ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Image segmentation

Robert Haase

Using materials from Marcelo Leomil Zoccoler and Johannes Soltwedel, PoL, TU Dresden

GEFÖRDERT VOM

Bundesministerium für Bildung und Forschung

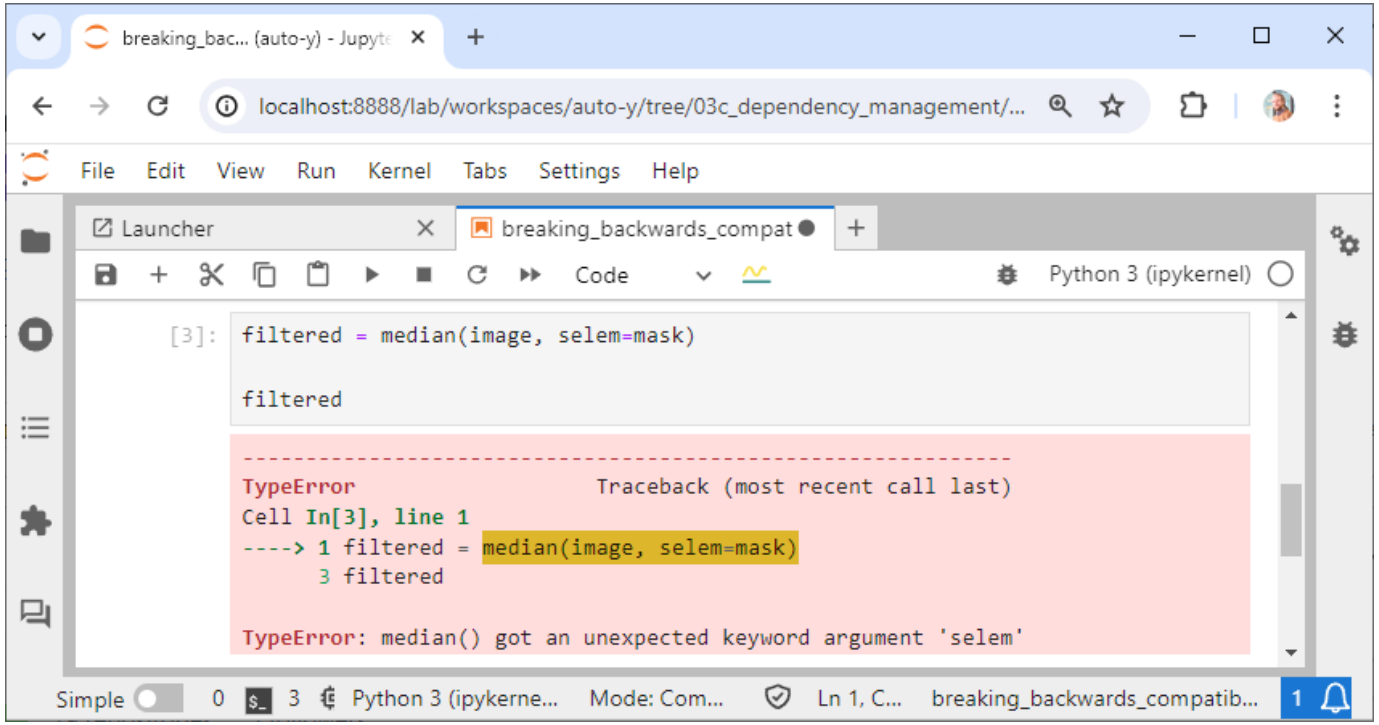
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Recap

How can one solve this problem?

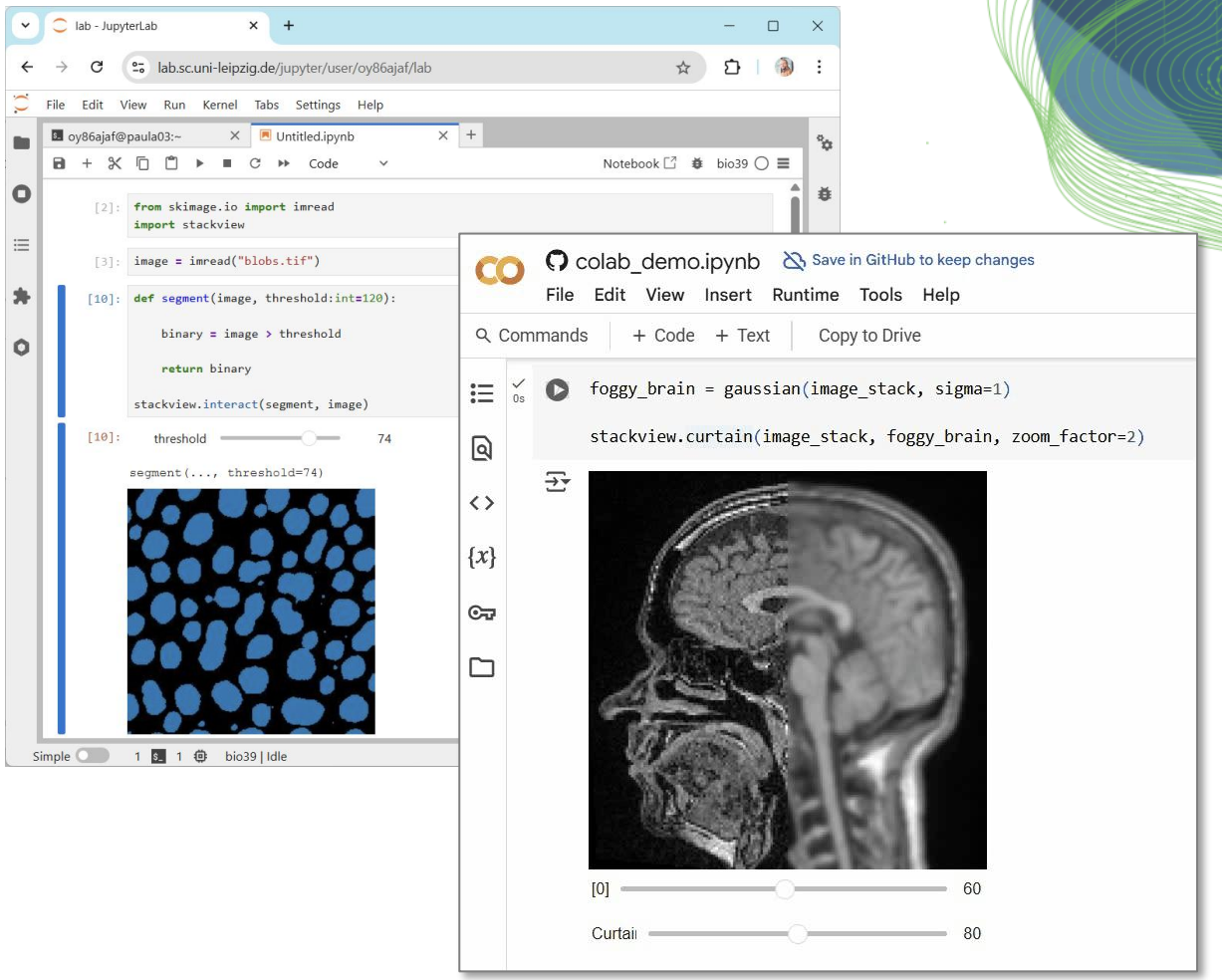
A) By modifying the code

B) By not modifying the code



Recap

- The stackview installation on the Jupyter Hub was broken. The SC/IT had to install ipywidgets in the GUI.
- For google colab there is a new stackview 0.16.0 release



- Both environments *should* work now.

Quiz (recap)

- Which of the following is a band-pass filter?

Gaussian

Median

Top-hat

Difference of Gaussian

Quiz (recap)

- Which of the following is a denoising filter?

Gaussian

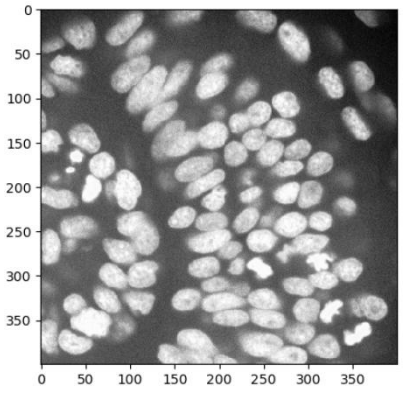
Median

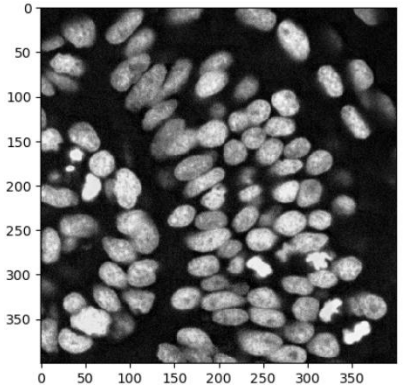
Top-hat

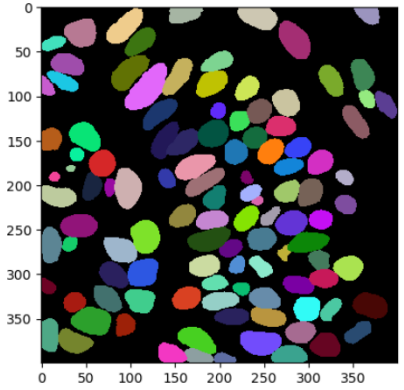
Difference of Gaussian

Lecture overview: Bio-image Analysis

- Image Data Analysis workflows
- Goal: **Quantify observations, substantiate conclusions with numbers**

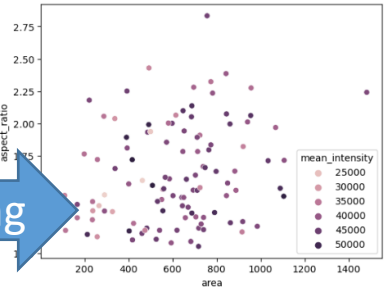






	area	mean_intensity	major_axis_length	minor_axis_length
0	594.0	40572.809764	28.611591	26.537947
1	645.0	43764.872868	33.511511	24.566916
2	1105.0	51970.561991	45.232031	31.456308
3	718.0	47015.487465	31.023274	29.5
4	791.0	49132.515803	36.382253	27.7
...	...	...	...	
105	238.0	30477.126050	20.252197	15.2
106	615.0	32886.154472	41.030760	19.8
107	110.0	33042.445455	14.366347	9.9
108	222.0	43304.180180	25.370549	11.6
109	167.0	43378.808383		

Plotting



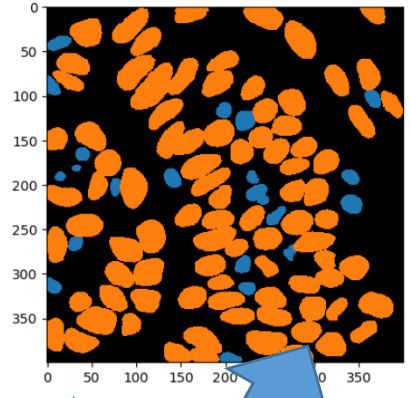


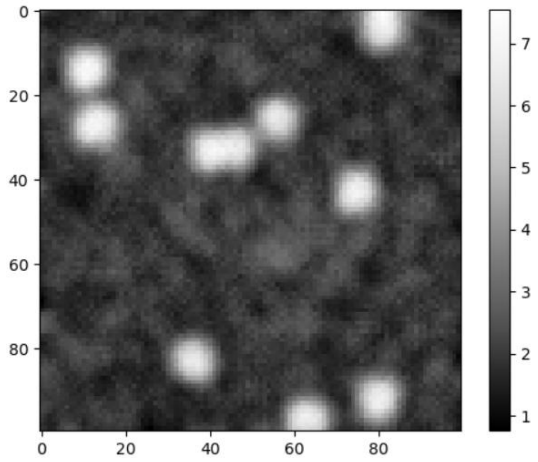
Image filtering → Image segmentation → Feature extraction → Object classification → Visualization

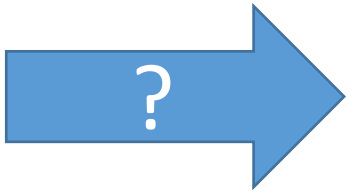
Segmentation: Binarization

Binarization, e.g. via thresholding

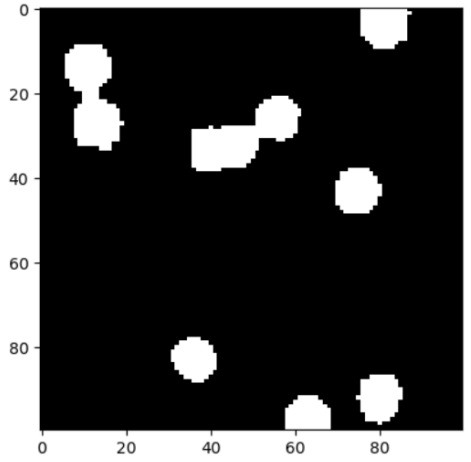
- Very basic and yet efficient segmentation technique
- Histogram based, to determine an intensity threshold above which pixels become white
- Not state-of-the-art in many fields anymore

