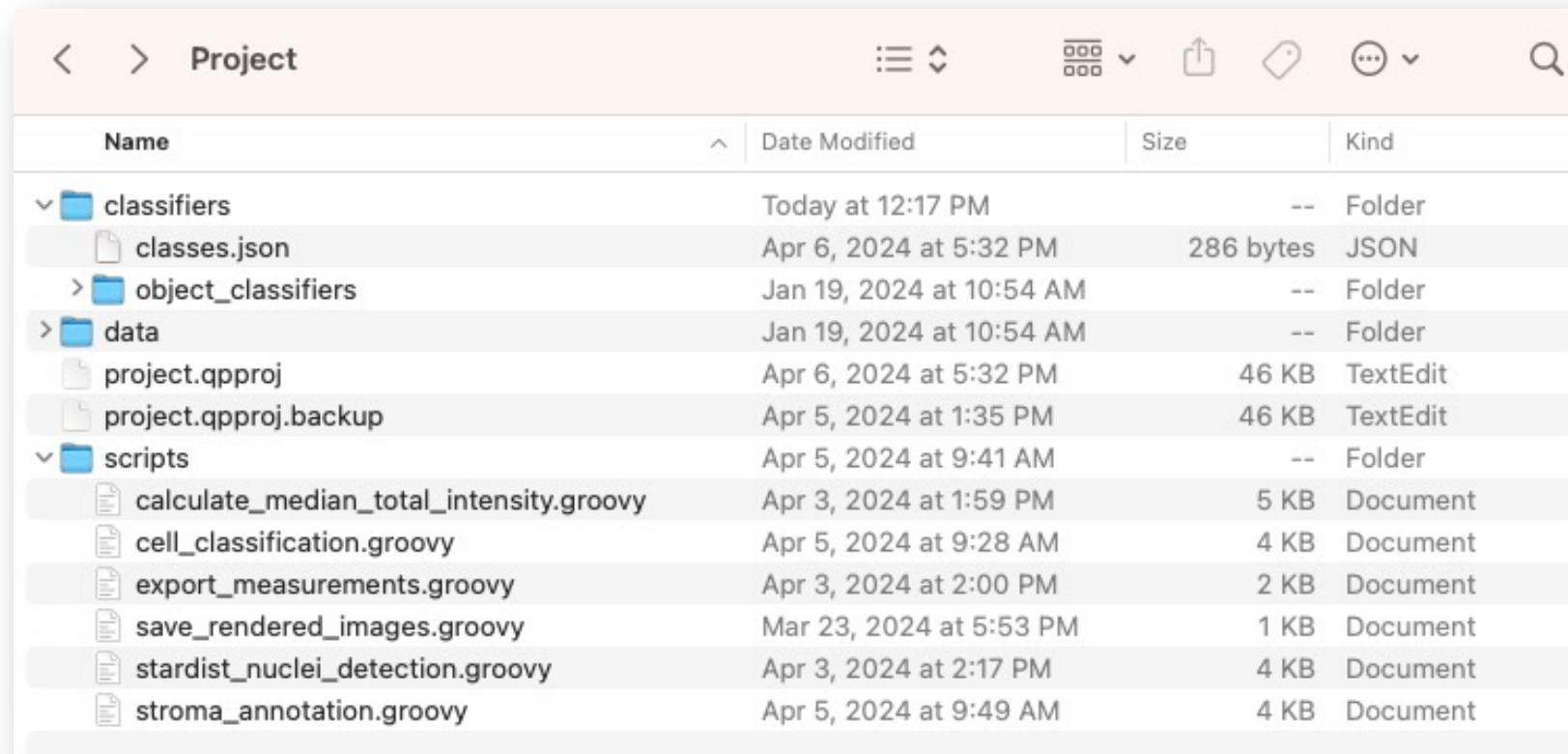
- Projects are the way to organize your work in QuPath
- In other words, they are folders
 - Group together images
 - Organize data, scripts, classifiers, etc
 - They only save data, not the original images
- Allow you to share your work with other QuPath users
 - Always send the images along!

How to create a project?



- *Create project* button
or
- File > Project... > Create project
- ! Make sure to create an **empty** folder for your project
 - Sometimes, you have to do this twice in the empty folder

Anatomy of a QuPath project

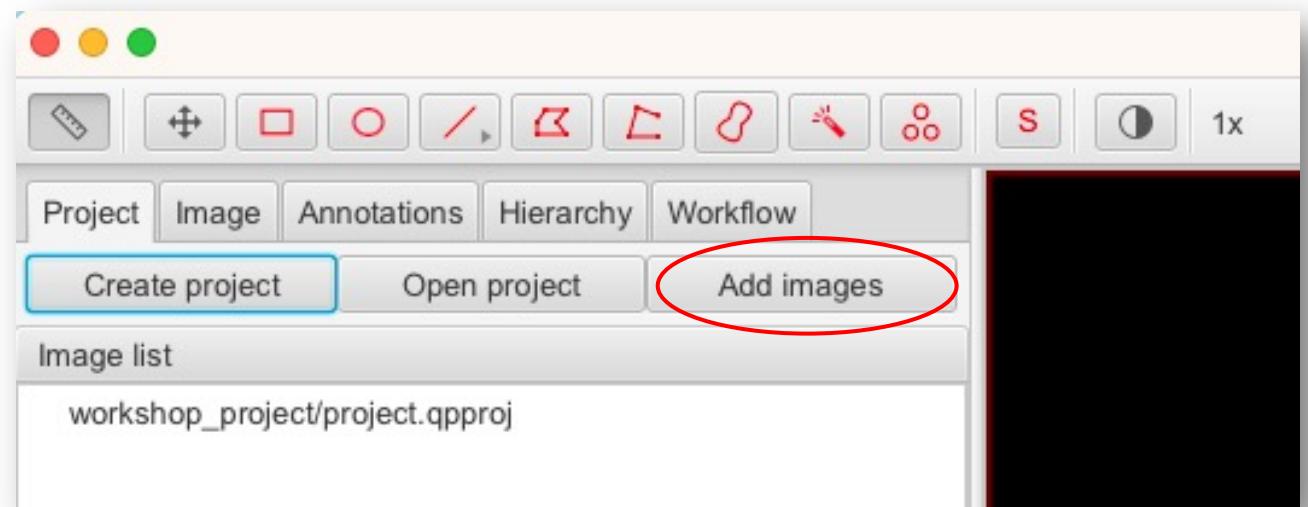


The screenshot shows the QuPath Project browser interface. At the top, there are navigation arrows, a 'Project' title, and various toolbar icons. Below is a table with columns for Name, Date Modified, Size, and Kind. The table lists several files and folders:

Name	Date Modified	Size	Kind
✓ classifiers	Today at 12:17 PM	--	Folder
classes.json	Apr 6, 2024 at 5:32 PM	286 bytes	JSON
> object_classifiers	Jan 19, 2024 at 10:54 AM	--	Folder
> data	Jan 19, 2024 at 10:54 AM	--	Folder
project.qpproj	Apr 6, 2024 at 5:32 PM	46 KB	TextEdit
project.qpproj.backup	Apr 5, 2024 at 1:35 PM	46 KB	TextEdit
✓ scripts	Apr 5, 2024 at 9:41 AM	--	Folder
calculate_median_total_intensity.groovy	Apr 3, 2024 at 1:59 PM	5 KB	Document
cell_classification.groovy	Apr 5, 2024 at 9:28 AM	4 KB	Document
export_measurements.groovy	Apr 3, 2024 at 2:00 PM	2 KB	Document
save_rendered_images.groovy	Mar 23, 2024 at 5:53 PM	1 KB	Document
stardist_nuclei_detection.groovy	Apr 3, 2024 at 2:17 PM	4 KB	Document
stroma_annotation.groovy	Apr 5, 2024 at 9:49 AM	4 KB	Document

Add an image to your project

1. Check your emails! Download this folder containing an example whole-slide image
2. Add an image
 - *Add images* button
 - Select the .vsi file

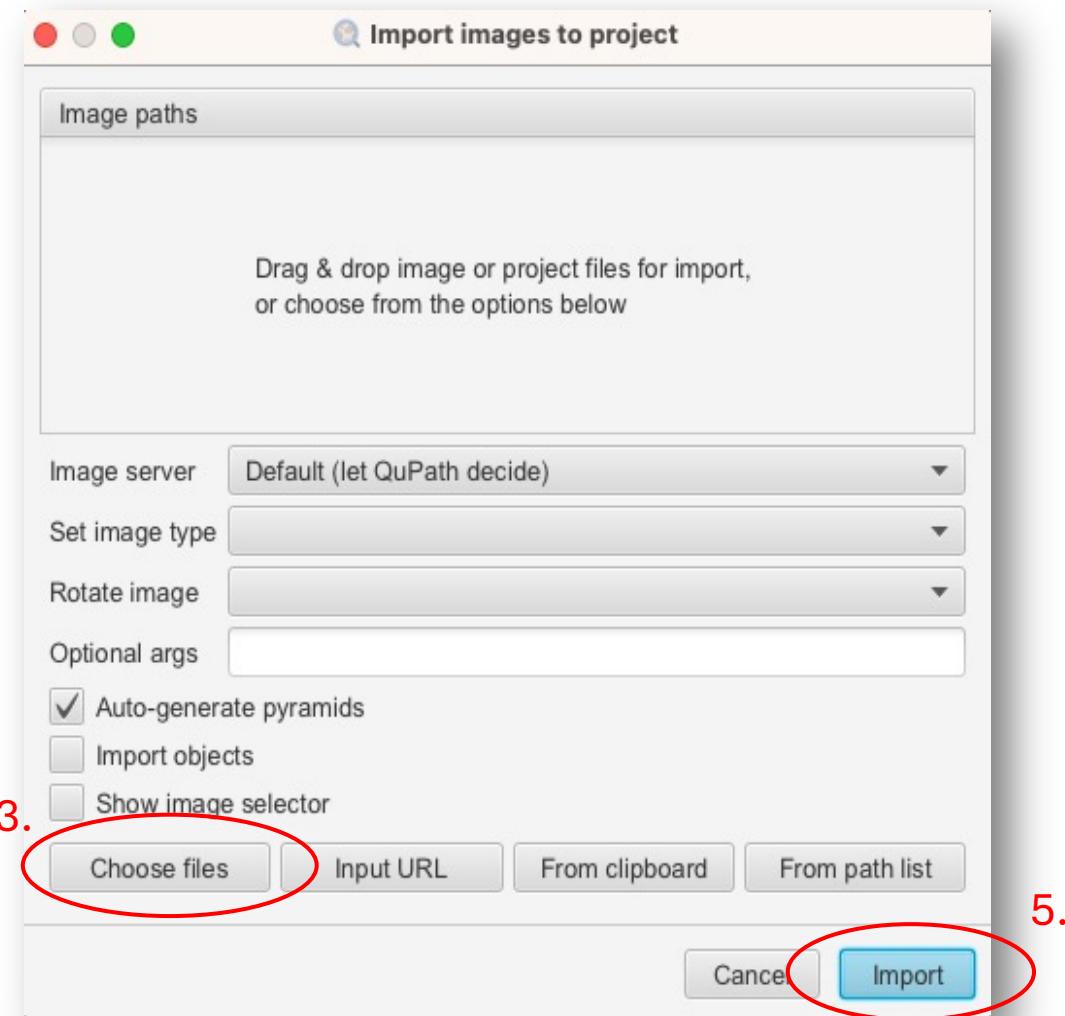


Add an image to your project

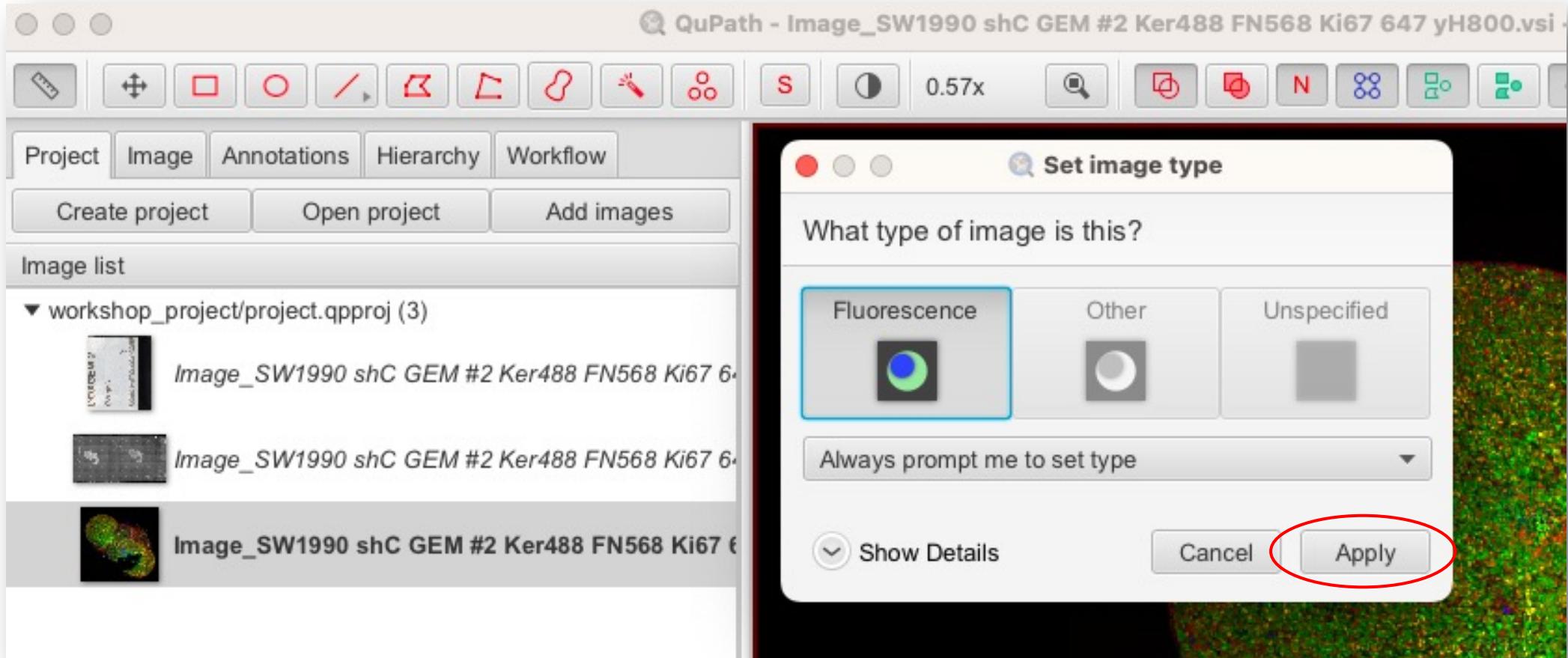
3. Select the .vsi image using *Choose files* or drag-and-drop

4. Use default settings

5. Click *import*



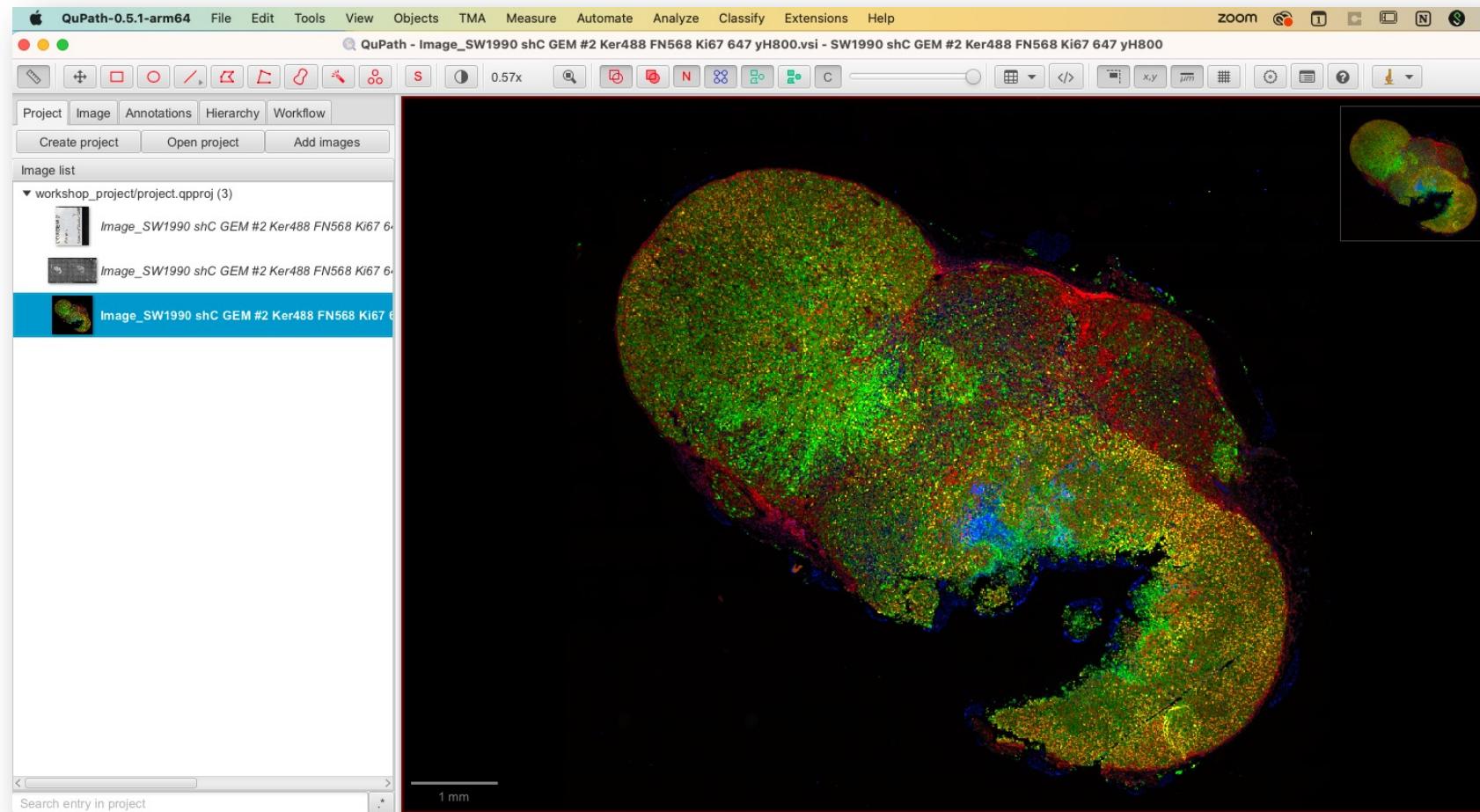
Set image type



- Other image types are supported: Brightfield H&E, H-DAB, other brightfield

Yay! We have a QuPath project with an image

→ *Double-click an image to open it in the viewer*



QuPath works on copies of your original files

- QuPath access the image pixels and metadata via an image server
 - Akin to a copy of the original file
- Manipulating files within a QuPath project will never modify the original files or pixels
 - Deleting, duplicating, processing, etc will not be reflected in your original files

QuPath projects are portable

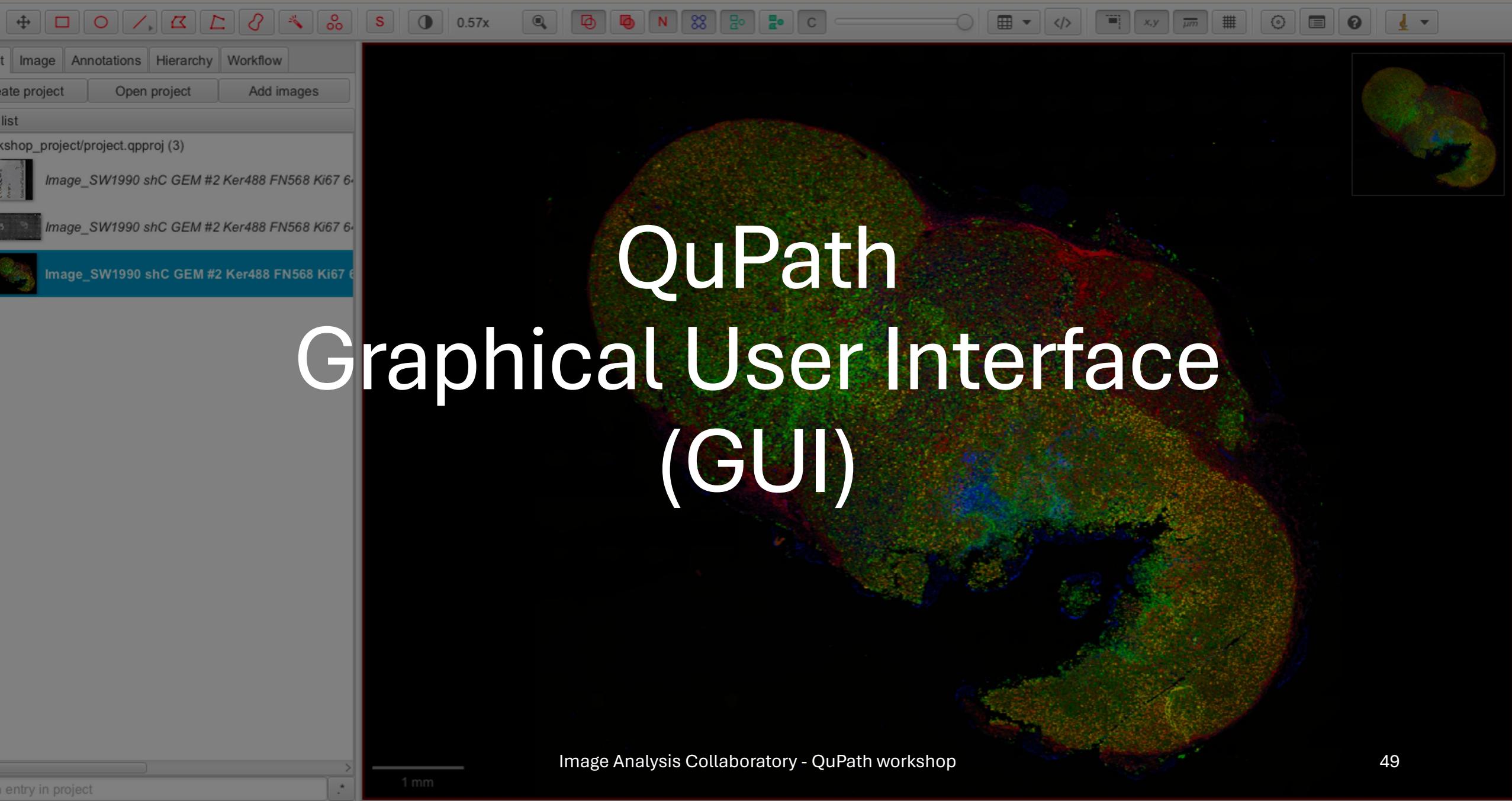
- Sharing a project:
 - Zip up the entire project directory
 - Email it to your collaborators

The project folder only contains QuPath objects and data, unless you had placed them there. Ensure that they can access the actual image files.

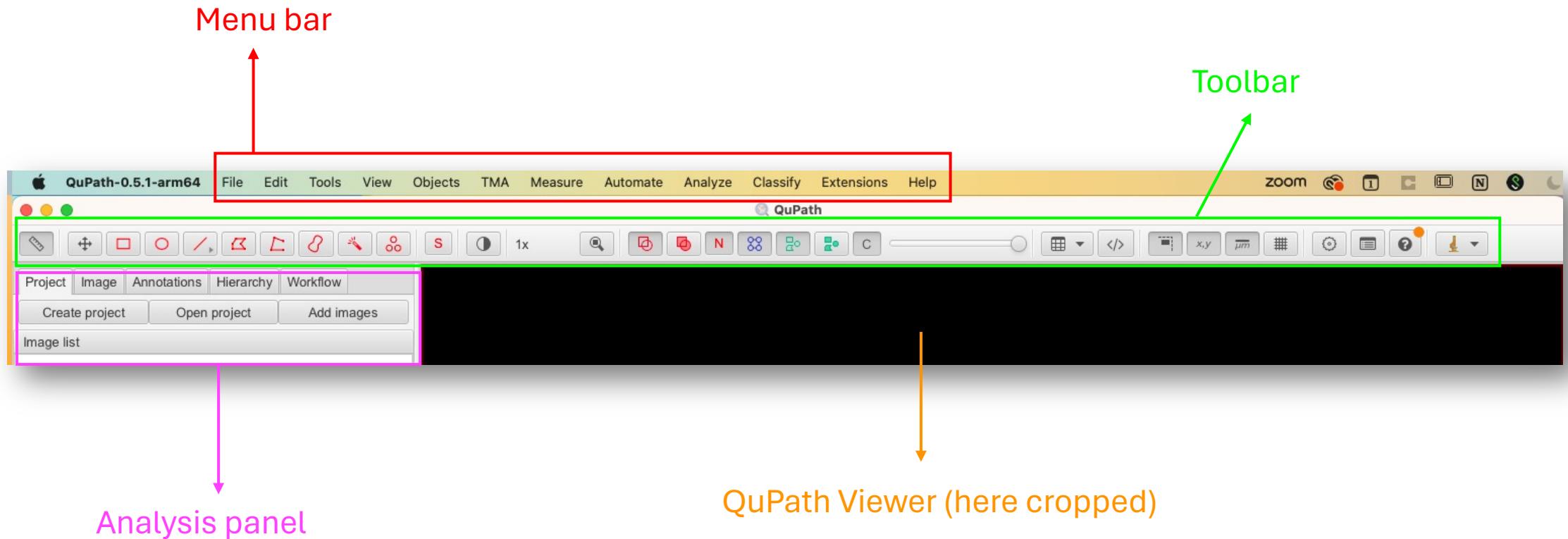
QuPath projects are portable

- Receiving a project:
 - The project still contains image paths specific to the local machine of the sender
 - If you move the image, you will be prompted to update the file path

QuPath - Image_SW1990 shC GEM #2 Ker488 FN568 Ki67 647 yH800.vsi - SW1990 shC GEM #2 Ker488 FN568 Ki67 647 yH800

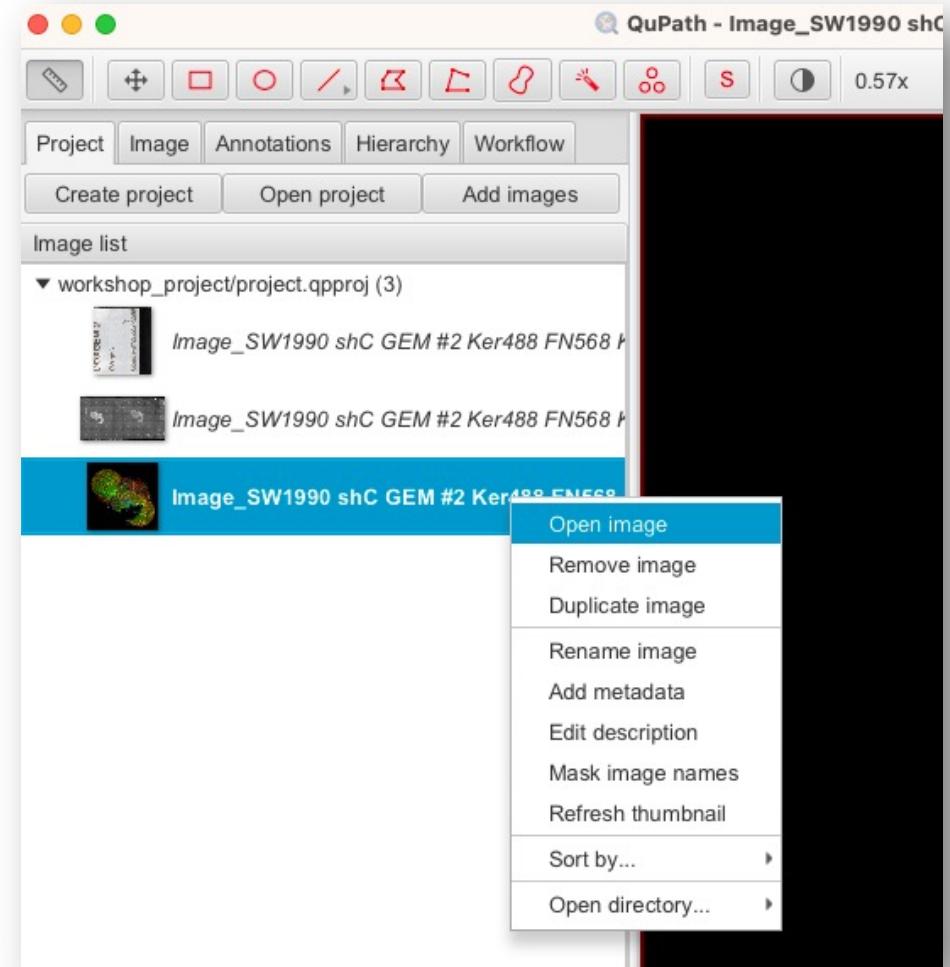


Graphic User Interface (GUI)



