A microscopy image showing a tissue sample with numerous cells. The cells are outlined in red, and their nuclei are stained blue. The background is dark.

# 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

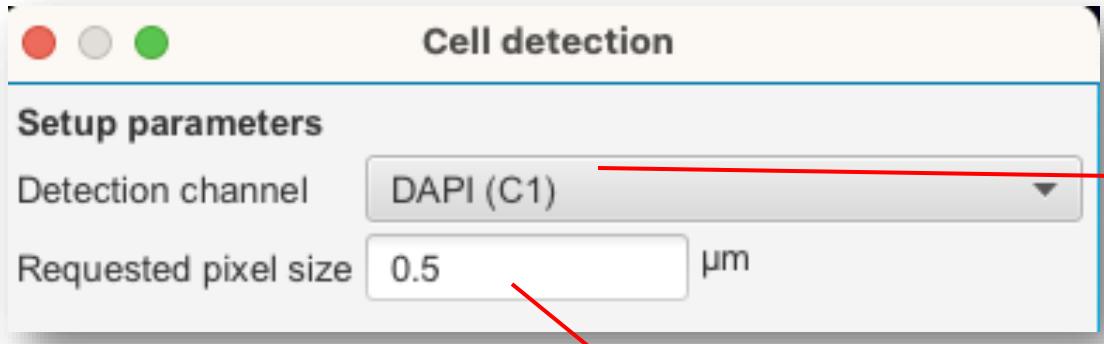
# Cell detection

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

The screenshot shows the QuPath software interface. The top menu bar includes Measure, Automate, Analyze, Classify, Extensions, and Help. The main window displays a fluorescence microscopy image with a red outline highlighting a specific region. The Analyze menu is open, showing various options: Estimate stain vectors, Tiles & superpixels, Cell detection (which is highlighted with a red circle), Calculate features, Spatial analysis, and Density maps. A sub-menu for Cell detection is also open, listing Positive cell detection, Subcellular detection (experimental), and Fast cell counts (brightfield). To the right of the image is the "Cell detection" configuration panel. It contains sections for Setup parameters (Detection channel: DAPI (C1), Requested pixel size: 0.5 μm), Nucleus parameters (Background radius: 8 μm, Use opening by reconstruction checked), Intensity parameters (Threshold: 100, Split by shape checked), Cell parameters (Cell expansion slider at 5 μm, Include cell nucleus checked), and General parameters (Smooth boundaries and Make measurements checked). A red circle highlights the "Run" button at the bottom of the panel.

Note that positive cell detection allows for multi-class segmentation on the fly

# Cell detection parameters



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

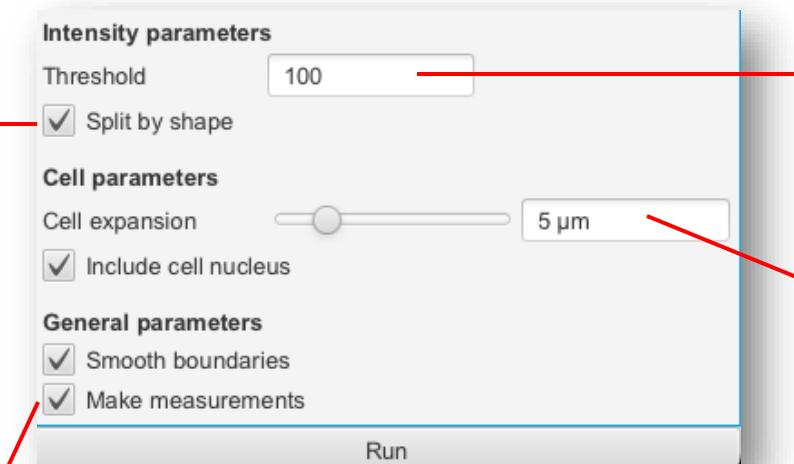
Background radius	8	µm
<input checked="" type="checkbox"/> Use opening by reconstruction		
Median filter radius	0	µm
Sigma	1.5	µm
Minimum area	10	µm <sup>2</sup>
Maximum area	400	µm <sup>2</sup>

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

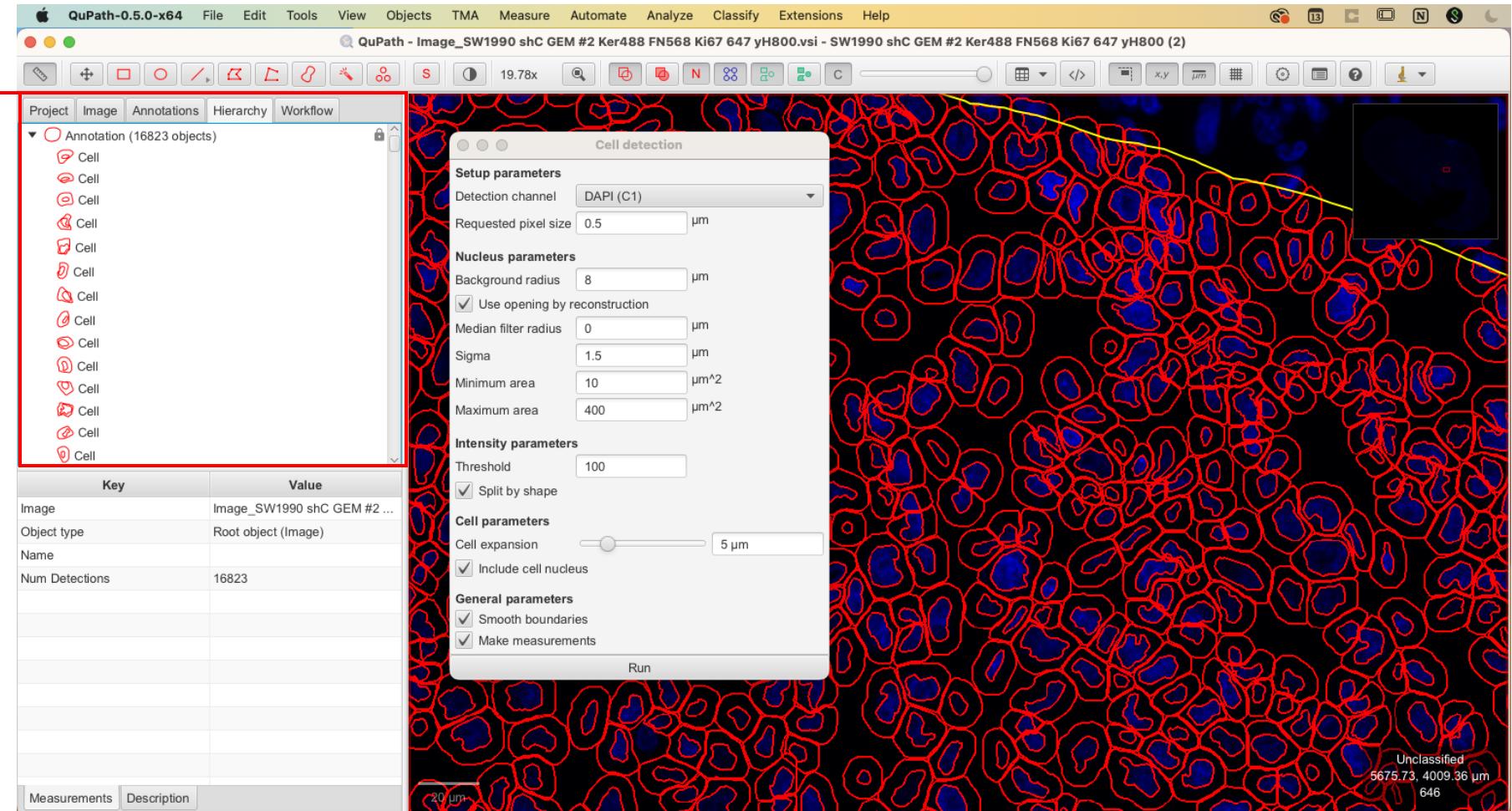
How much to expand nuclei to get cell boundaries