Intensity image





Binary image

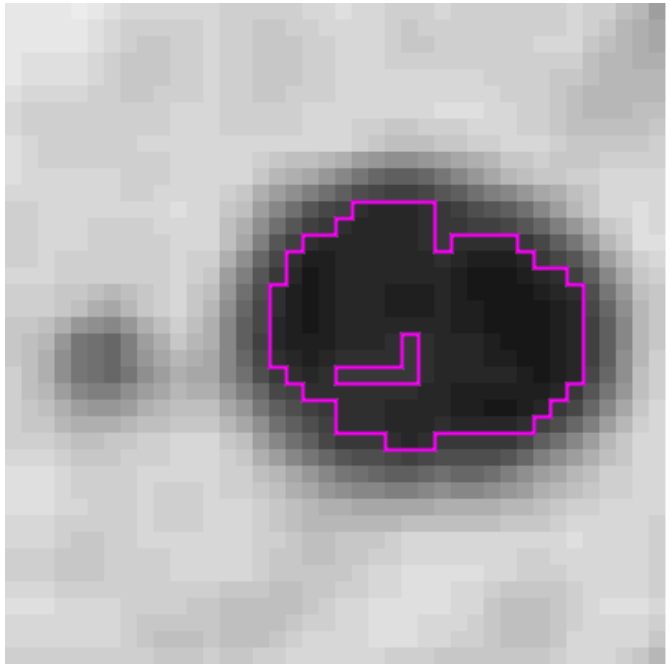


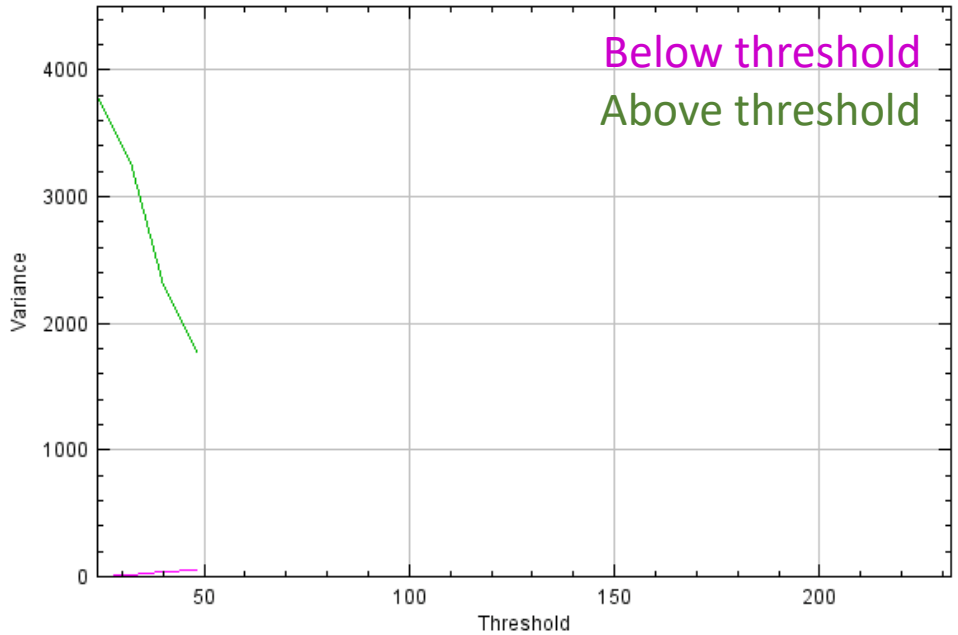
Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.

$$Var(I) = \sum_{i \in I} g_i - \bar{g}_I \qquad \bar{g}_I = \sum_{i \in I} \frac{g_i}{n_I}$$

$Var(I)$ … Variance in image I
g_i … grey value of a pixel i
$\bar{g}_I$ … mean grey value of the whole image I
n_I … number of pixels in Image I



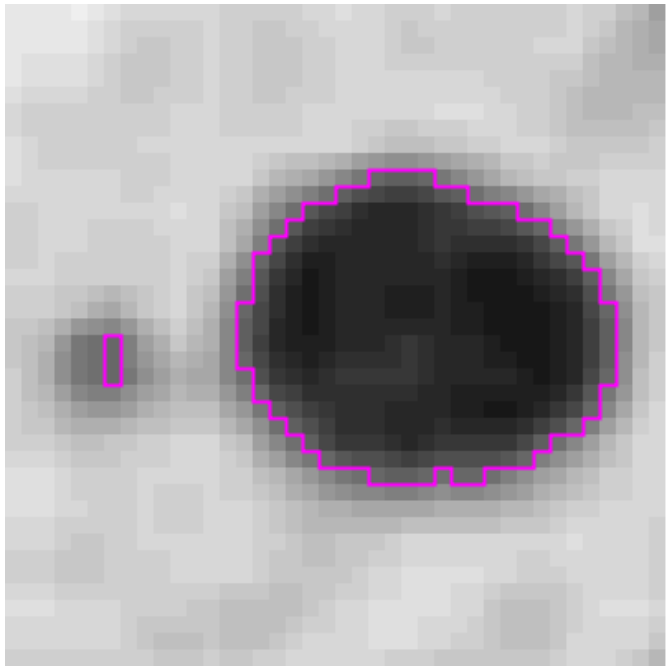


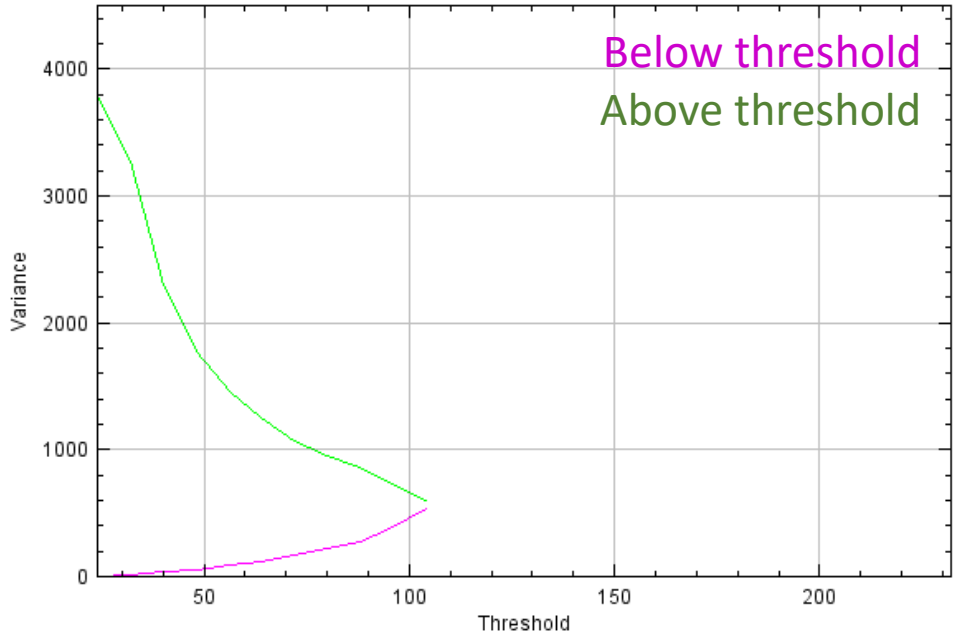
Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.

$$Var(I) = \sum_{i \in I} g_i - \bar{g}_I \qquad \bar{g}_I = \sum_{i \in I} \frac{g_i}{n_I}$$

$Var(I)$ … Variance in image I
g_i … grey value of a pixel i
$\bar{g}_I$ … mean grey value of the whole image I
n_I … number of pixels in Image I



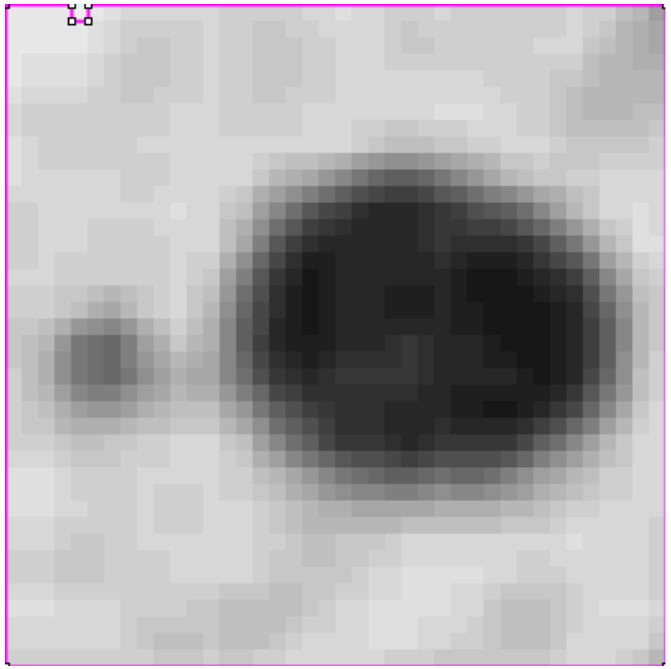


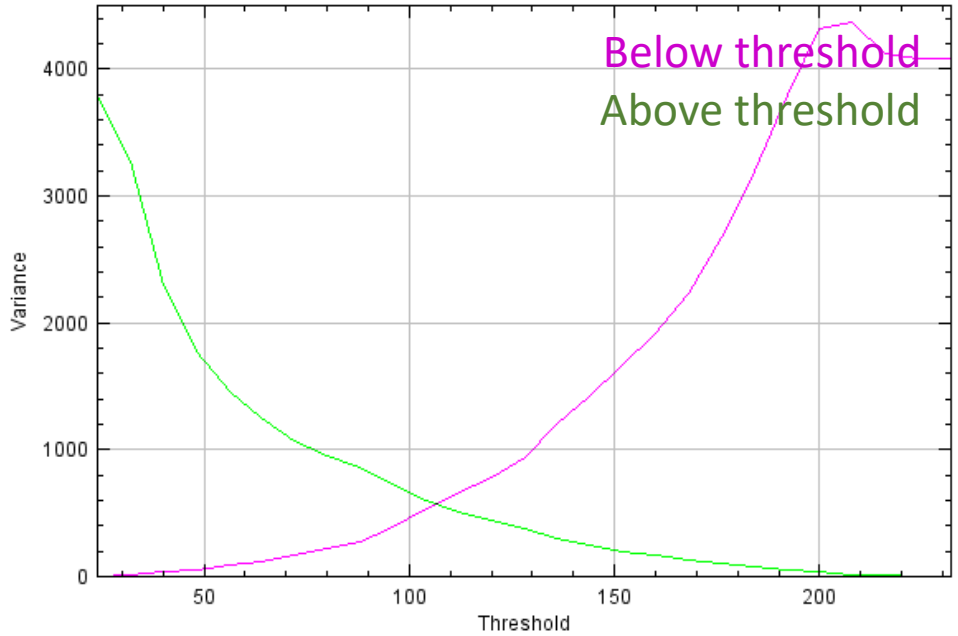
Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.

$$Var(I) = \sum_{i \in I} g_i - \bar{g}_I \qquad \bar{g}_I = \sum_{i \in I} \frac{g_i}{n_I}$$

$Var(I)$ … Variance in image I
g_i … grey value of a pixel i
$\bar{g}_I$ … mean grey value of the whole image I
n_I … number of pixels in Image I