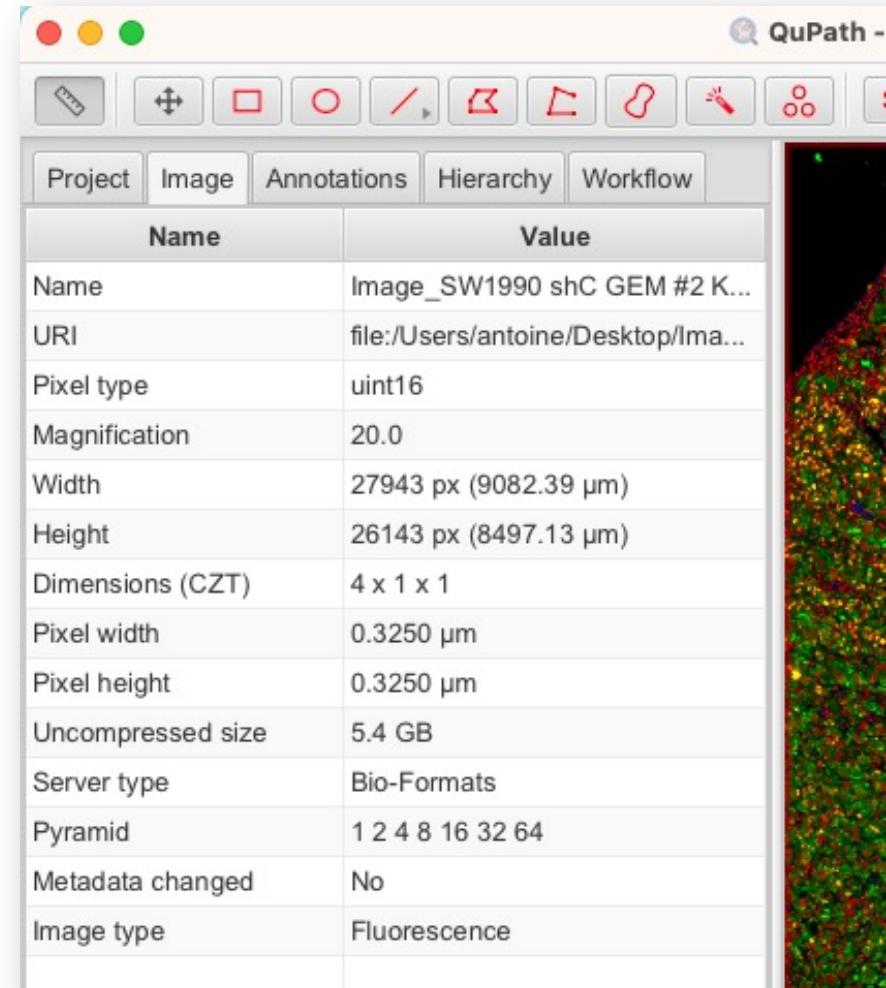
Analysis Panel

- *Project tab > right-click on an image*
 - *Open, remove, rename and duplicate images*
 - *Edit metadata*

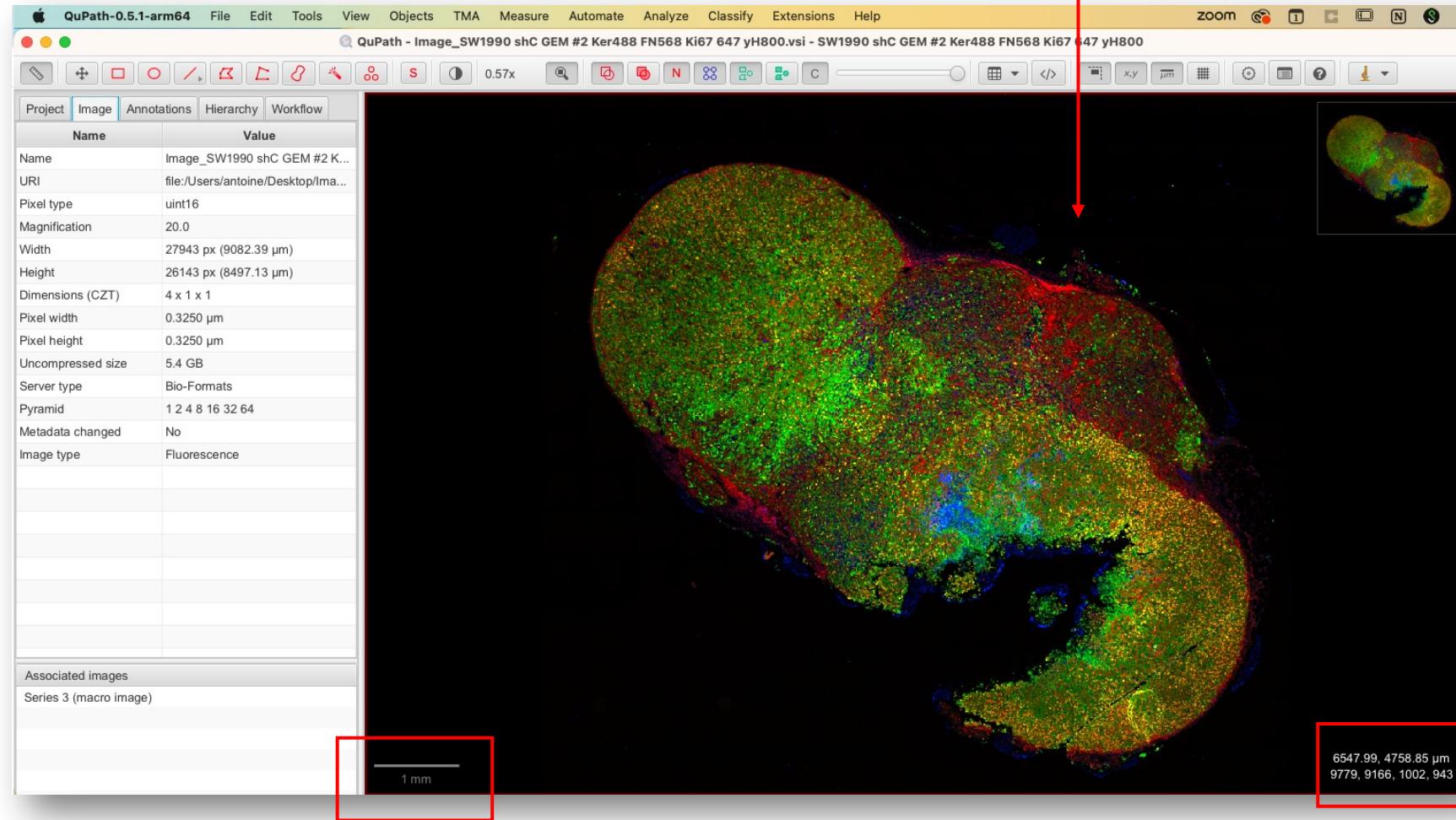


Analysis Panel

- *Image tab*
 - Name and image file path
 - Magnification: 20x
 - Pixel type, width and height are crucial for scale calibration
 - Dimensions: 4 channels + 2D
 - Pyramid: level of downsampling in the viewer
 - Image type: previously set to fluorescence

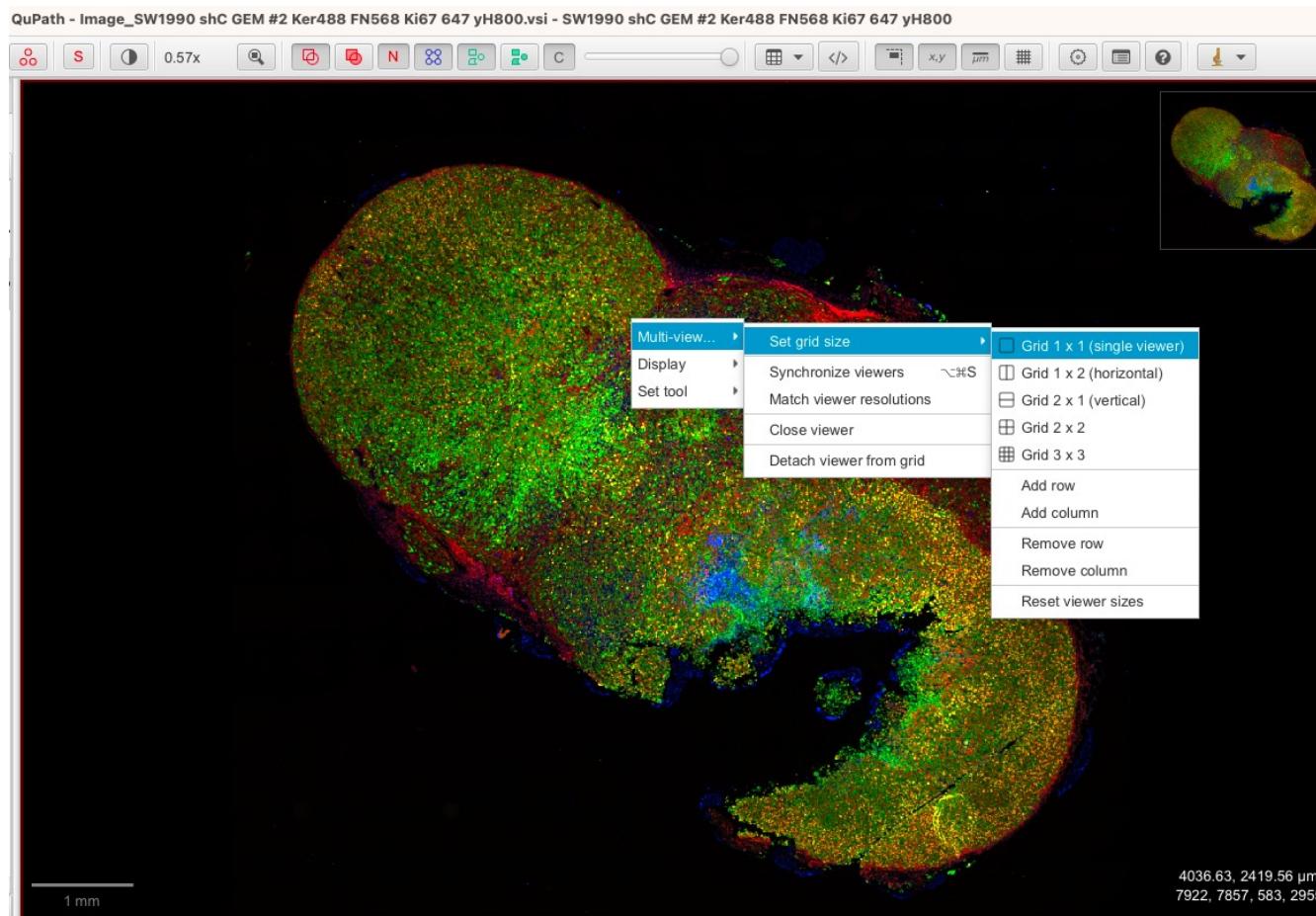


QuPath viewer

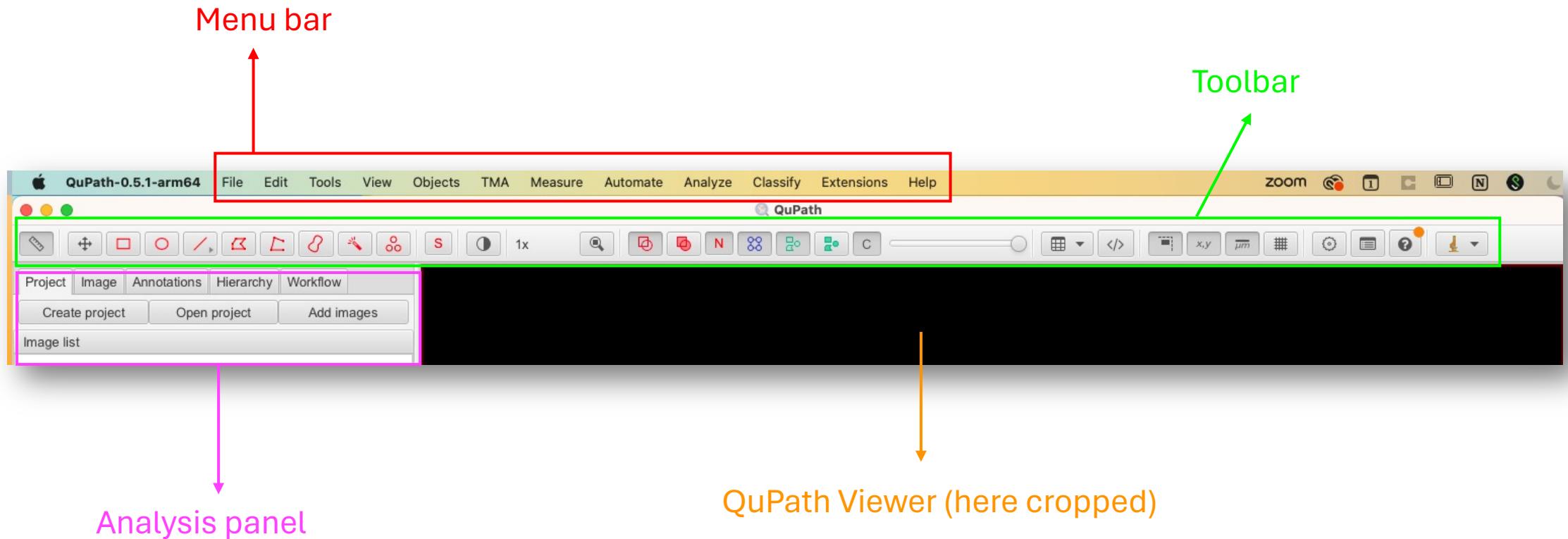


Multi-viewer

- Right-click in the viewer



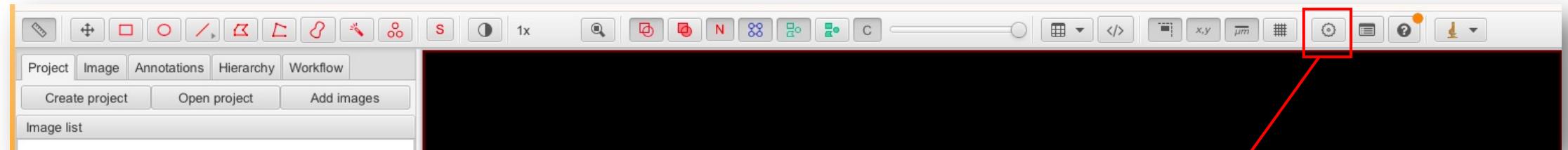
Graphic User Interface (GUI)



Toolbar

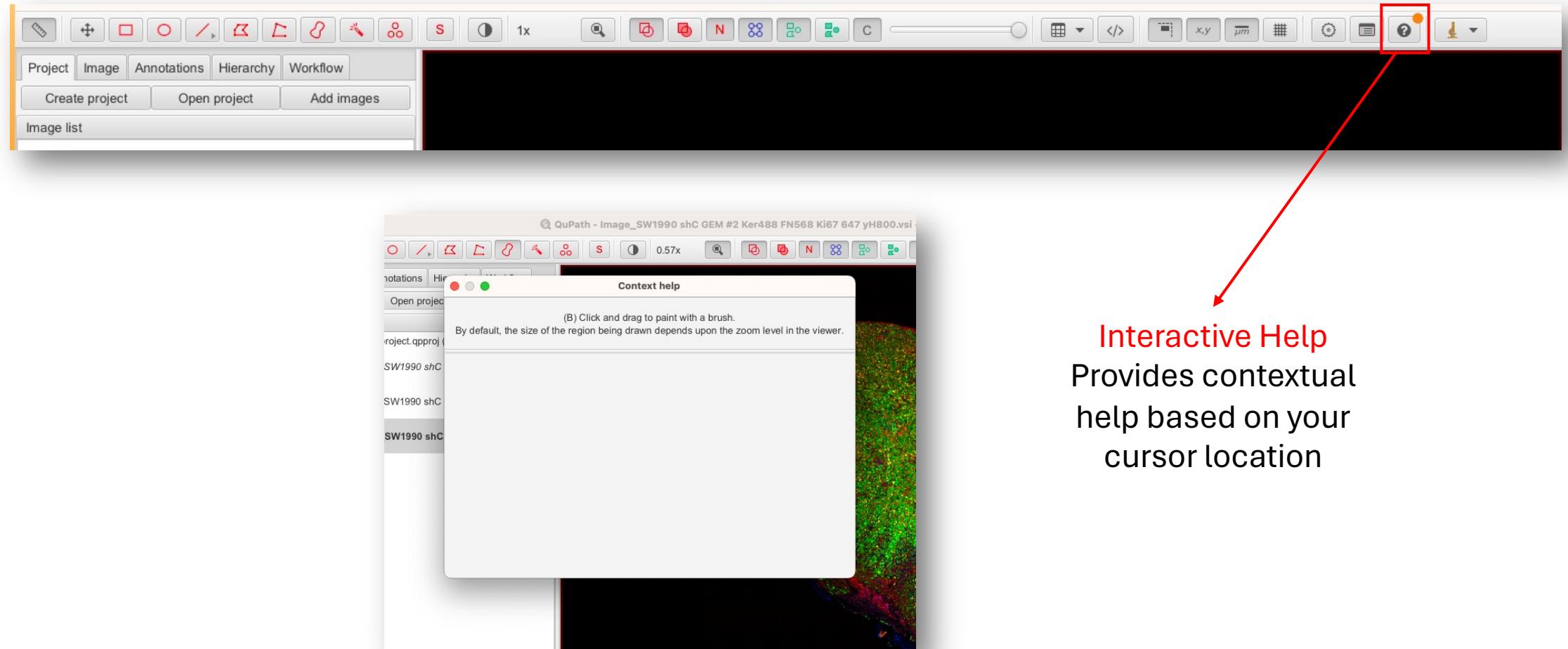


Toolbar



Preferences
Settings, GUI
customization,
extensions, ...

Toolbar



Example when my cursor is on the paint brush tool

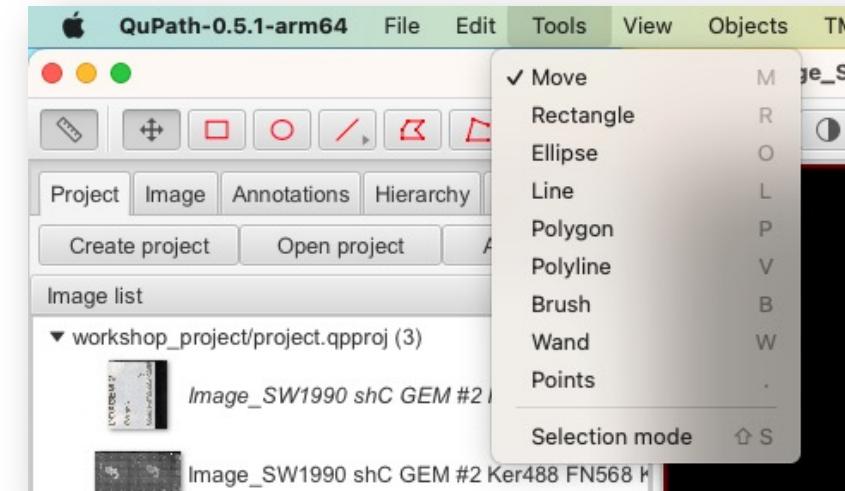
Toolbar

M R O B W Shift+S



Annotation tools

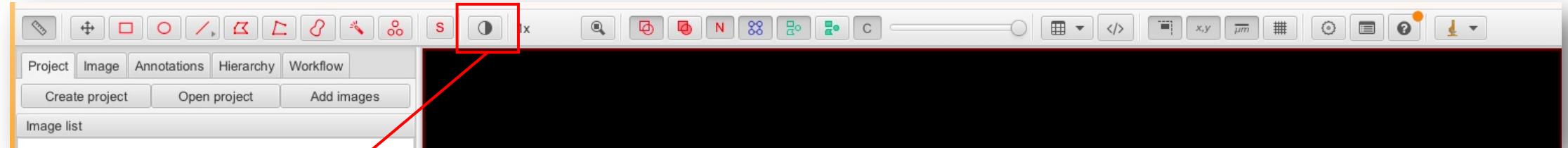
- **M** key: move tool
- **R** key: draw a rectangle annotation
- **O** key: draw an ellipse annotation
- **B** key: paint with a brush
- **W** key: draw with a wand tool
- And many more!



Annotation tools are also accessible in the *Tools* menu

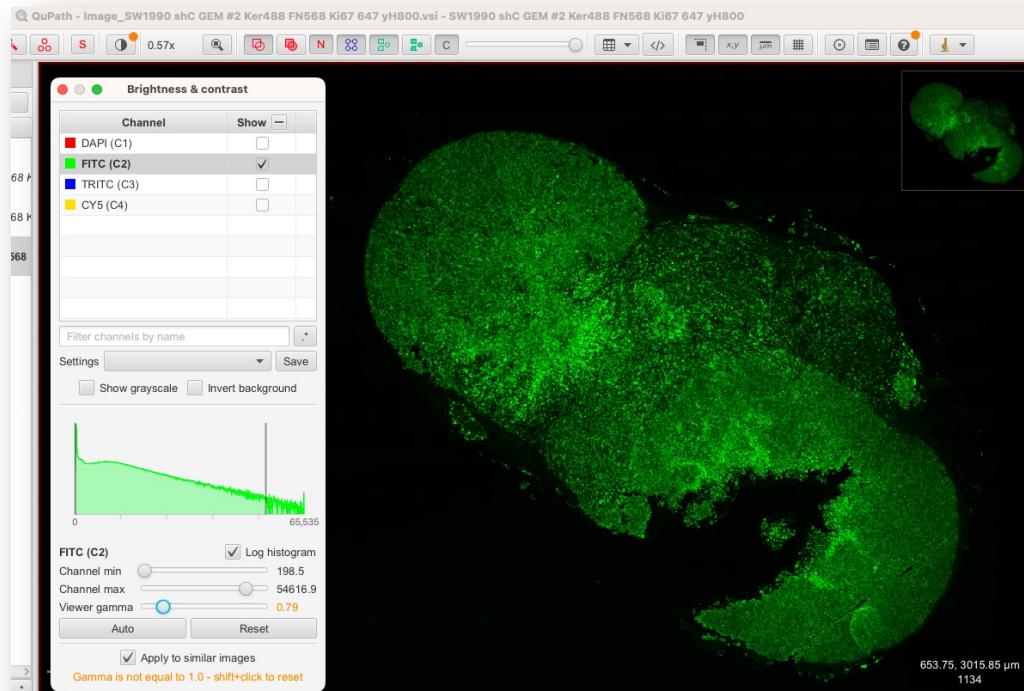
Toolbar

Shift+C



Brightness and contrast

- Toggle on/off channels
- Adjust LUT range
- Visualize intensity histogram

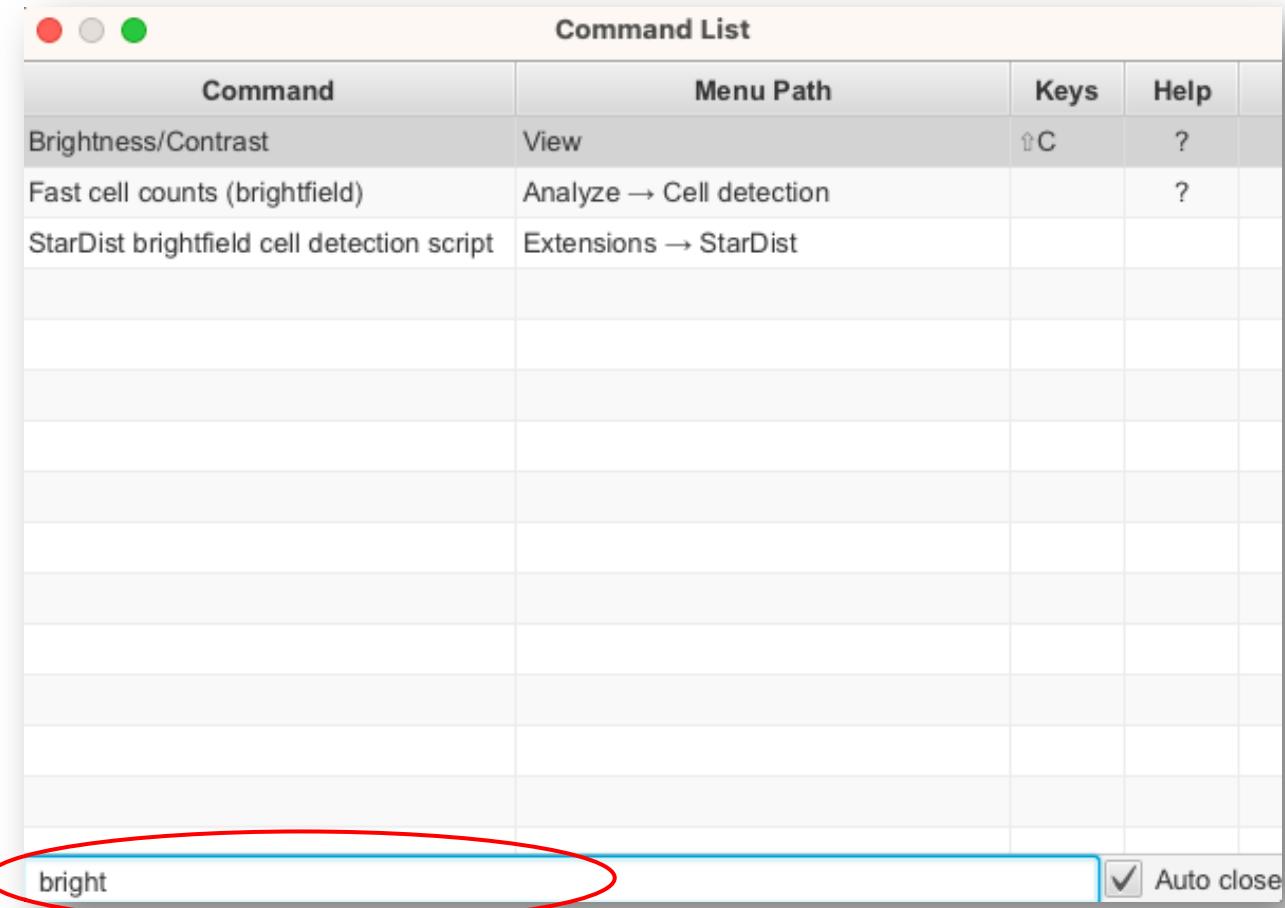


Example for FITC (Keratin) channel

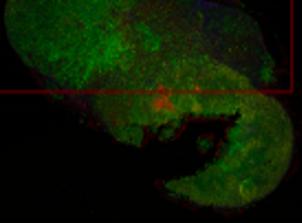
QuPath pro-tip: command list

Command/Control + L
Opens a dialogue to
search for any command
using keyword

*For example, search for
'brightness'*



Exercises 1: QuPath projects and GUI



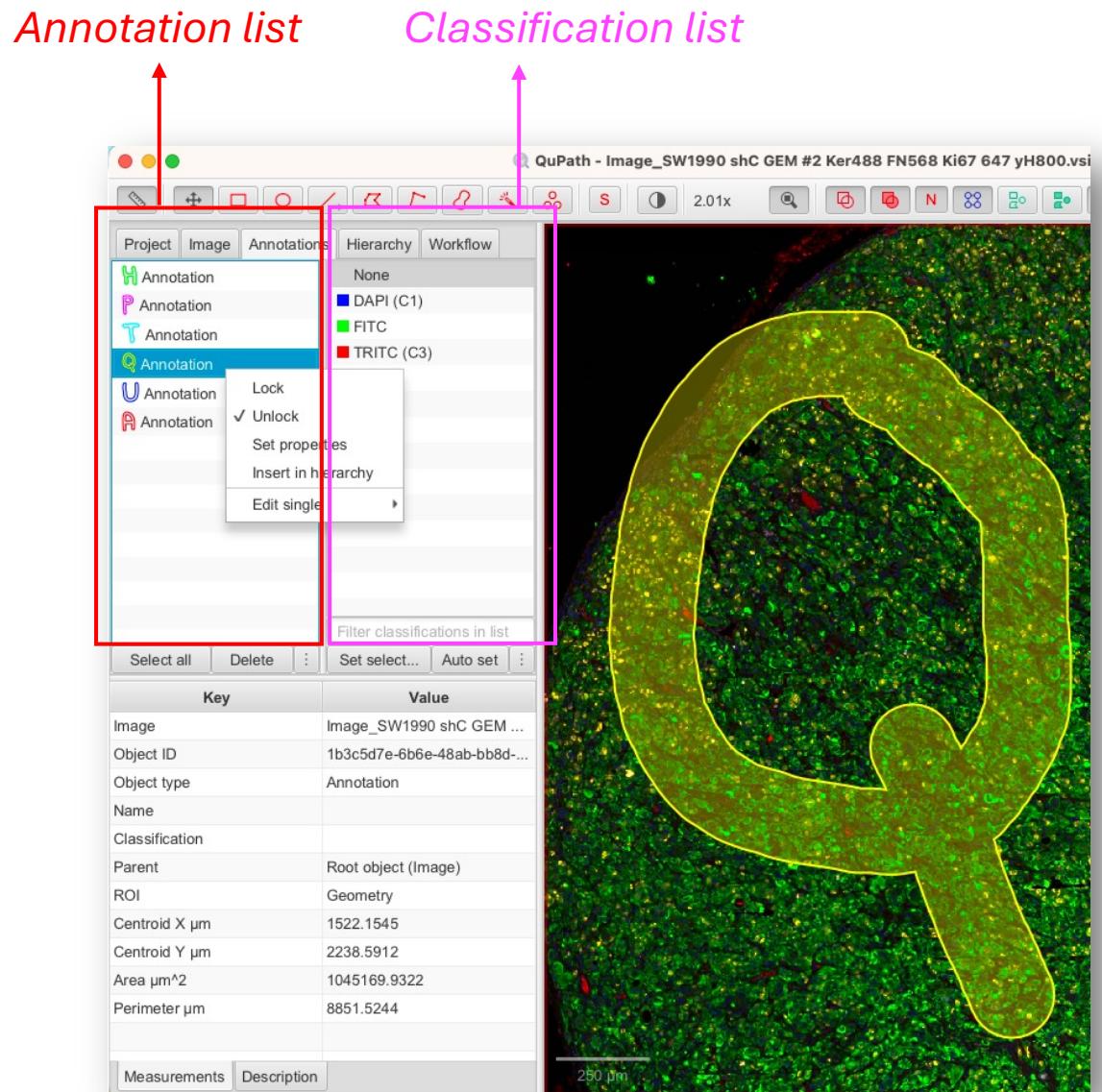
Introducing objects: annotations and detections

Key concept: QuPath objects

- **Objects** are a ‘thing’ in an image which encapsulates not only its shape but also some properties about it
- **Annotations:** Objects that you usually create yourself, by drawing on the image
 - They are flexible, up to ~100 per image
 - Can be edited
 - Often used to define regions
- **Detections:** Objects that QuPath usually creates for you
 - They are efficient, up to ~millions per image
 - Can be deleted but not edited
 - Often used to define cells

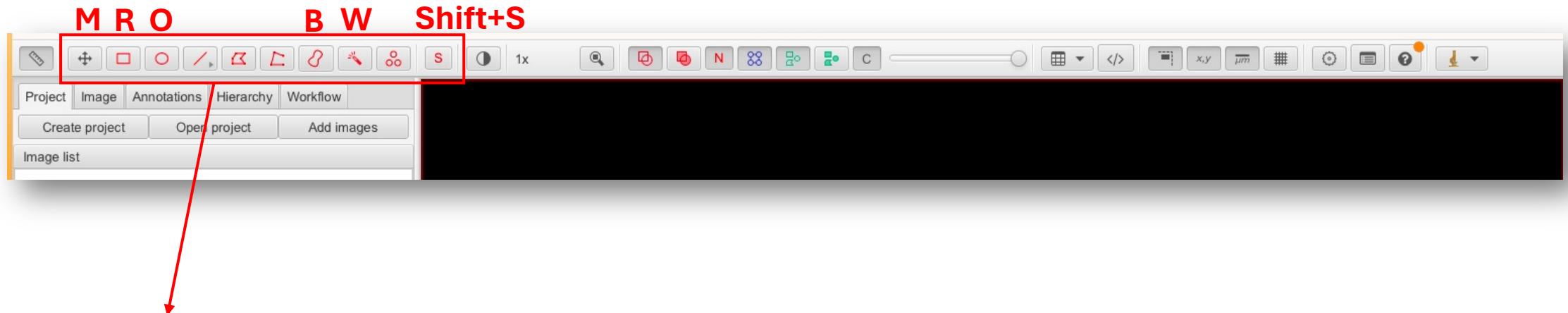
Analysis Panel

- *Annotations* tab
 - Annotation list lets you select, delete
 - Right-click to **lock** or edit properties (name, color)
 - Shift or Command/Control to multi-select



How to create manual annotations?