- 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

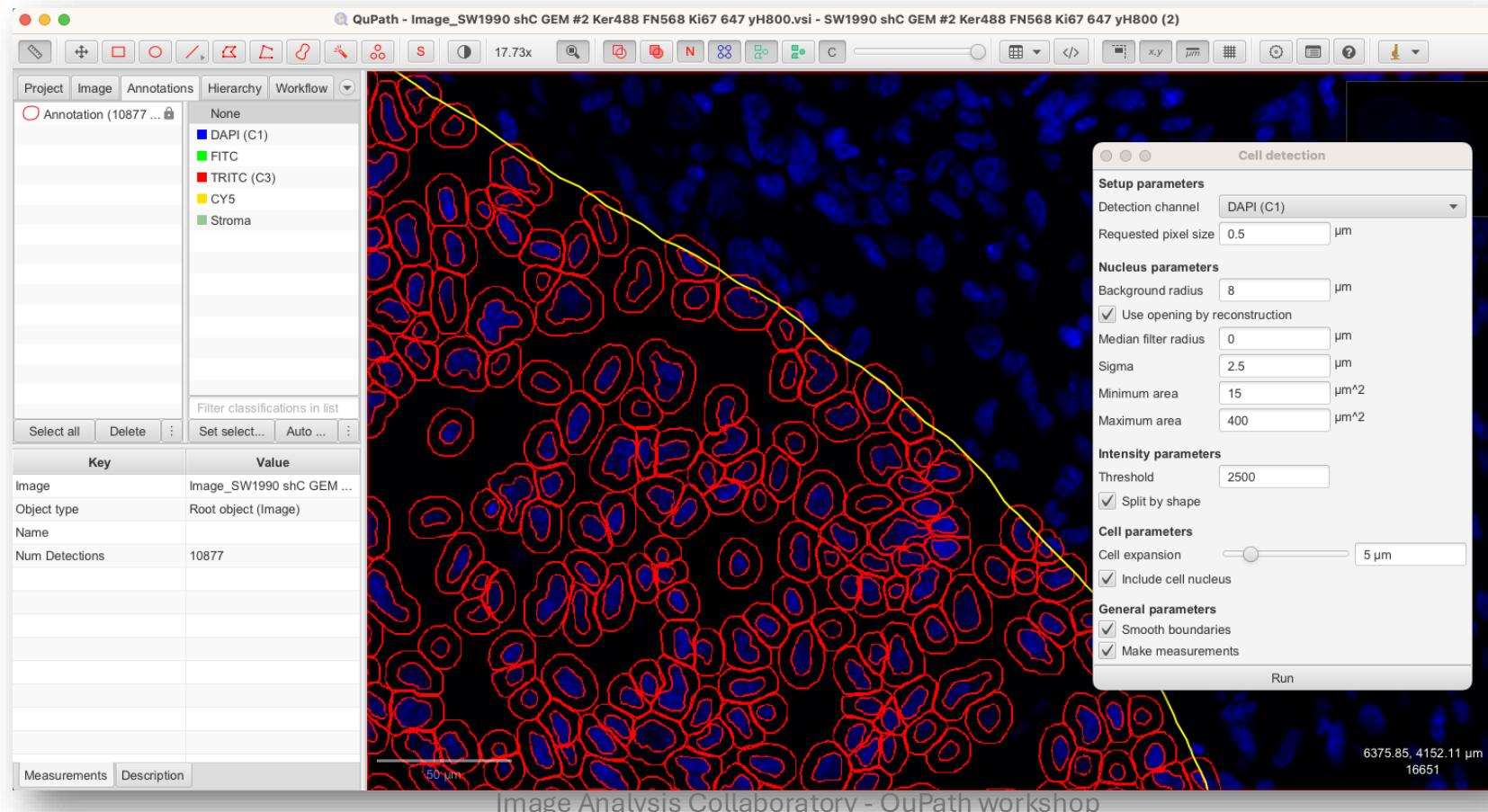
- QuPath allows to nest objects in one another to organize your projects
  - Child-parent link
  - Very useful to organize and restrict the analysis to parts of an image
  - Can be used to restrict image processing within a ROI or a detected tissue region

**Practice time**

## Exercise 3.a: QuPath cell detection

# Exercise: explore parameters

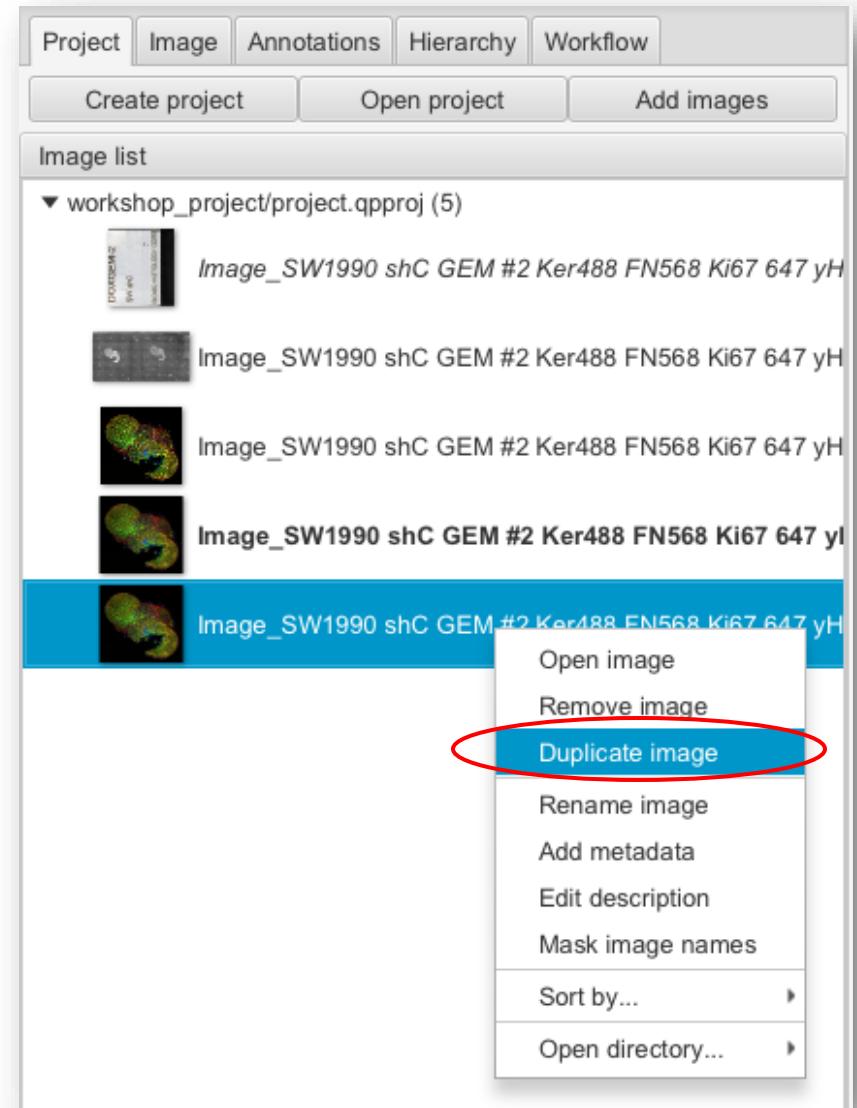
- I found that default parameters tend to over-segment nuclei so adapted the parameters to be slightly stricter (min area and threshold increased)