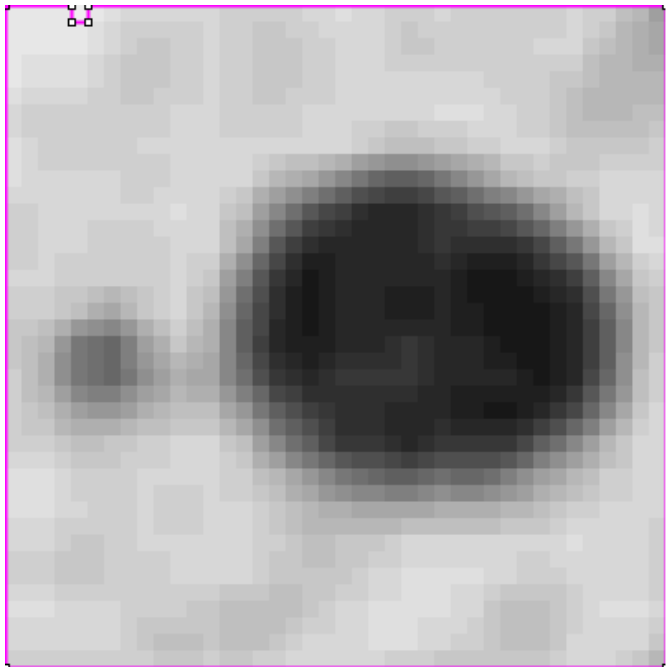


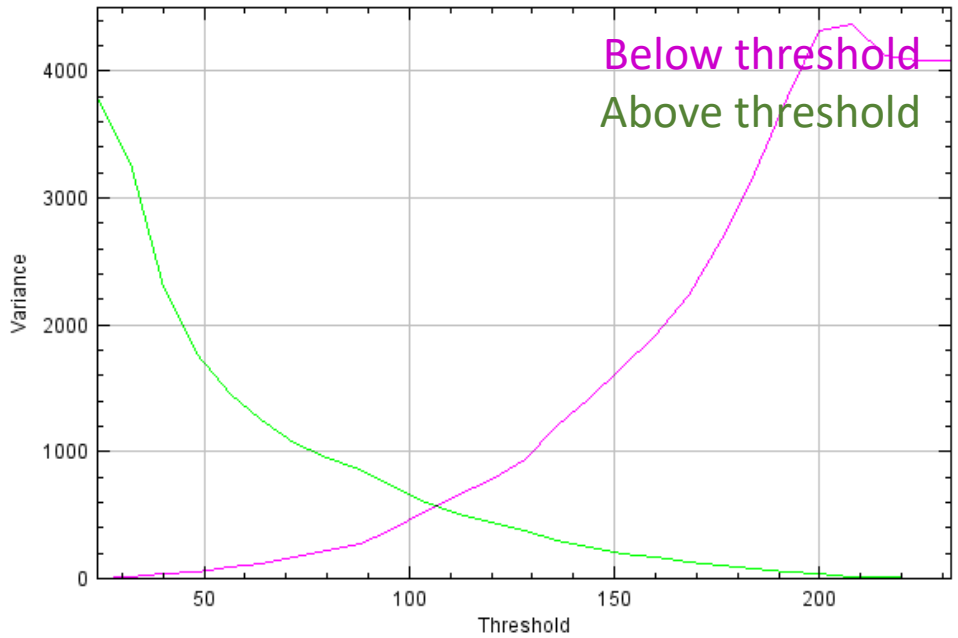


Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.
- Weighted (!) sum variance

$$Var'(I) = \frac{n_A}{n_I} Var(A) + \frac{n_B}{n_I} Var(B) \qquad I = A \cup B$$

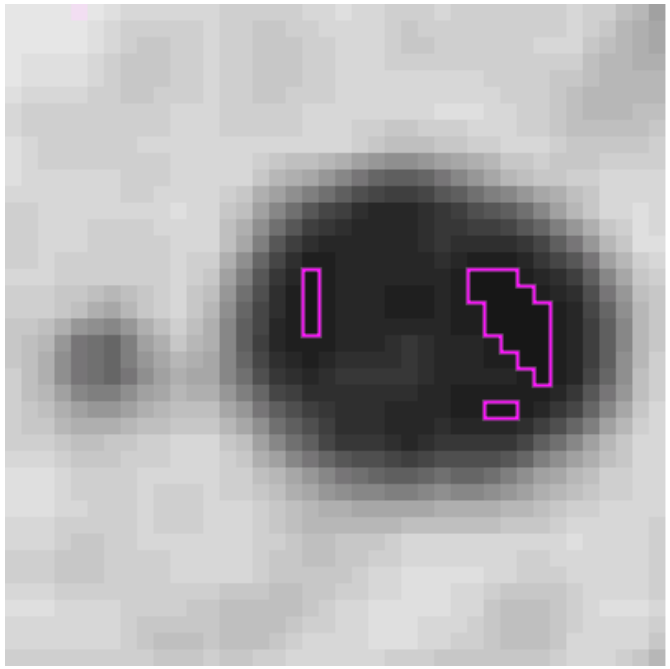


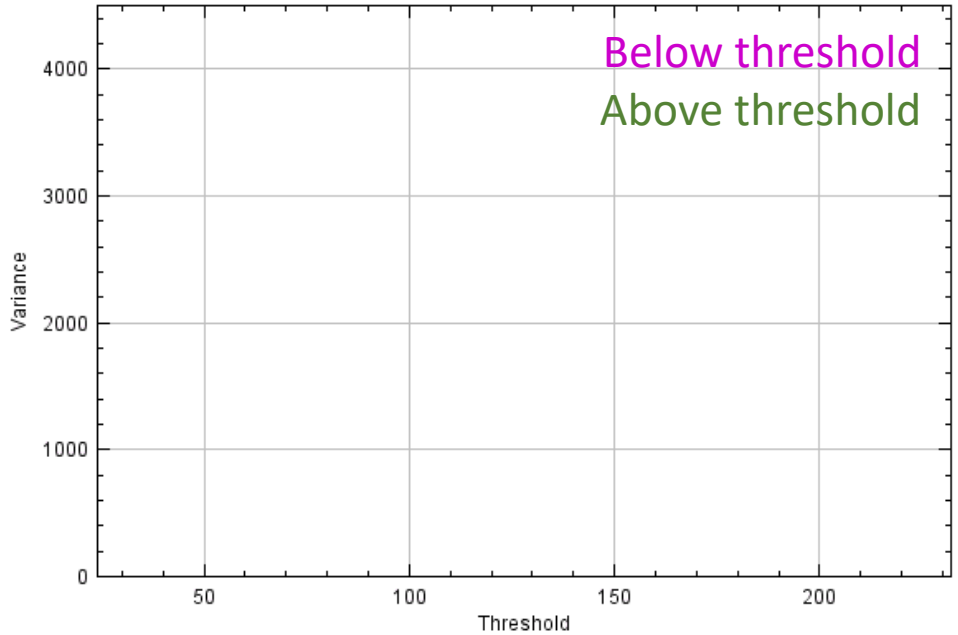


Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.
- Weighted (!) sum variance

$$Var'(I) = \frac{n_A}{n_I} Var(A) + \frac{n_B}{n_I} Var(B) \qquad I = A \cup B$$

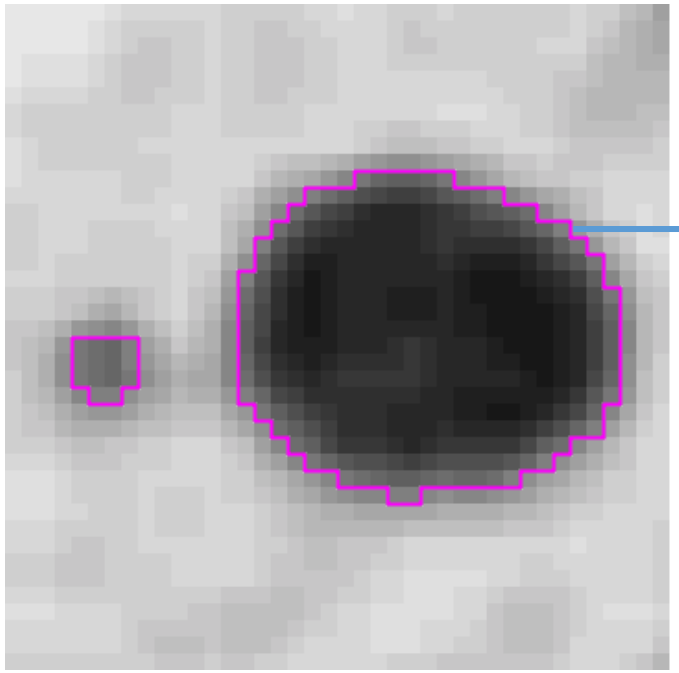


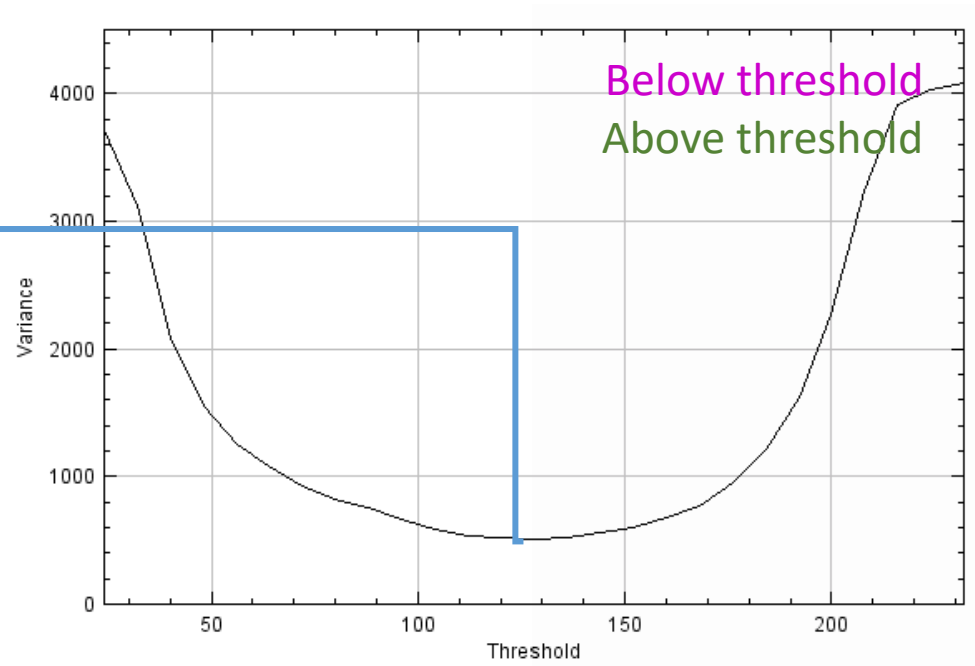


Thresholding: Otsu's method

- Searching for a threshold where the variance in both classes (above/below threshold) becomes minimal.
- Weighted (!) sum variance

$$Var'(I) = \frac{n_A}{n_I} Var(A) + \frac{n_B}{n_I} Var(B) \qquad I = A \cup B$$





Citing Otsu's method / implementation

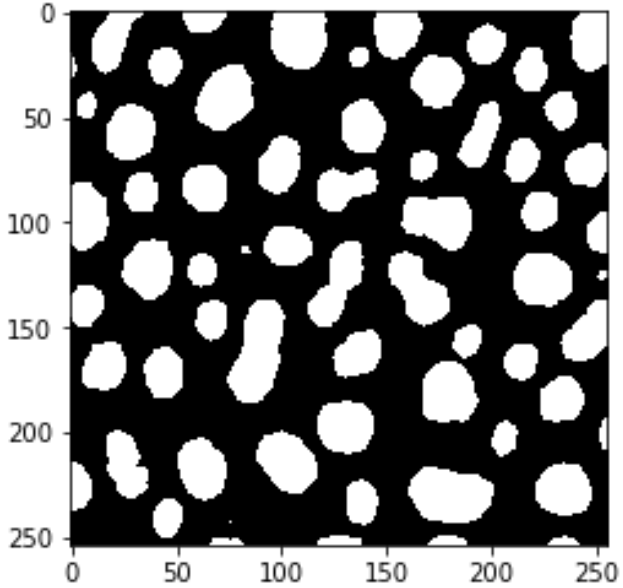
- Cite the thresholding method of your choice properly

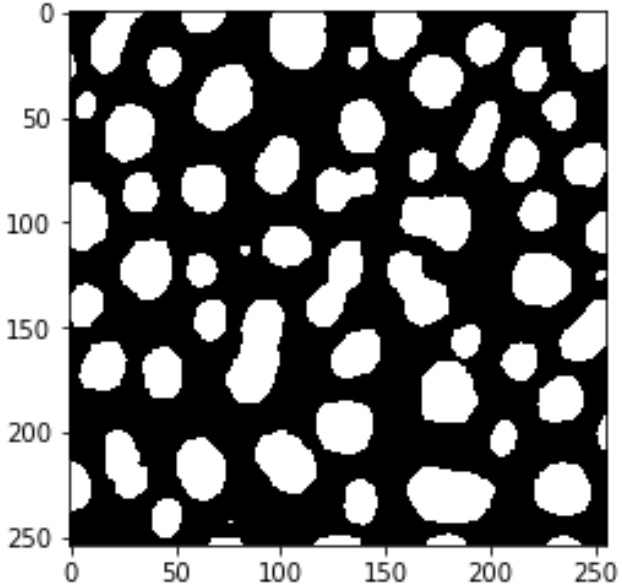
"We segmented the cell nuclei in the images using Otsu's thresholding method (Otsu et Al. 1979) implemented in scikit-image 0.19.3 (van der Walt et al. 2014)."