Select one of the annotation tools from the toolbar then scribble on the image!



Annotation tools

- M key: move tool
- R key: draw a rectangle annotation
- O key: draw an ellipse annotation
- B key: paint with a brush
- W key: draw with a wand tool
- And many more!

**Remember to always
lock your annotation
to prevent accidental
editing!**

Exercises 2: QuPath manual annotations

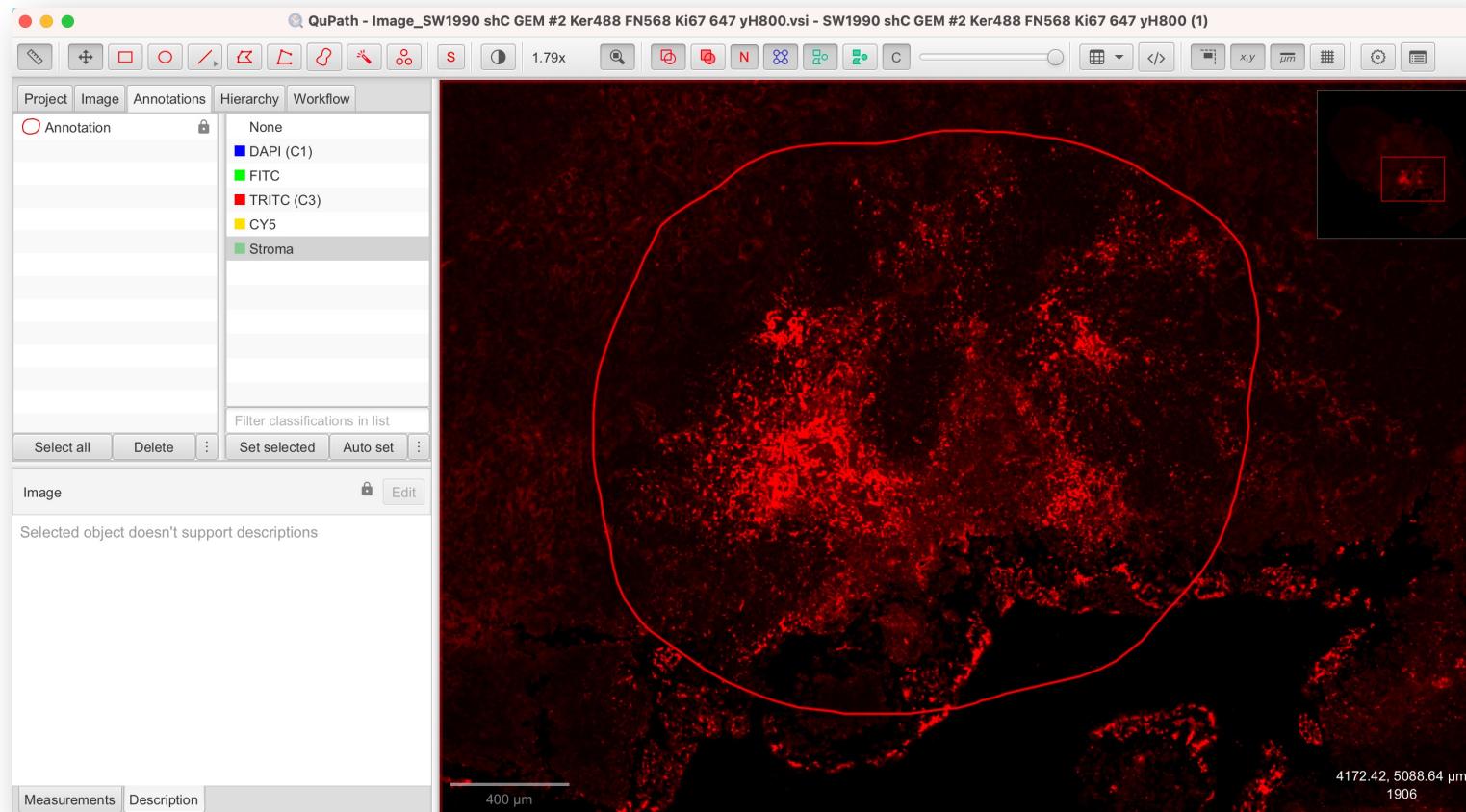
Recreate these annotations

Decide on which annotations tool from the toolbar is best to do so



Create a region of interest with the annotation tools

In the TRITC channel (fibronectin), create a region of interest that enclose high-fibronectin content regions

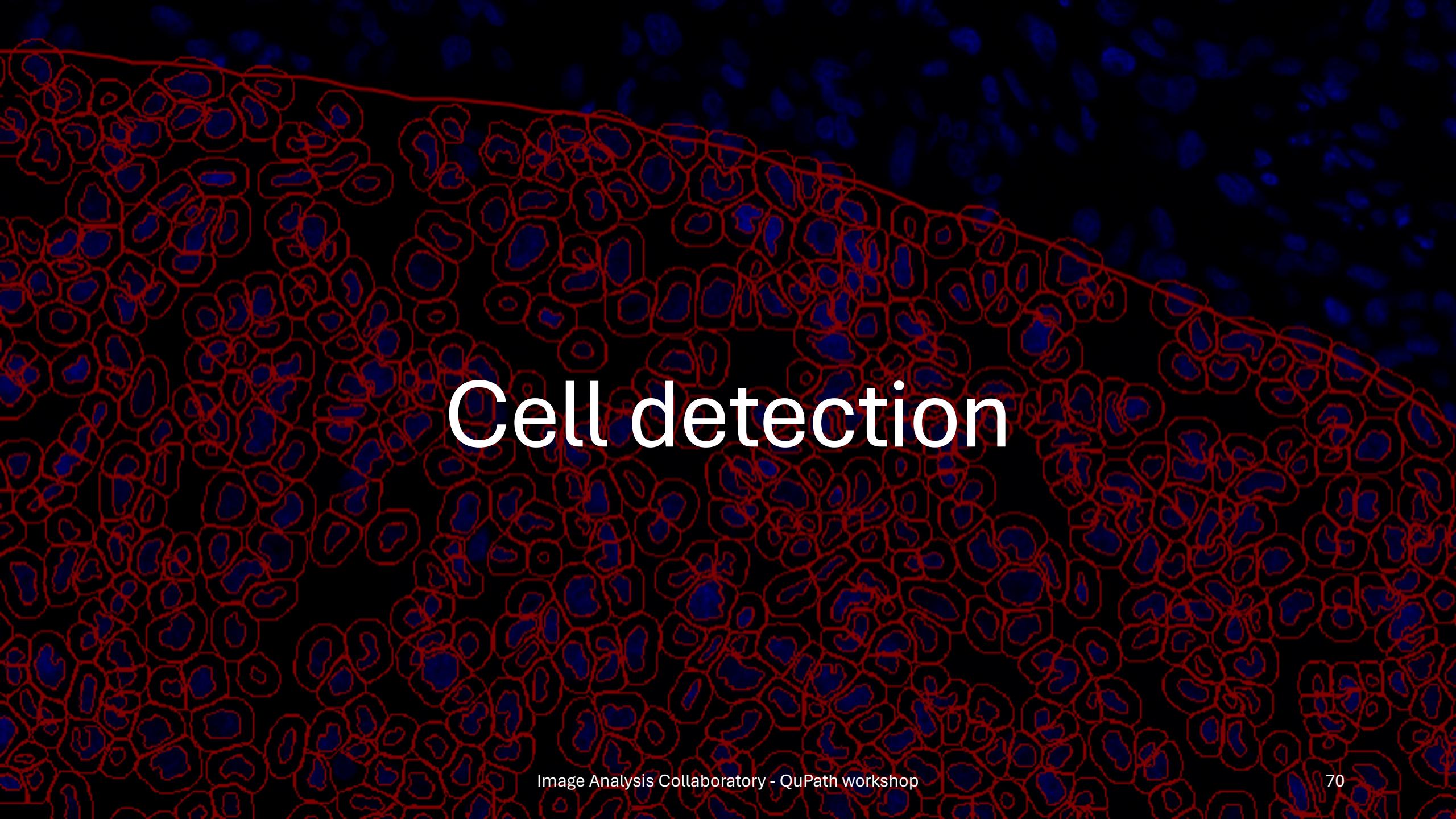


Once you have finished your annotation, **lock** it:

Right-click in the viewer
> Annotations > Lock

or

Right-click on the annotation in the analysis panel > Lock



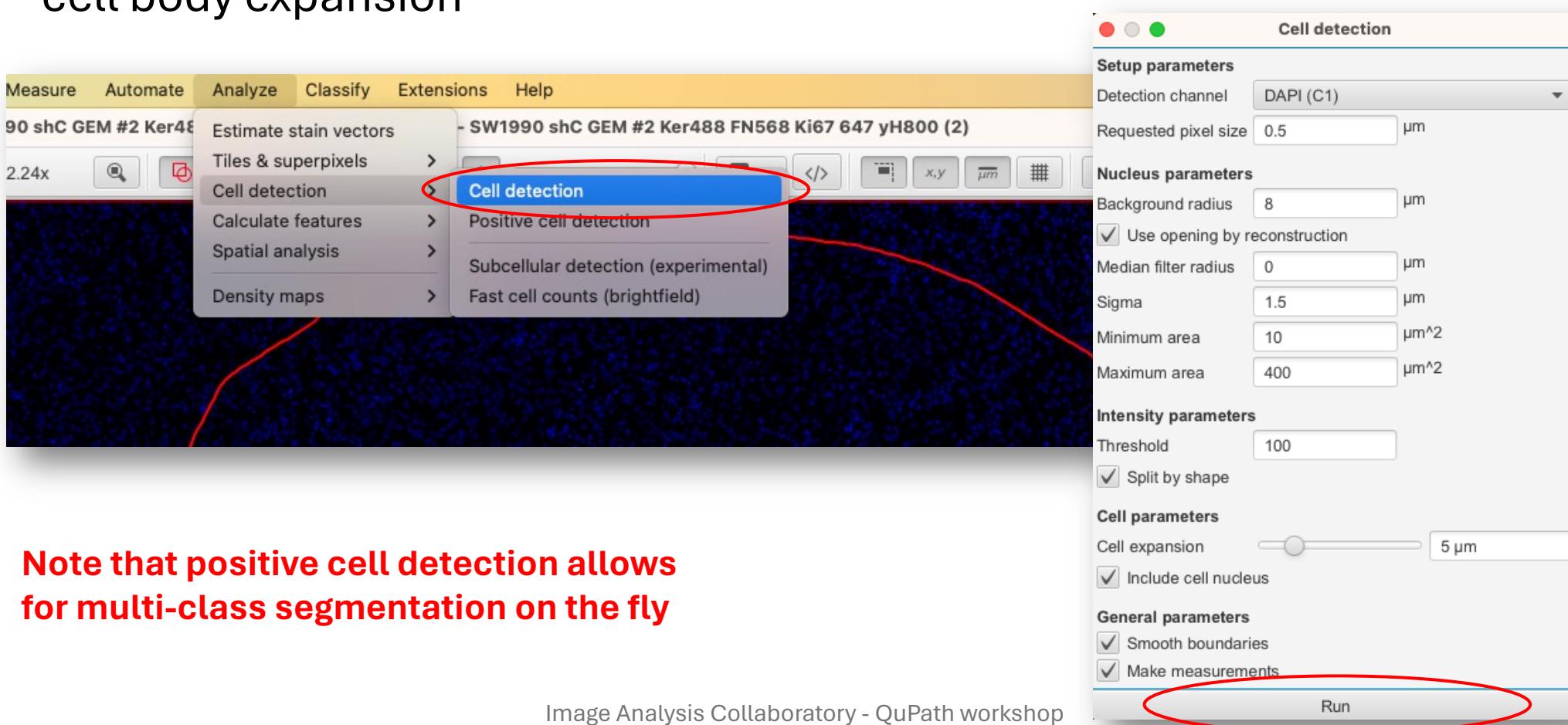
Cell detection

Cell detection

- QuPath offers three main options:
 1. Built-in cell segmentation algorithm, based on nucleus thresholding and cell body expansion
 2. StarDist as an extension (DL)
 3. Cellpose as an extension (DL) – not covered here
- All yield *Cell Detections* objects that will have shape and intensity measurements for nucleus, cell and membrane
- Detection can be computationally intensive so we will start from the region of interest

Cell detection

1. Built-in cell segmentation algorithm, based on nucleus thresholding and cell body expansion



Cell detection parameters

Cell detection

Setup parameters

Detection channel DAPI (C1)

Requested pixel size 0.5 μm

Channel e.g. DAPI

The resolution of the image used in the segmentation algorithm

- Enter **0** for full resolution
- Default **0.5** typically good trade-off between cost and details

Cell detection parameters

Radius of median filter
(smoothing):

- Enter **0** to disable
- Higher values will smooth off details

Radius of gaussian filter
(smoothing):

- Enter **0** to disable
- Higher values will smooth off details

Nucleus parameters

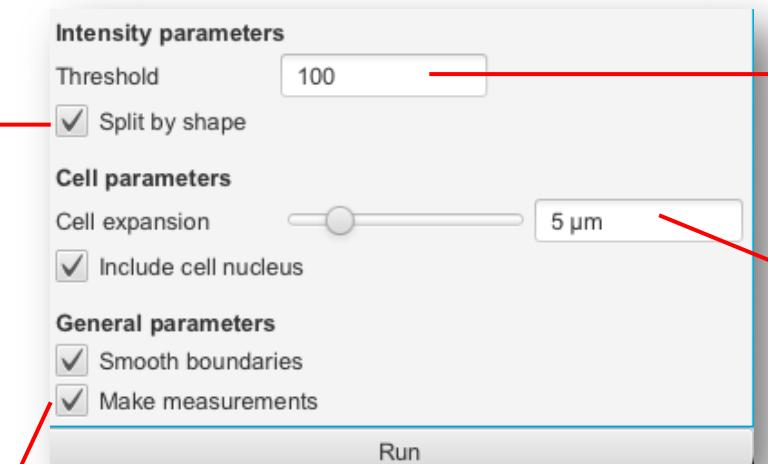
Background radius	8	µm
<input checked="" type="checkbox"/> Use opening by reconstruction		
Median filter radius	0	µm
Sigma	1.5	µm
Minimum area	10	µm ²
Maximum area	400	µm ²

Radius of area used for background subtraction

Allowed area interval for detections; nuclei detection is removed if outside of the interval

Cell detection parameters

Uses roundness of detections shape to split clusters/clumps; keep it ticked for most usages



If ticked, will generate measurements specific to each detected nuclei and inferred cytoplasm

Minimum signal intensity of nuclei relative to background

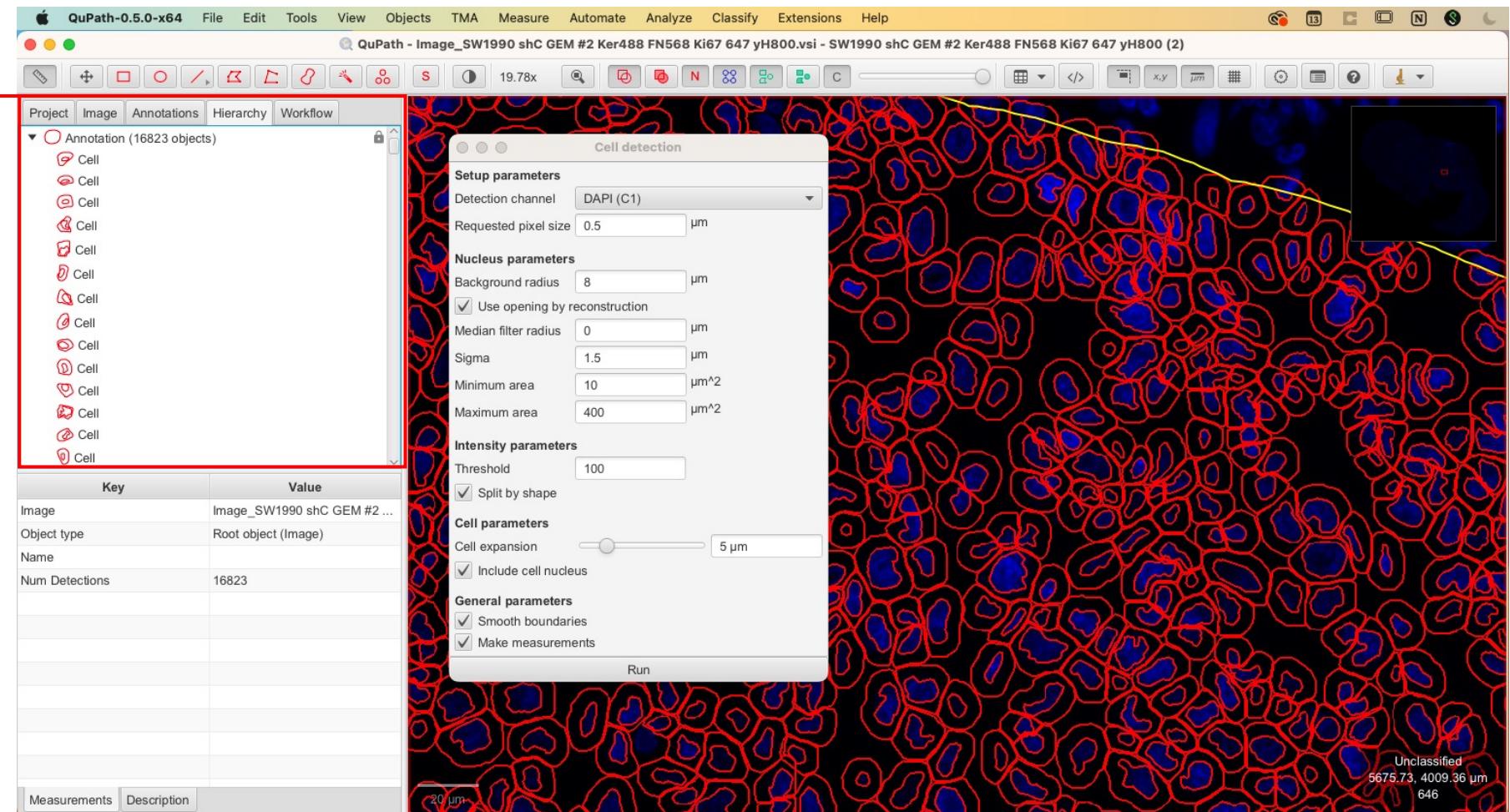
How much to expand nuclei to get cell boundaries

- Enter **0** to disable
- Enter small values **0 – 2** for peri-nuclear measurements
- Enter values **~5** for cytoplasm measurements, depending on tissues

Cell detection with default parameters

Hierarchy tab

- Detection list
 - Nested in its parent annotation (ROI)
 - Note the cell count



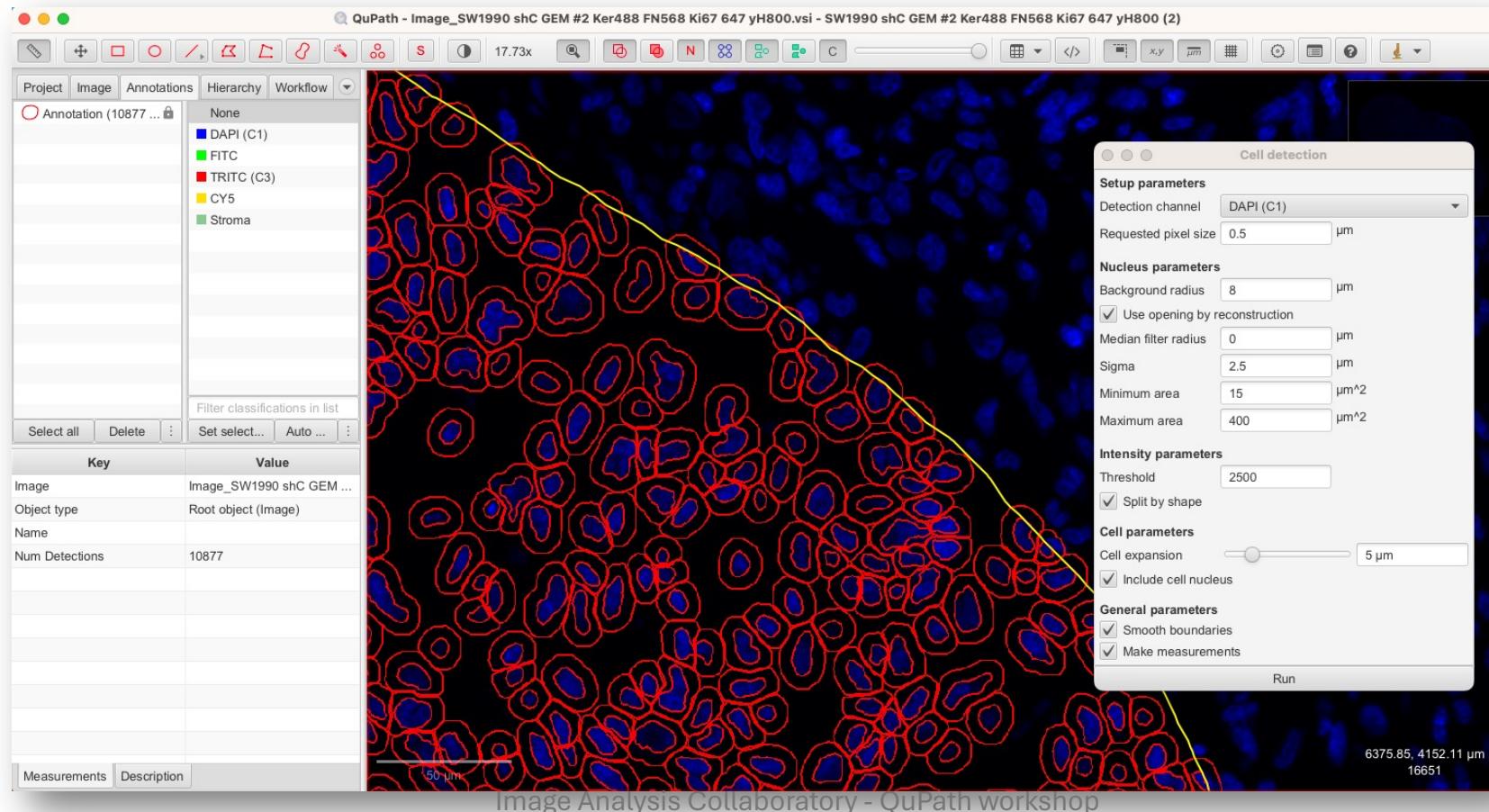
Note on the hierarchy of objects in QuPath

- QuPath allows to nest objects in one another to organize your projects
 - Sort of child-parent link
 - Very useful to organize and restrict the analysis to parts of an image
 - Can be used to restrict image processing within a ROI or a detected tissue region

Exercise 3.a: QuPath cell detection

Exercise: explore parameters

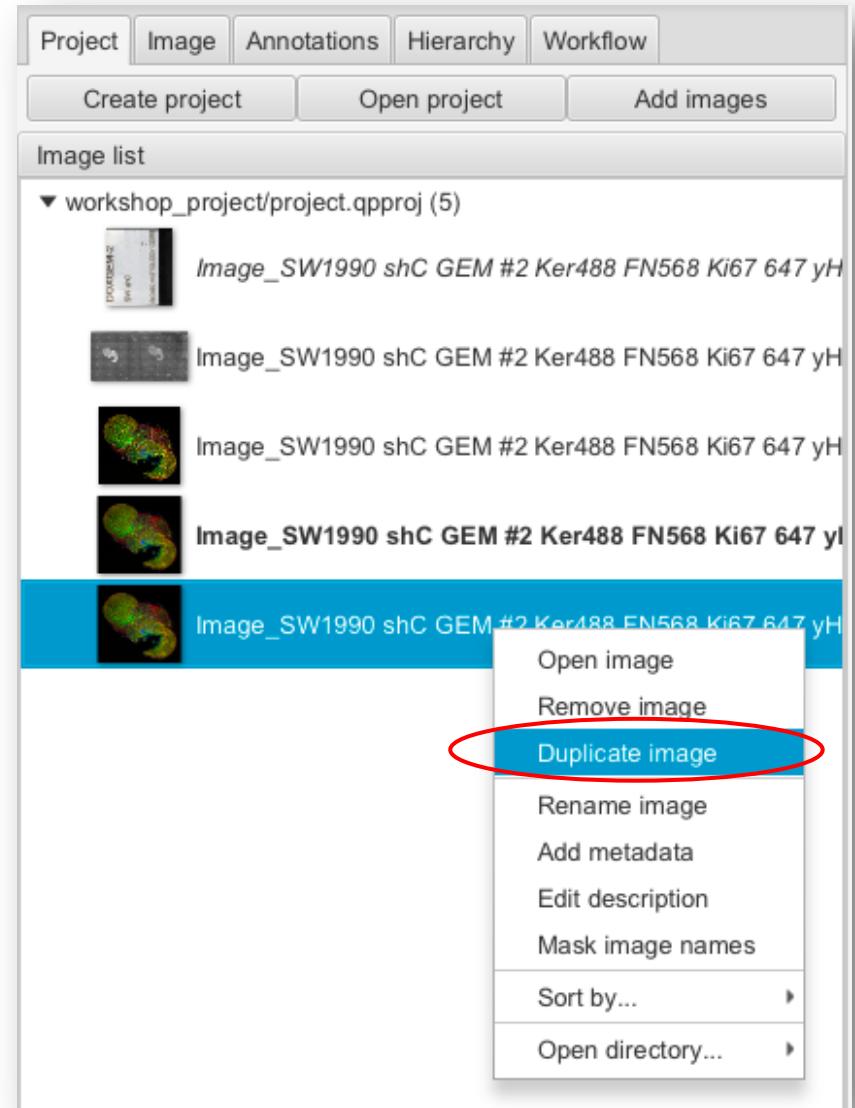
- I found that default parameters tend to over-segment nuclei so adapted the parameters to be slightly stricter (min area and threshold increased)



Duplicate your image

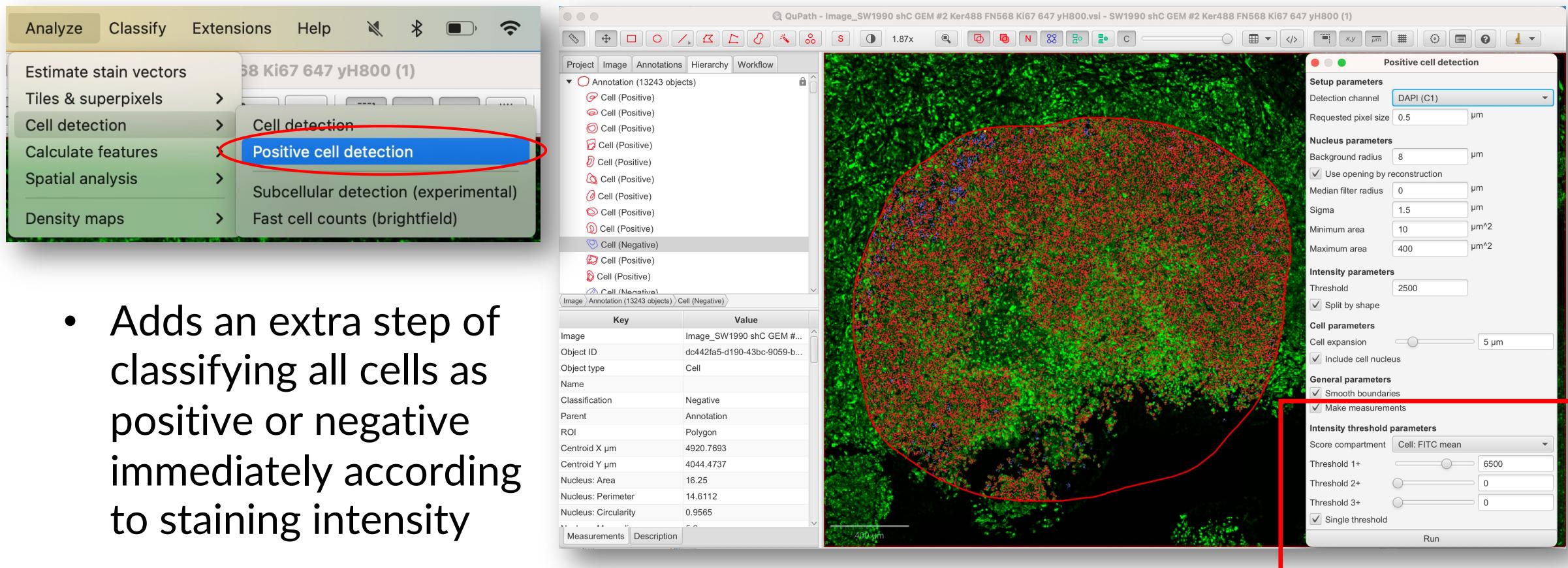
- Copy your cell detection results for future work on it
- *Project tab > Image list > Option+click or right-click on the image name > Duplicate image*