# Duplicate your image

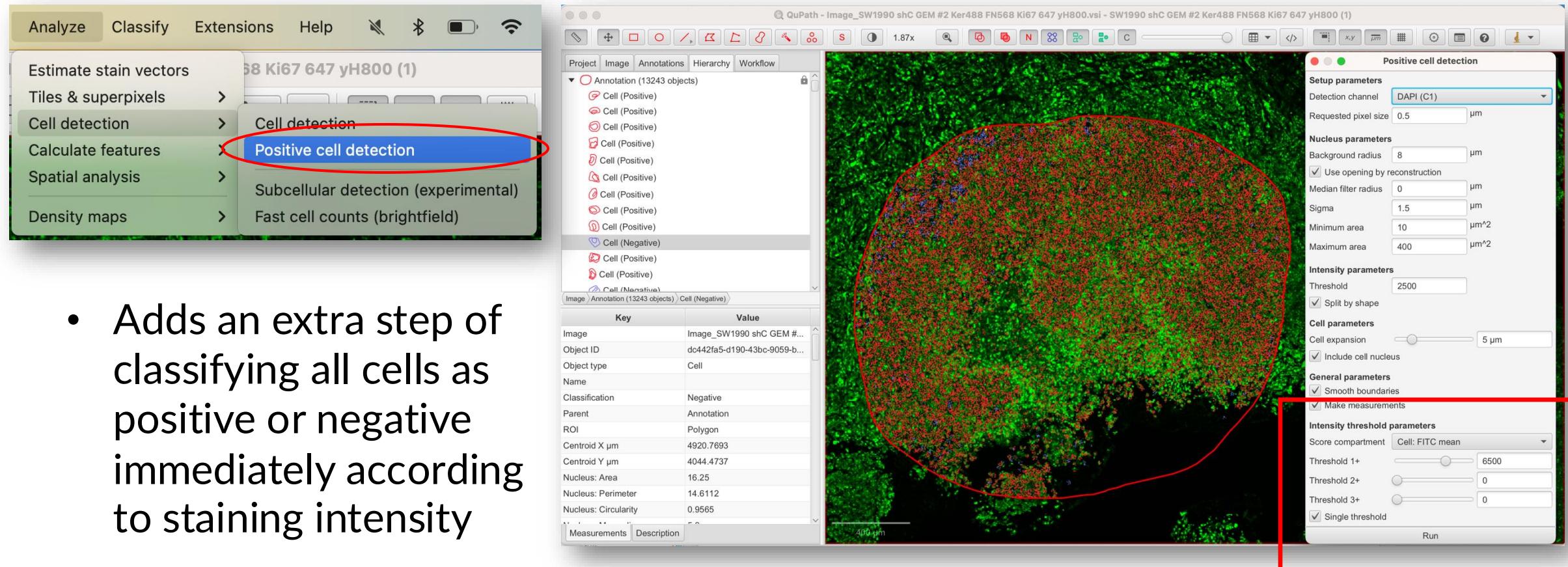
- Copy your cell detection results for future work on it
- *Project tab > Image list > Option+click or right-click on the image name > Duplicate image*

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



# Detecting cells with an extra condition

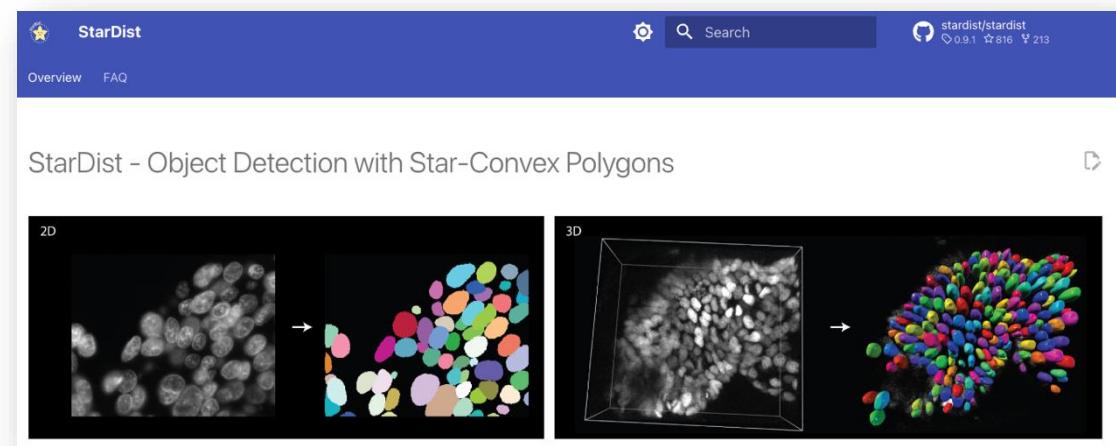
- *Analyze > Cell detection > Positive cell detection*



- Adds an extra step of classifying all cells as positive or negative immediately according to staining intensity