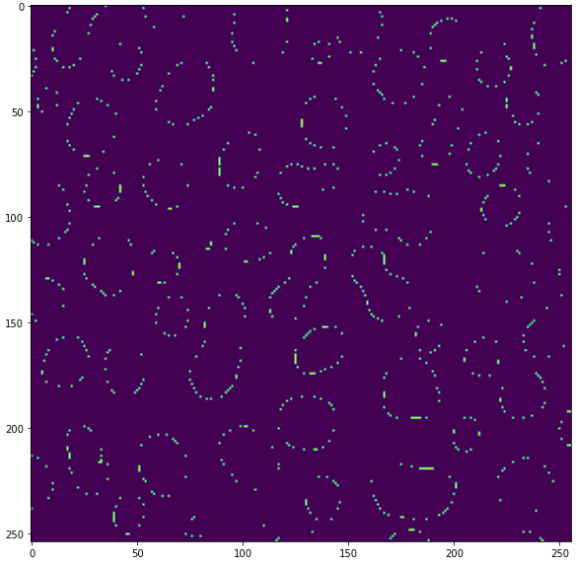
Quiz: Why do we need to report this so detailed?

Reproducibility of Otsu's method

- Comparing scikit-image versus ImageJ







Different pixels:

```
np.sum(difference)
```

```
830
```

Quiz

- Binary erosion is identical with which filter?

0	0	0	0	0
0	1	1	1	0
0	1	1	1	0
0	1	1	1	0
0	0	0	0	0

Erosion

0	0	0	0	0
0	0	0	0	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

Mean

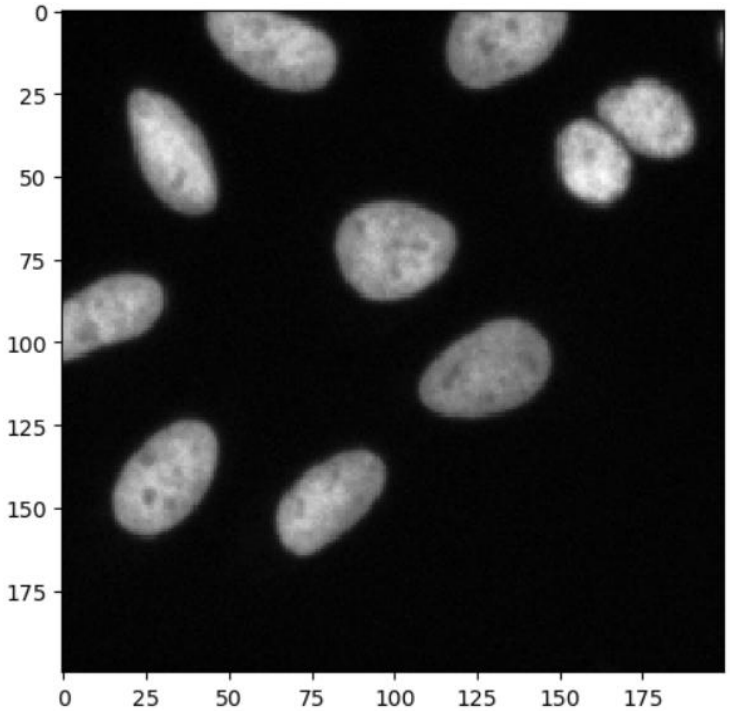
Median

Minimum

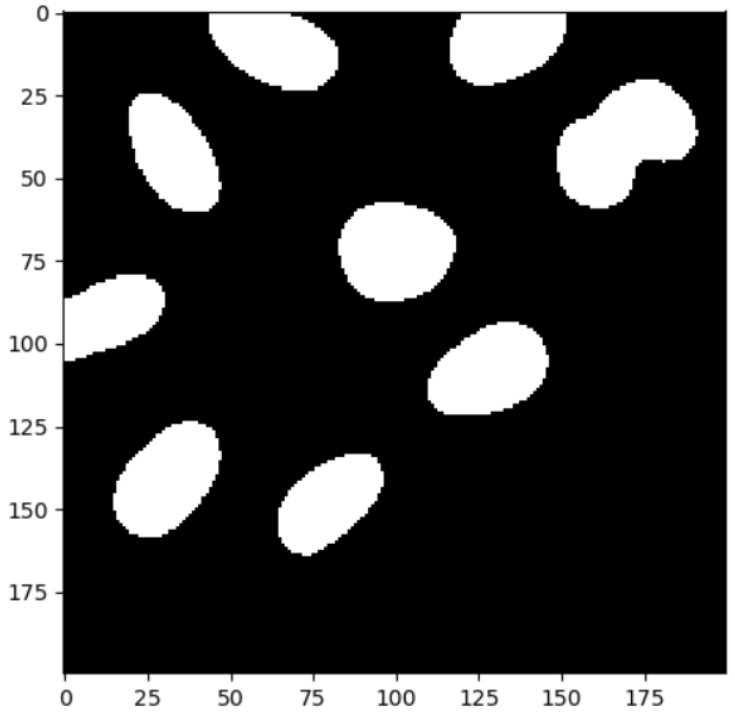
Maximum

Terminology

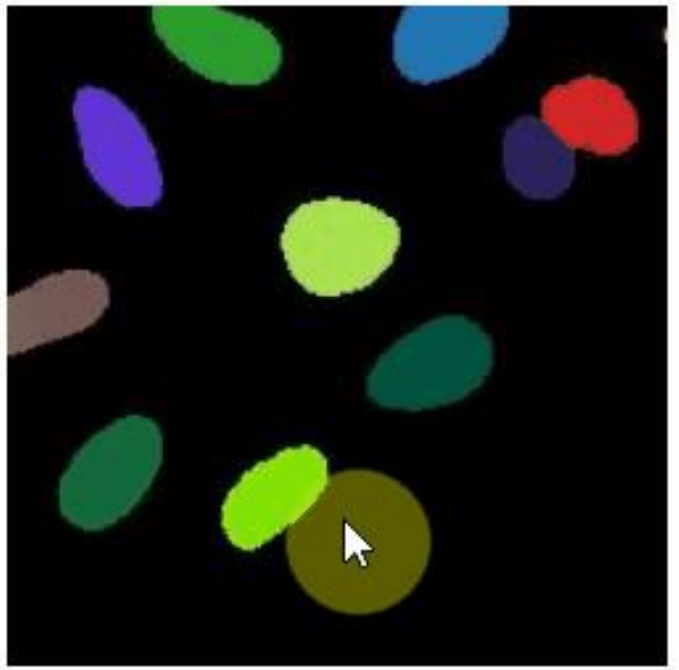
Intensity image



Binary image

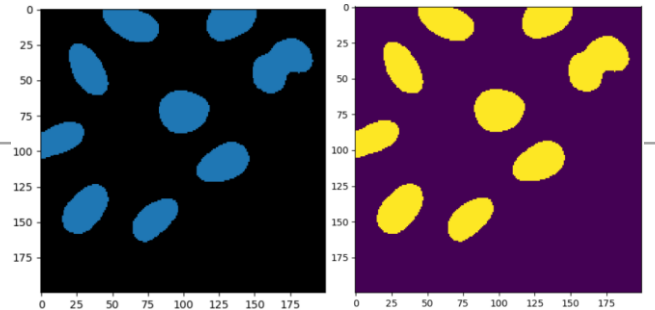


Label image



[y=152, x=92] = 0

No matter how they are displayed



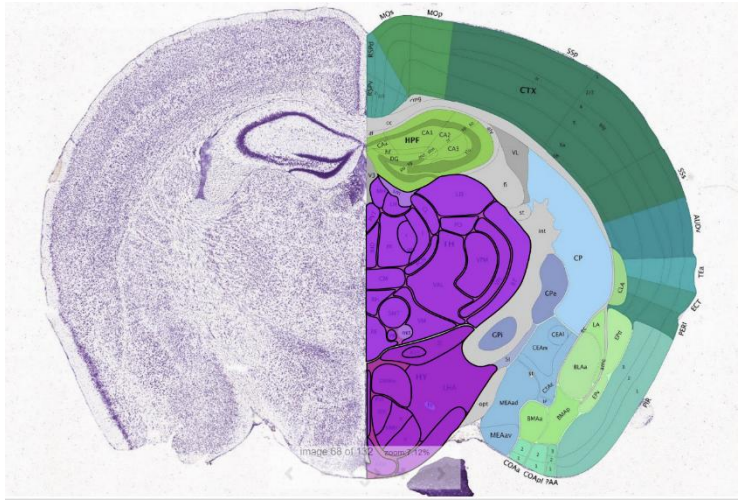
Terminology

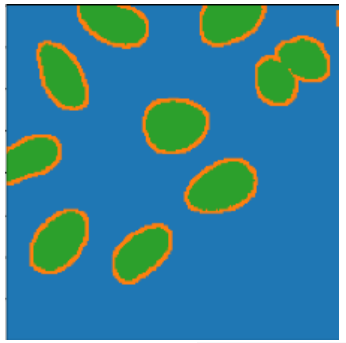
Instance segmentation

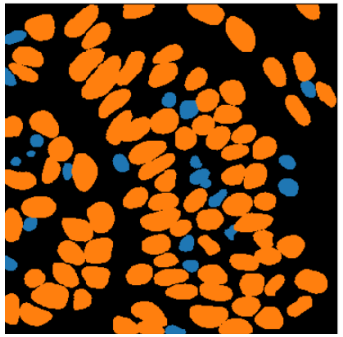
Instances:

- Cells, nuclei, cats, dogs, cars, trees

Semantic segmentation







Regions:

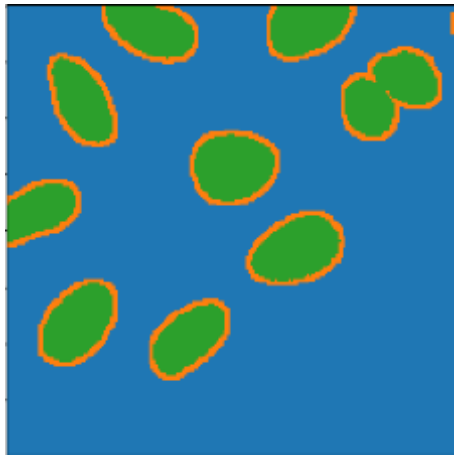
- Anatomical, geographical
- All pixels belonging to the same type of object have the same value

Source: Allen Mouse Brain Atlas [dataset]. Available from mouse.brain-map.org. Allen Institute for Brain Science (2011).

Terminology

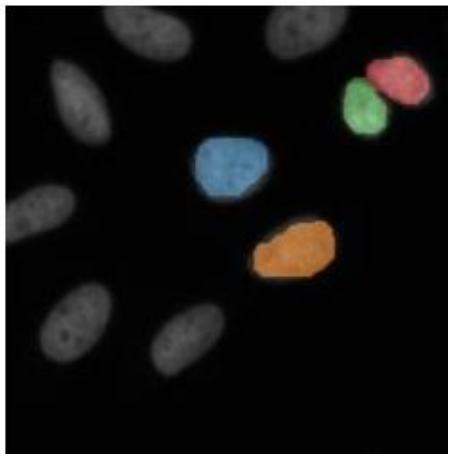
- Annotations are typically drawn by humans (e.g. to train machine learning models)