**It duplicates QuPath objects,
not the actual image**



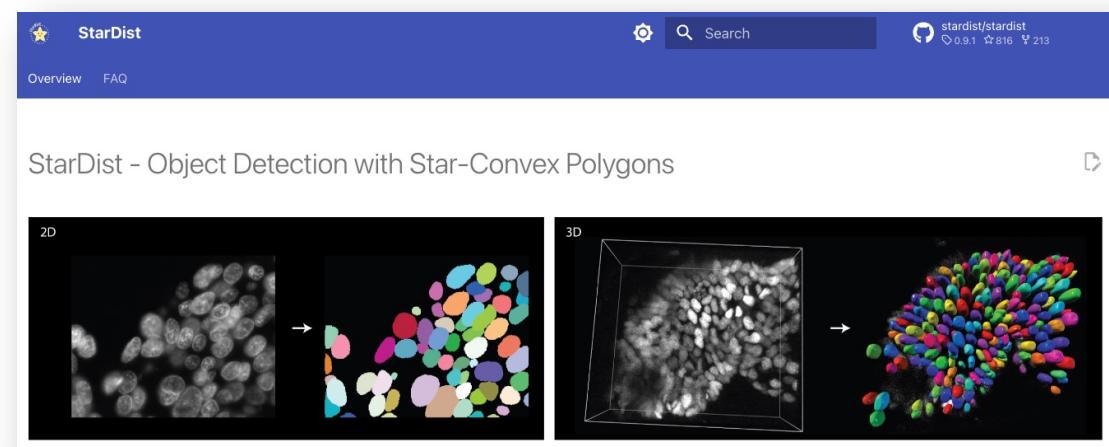
Detecting cells with an extra condition

- *Analyze > Cell detection > Positive cell detection*



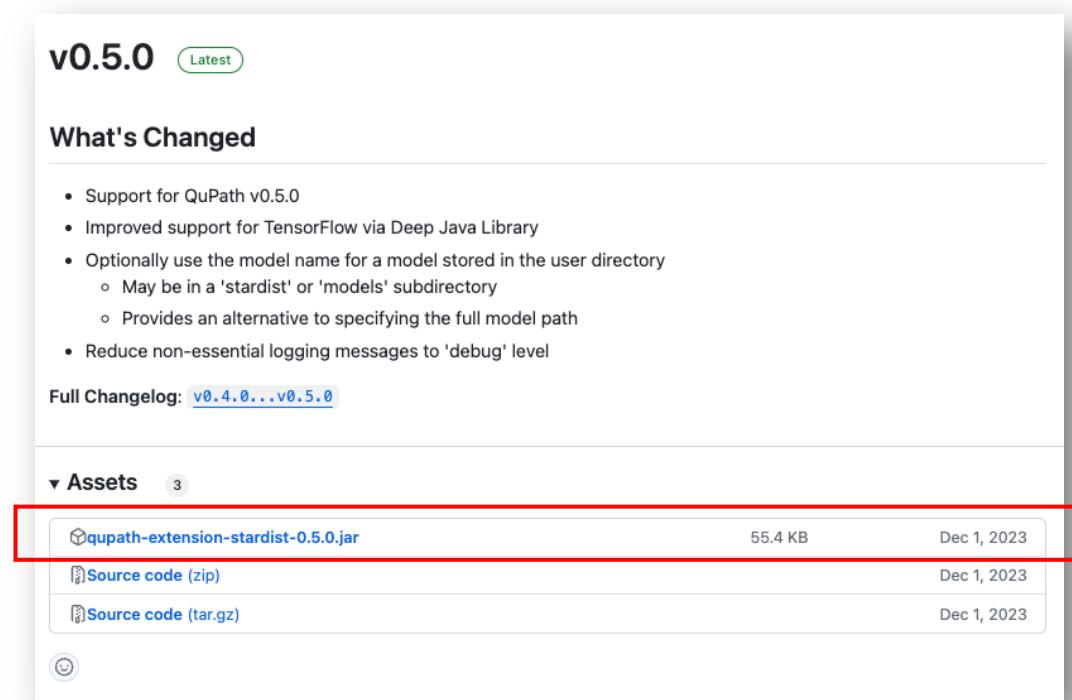
Deep learning-based cell segmentation

- DL-based methods can typically capture more complex patterns, tend to mitigate human bias such as threshold hand-picking
- **However**, they are more computationally expensive and often need fine-tuning or re-training for specific applications
- StarDist is a deep learning model trained to detect specific kinds of nuclei in different kinds of image



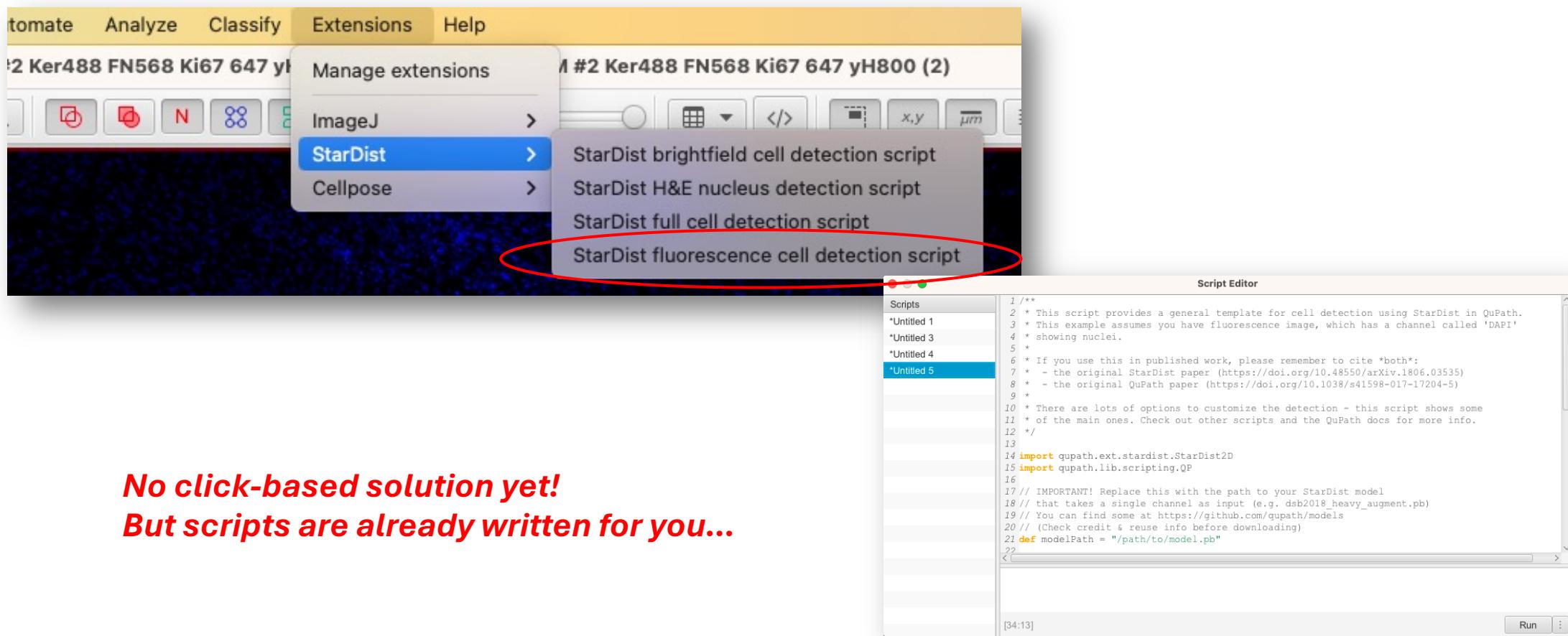
Installing StarDist extension in QuPath

- Browse to
<https://github.com/qupath/qupath-extension-stardist/releases>
- Download the .jar file compatible with your QuPath version
 - For this workshop, get [qupath-extension-stardist-0.5.0.jar](#)
- Drag and drop the .jar file onto QuPath main window, and... that's it!



Using StarDist extension in QuPath

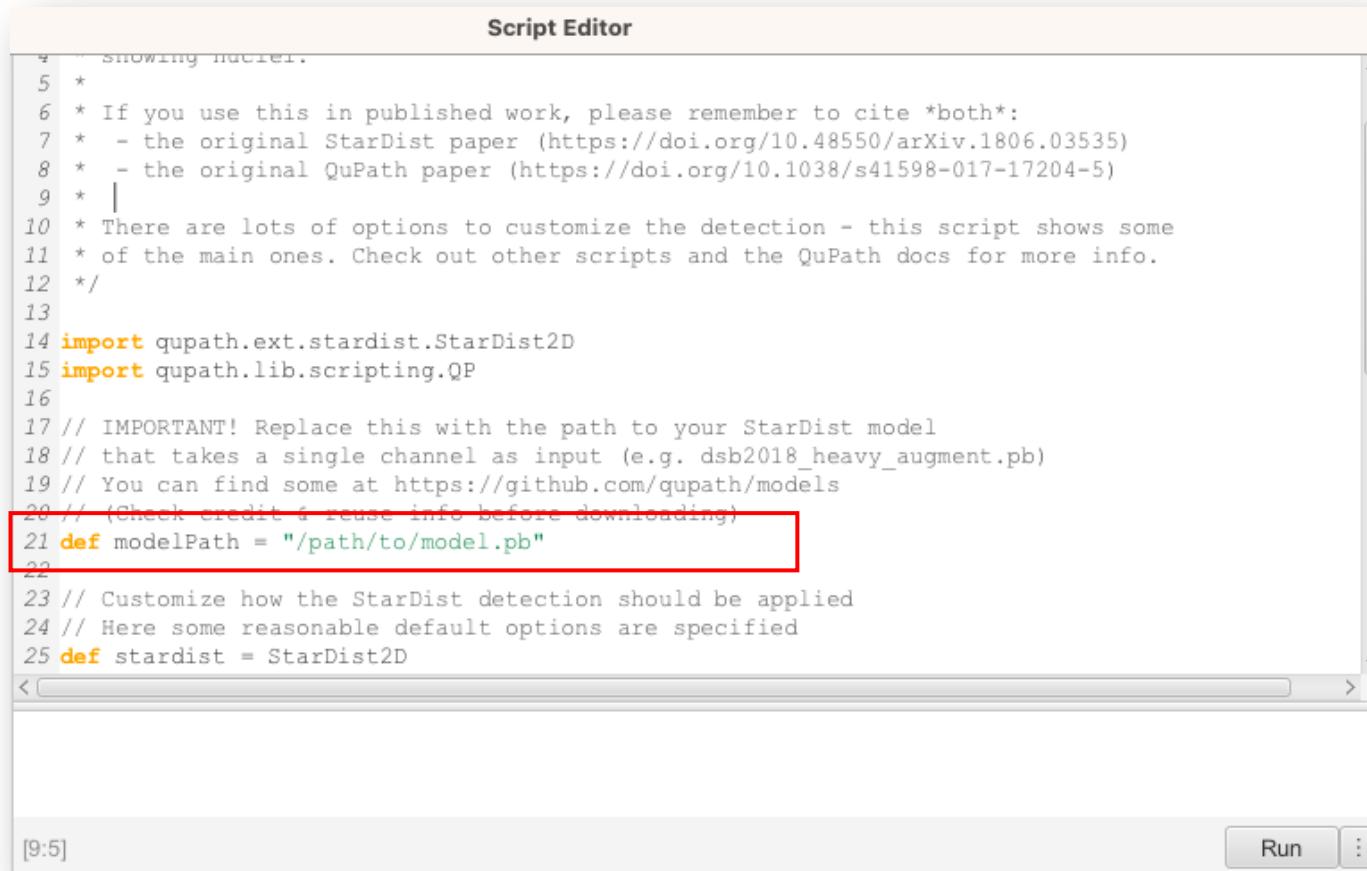
- Go to *Extensions* tab > *StarDist* > *StarDist fluorescence cell detection script*



No click-based solution yet!
But scripts are already written for you...

Using StarDist extension in QuPath

- Requires to load a pre-trained model (basically the weights)



The screenshot shows the QuPath Script Editor window. The title bar says "Script Editor". The code editor contains a Python script for performing StarDist detections. The script includes comments about citation, detection options, and model paths. A red box highlights the line "21 def modelPath = "/path/to/model.pb"".

```
# ... showing nuclei.
5 *
6 * If you use this in published work, please remember to cite *both*:
7 * - the original StarDist paper (https://doi.org/10.48550/arXiv.1806.03535)
8 * - the original QuPath paper (https://doi.org/10.1038/s41598-017-17204-5)
9 *
10 * There are lots of options to customize the detection - this script shows some
11 * of the main ones. Check out other scripts and the QuPath docs for more info.
12 */
13
14 import qupath.ext.stardist.StarDist2D
15 import qupath.lib.scripting.QP
16
17 // IMPORTANT! Replace this with the path to your StarDist model
18 // that takes a single channel as input (e.g. dsb2018_heavy_augment.pb)
19 // You can find some at https://github.com/qupath/models
20 // (Check credit & reuse info before downloading)
21 def modelPath = "/path/to/model.pb"
22
23 // Customize how the StarDist detection should be applied
24 // Here some reasonable default options are specified
25 def stardist = StarDist2D
```

Note: StarDist is rather computationally expensive, typically can take ~ 5 min for 100k detections

StarDist for 2D segmentation of DAPI-stained nuclei

- Some pre-trained StarDist models are freely available as `.pb` files (frozen)
- Go to <https://github.com/qupath/models/raw/main/stardist> and download the `dsb2018_heavy_augment.pb` model

StarDist models

Here you can find pre-trained StarDist models as frozen `.pb` files that are compatible with OpenCV's DNN module.

This means they can be used in QuPath via the [QuPath StarDist extension](#) without any requirement to install TensorFlow.

Downloads

The converted model files are

- [dsb2018_heavy_augment.pb](#) - single channel
- [dsb2018_paper.pb](#) - single channel
- [he_heavy_augment.pb](#) - RGB images

***dsb2018_heavy_augment.pb* is pre-trained for 2D fluorescence images (one detection channel)**

Using StarDist extension in QuPath

- Change the value of the **modelPath** variable to an actual StarDist model path in the script

Change the
channel
name

The screenshot shows the QuPath Script Editor window. The code is as follows:

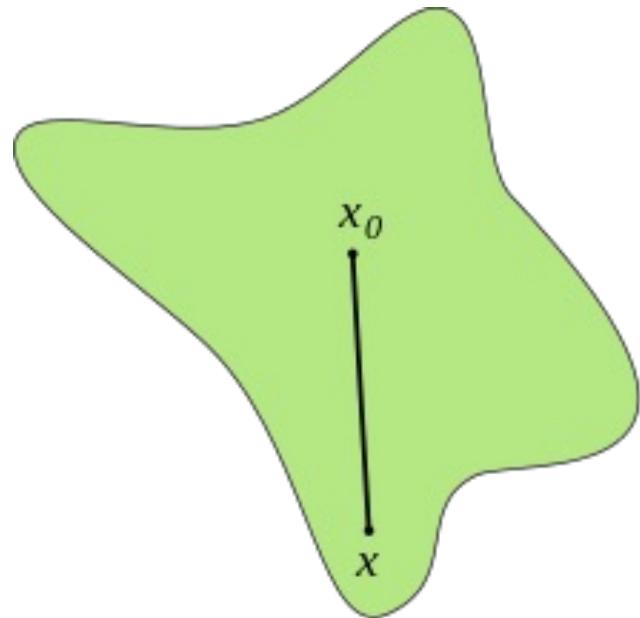
```
19 // You can find some at https://github.com/qupath/models
20 // (Check credit & reuse info before downloading)
21 def modelPath = "/Users/antoine/Desktop/test_qupath_workshop/models/dsb2018_heavy_augment.pb"
22
23 // Customize how the StarDist detection should be applied
24 // Here some reasonable default options are specified
25 def stardist = StarDist2D
26     builder(modelPath)
27     .channels('DAPI (C1)')           // Extract channel called 'DAPI'
28     .normalizePercentiles(1, 99)    // Percentile normalization
29     .threshold(0.5)                // Probability (detection) threshold
30     .pixelSize(0.5)                // Resolution for detection
31     .cellExpansion(5)              // Expand nuclei to approximate cell boundaries
32     .measureShape()                // Add shape measurements
33     .measureIntensity()            // Add cell measurements (in all compartments)
34     .build()
35
INFO: Done!
```

A red box highlights the line `def modelPath = "/Users/antoine/Desktop/test_qupath_workshop/models/dsb2018_heavy_augment.pb"`. A red arrow points from the text "Change the channel name" to the line `.channels('DAPI (C1)')`.

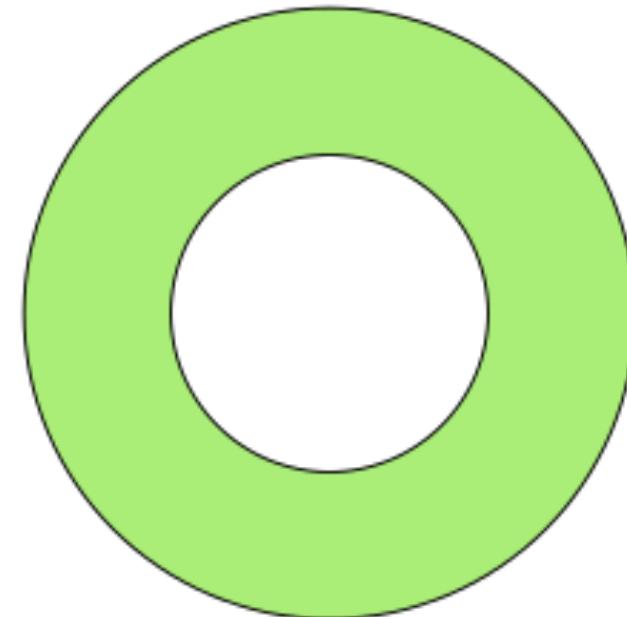
Make sure to
select the ROI in
QuPath before
running the script.

Exercise 3.b: QuPath cell detection with StarDist

Compare StarDist to threshold-based cell detection, what do you observe?



StarDist can segment



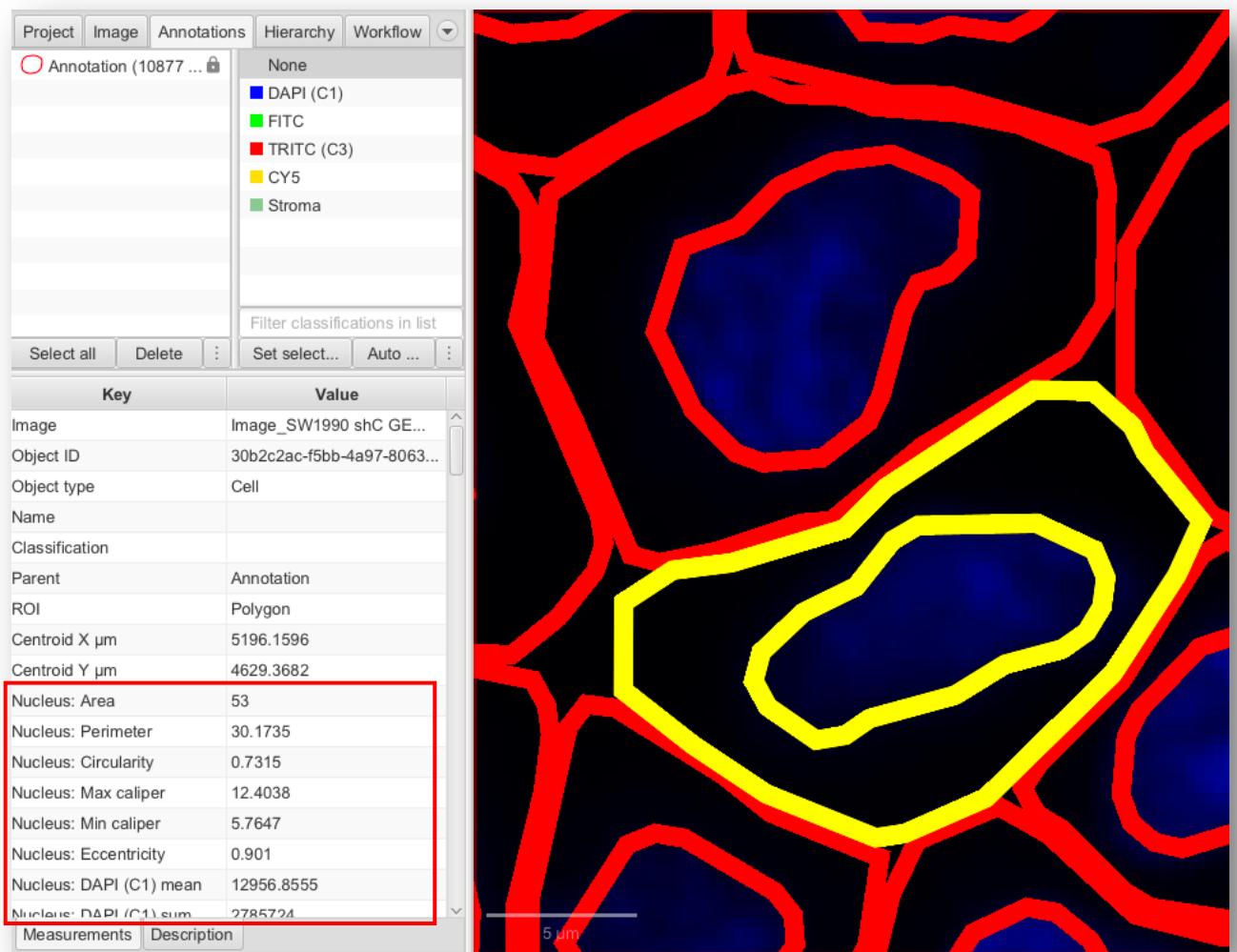
StarDist can not segment

Cell detection measurements

Detection measurements

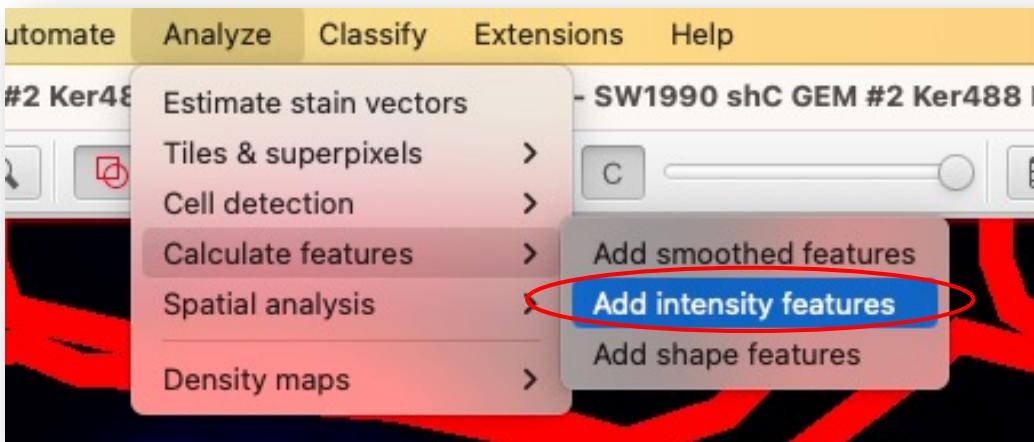
- Each detection object (i.e. a cell) has its measurement list
 - Intensity features
 - Haralick (texture) features
 - Shape features
 - Smoothed features
- *Annotations* tab > select a cell in the viewer > inspect its measurements list

By default, basic intensity and shape features are calculated



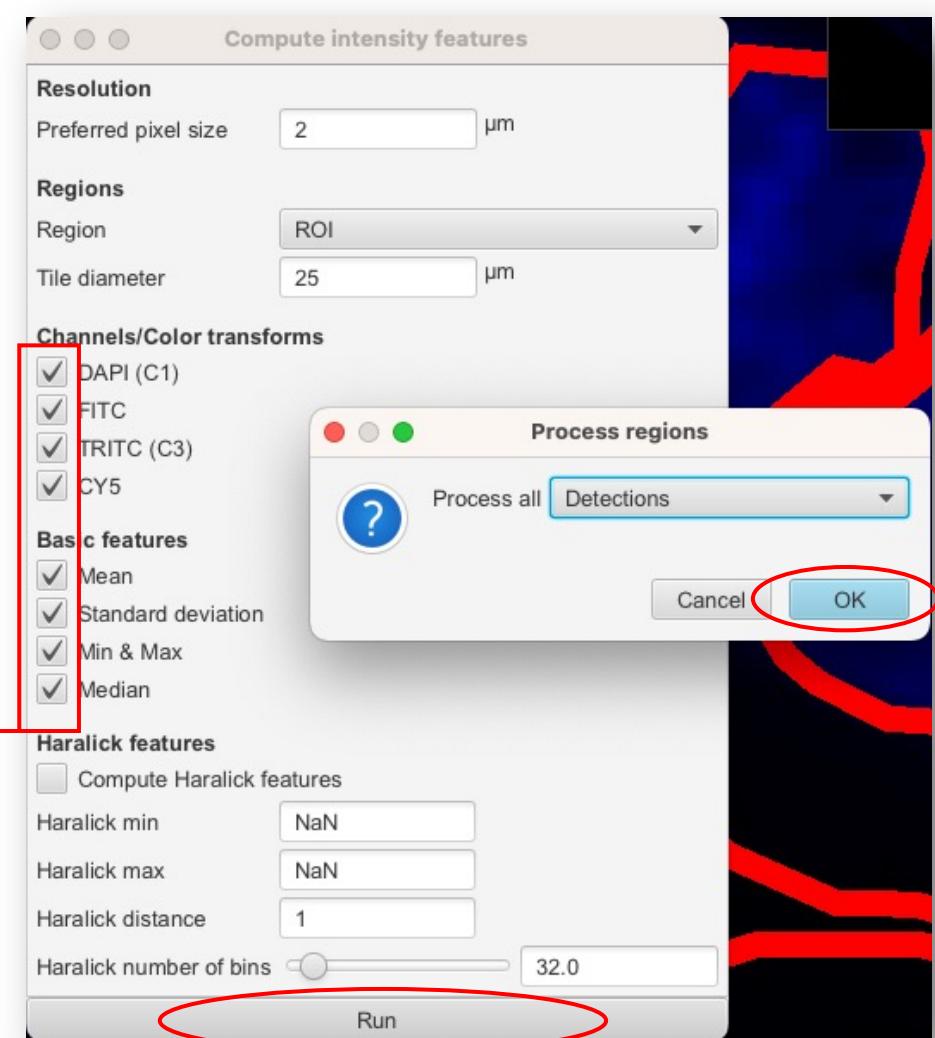
Calculating measurements

- *Analyze > Calculate features > Add intensity features*



Tick boxes of the channels and
features of interest

Need a custom feature? Script it!



Visualizing measurements

- *Measure > Show detection measurements*

The screenshot shows the QuPath software interface. On the left, a navigation bar includes 'Objects', 'TMA', 'Measure' (which is highlighted in green), and 'Automate'. A context menu is open under 'Measure', with 'Show detection measurements' highlighted in blue and circled in red. Other options in the menu include 'Show measurement maps', 'Show measurement manager', 'Show TMA measurements', 'Show annotation measurements', 'Grid views...', and 'Export measurements'. The main workspace displays a digital slide with several cells outlined in red. To the right is a detailed table of 'Detections'.