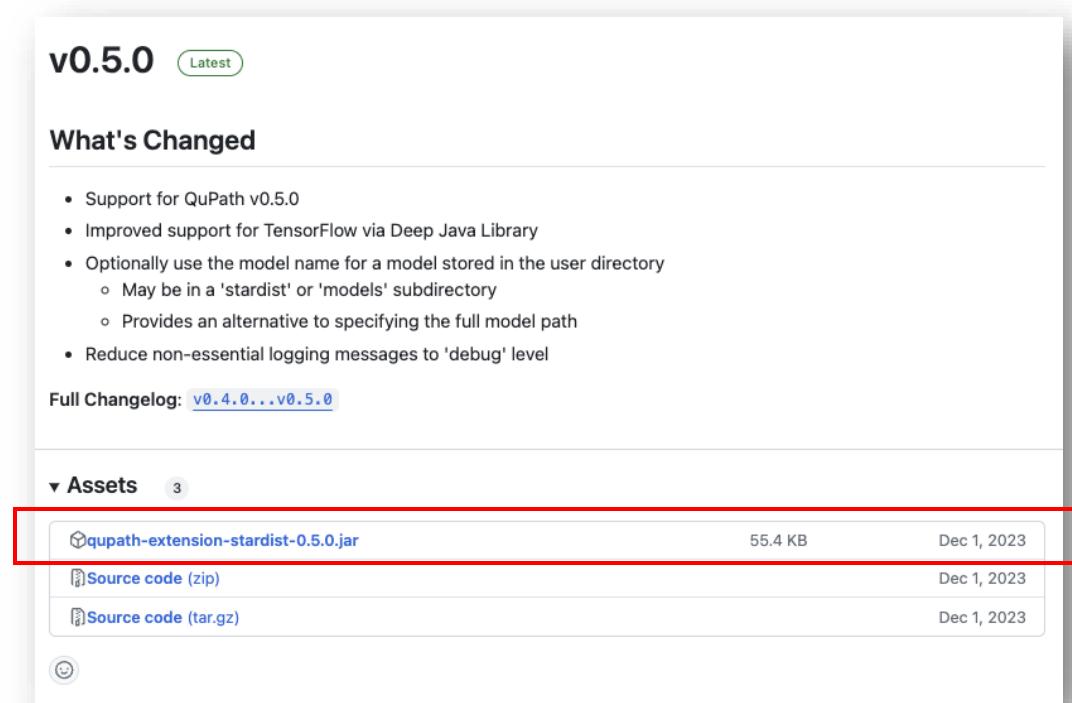
# Deep learning-based cell segmentation

- 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



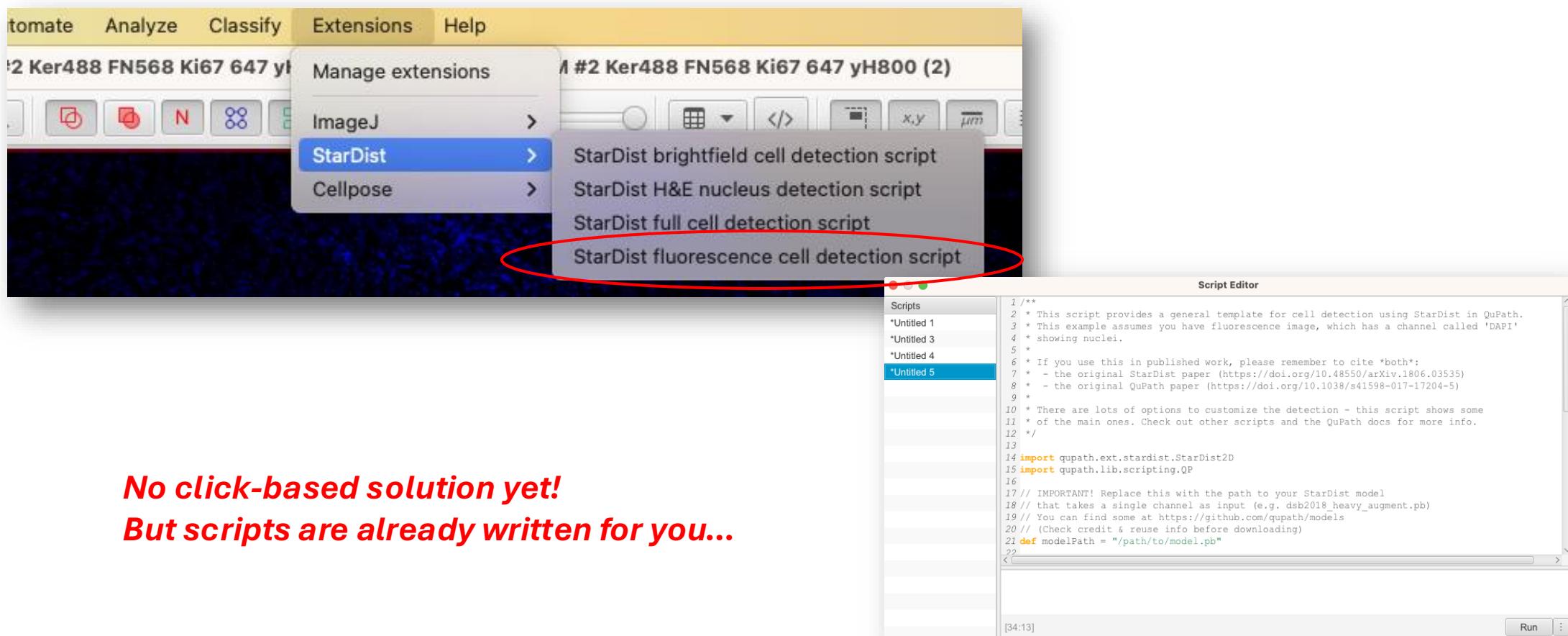
# Installing StarDist extension in QuPath

- 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!