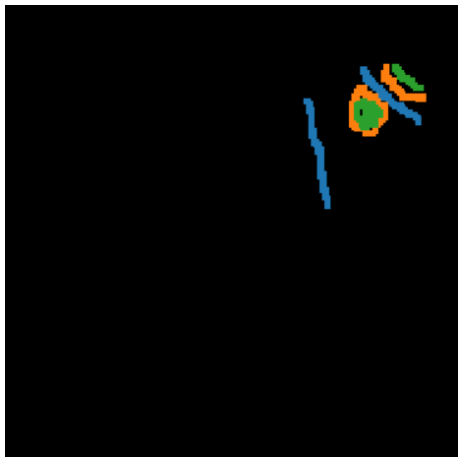
Instance segmentation



Semantic segmentation

Sparse instance annotation

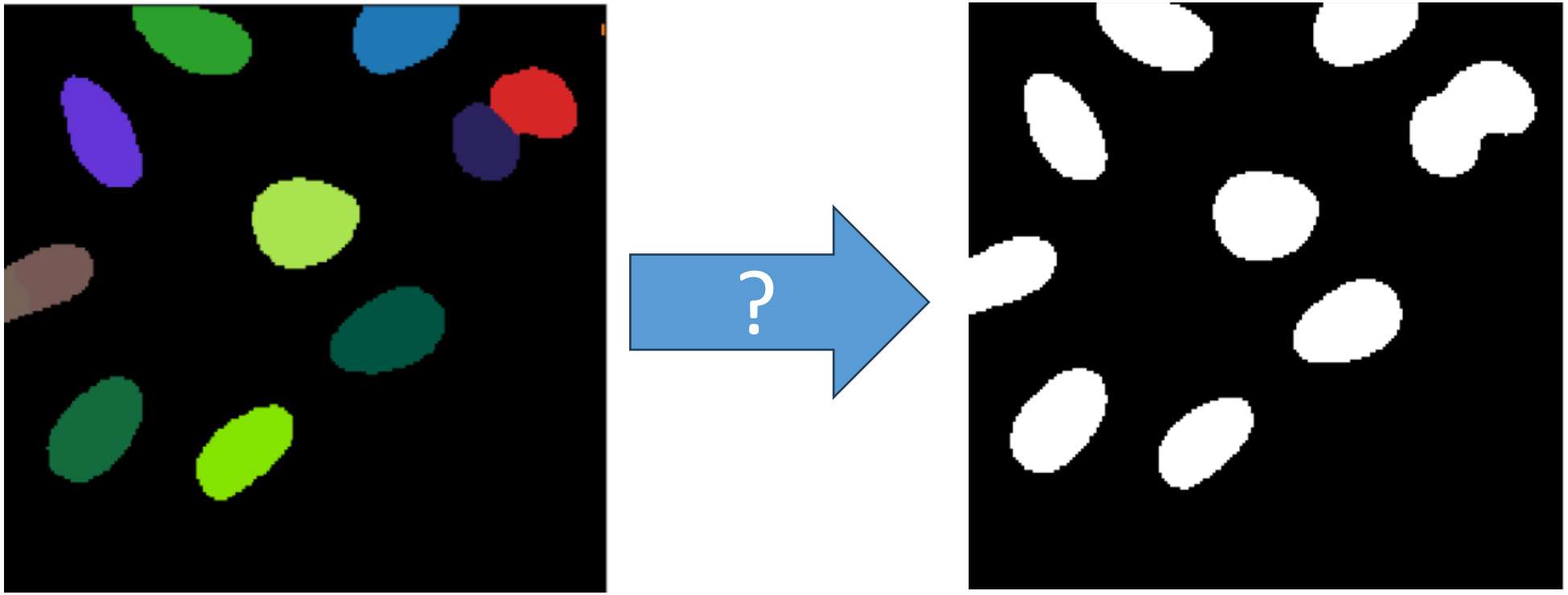




Sparse semantic annotation

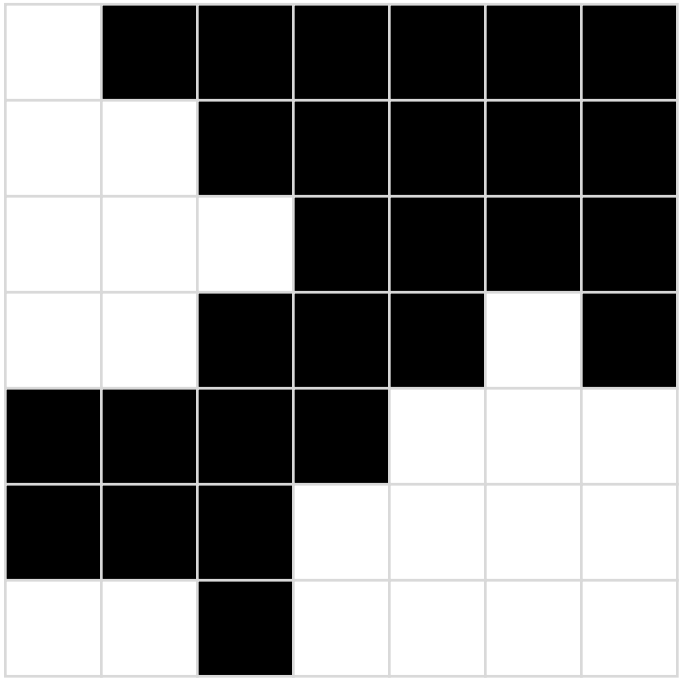
Quiz

- How could I turn a label image into a binary image?



Connected component labelling

- In order to allow the computer differentiating objects, connected component analysis (CCA) is used to mark pixels belonging to different objects with different numbers
- Background pixels are marked with 0.
- The maximum intensity of such a label map corresponds to the number of objects.



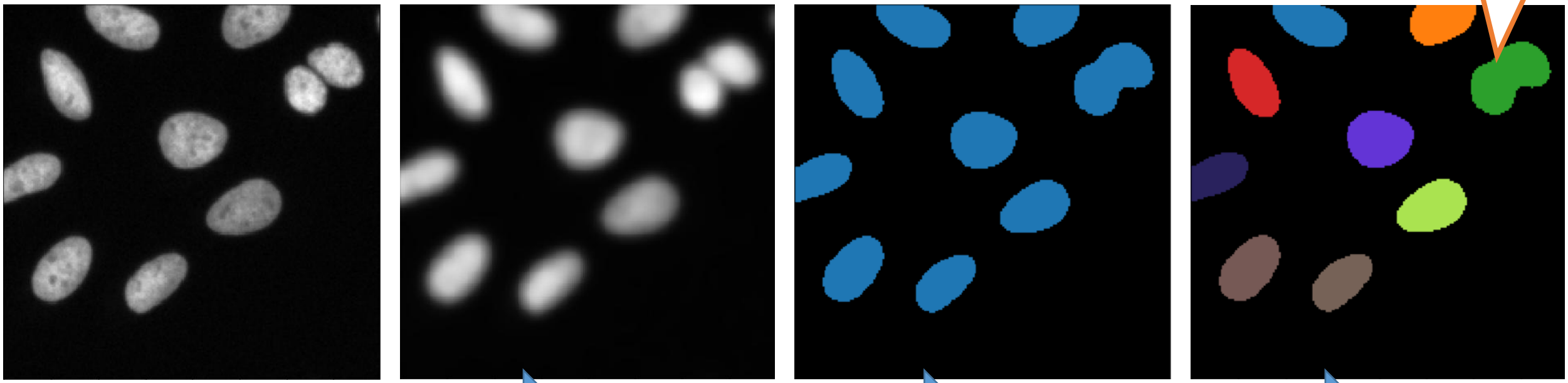
CCA

1	0	0	0	0	0	0
1	1	0	0	0	0	0
1	1	1	0	0	0	0
1	1	0	0	0	3	0
0	0	0	0	3	3	3
0	0	0	3	3	3	3
2	2	0	3	3	3	3

Common image segmentation workflows

- Presumably the most common segmentation algorithm used for fluorescence microscopy images:
  - Gaussian blur, Otsu's Threshold, Connected Component Labeling

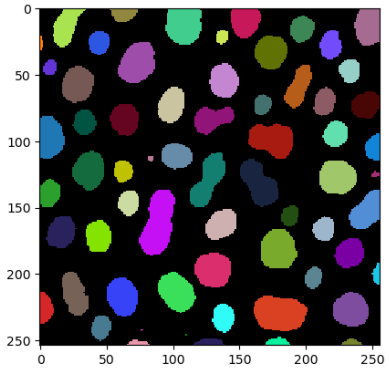
Limitation: Dense objects



Denoising → Binarization → Labeling

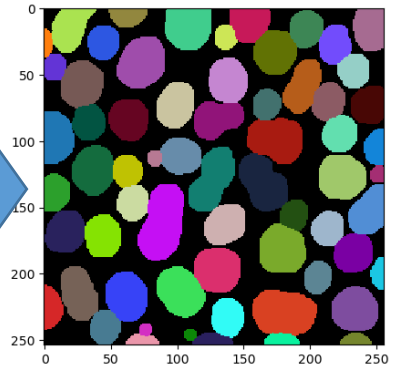
Voronoi-Tesselation

- For separating objects using spatial constraints (not intensity-based)