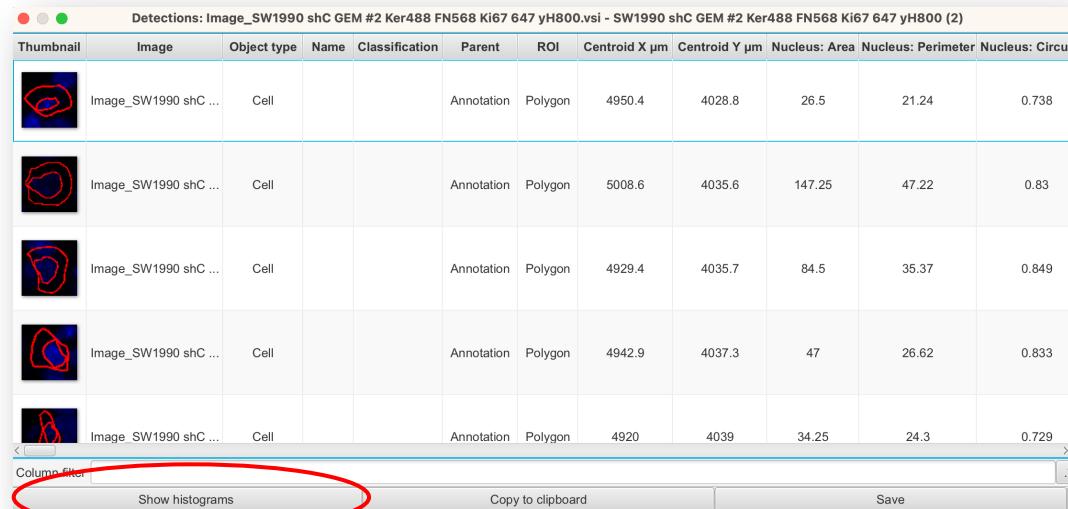
Columns: measurements

Rows: cells

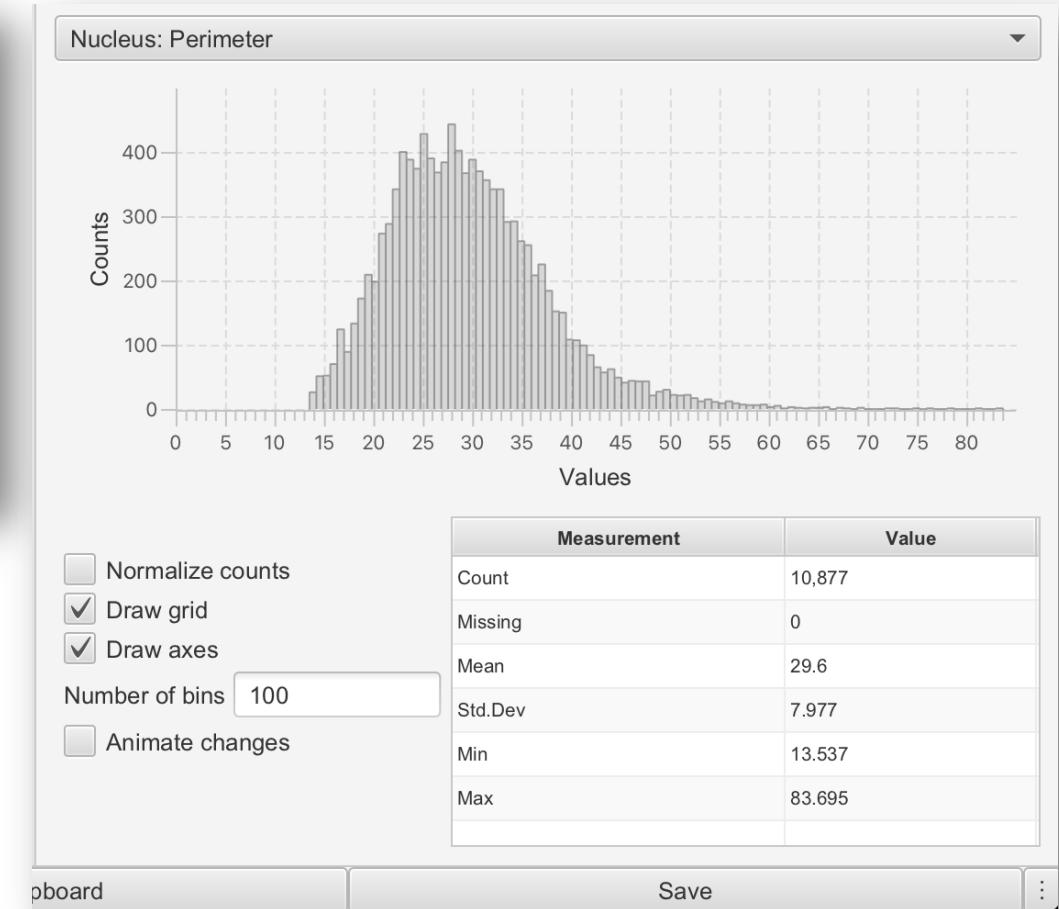
Thumbnail	Image	Object type	Name	Classification	Parent	ROI	Centroid X µm	Centroid Y µm	Nucleus: Area	Nucleus: Perimeter	Nucleus: Circular.	
	Image_SW1990 shC ...	Cell				Annotation	Polygon	4950.4	4028.8	26.5	21.24	0.738
	Image_SW1990 shC ...	Cell				Annotation	Polygon	5008.6	4035.6	147.25	47.22	0.83
	Image_SW1990 shC ...	Cell				Annotation	Polygon	4929.4	4035.7	84.5	35.37	0.849
	Image_SW1990 shC ...	Cell				Annotation	Polygon	4942.9	4037.3	47	26.62	0.833
	Image_SW1990 shC ...	Cell				Annotation	Polygon	4920	4039	34.25	24.3	0.729

Visualizing measurement distributions

- *Measure > Show detection measurements*

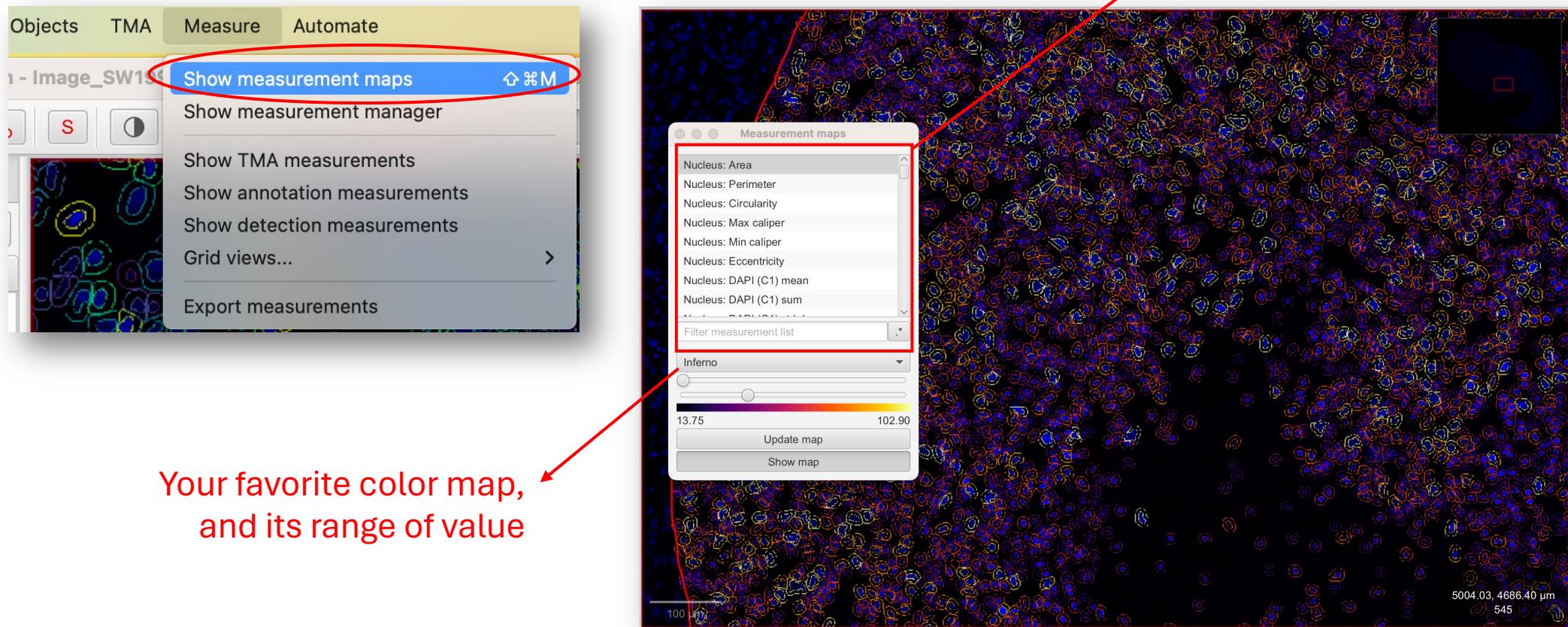


FYI, it is not possible to export distribution plots...



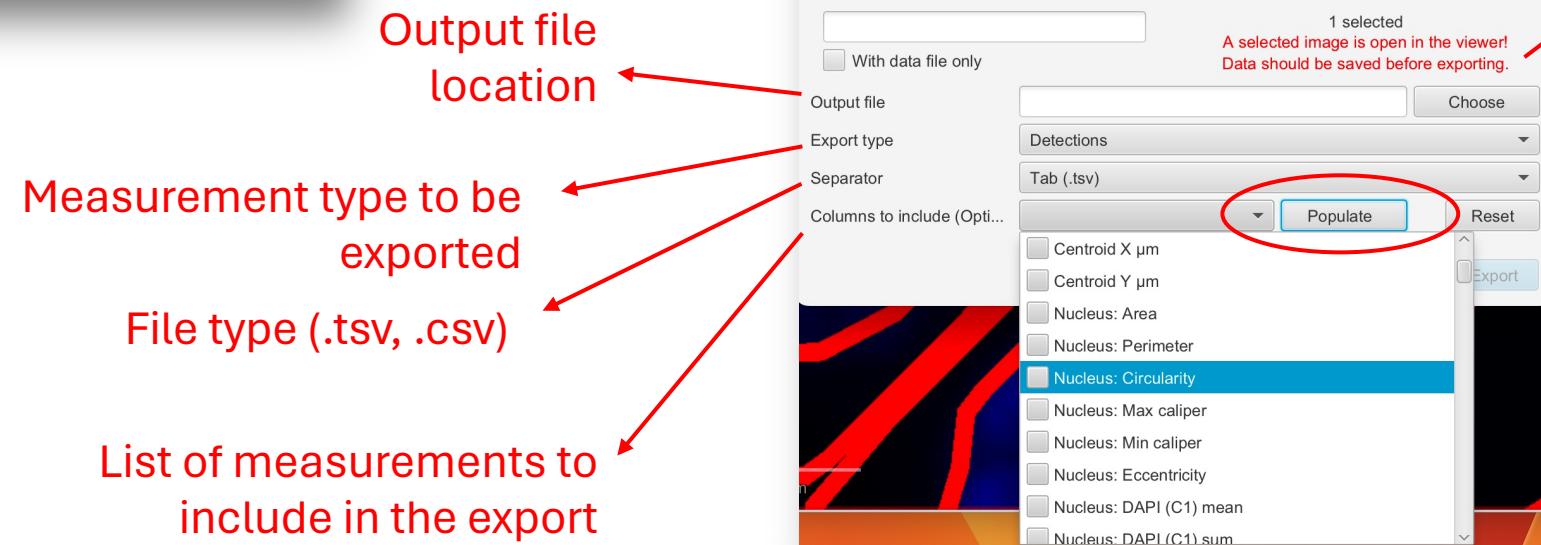
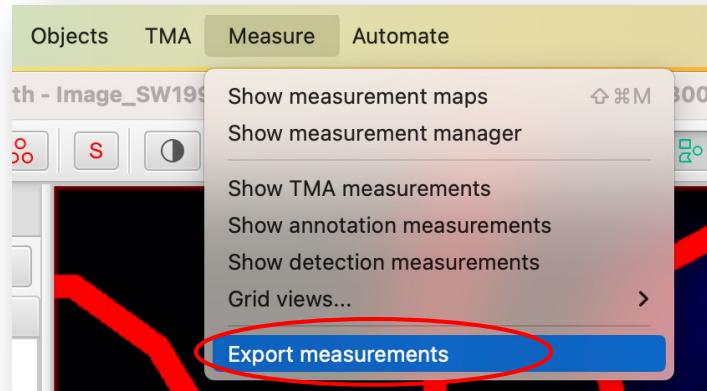
Visualizing measurements as heat maps

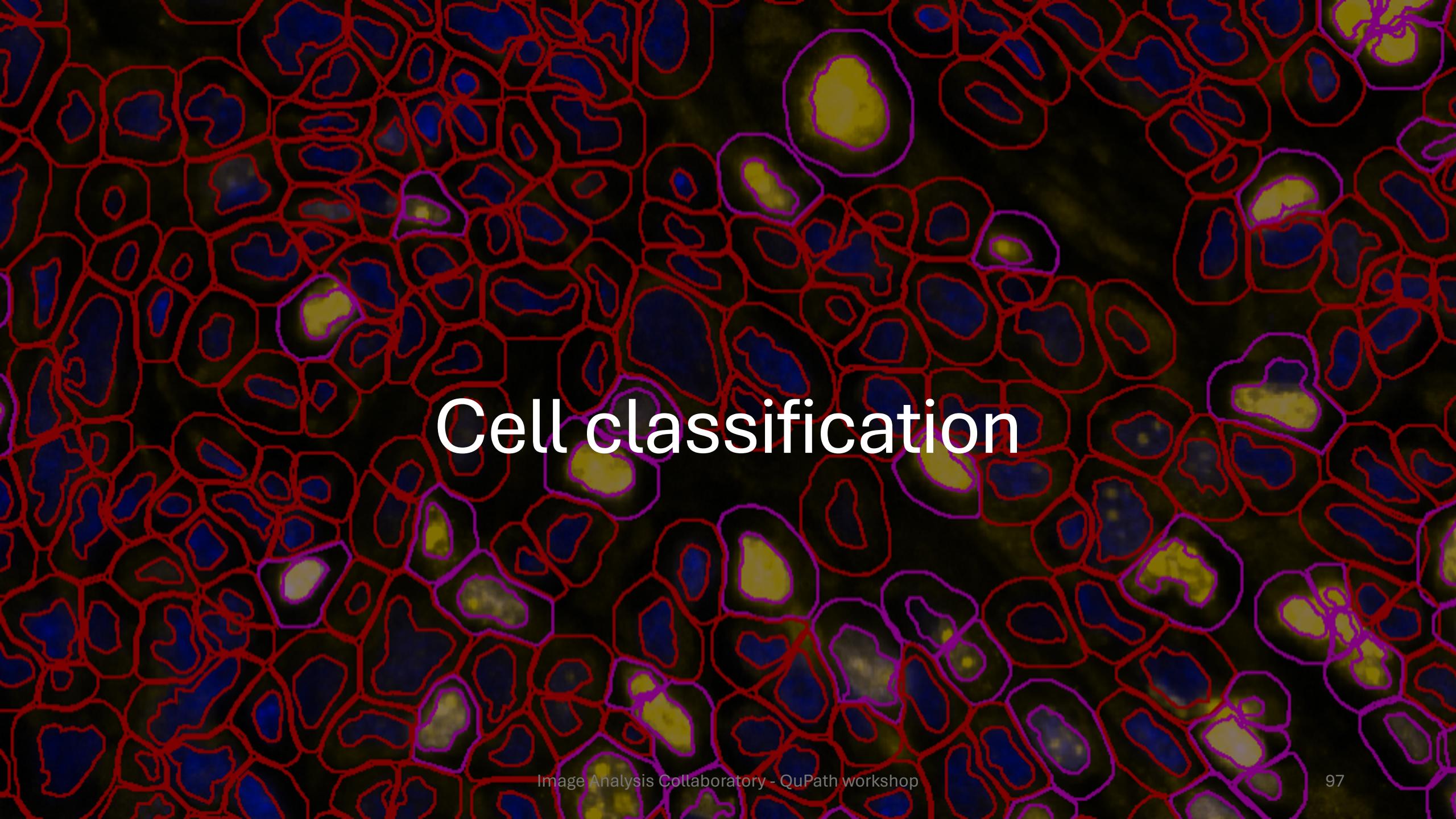
- *Measure > Show measurement maps*



Export measurements the right way

- *Measure > Export measurements*
- Drag an image from Available to Selected





Cell classification

What is cell classification?

explain in one sentence to a toddler what is cell classification in the context of image analysis

< 5/5 >



Cell classification is like sorting different kinds of toys into separate boxes based on how they look.

Thank you, I'm sure students will love being compared to toddlers

< 3/3 >



It's all about finding the right analogy for understanding! If it helps simplify a complex concept, then it's worth it.



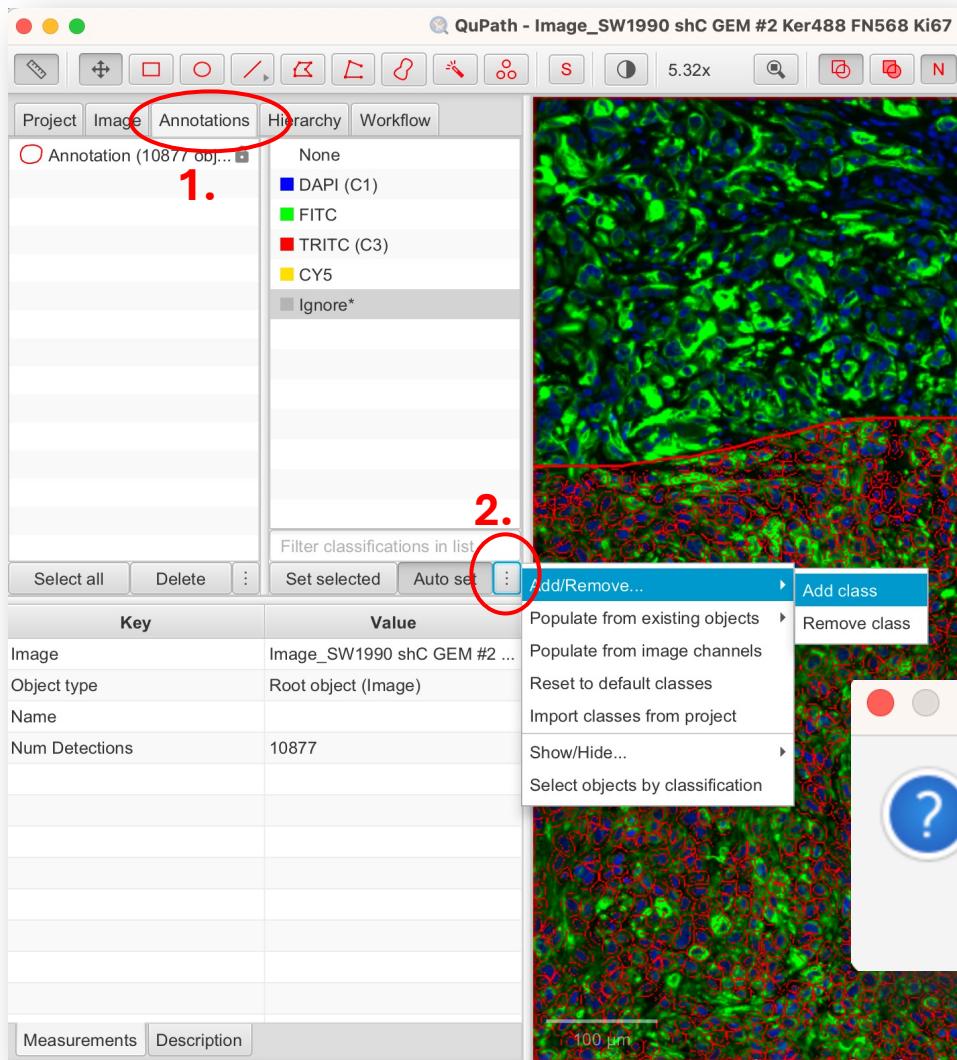
What is cell classification?

- Cell classification in image analysis is sorting different types of cells into groups based on their features or characteristics, such as shape, color, position, neighbors, etc.
- As a rule, if visually you can't discriminate cells in your images, then your algorithm might be separating cell on very fine differences (*overfitting*)

Cell classification in QuPath

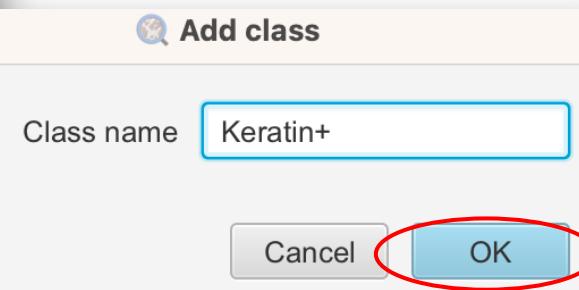
- Single measurement classifier
- Composite thresholder: combine single measurement thresholders together
- Train a machine learning model

Creating classes



- *Annotations tab > Classification list > ... > Add/Remove... > Add class*

**Default classes are channel names.
We'll change that.**



3.

Create a second class named ‘Ki67+’

The screenshot shows the QuPath software interface for image analysis. The main window displays a fluorescence microscopy image with blue (DAPI), green (FITC), and red (TRITC) channels. The top navigation bar includes Project, Image, Annotations (highlighted with a red circle), Hierarchy, and Workflow. A legend on the left identifies the channels: None, DAPI (C1), FITC, TRITC (C3), CY5, and Ignore*. Below the image is a table with project metadata: Image_SW1990 shC GEM #2 Ker488 FN568 Ki67+, Object type: Root object (Image), Name: 10877, and Num Detections: 10877. A context menu is open at the bottom center, with step 2 highlighting the 'Add/Remove...' option. A sub-menu for 'Add class' is shown, with step 3 highlighting the 'OK' button. A callout box with step 1 highlights the 'Annotations' tab. A red box also highlights the 'Add/Remove...' button in the main menu.

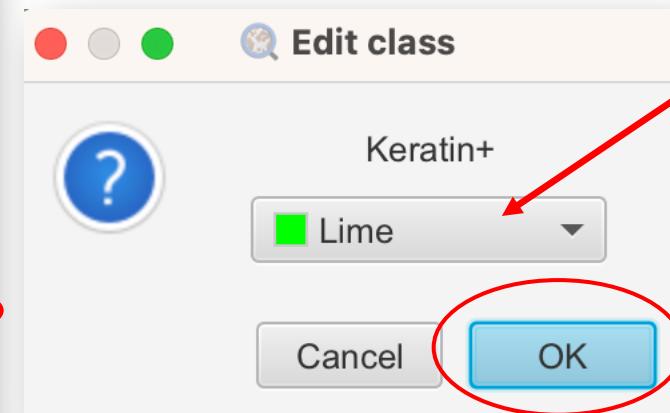
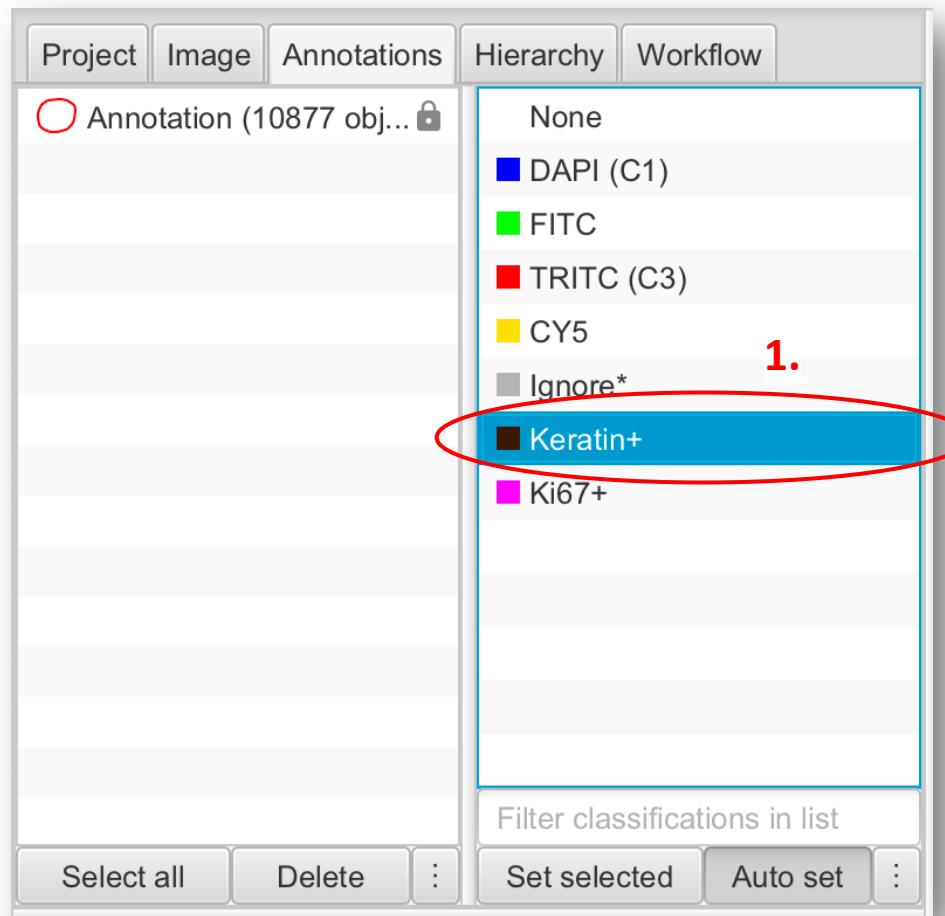
- *Annotations* tab > Classification list > > Add/Remove... > Add class

**Default classes are channel names.
We'll change that.**

3.

Change the color of a class

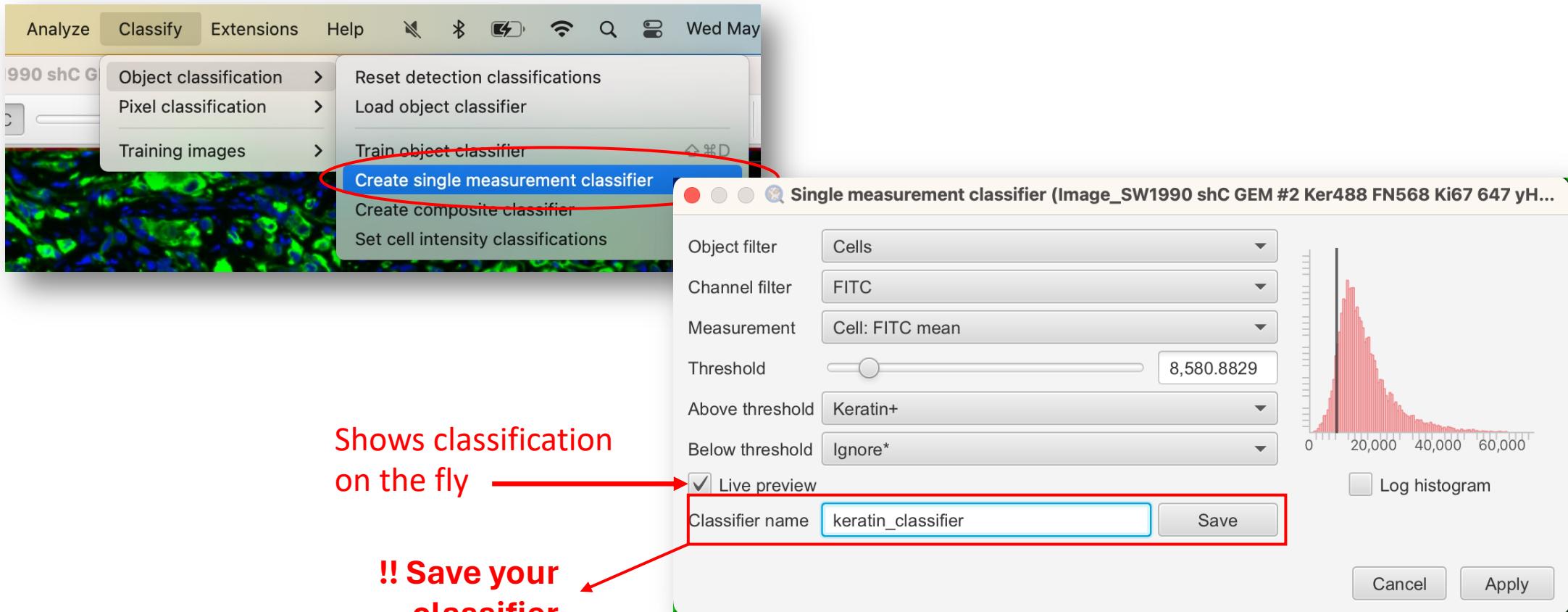
- Double click on the class > Edit class > Choose a new color > *OK*



By default, classes are populated by image channels.