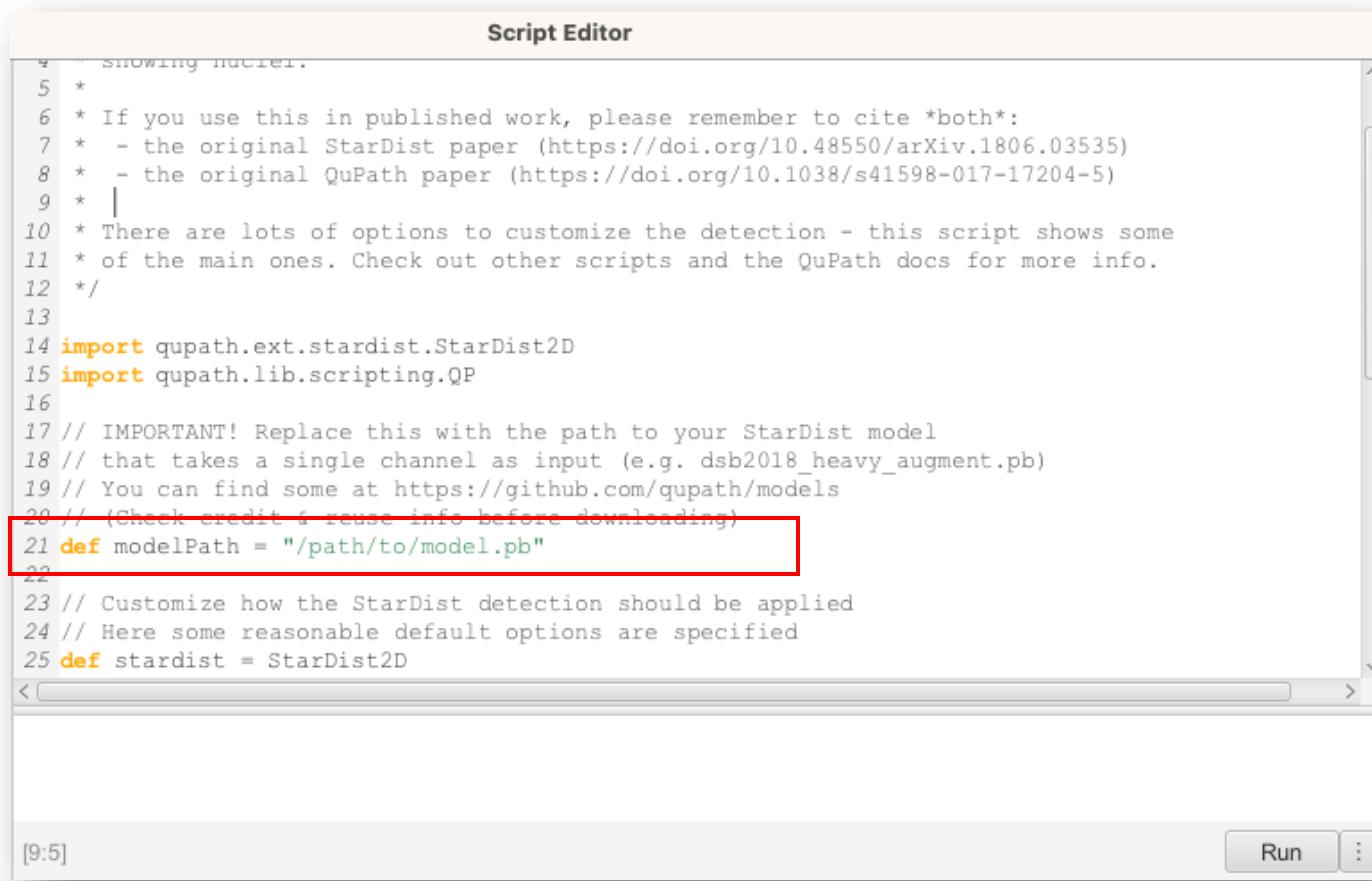
# Using StarDist extension in QuPath

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



# 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 a note about the model path. 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)**

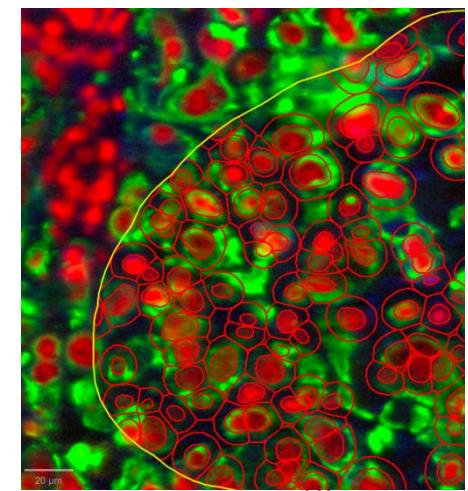
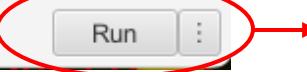
# Using StarDist extension in QuPath

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

Change the channel name

```
Script Editor
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!
[27:24] Stopped: 0:00:12
```

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

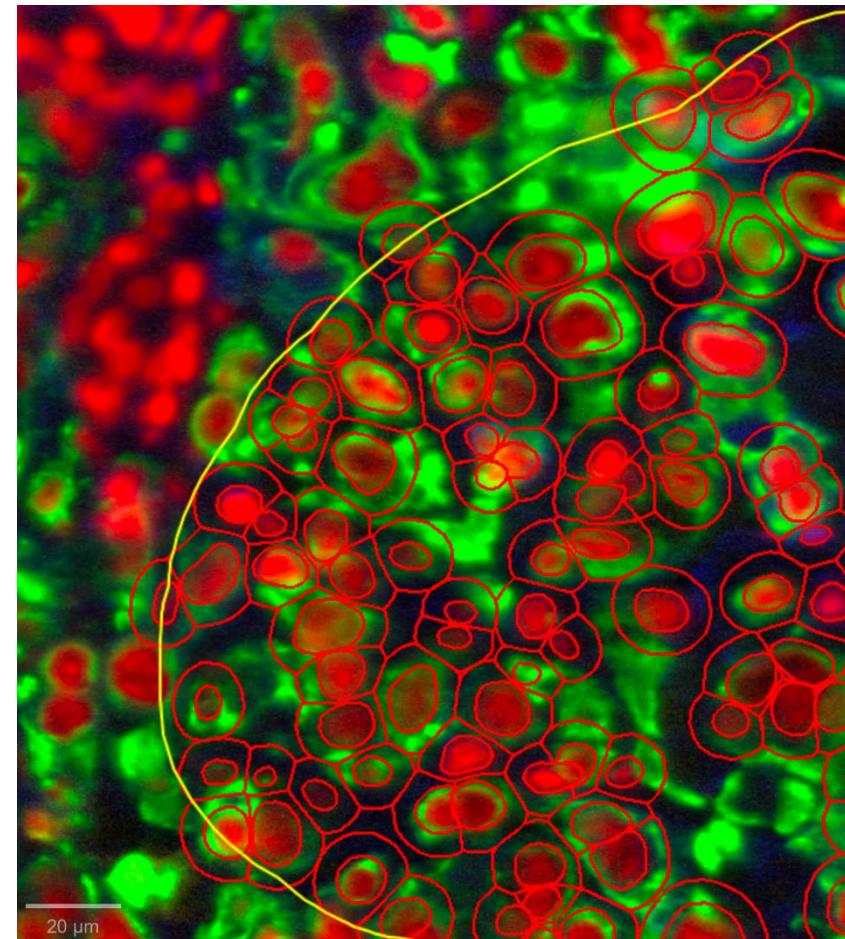


# Allow cell boundaries to bleed over the ROI

- Add `.constrainToParent(false)`

```
def stardist = StarDist2D  
    .builder(modelPath)  
    .channels('DAPI')  
    .normalizePercentiles(1, 99)  
    .threshold(0.5)  
    .pixelSize(0.5)  
    .cellExpansion(5)  
    .measureShape()  
    .measureIntensity()  
    .constrainToParent(false)||  
    .build()
```