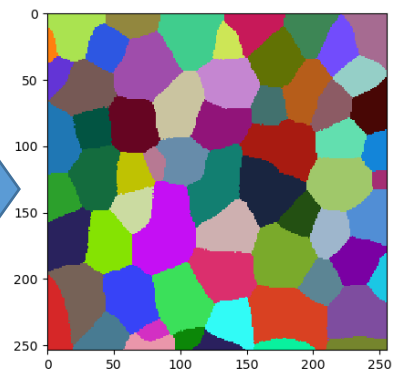


Label-image

Dilate labels

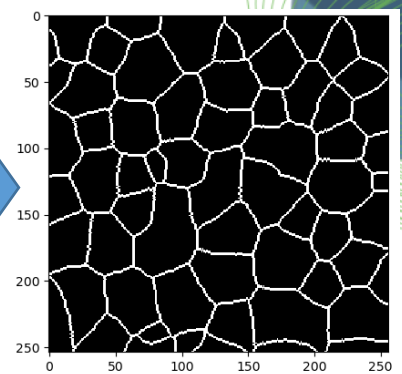


Dilate labels



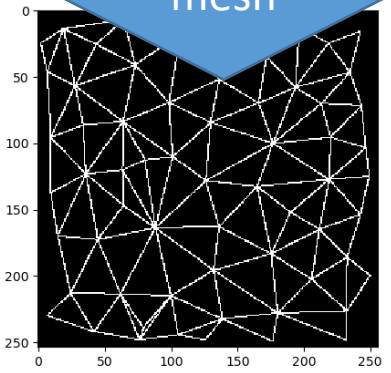
Voronoi-label-image

Edge detection

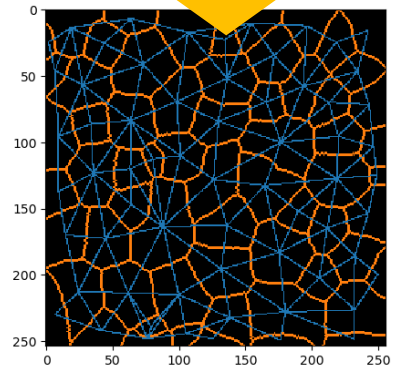


Label-edge-image

Centroid mesh



Touching neighbor mesh

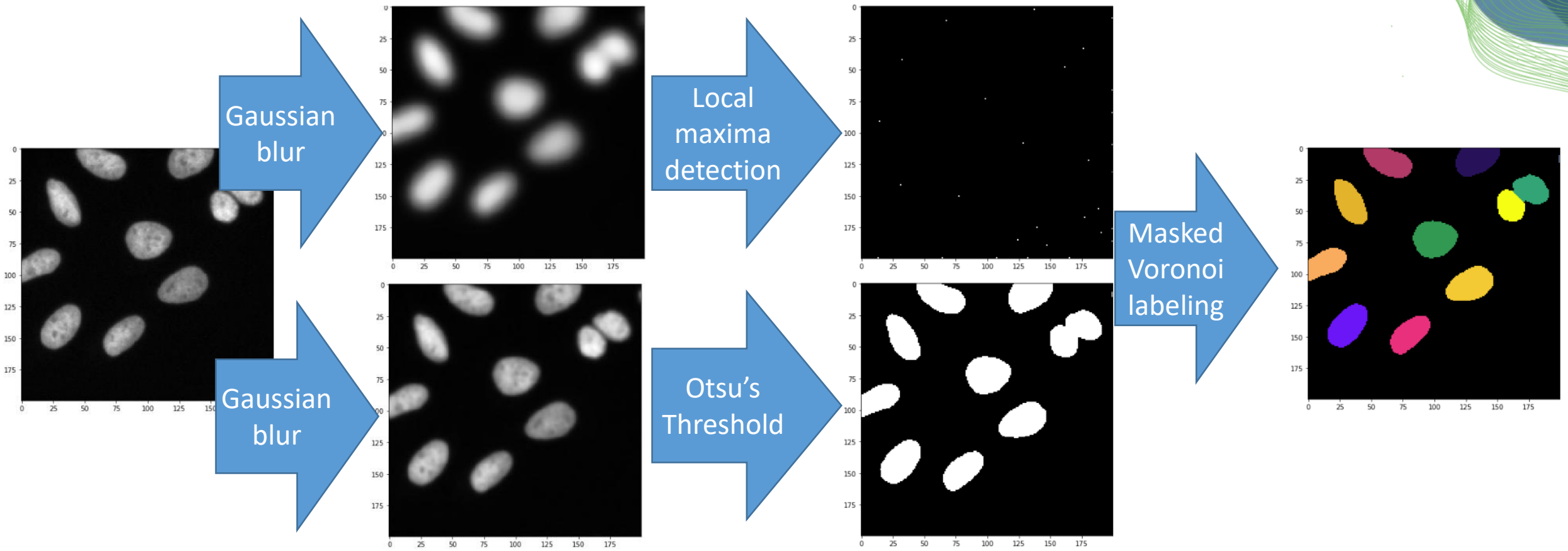


Voronoi-Tesselation

Delauney-Triangulation

Common image segmentation workflows

- Combination of Gaussian blur, Otsu's Threshold and Voronoi-labeling

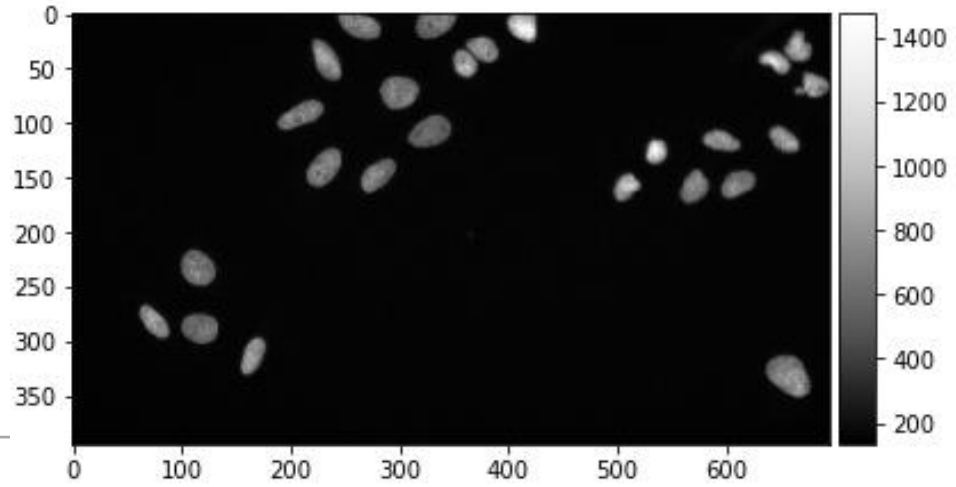


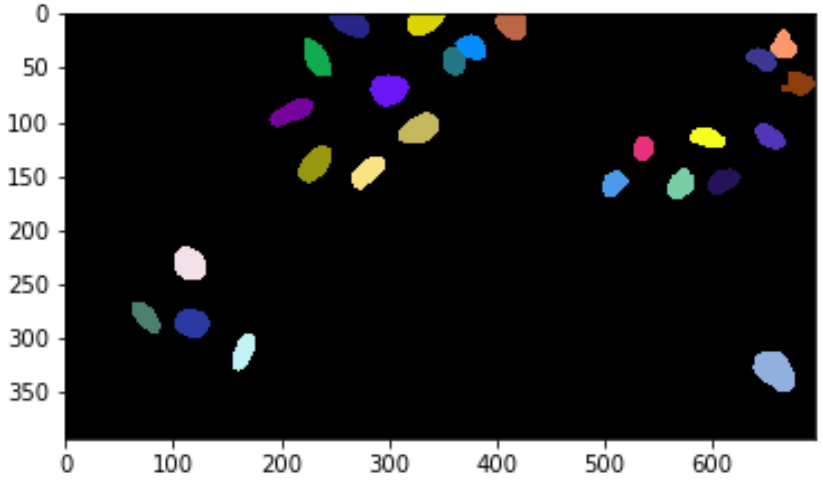
Voronoi-Otsu-Labeling

- Gaussian-Blur
- Otsu-Thresholding
- Spot-detection
- Watershed on the binary image

… in a single line of code:

```
segmented = nsbatwm.voronoi_otsu_labeling(input_image,
                                          spot_sigma=5,
                                          outline_sigma=1
                                         )
segmented
```





nsbatwm made image

shape	(395, 695)
dtype	int32
size	1.0 MB
min	0
max	25

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/20_image_segmentation/11_voronoi_otsu_labeling.html
https://www.youtube.com/watch?v=evgRgDfVuEc

Anisotropy

- Some [segmentation] algorithms have prerequisites…

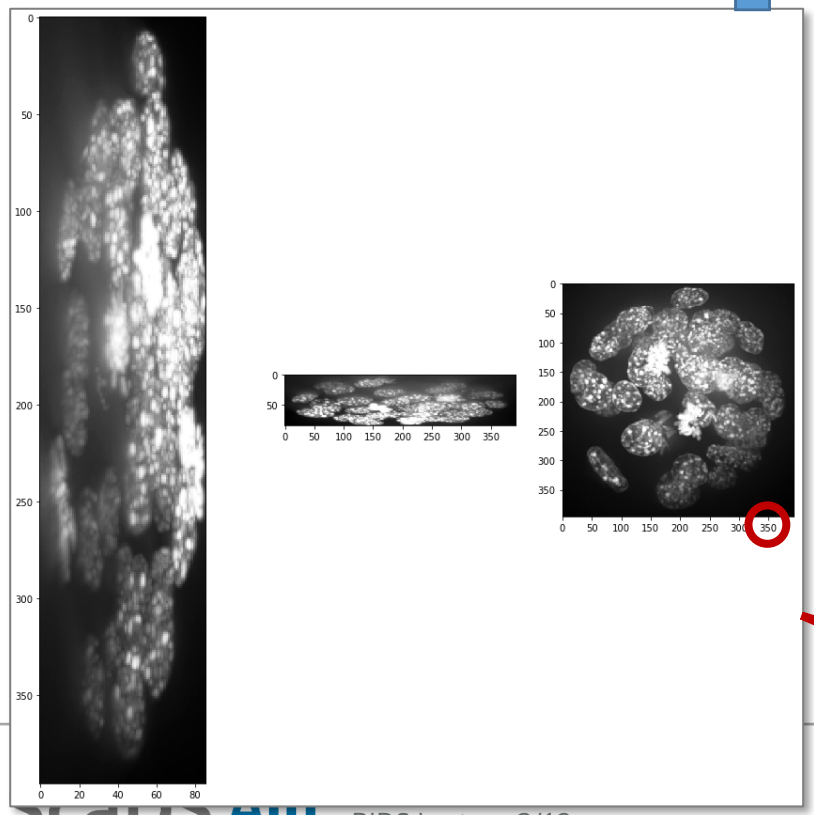
```
[1]: import pyclesperanto_prototype as cle

[ ]: cle.voronoi_otsu_labeling(

[ ]:
```

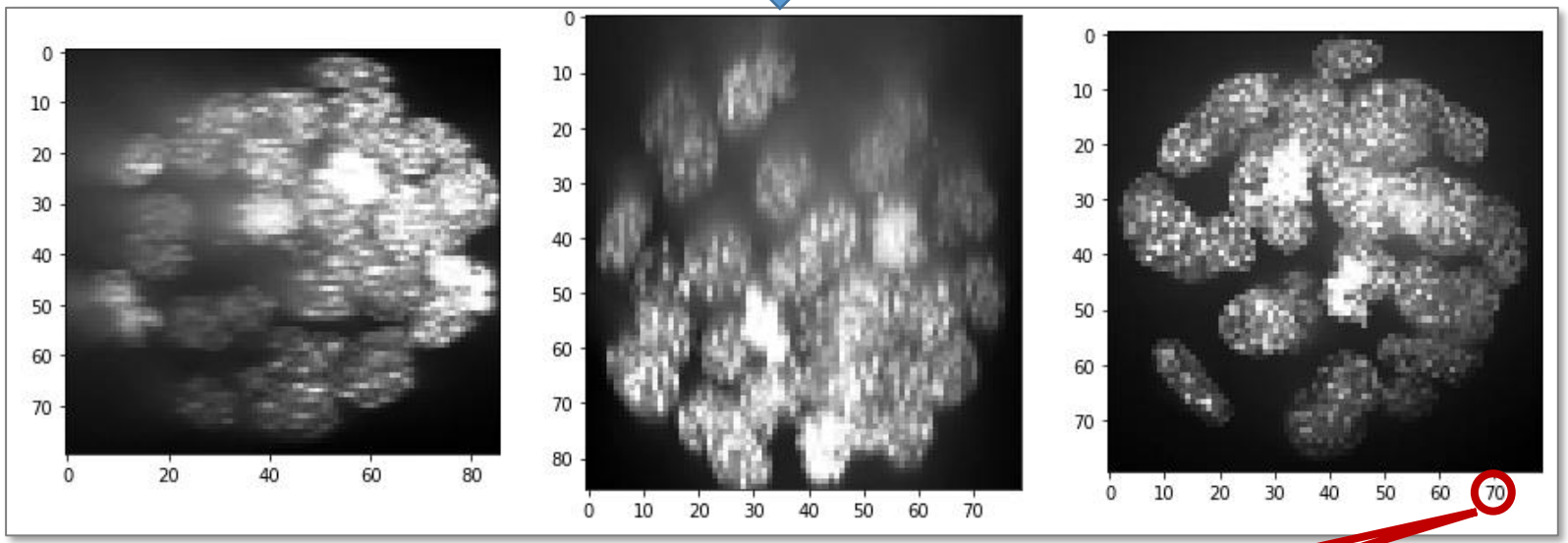
```
Docstring:
Labels objects directly from grey-value images.

The two sigma parameters allow tuning the segmentation result. Under the hood,
this filter applies two Gaussian blurs, spot detection, Otsu-thresholding [2] and Voronoi-labeling
[3]. The
thresholded binary image is flooded using the Voronoi tesselation approach starting from the found
local maxima.

Notes
-----
* This operation assumes input images are isotropic.

Parameters
----------
source : Image
    Input grey-value image
label_image_destination : Image, optional
    Output image
spot_sigma : float, optional
```

Reslicing / scaling / sampling

- Resample image data to a specific voxel size

```
resampled = cle.scale(input_image, factor_x=voxel_size_x, factor_y=voxel_size_y, factor_z=voxel_size_z, auto_size=True)

show(resampled)
```