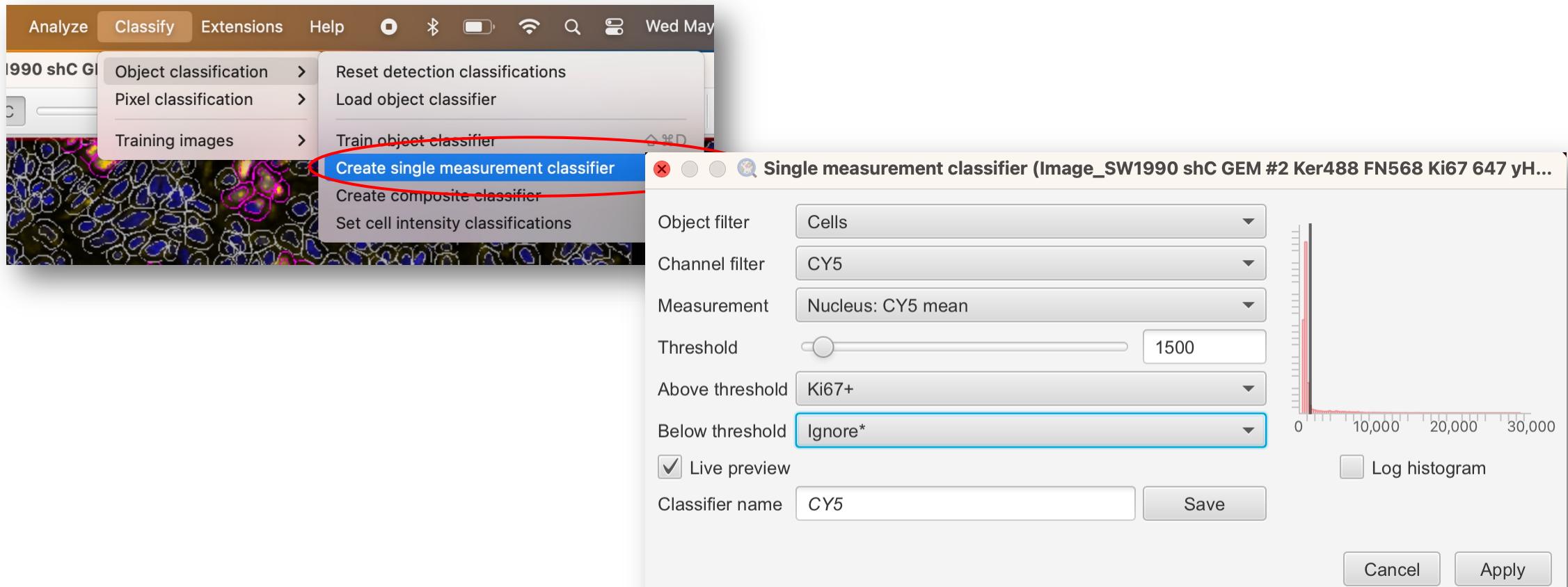
Simple measurement classifier on Keratin signal intensity (FITC channel)

- *Classify > Object classification > Create single measurement classifier*



Simple measurement classifier on Ki67 signal intensity (CY5 channel)

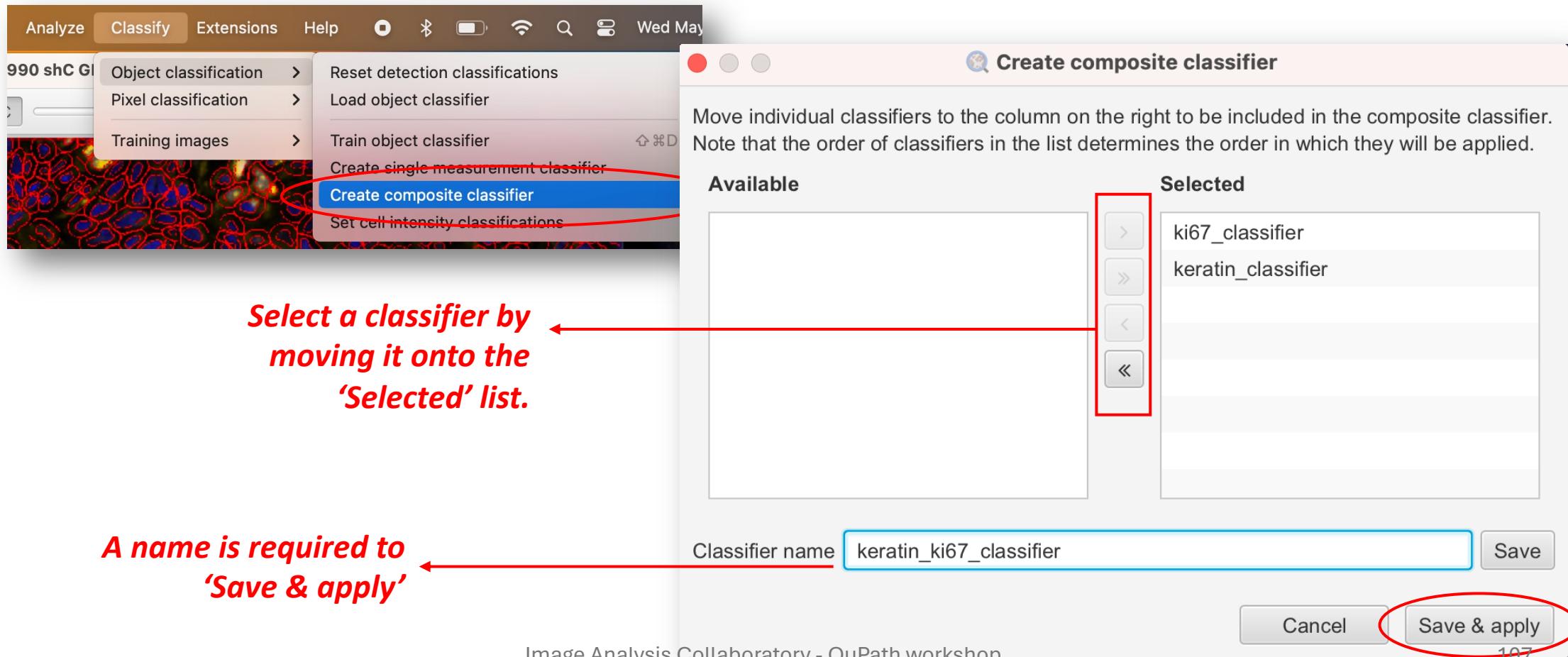
- *Classify > Object classification > Create single measurement classifier*



Exercise 4.a: single-measurement classifier

Combine single measurement classifiers into a composite classifier

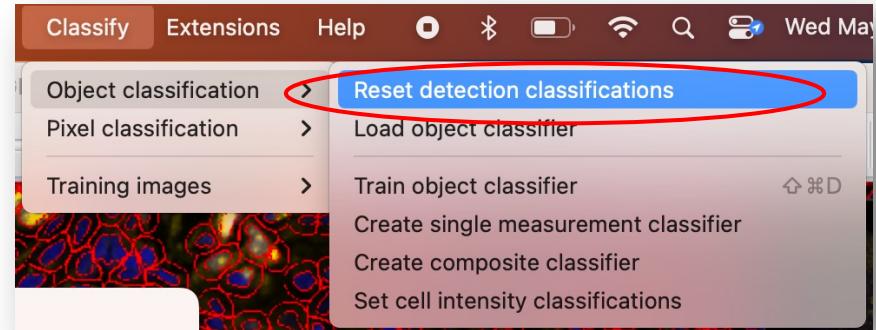
- *Classify > Object classification > Create composite classifier*



Exercise 4.b: composite classifier

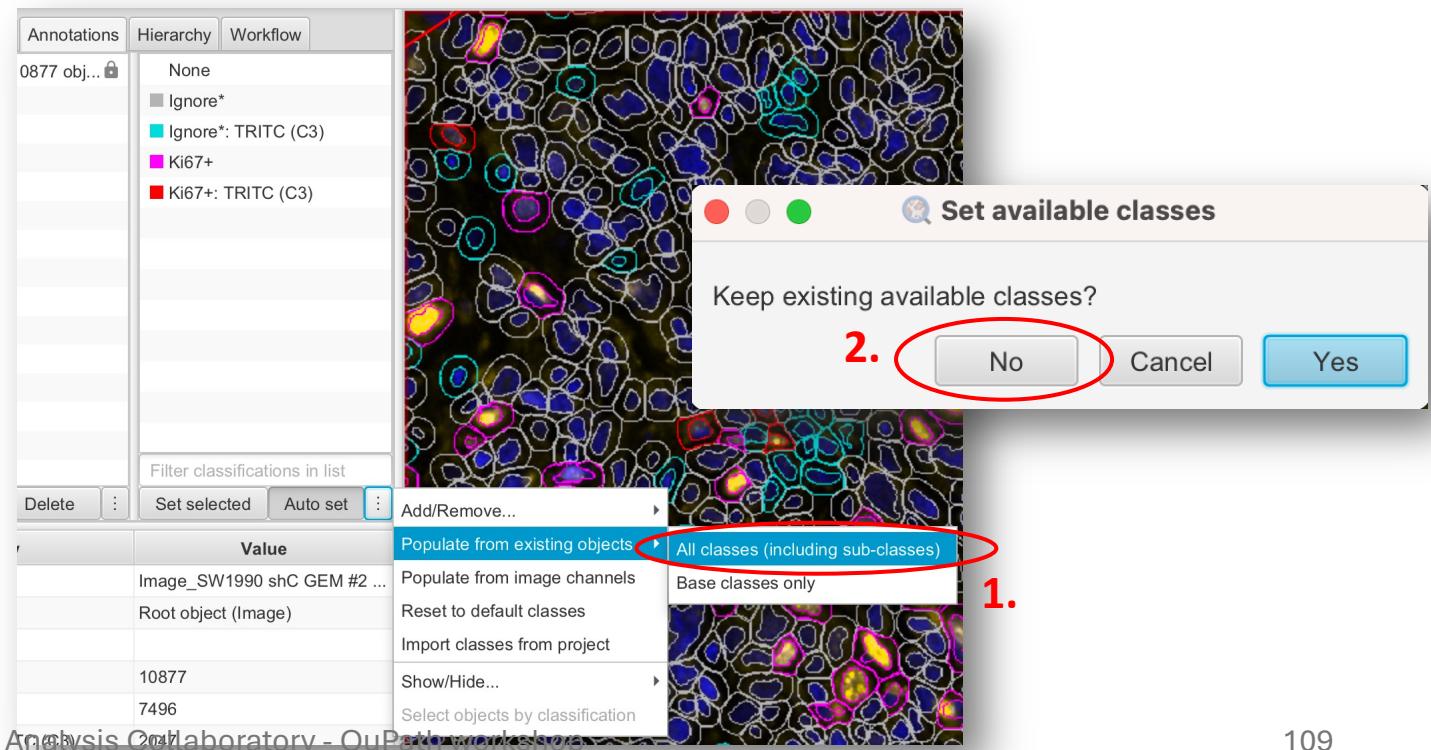
Reset detection classes

- *Classify > Object classification > Reset detection classifications*



Populate classes in the classification panel

- *Annotations tab > Classification list > [...] > Populate from existing objects > All classes (including sub-classes)*



Object classification using machine learning

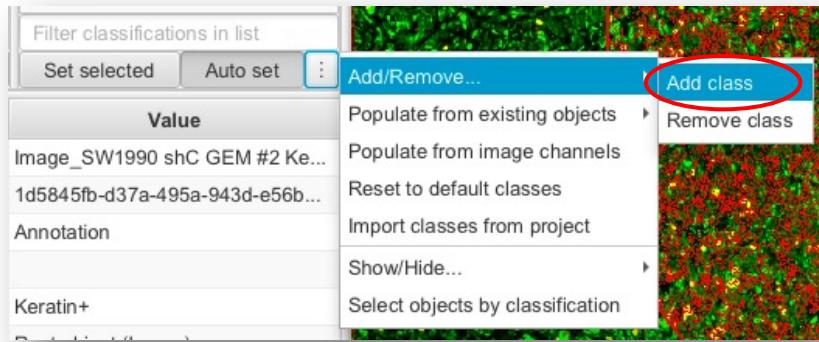
- Detections (and annotations) can be classified into classes using a ML classifier
- Classification requires measurements!
- Object classifiers are trained using manual annotations of 2 or more classes
 - Need to create some training data points
- **Live demo of object classification using ML**

Reset your detection classes!



Train an object classifier: create classes

- *Annotations* tab > Classification list >  > Add/Remove... > Add class



- Create 4 classes:
 - Keratin+
 - Keratin-
 - Ki67+
 - Ki67-

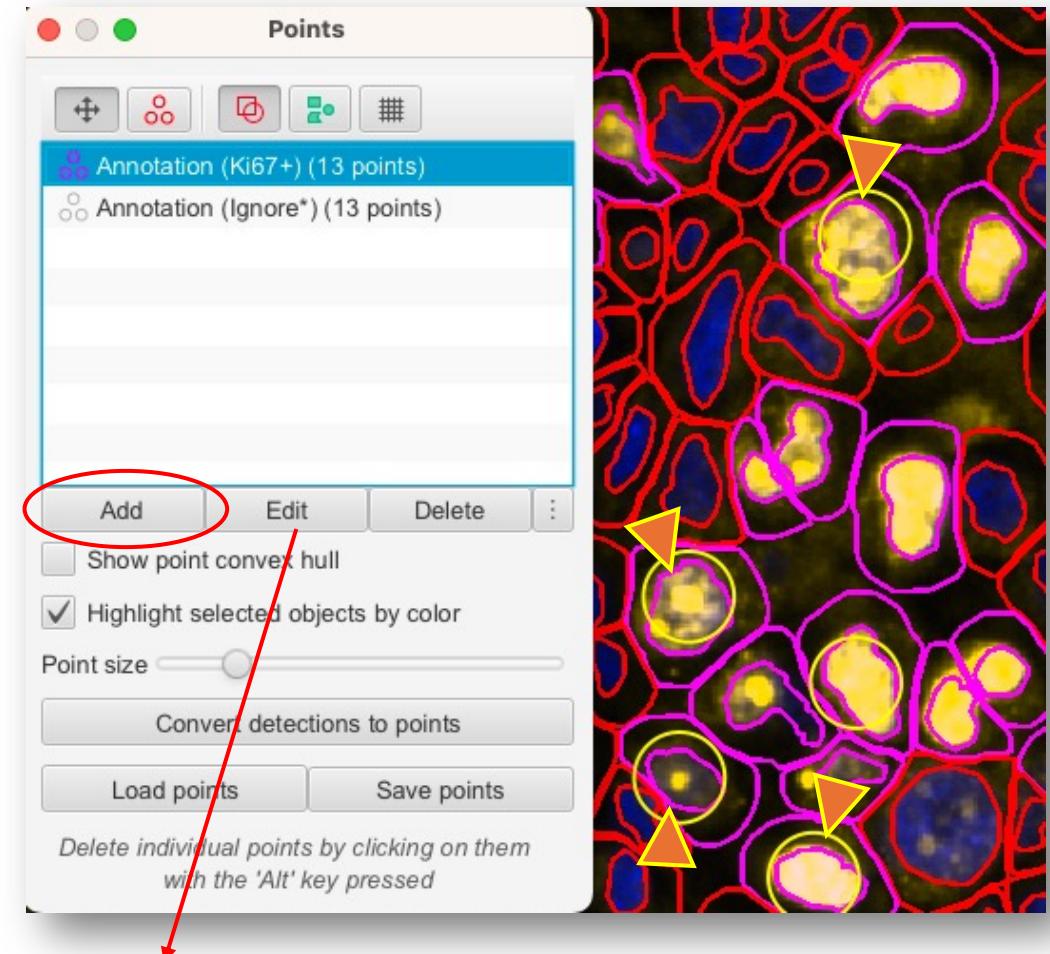


Train an object classifier: training data points

-  > Add > Label ~10 for each class

To remove a single point:
Option + click (Mac) or left-click

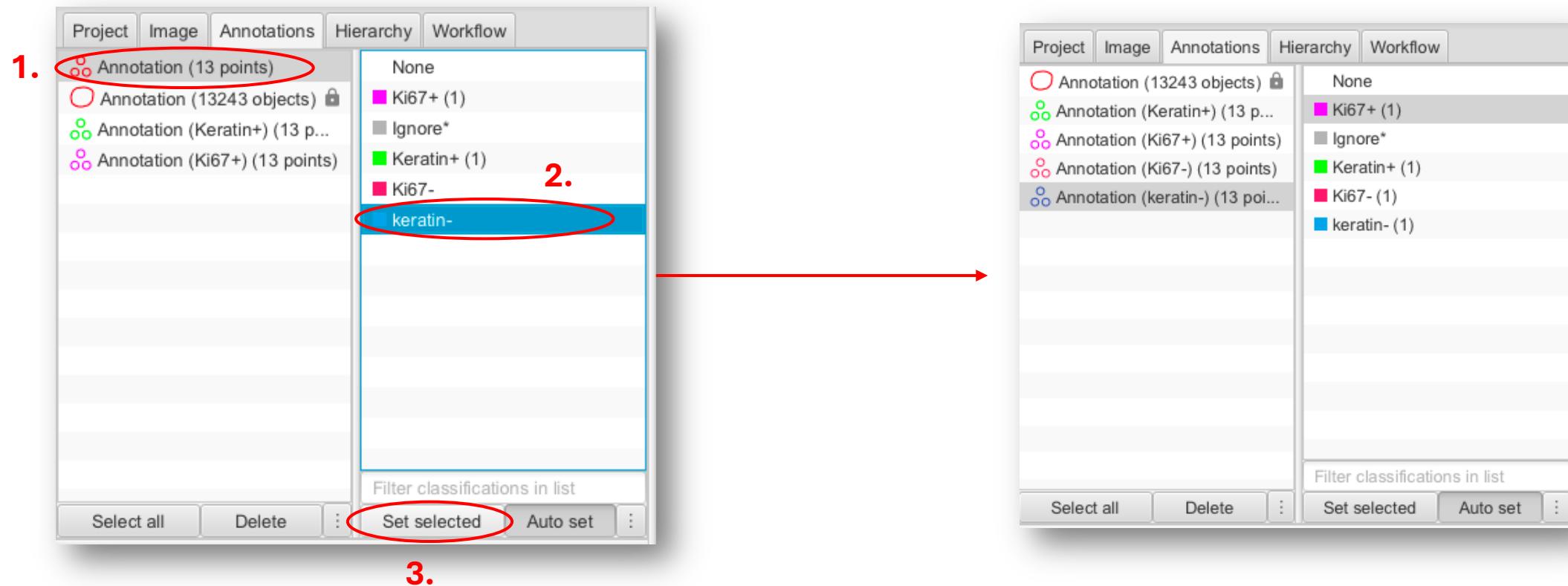
- Assign each training data set a class:
 - Select the annotation set
 - Select the class



Click edit to change color

Train an object classifier: training data points

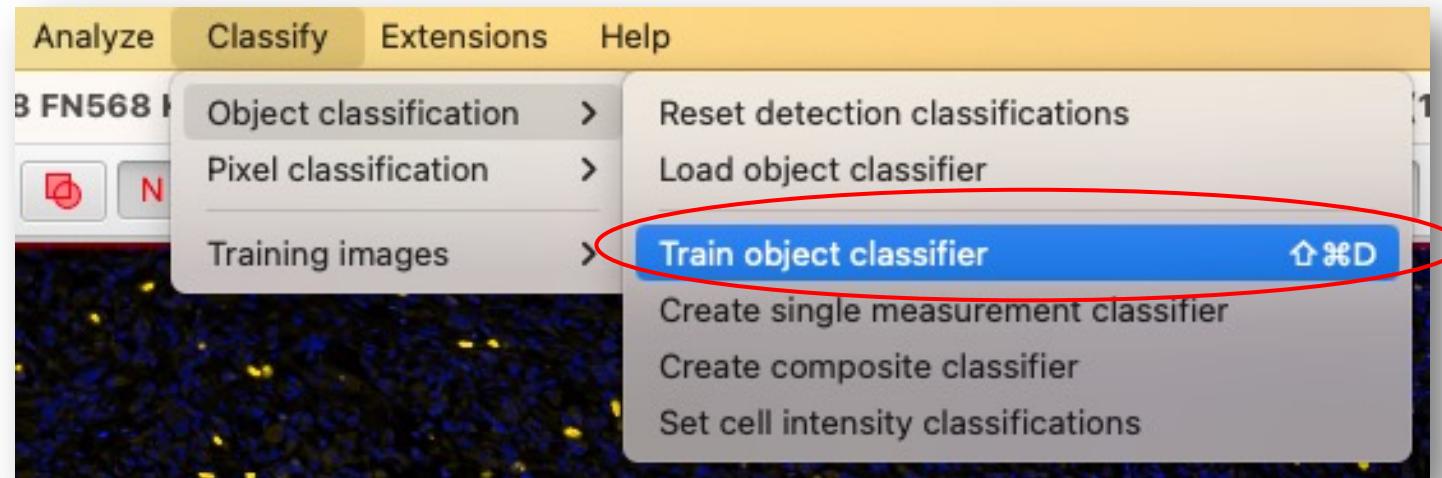
- Assign each training data set a class in the *Annotations* tab



Make sure to lock your annotation: Ctrl+click > Lock

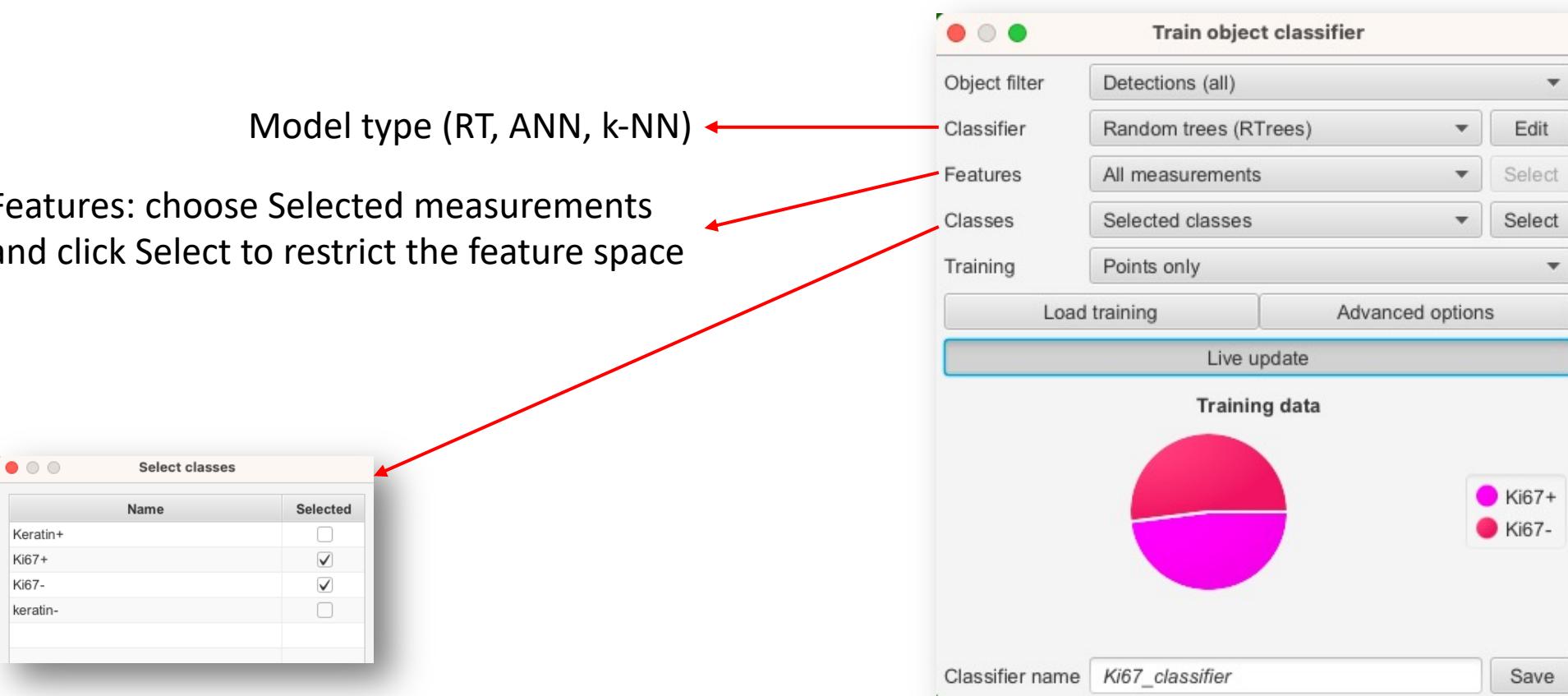
Train an object classifier

- *Classify > Object classification > Train object classifier*



Train an object classifier

- *Classify > Object classification > Train object classifier*



Name it to save it.

Combine multiple ML classifiers together

- *Classify > Object classification > Create composite classifier*

Move individual classifiers to the column on the right to be included in the composite classifier. Note that the order of classifiers in the list determines the order in which they will be applied.

Available	Selected
ML_ki67_keratin_classifier	>
test_classifier	»
keratin_ki67_classifier	<
	«

Classifier name Save

Cancel **Save & apply**

Project Image Annotations Hierarchy Workflow

- None
- Annotation (13243 objects)
- Annotation (Keratin+) (16 points)
- Annotation (Ki67+) (13 points)
- Annotation (Ki67-) (13 points)
- Annotation (keratin-) (13 points)

Filter classifications in list

Key	Value
Image	Image_SW1990 shC GEM #2 K...
Object ID	1d5845fb-d37a-495a-943d-e56...
Object type	Annotation
Name	
Classification	Keratin+
Parent	Root object (Image)
ROI	Points
Centroid X µm	5173.7928
Centroid Y µm	4524.1271
Num Detections	0
Num Ki67+: Keratin+	0
Num Ki67+: keratin-	0

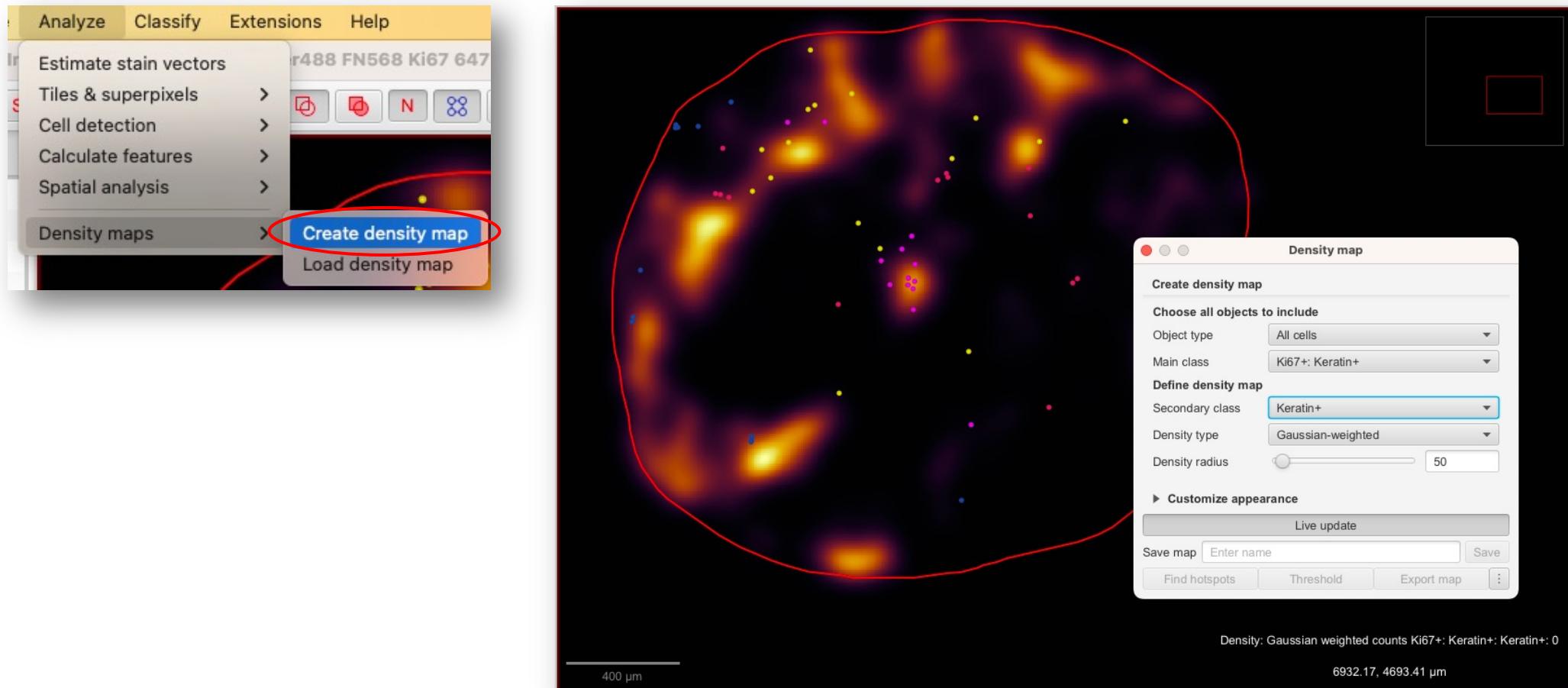
8 resulting classes!
+ ignore*

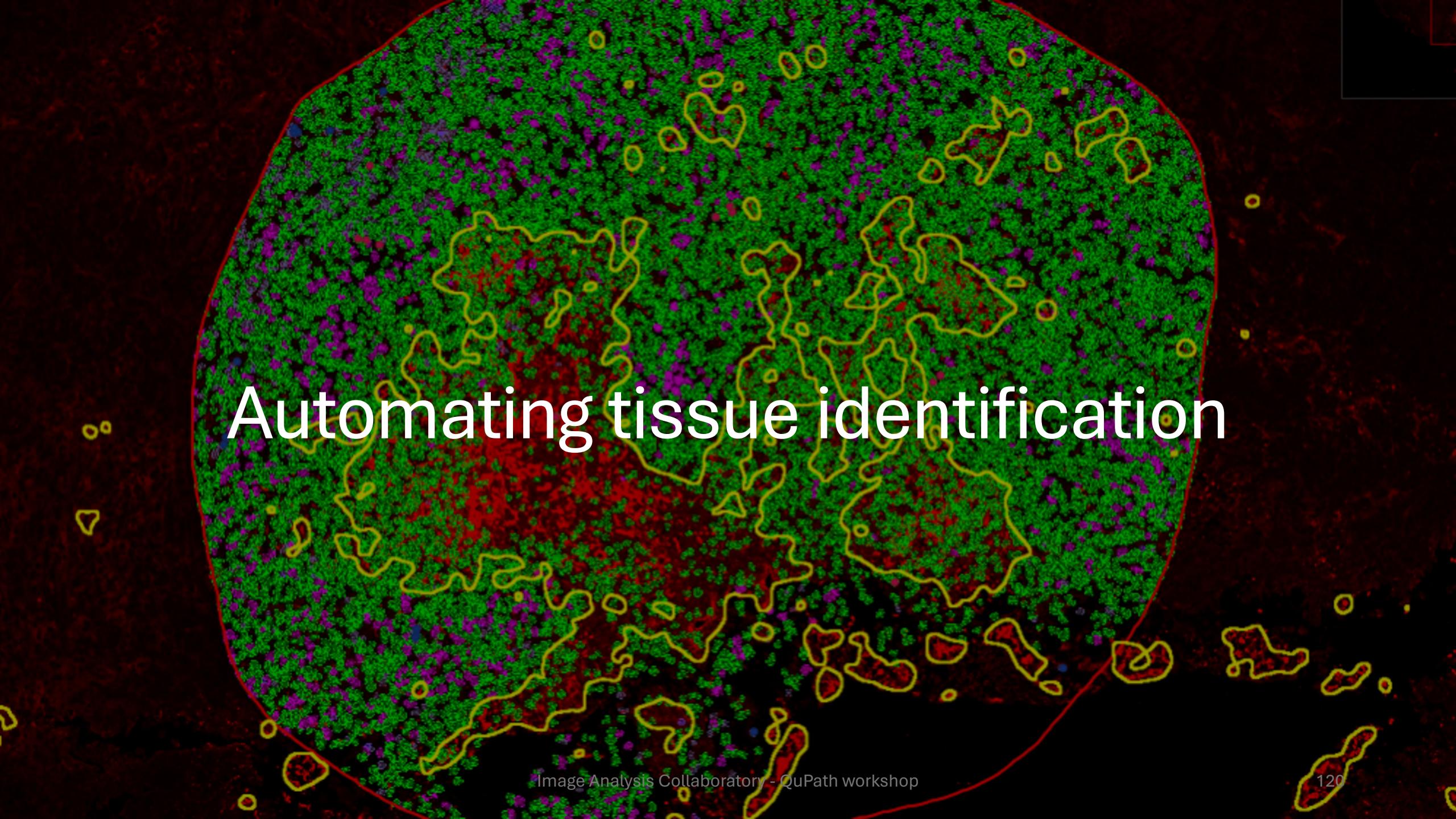
Refine your classifier

- Add more training data points
 - Classification results will change in real time if ‘Live update’ option is enabled
- Typically, *fewer*, but *well-chosen* features provides more robust results