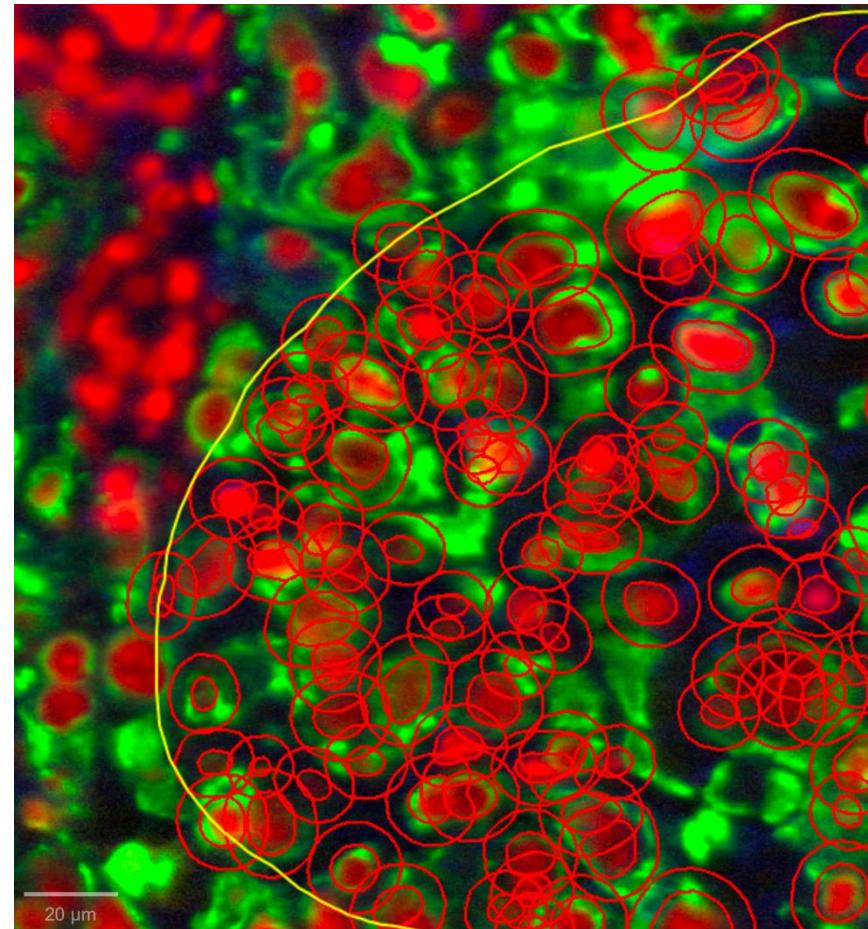
Add this  
line →



# Do not constrain cell expansion with neighbors

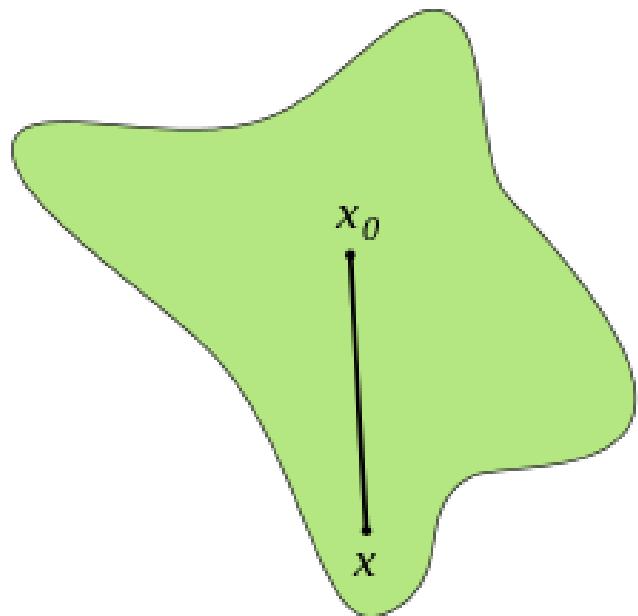
- Add `.ignoreCellOverlaps(true)`

```
def stardist = StarDist2D  
    .builder(modelPath)  
    .channels('DAPI')  
    .normalizePercentiles(1, 99)  
    .threshold(0.5)  
    .pixelSize(0.5)  
    .cellExpansion(5)  
    .measureShape()  
    .measureIntensity()  
Add this line → .ignoreCellOverlaps(true)  
    .build()
```