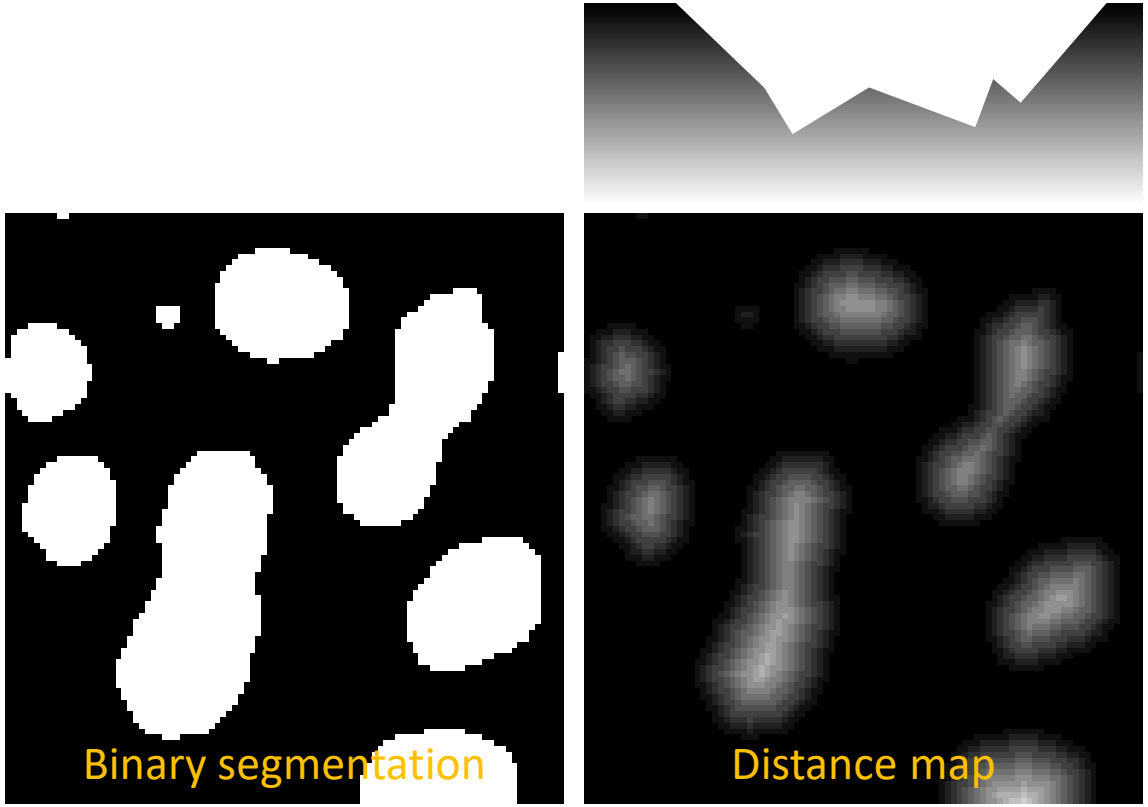




Down-sampling versus up-sampling,
in this case:
670kByte versus 89mByte

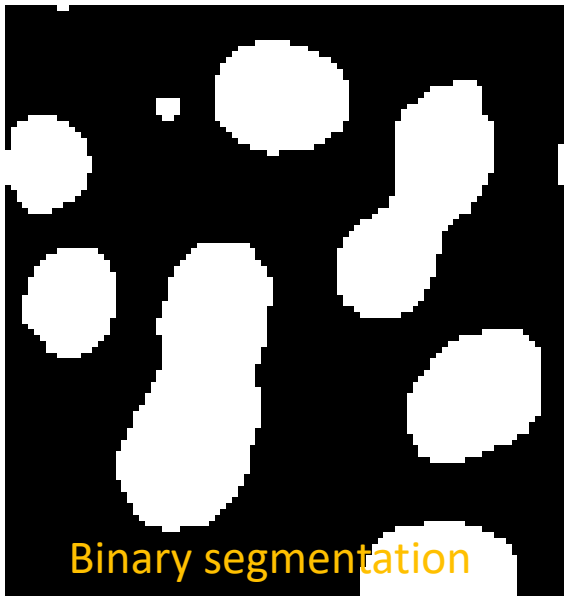
Watershed

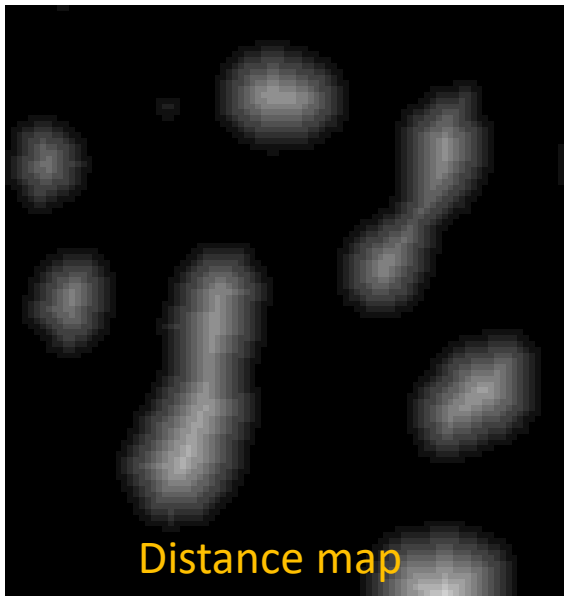
- The watershed algorithm for binary images allows cutting one object into tow where it’s reasonable.



Watershed

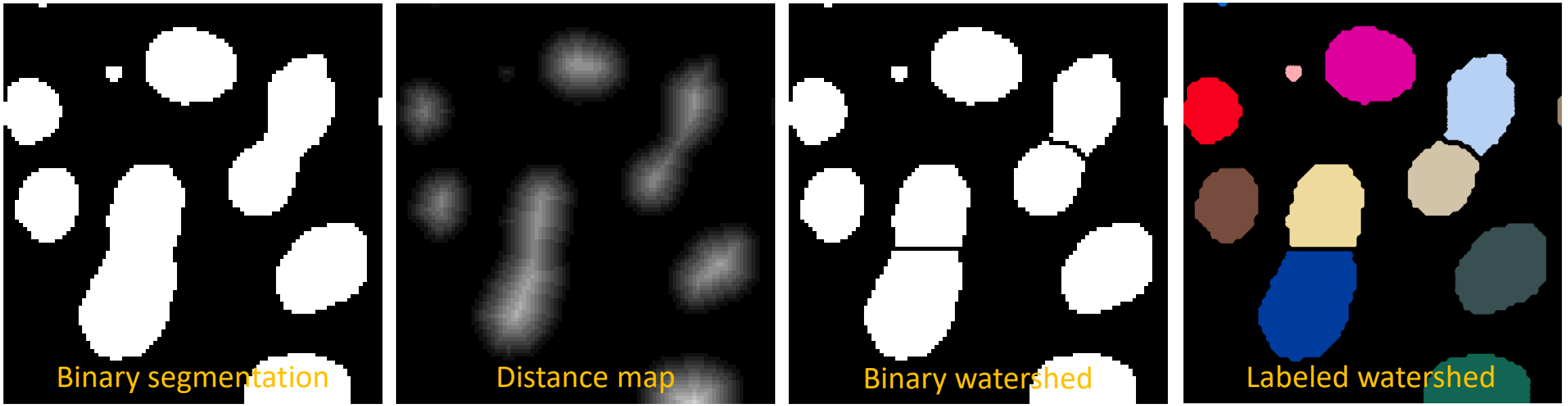
- The watershed algorithm for binary images allows cutting one object into tow where it’s reasonable.





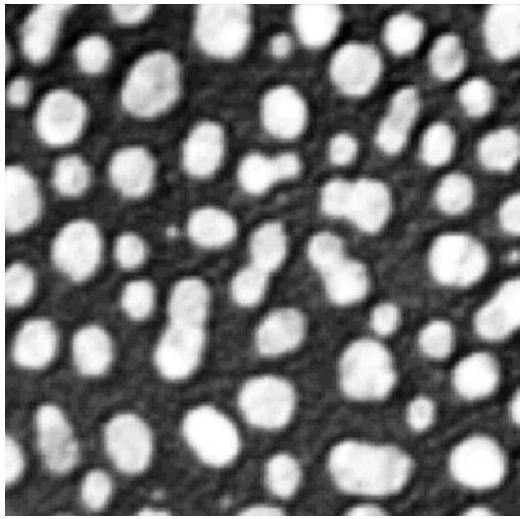
Watershed

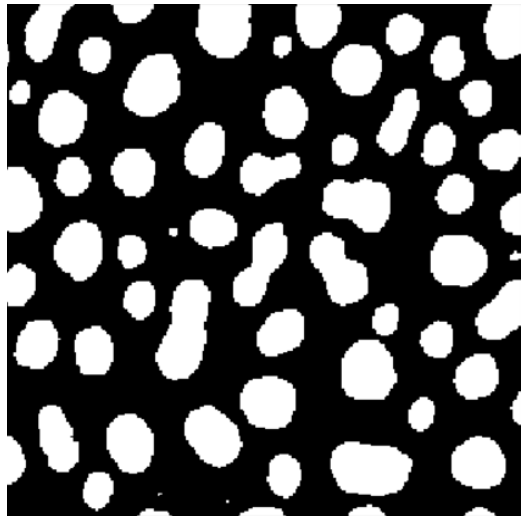
- The watershed algorithm for binary images allows cutting one object into two where it’s reasonable.
- The distance-maps are typicall made from binary images. It does not take the original image into account!



Watershed use-cases

- Split dense objects

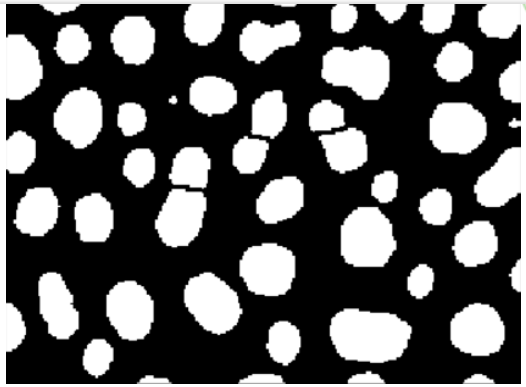




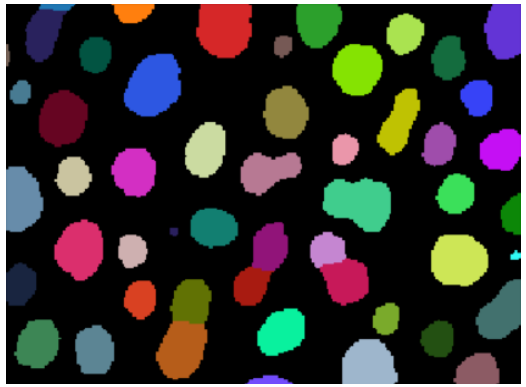
Binarization

"Watershed"

```
split_objects = nsbatwm.split_touching_objects(binary)
split_objects
```



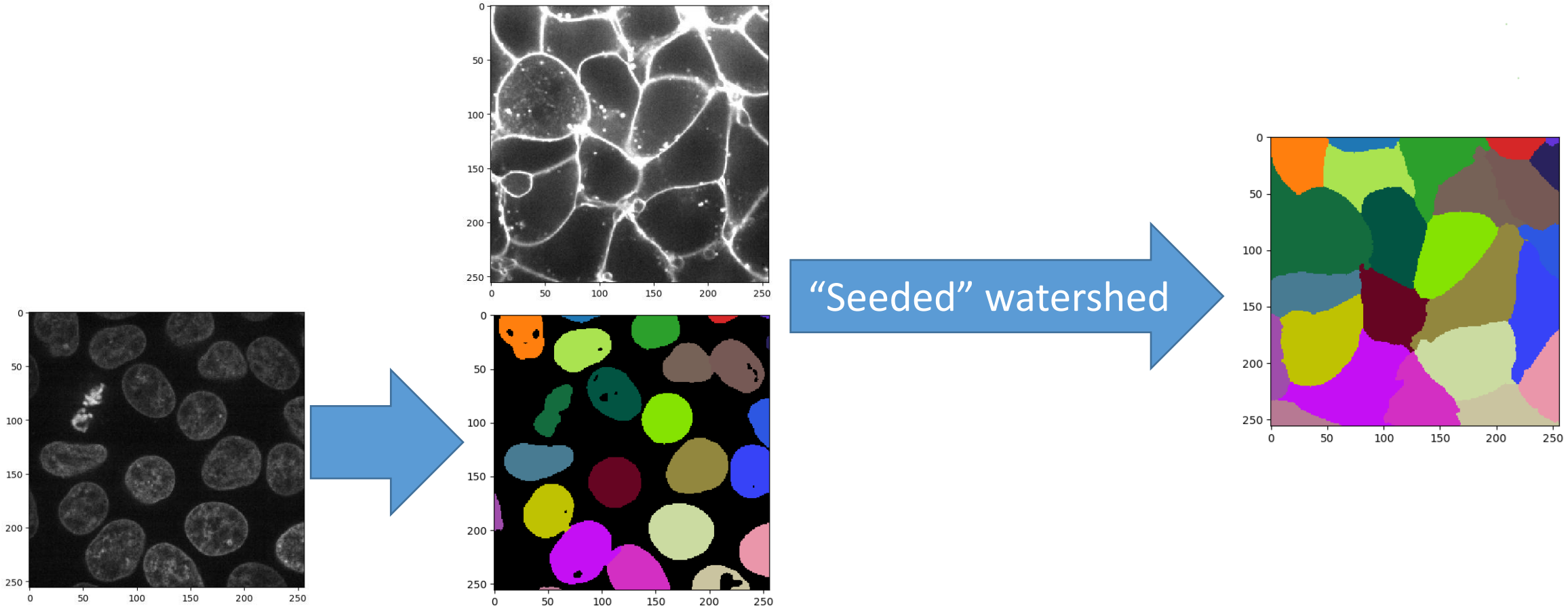
"Watershed"



```
touching_labels = nsitk.touching_objects_labeling(binary)
touching_labels
```

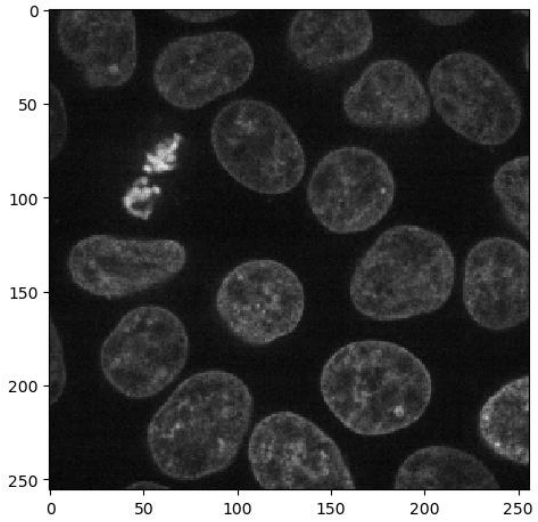
Watershed use-cases

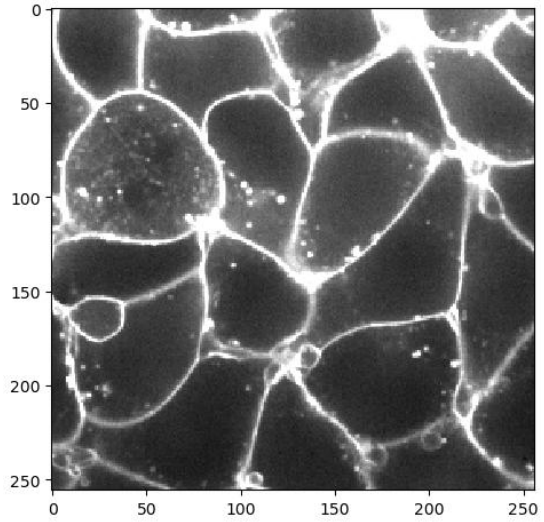
- Seeded watershed: Flood regions from pre-defined seeds
- Example: Flood membrane image from nuclei positions (no distance map)