Visualizing results using density maps

- Analyze > Density maps > Create density maps

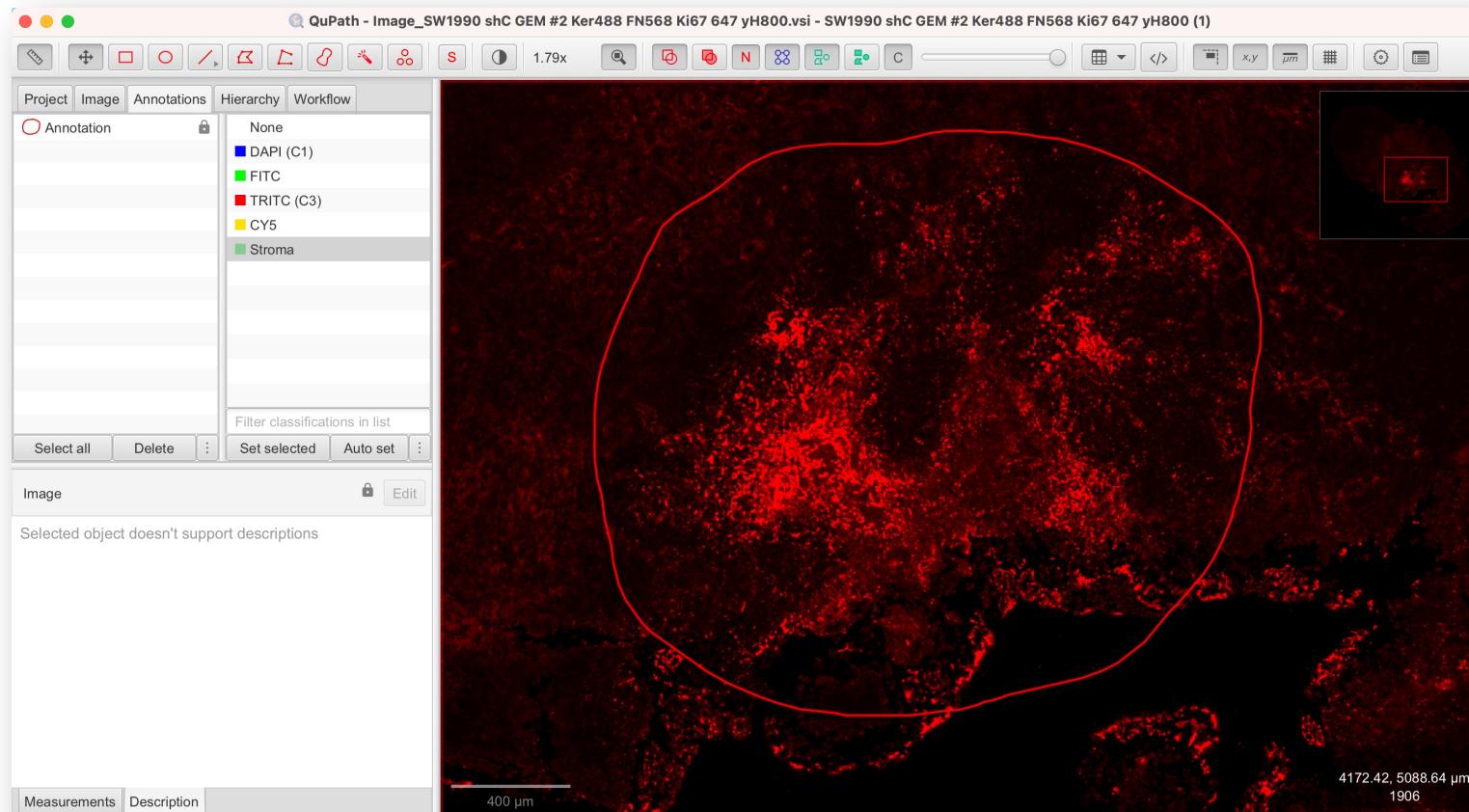




Automating tissue identification

Creating a region of interest

In the TRITC channel (fibronectin), create a region of interest that enclose high-fibronectin content regions aka stromal regions



Once you have finished your annotation, **lock** it:

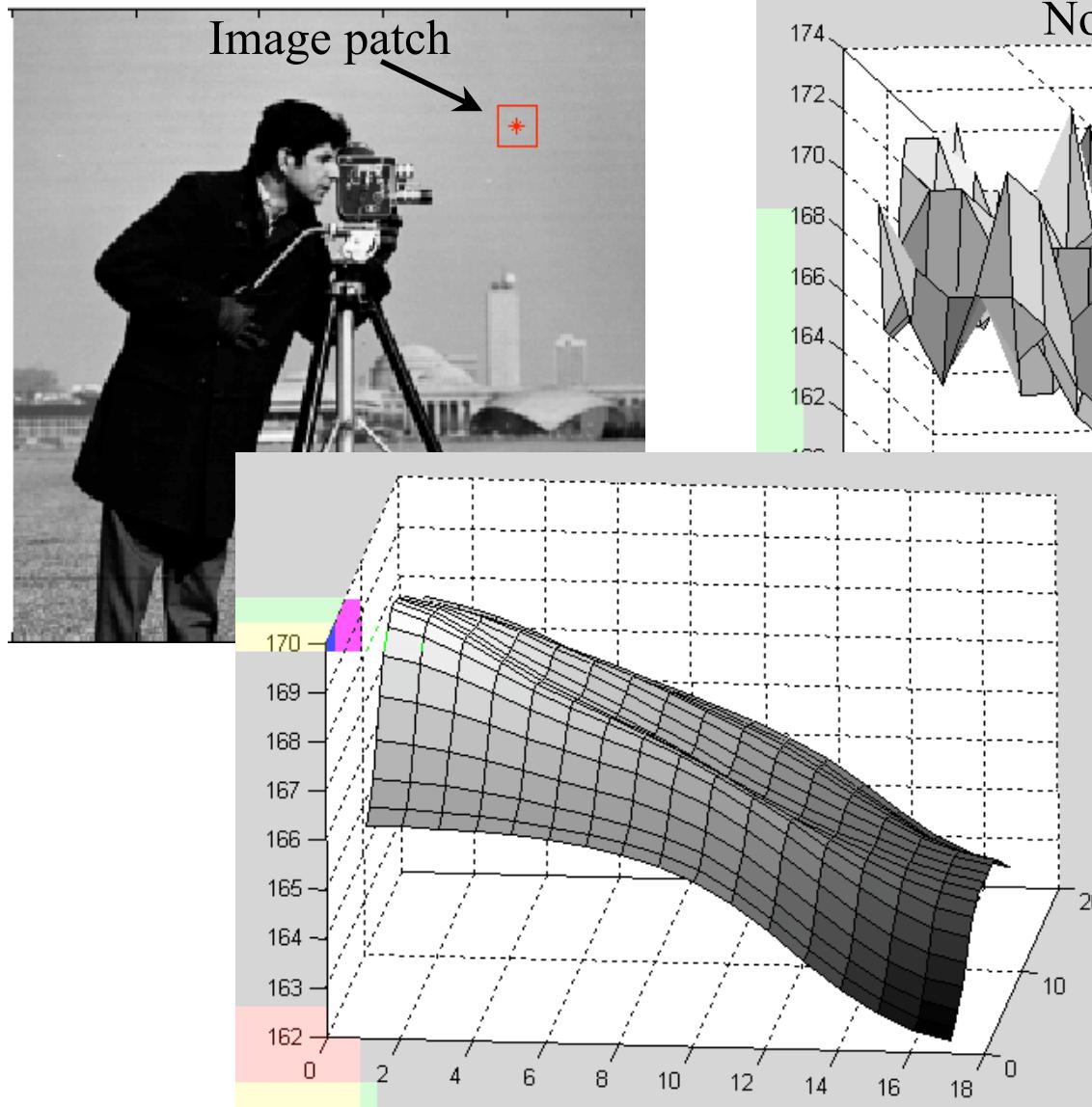
Right-click in the viewer
> Annotations > Lock

or

Right-click on the annotation in the analysis panel > Lock

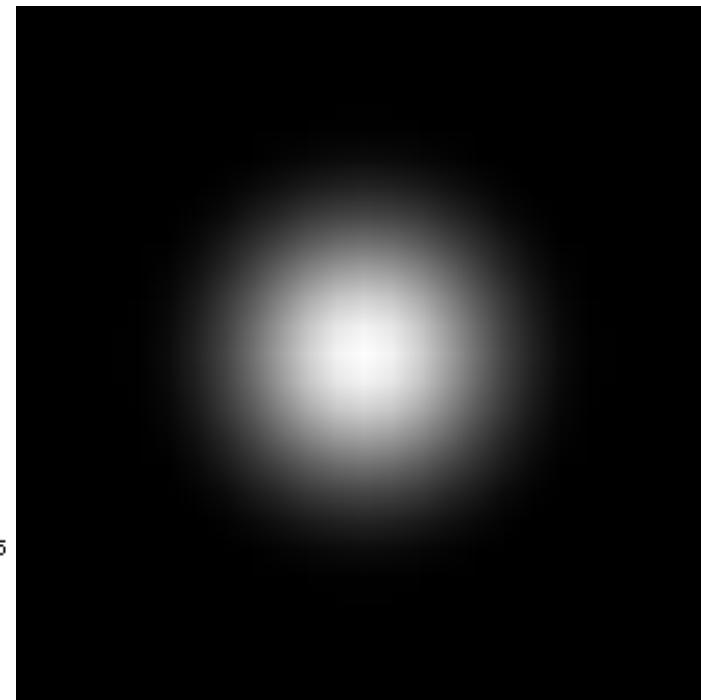
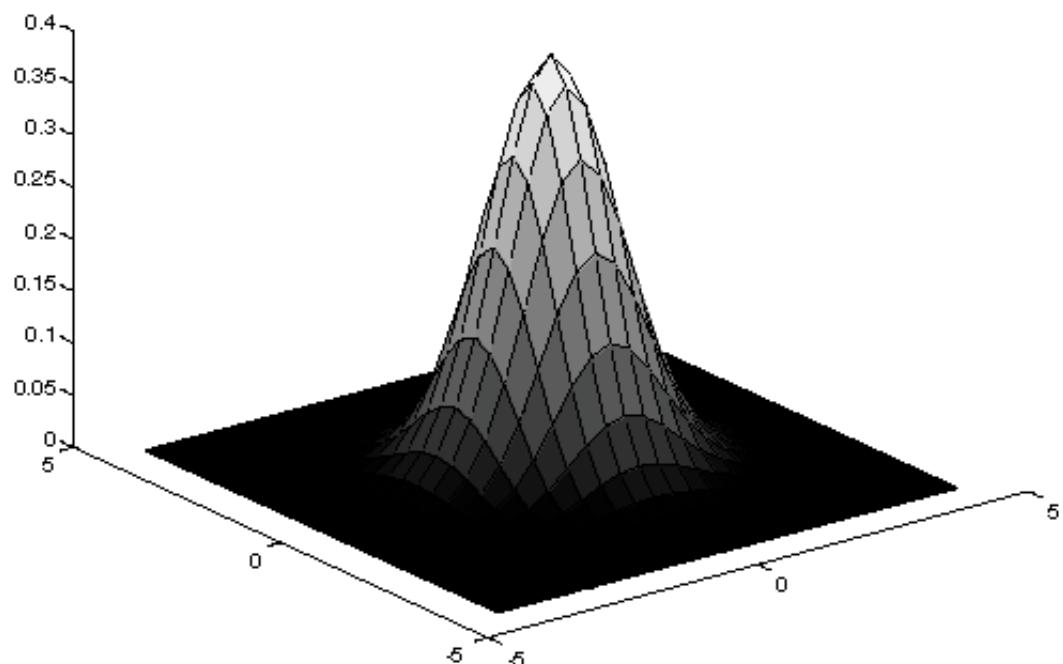
Notes on smoothing

Today: Smoothing Reduces Noise



Gaussian Smoothing Filter

An isotropic (circularly symmetric) Gaussian:



Gaussian Smoothing Example



original



$\sigma = 3$

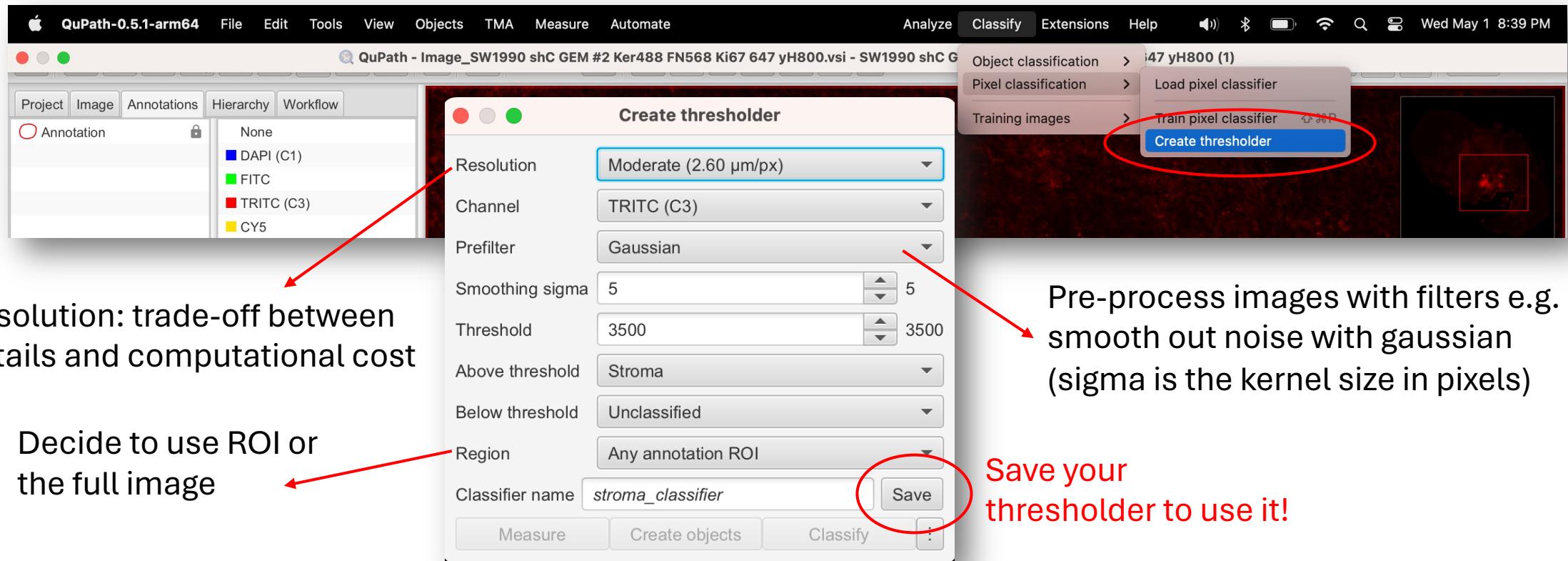
Gaussian Smoothing at Different Scales



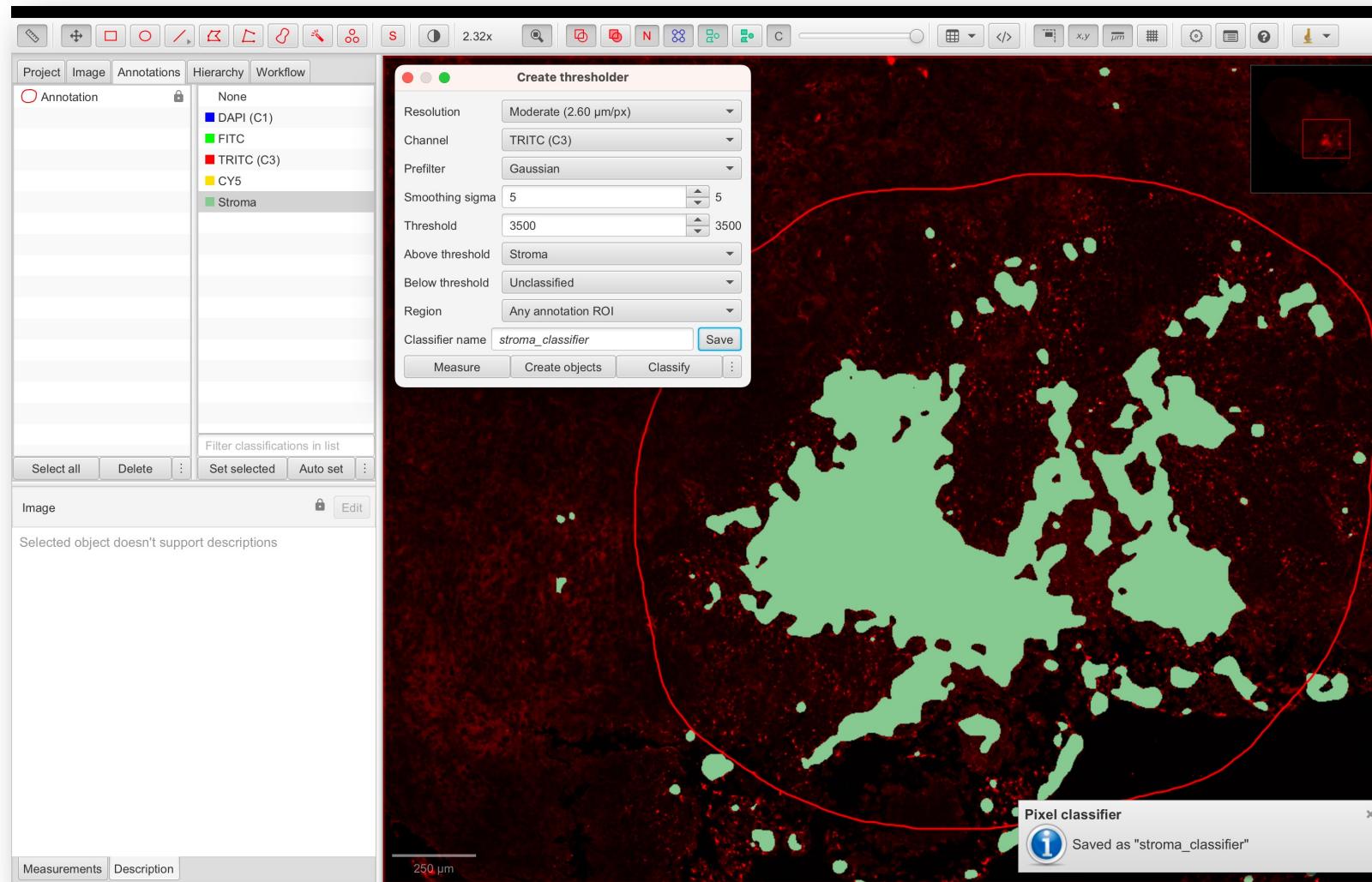
**Balancing act: smooth enough to “clean up”
the noise, but not so much as to remove
important image gradients.**

Pixel-based tissue annotation

- Simplest case of annotation: every pixel get assigned a class based on its intensity value – **or is a given pixel above or below a certain numeric value?**

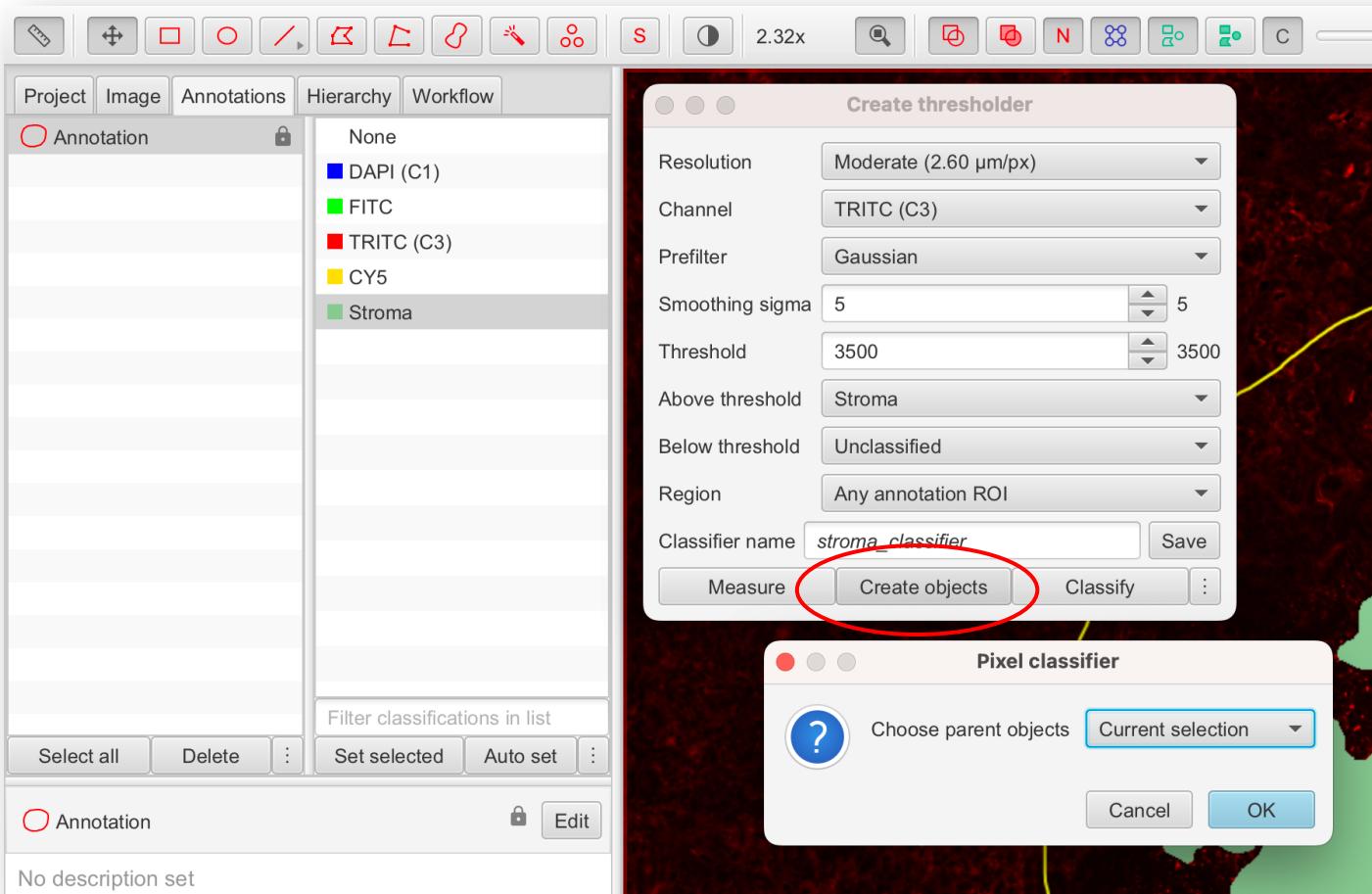


Interactive visualization of thresholding results

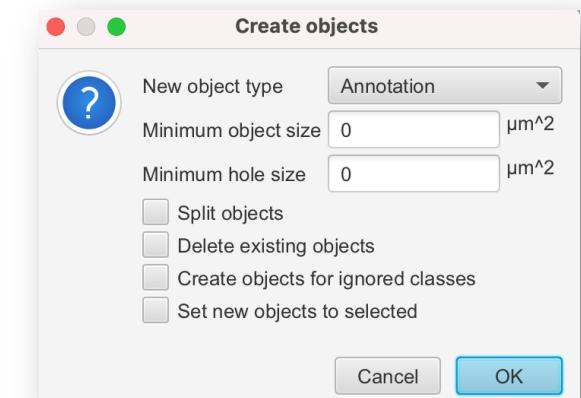


Try varying the value
of the different
parameters!

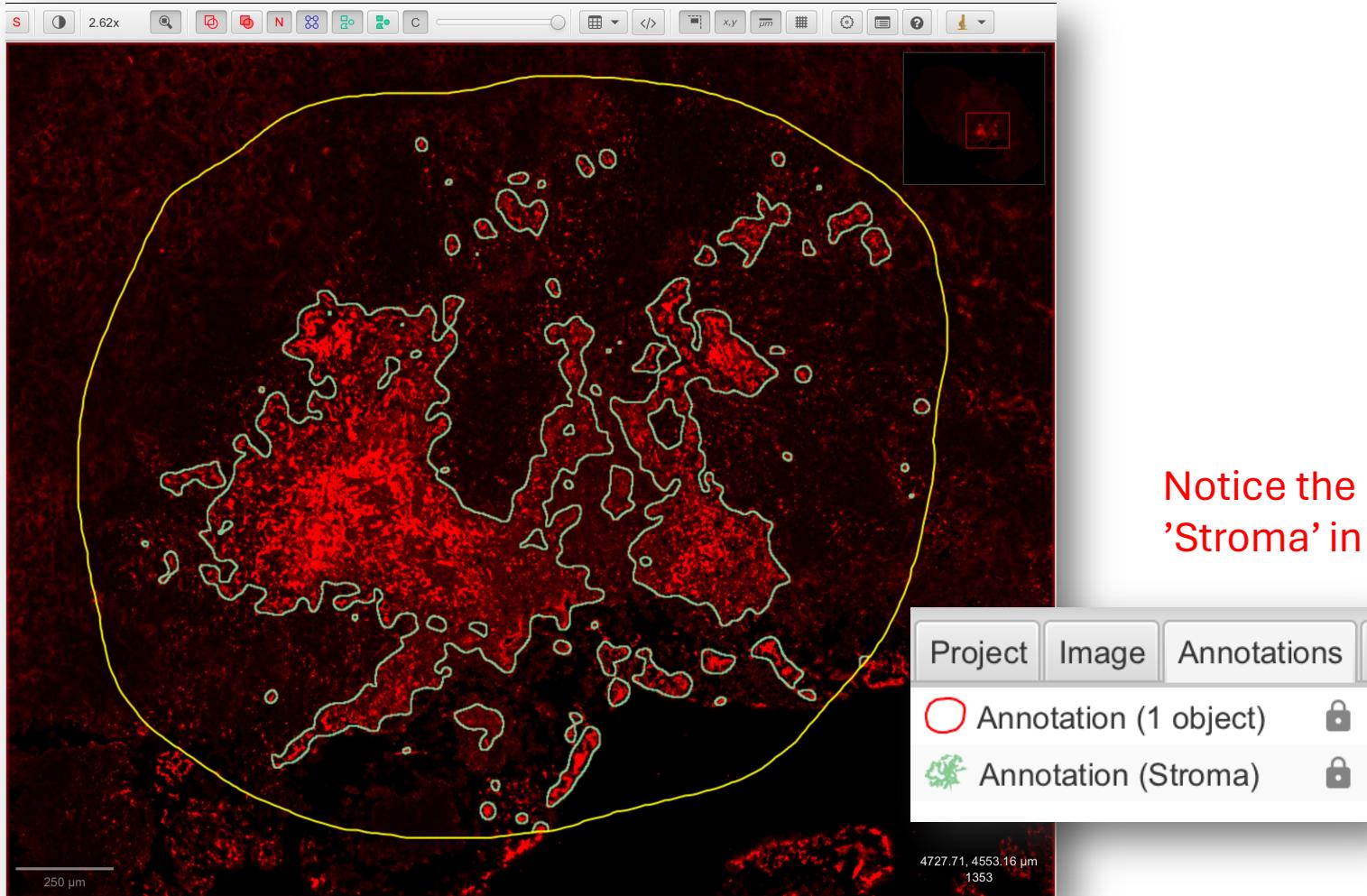
Create annotations from pixel classifier



- Real-time visualization of results, once happy with it:
 1. Save your thresholder
 2. Select ROI
 3. Click *Create objects*
 4. Keep default parameters > OK