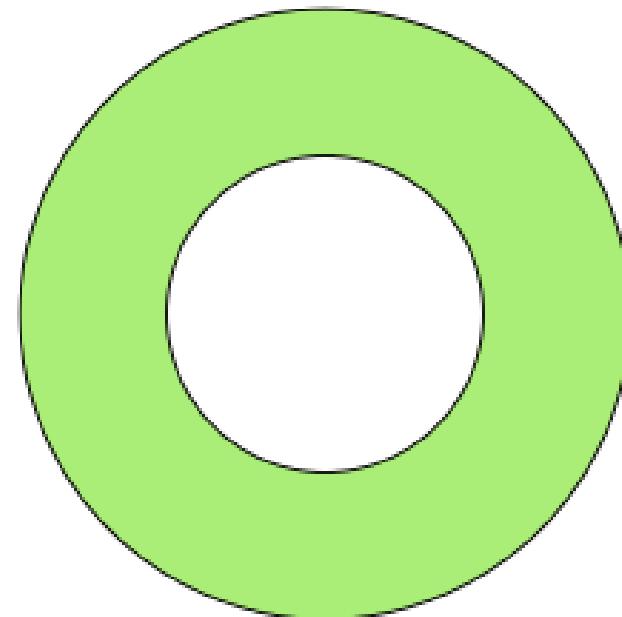


# Exercise 3.b: QuPath cell detection with StarDist

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



StarDist can segment



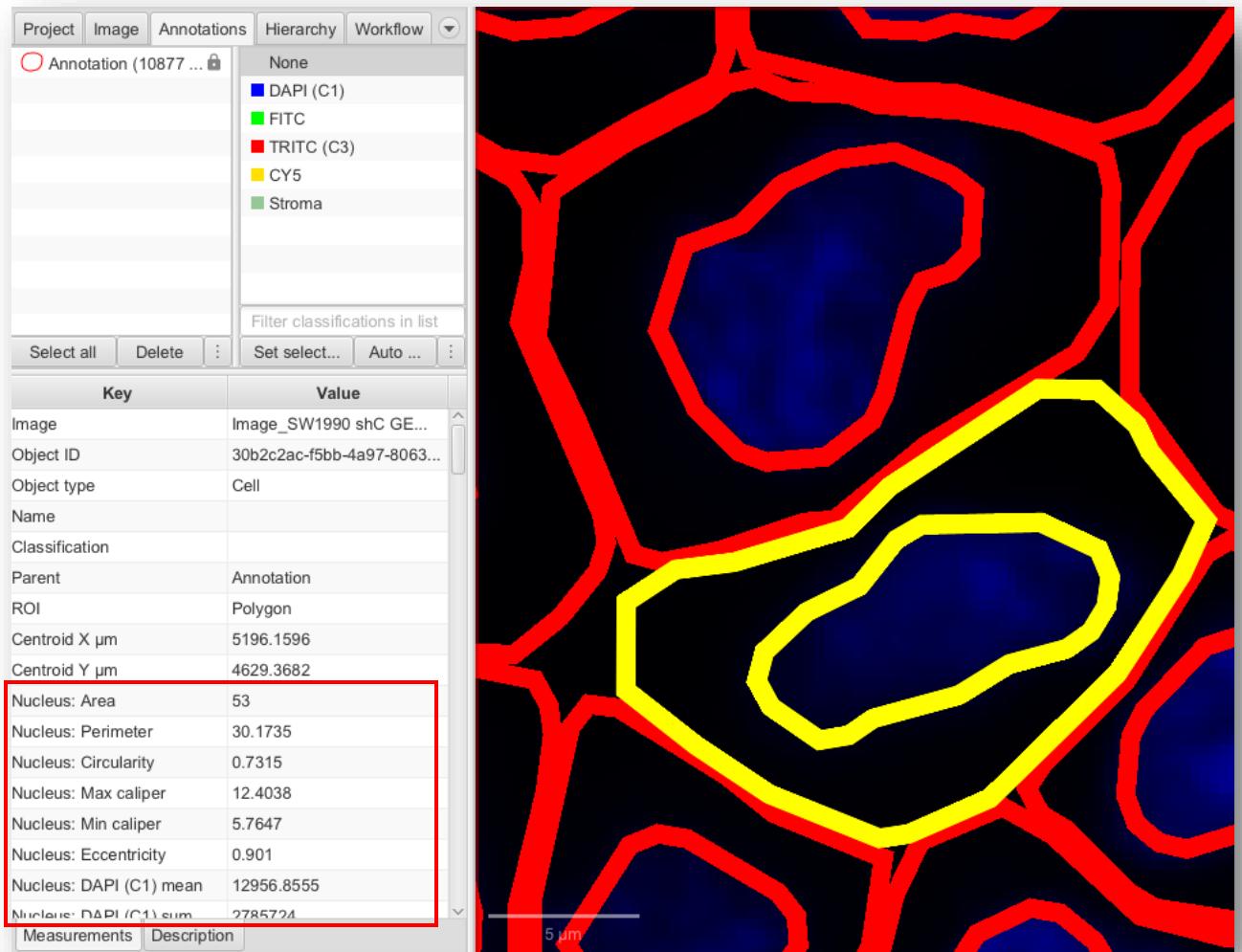
StarDist can **not** segment

# Cell detection measurements

# Detection measurements

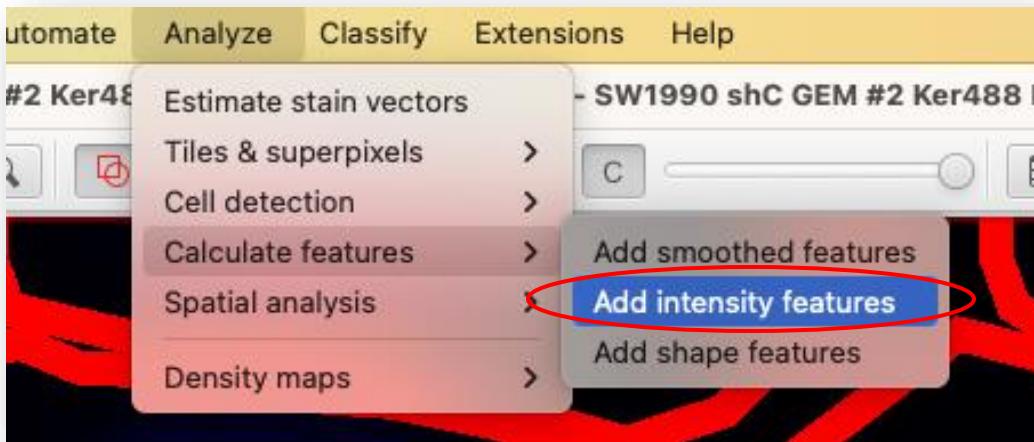
- Each detection object (i.e. a cell) has its measurement list
  - Intensity features
  - Haralick (texture) features
  - Shape features
  - Smoothed features
- *Annotations* tab > select a cell in the viewer > inspect its measurements list

By default, basic intensity and shape features are calculated



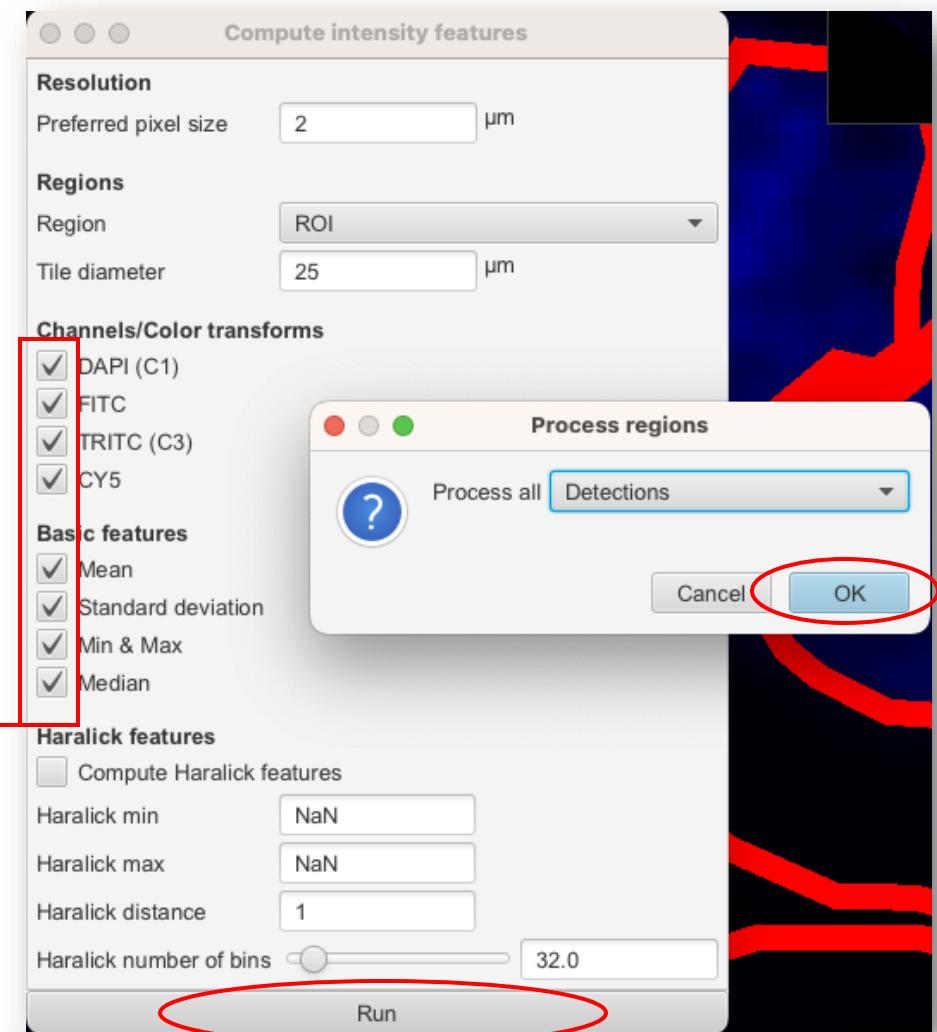
# Calculating measurements

- *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

Objects TMA Measure Automate

Path - Image\_SW1990.vsi - SW1990

Show measurement maps  
Show measurement manager  
Show TMA measurements  
Show annotation measurements  
**Show detection measurements** (highlighted)  
Grid views...  
Export measurements