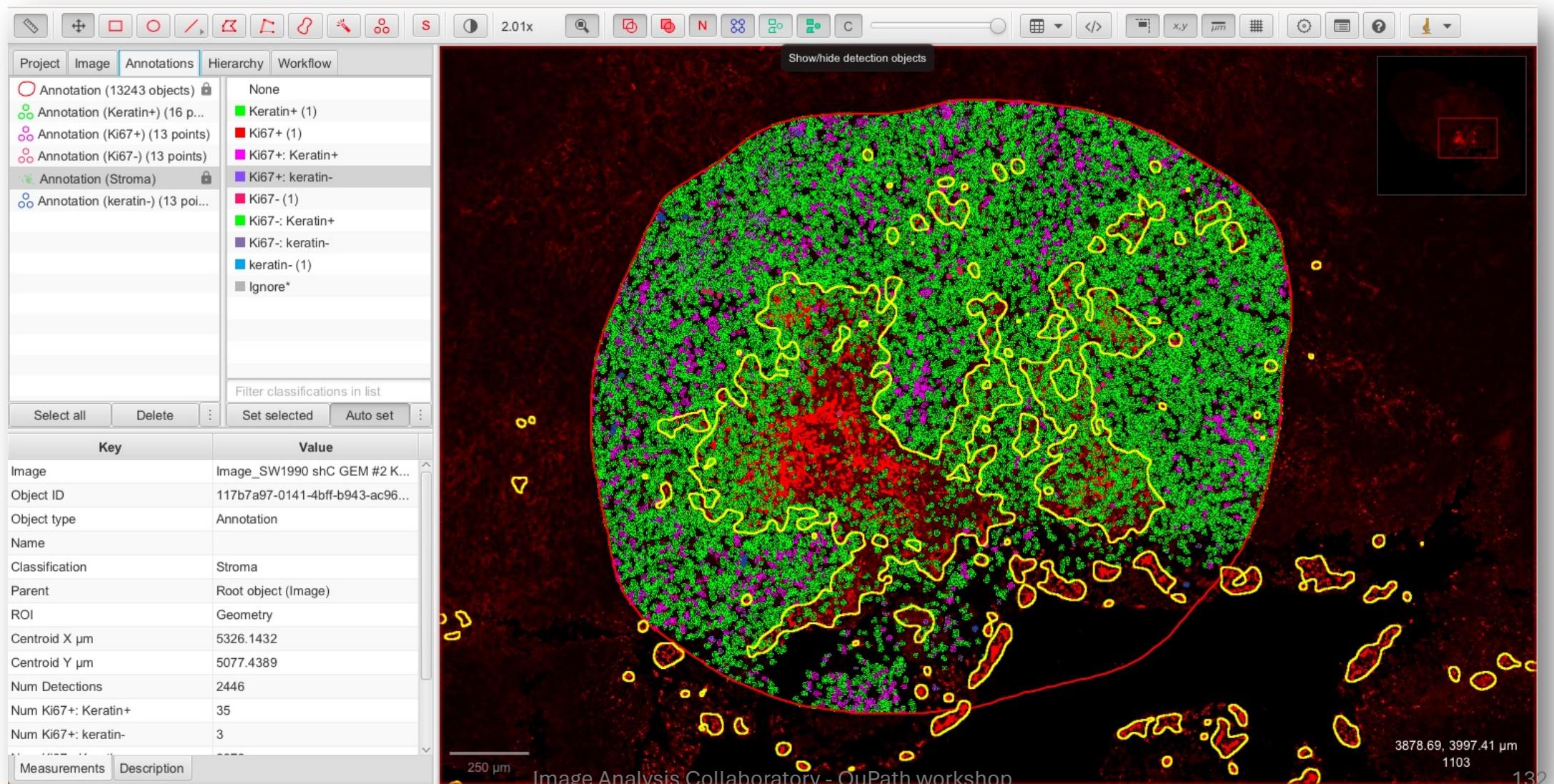


Create annotations from pixel classifier



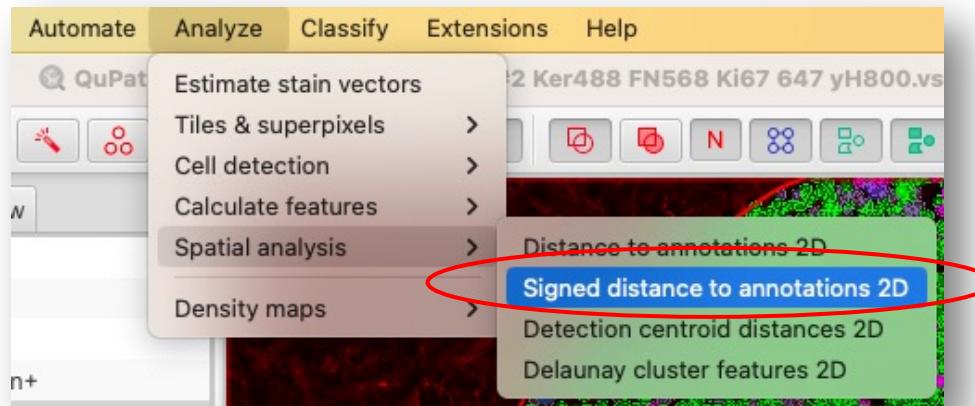
Notice the new annotation named
'Stroma' in the *Annotations* list

Fully annotated image



Spatial information: signed distance

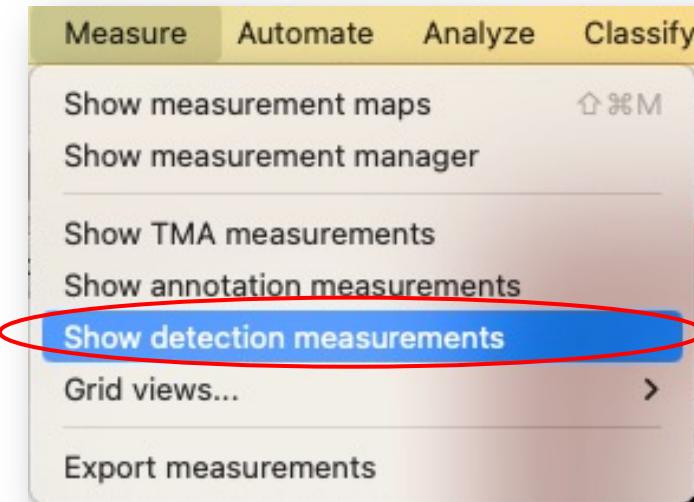
- *Analyze > Spatial analysis > Signed distance to annotations 2D*



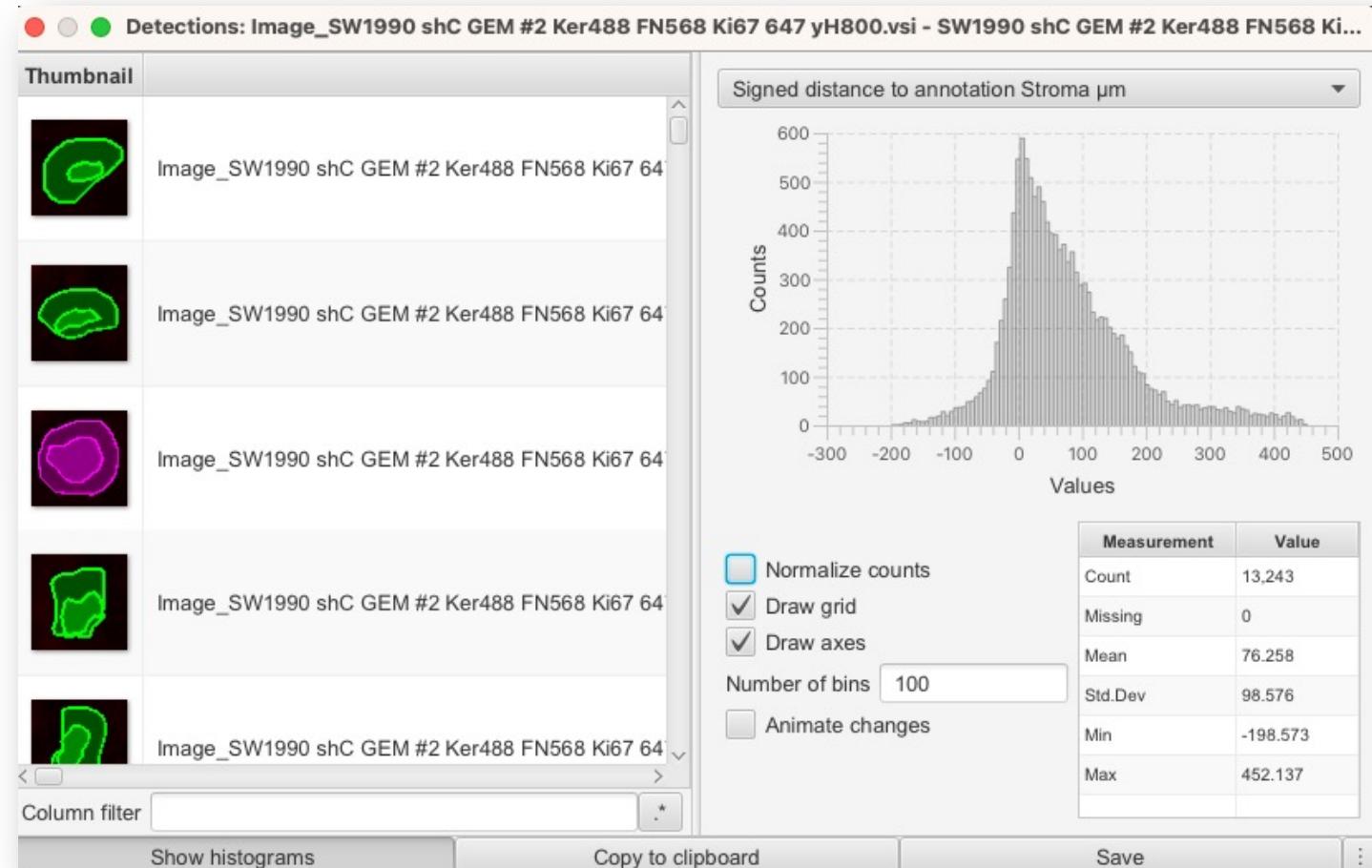
- Calculates the signed distance (2D euclidian) between cells and annotations
 - If a cell lies inside the annotation: negative distance
 - If a cell lies outside the annotation: positive distance

Spatial information: signed distance

- *Measure > Show detection measurements*



Export measurements table and use Python/R for visualization based on classes



```
String message = "Hello, Groovy!"  
int age = 25  
double pi = 3.14  
boolean isGroovy = true  
List<Integer> numbers = [1, 2, 3, 4, 5]  
Map<String, Object> person = [name: "John", age: 30, city: "New York"]
```

Scripting, workflows and batch processing

Scripting in QuPath

- QuPath uses **Groovy**, a scripting language with Java-like syntax
- Some fun facts about Groovy:
 - Created by James Strachan in 2003
 - Open-source (under the Apache License 2.0)
 - Groovy is a superset of Java and its syntax is Java-like
 - Bonus: dynamically typed (vs Java being statically typed)

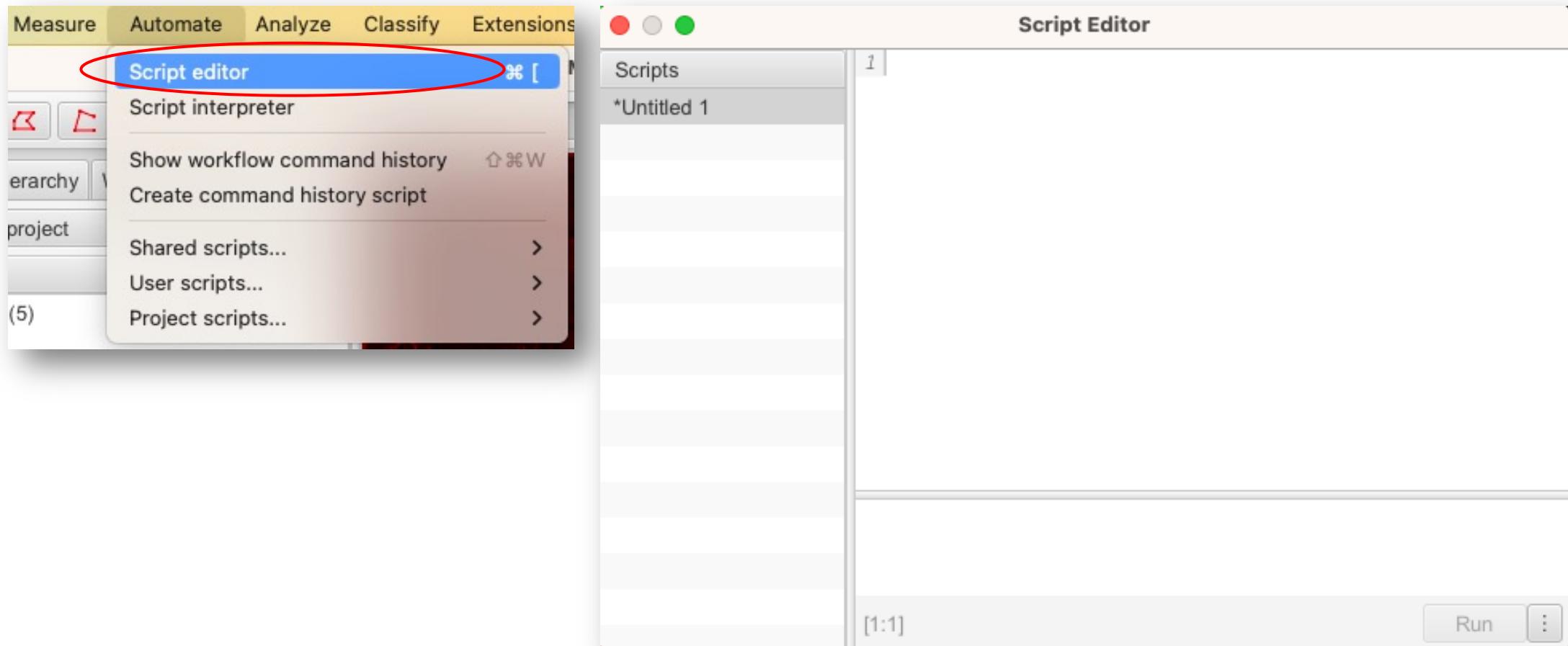
```
groovy

// Iterate over a range of numbers
for (int i = 0; i < 5; i++) {
    println("Index: $i")
}
```

for loop in Groovy

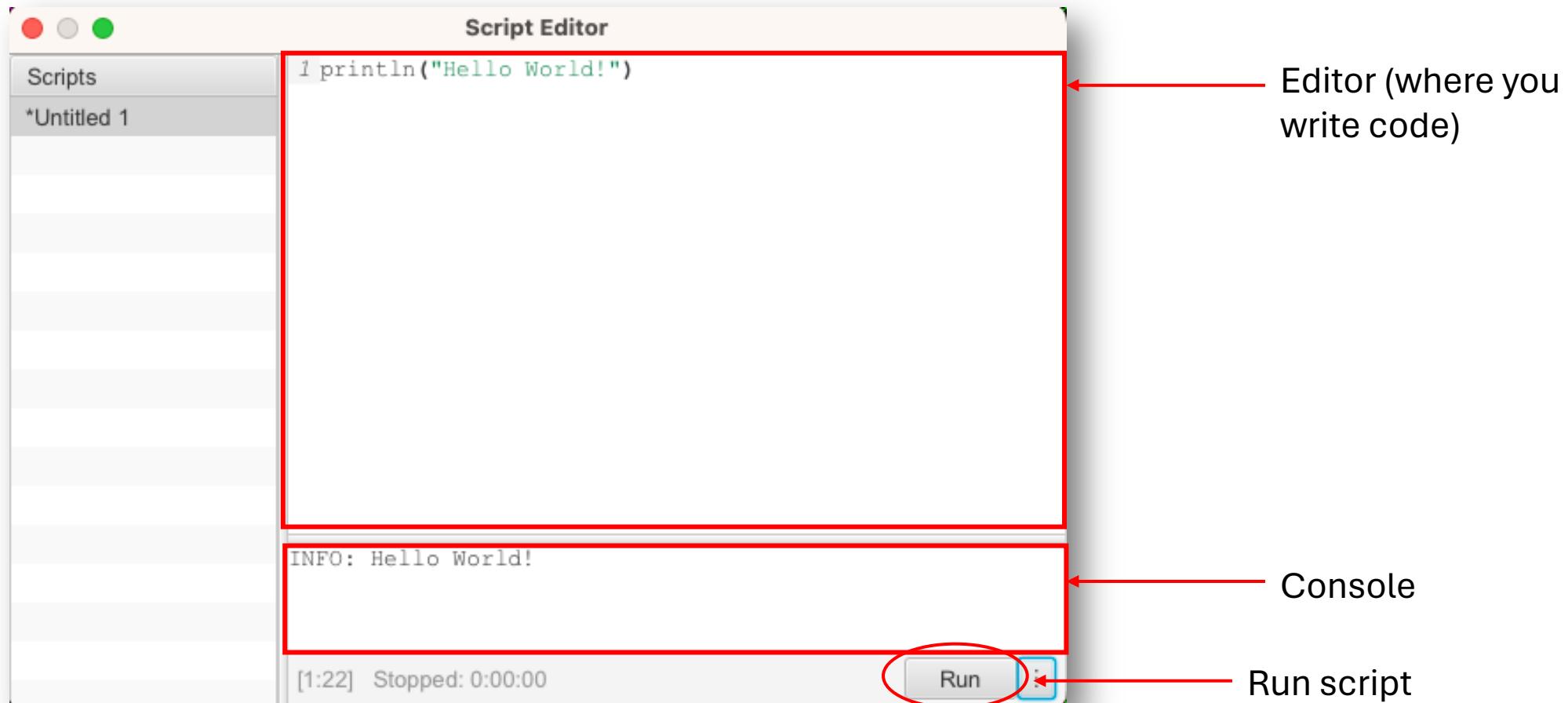
Scripting in QuPath

- *Automate > Script editor*



Hello World!

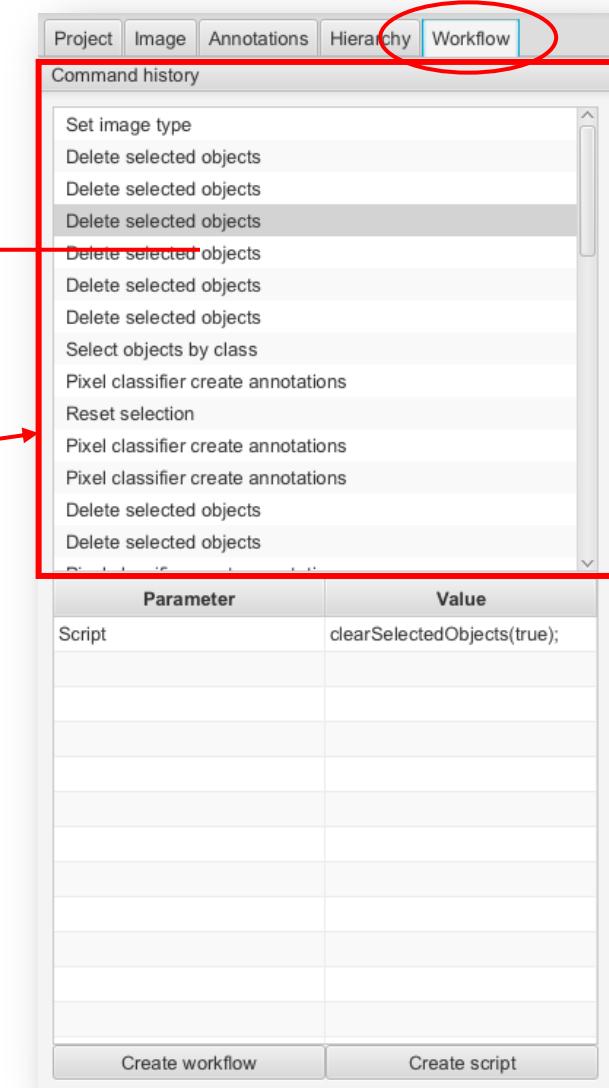
- *Automate > Script editor*



Automate your workflows without coding

QuPath uses **Workflows** to represent sequences of steps that have been applied to an image (commands run but also the parameters used).

- *Analysis panel > Workflow tab*
- The *Command history* is a record of most processing that has been done to the currently open image



Clean your workflow for cell detection and classification

1. Workflow interface showing a list of steps: Set image type, Cell detection, Run object classifier. A 'Create script' button at the bottom is circled in red.

2. Script Editor interface showing the following code:

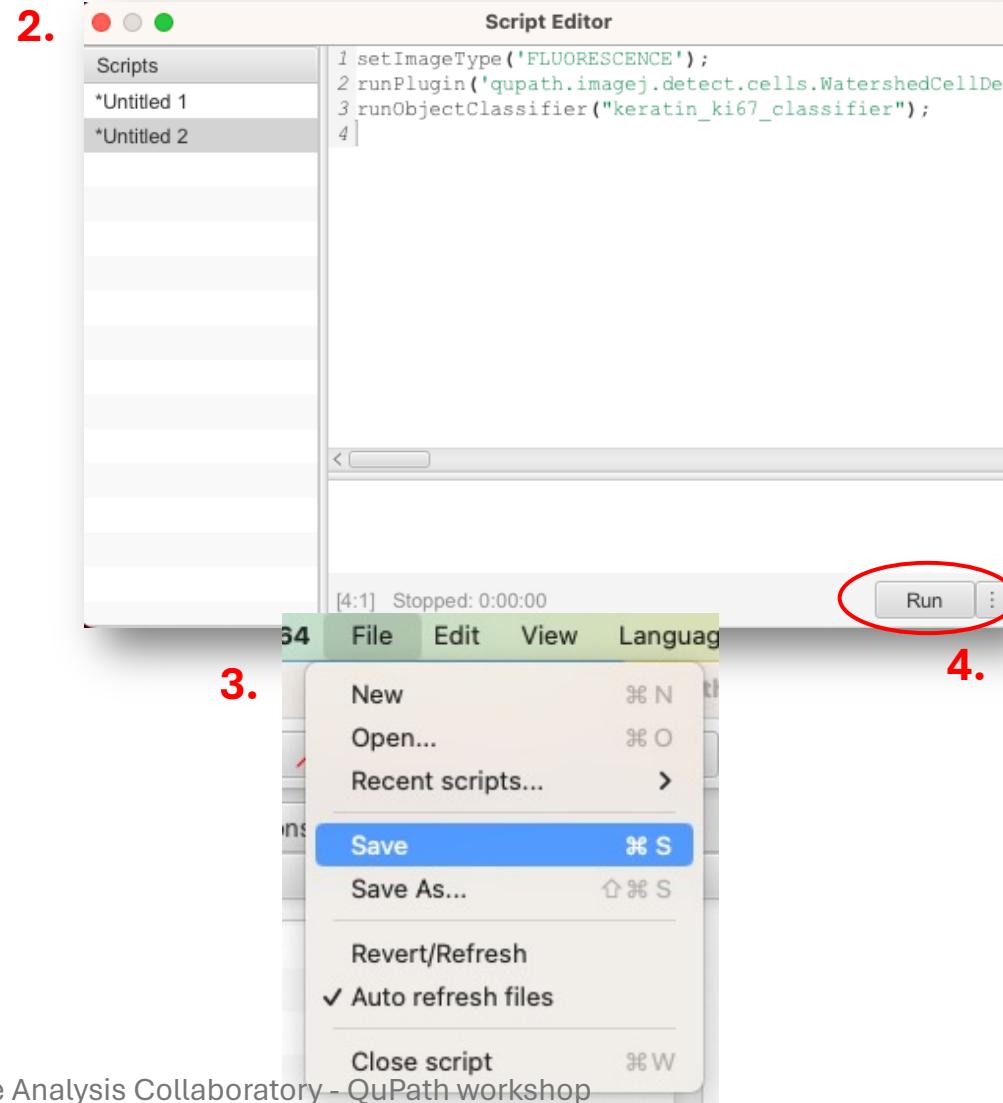
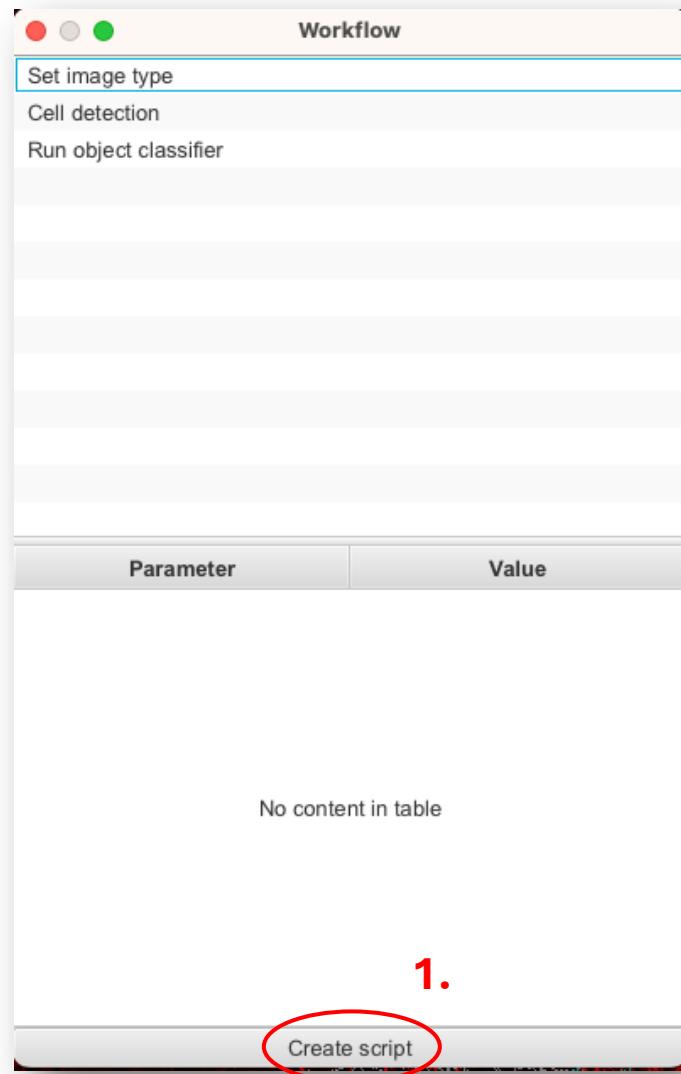
```
1 setImageType('FLUORESCENCE');
2 runPlugin('qupath.imagej.detect.cells.WatershedCellDet
3 runObjectClassifier("keratin_ki67_classifier");
4 |
```

The 'Run' button is visible at the bottom right of the editor window.

A context menu is open over the 'Cell detection' step in the workflow, listing options: Copy commands, Move up, Move down, and Remove selected items.

Edit the sequence of steps in the workflow using right-click

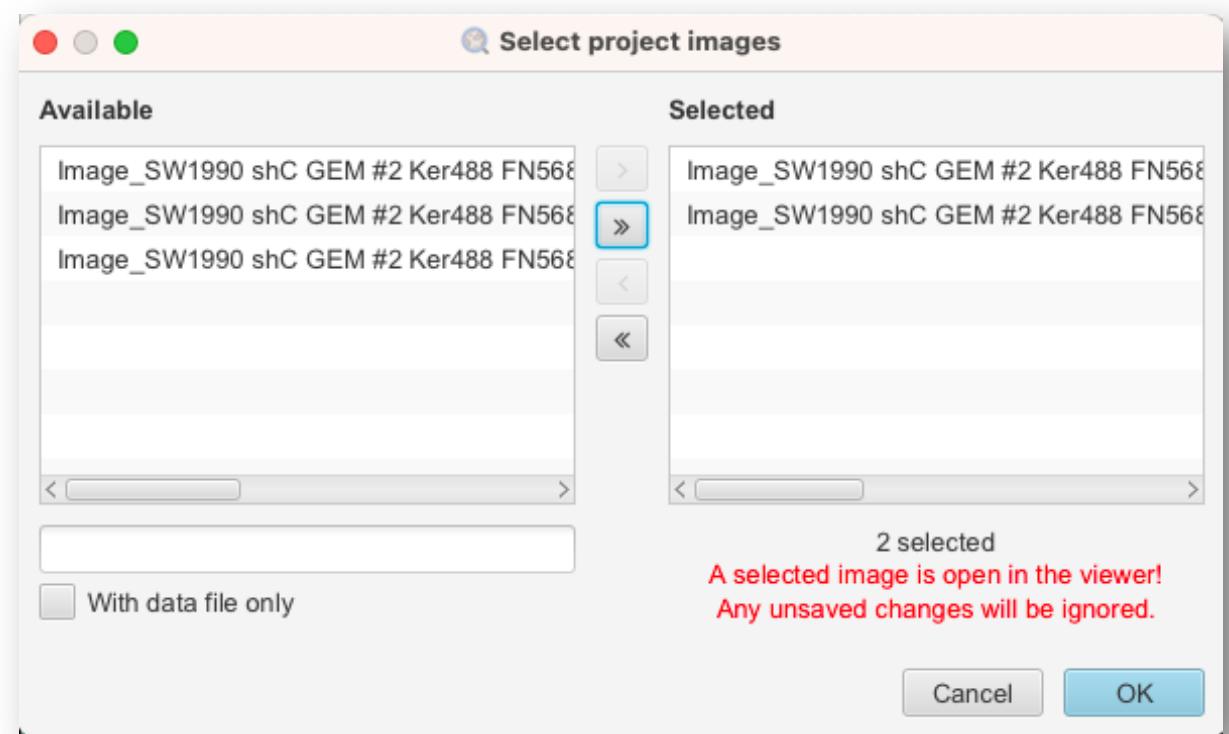
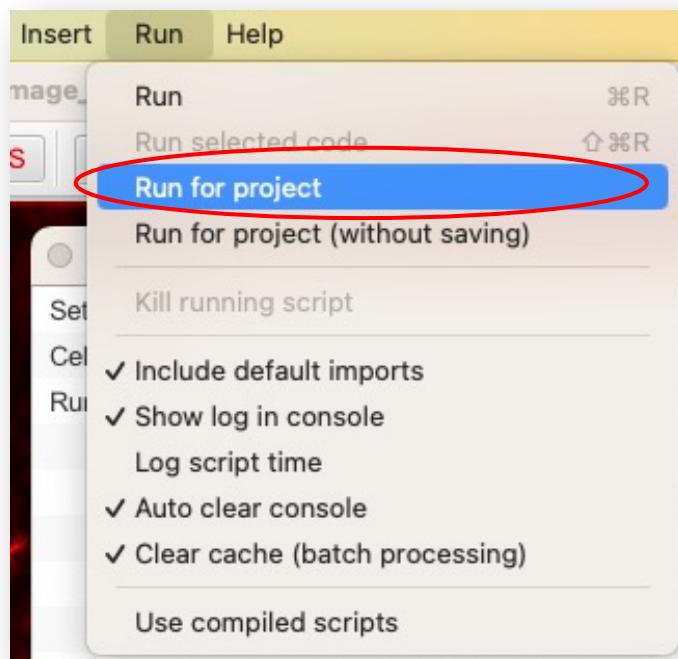
Save and run a script



Scripts can be repeated on a batch of images

QuPath allows for batch processing: scripts will run on multiple images loaded in the project.

- *Run > Run for project*



Select images you wish to run the script on.

Further resources

- QuPath documentation
 - Scripting: <https://qupath.readthedocs.io/en/latest/docs/scripting/overview.html>
 - QuPath's API docs: <https://qupath.github.io/javadoc/docs/>
- The Forum
 - Where to contact the developers of most image analysis tools
 - If you have a question, likely someone else already asked