**Columns: measurements** (highlighted)

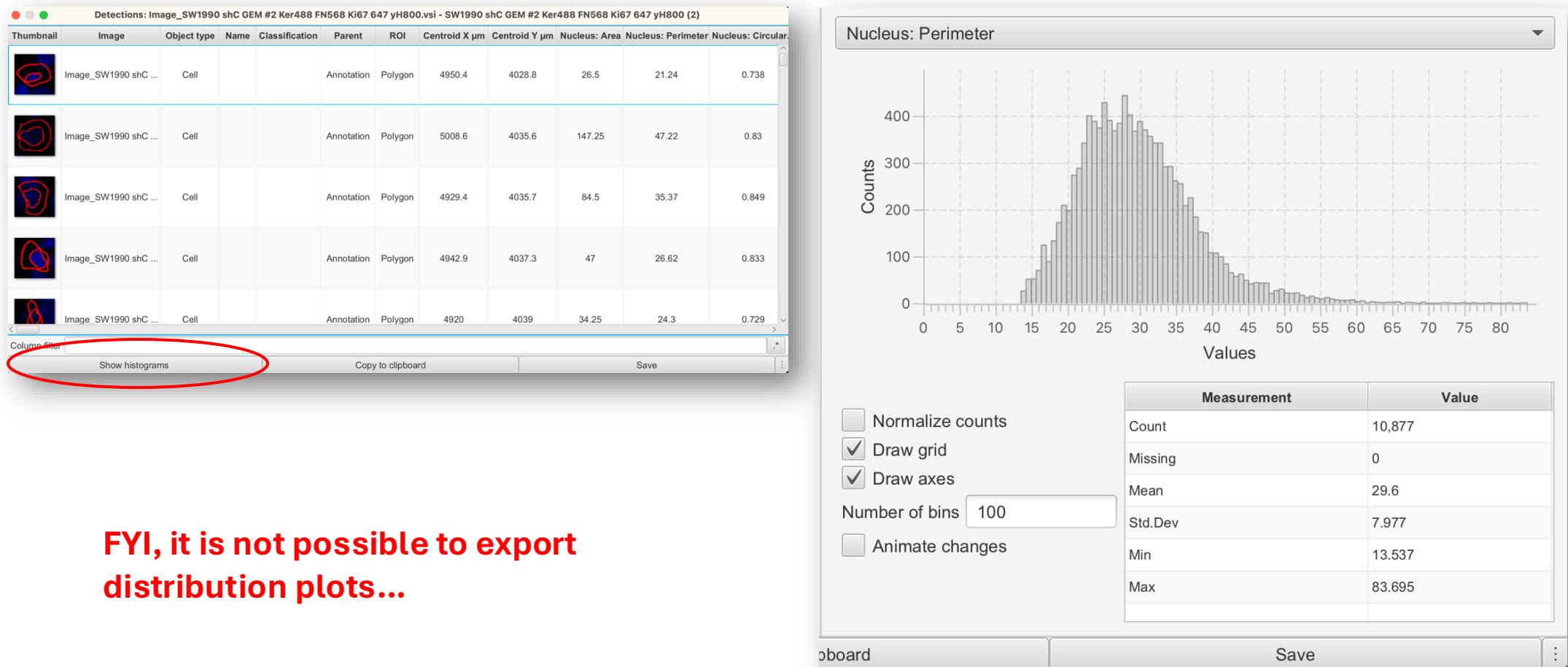
**Rows: cells**

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

Show histograms Copy to clipboard Save

# Visualizing measurement distributions

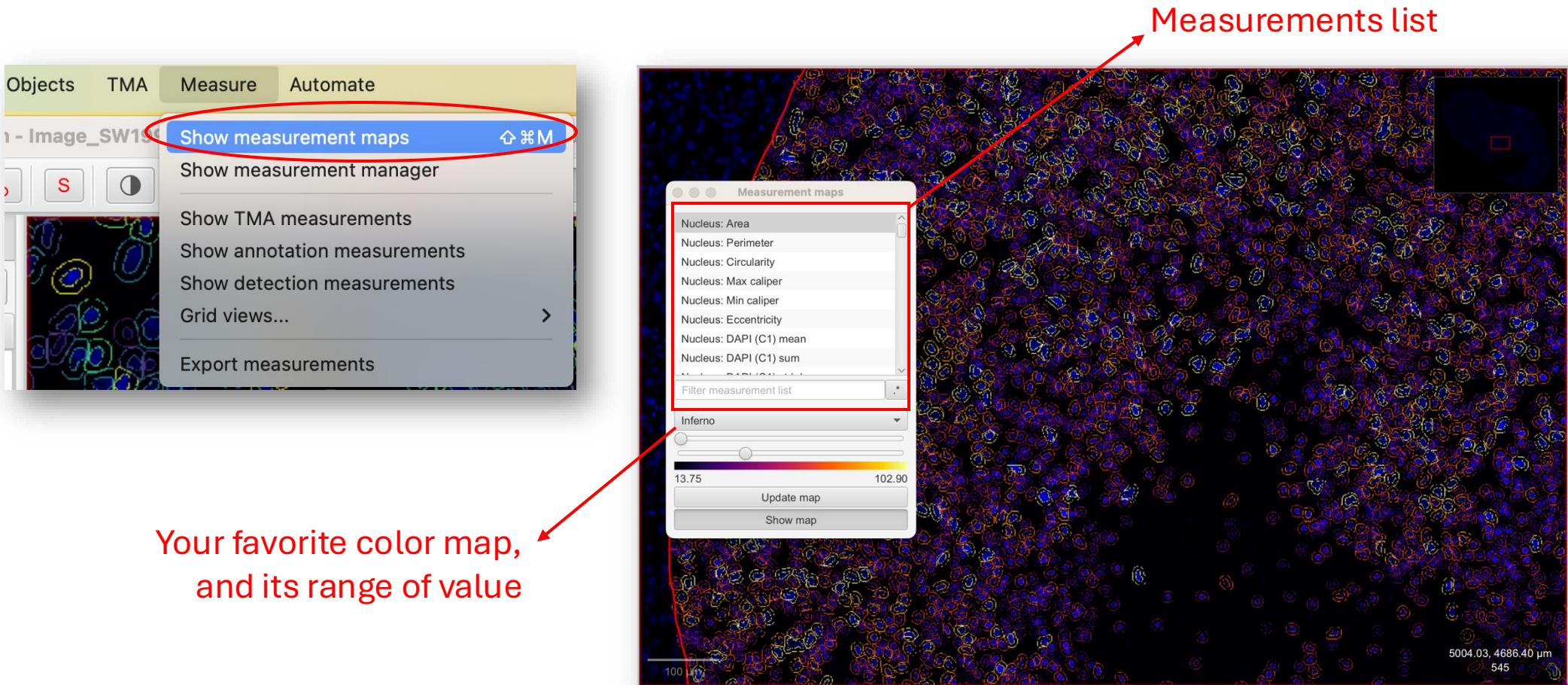
- *Measure > Show detection measurements*



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

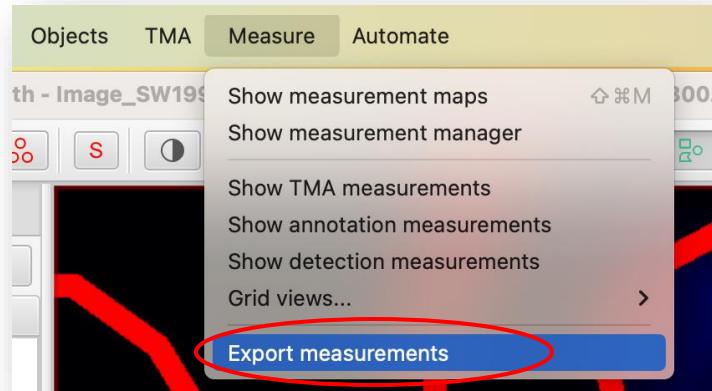
# Visualizing measurements as heat maps

- *Measure > Show measurement maps*



# Export measurements the right way

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



Output file location

Measurement type to be exported

File type (.tsv, .csv)

List of measurements to include in the export