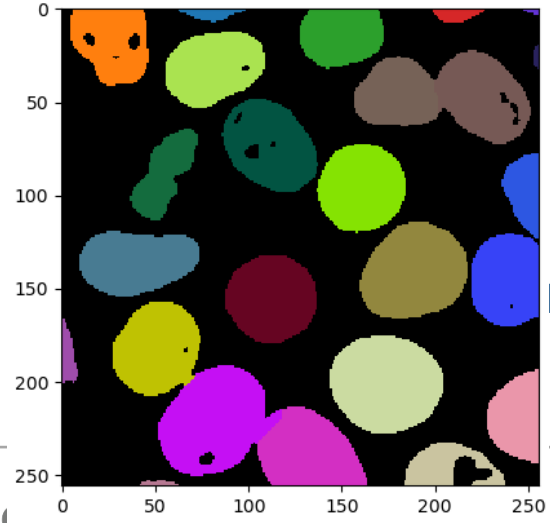


Watershed

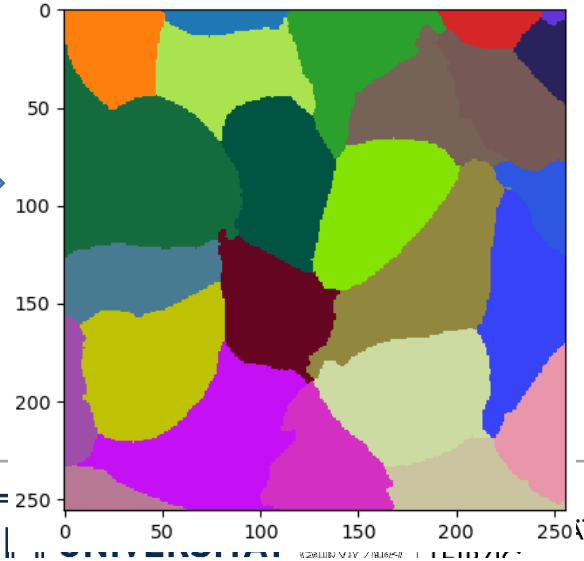
- … in Python practice







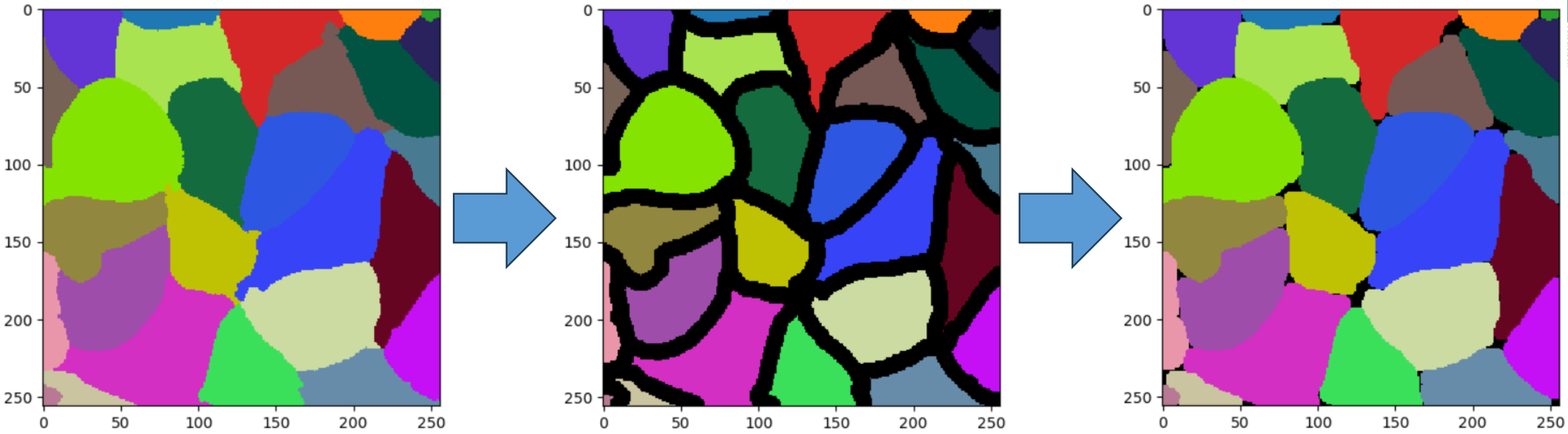

```
labeled_cells = seeded_watershed(membrane_channel, labeled_nuclei)

labeled_cells
```



https://www.napari-hub.org/plugins/napari-segment-blobs-and-things-with-membranes#seeded-watershed

Label post-processing / morphological operations

- … similar to morphological operations on binary images



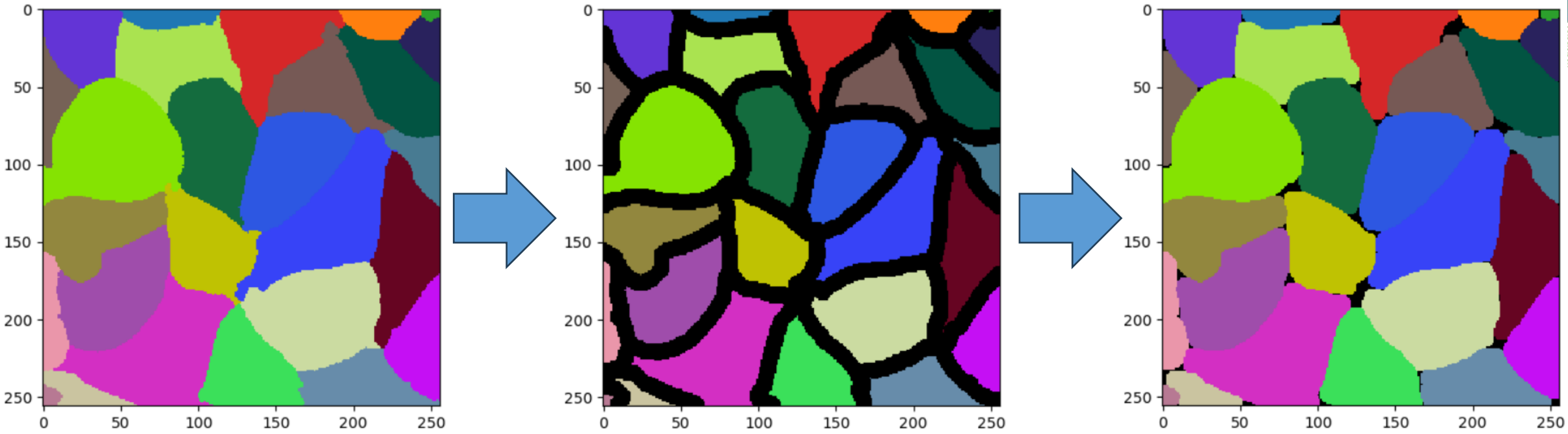
Original

Eroding labels

Dilating Labels

Label post-processing / morphological operations

- … similar to morphological operations on binary images



Original

Eroding labels

Dilating Labels

This combination is called …?

Opening

Closing

Label post-processing / morphological operations

- … similar to morphological operations on binary images

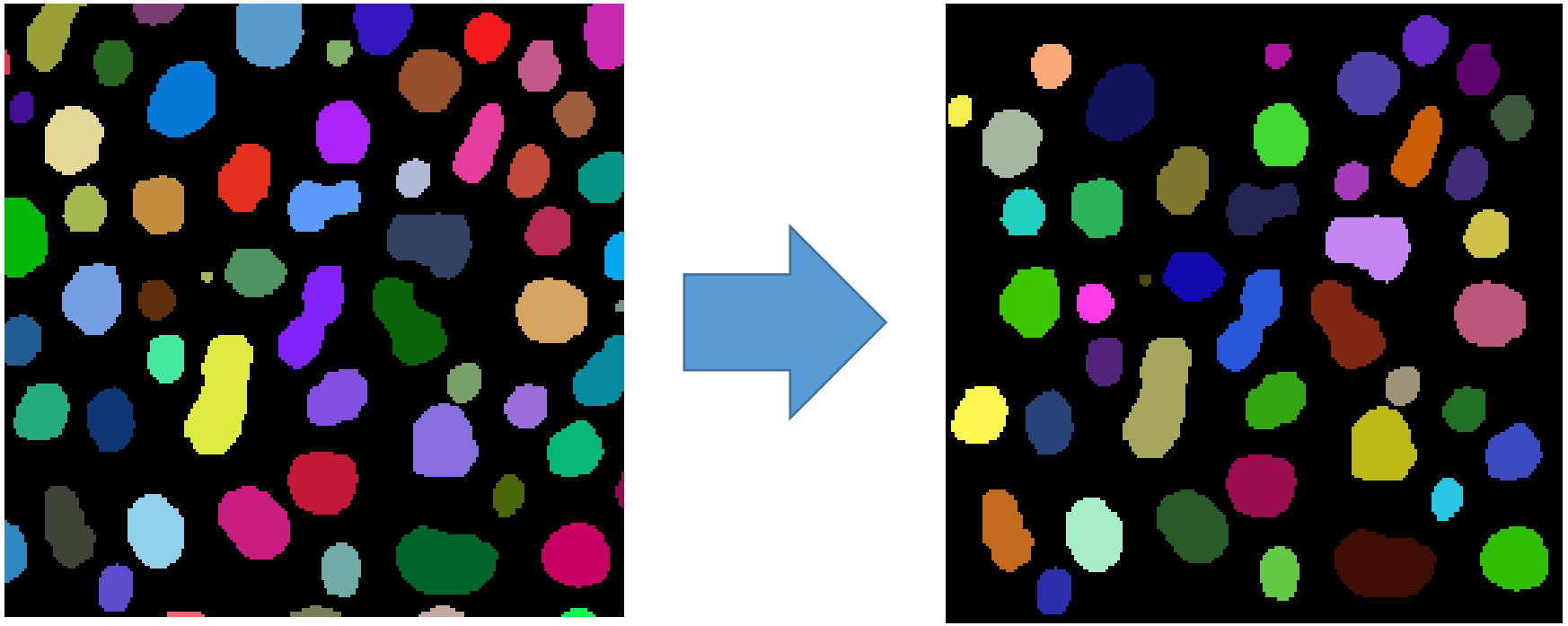
Original

Opening labels

Smoothing Labels

Label post-processing / selections

- Remove objects at the image border



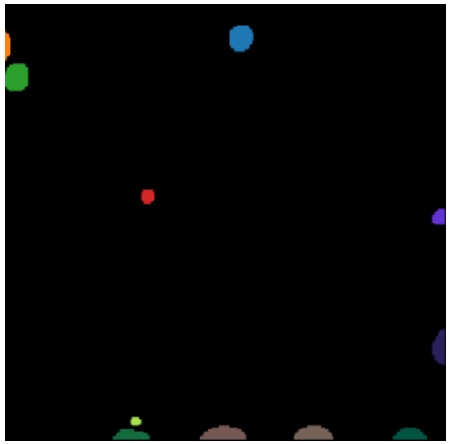
Quiz: Why is it a good idea to do this before measuring object properties?

Label post-processing / selections

- Excluding small / large objects
- Common correction-step in case segmentations contain noise-related small particles

Exclude small objects

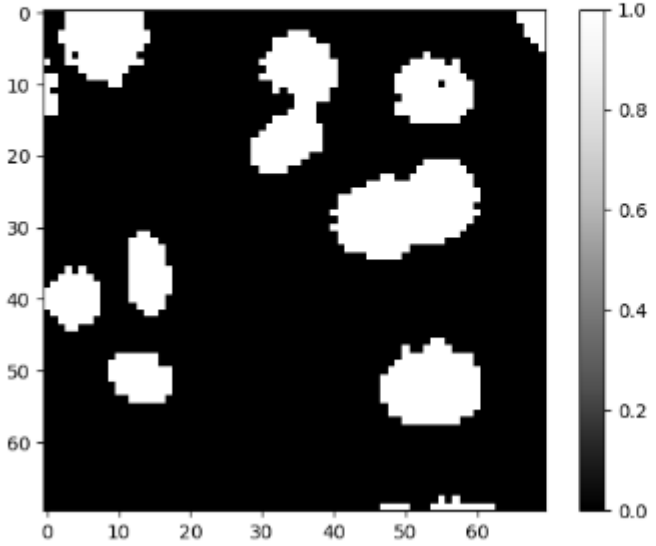
Exclude large objects



Pitfalls in Python

- Image results of some libraries do not need to be imshow'ed

```
binary_image = cle.threshold_otsu(image)
binary_image
```



cle._ image

shape	(70, 70)
dtype	uint8
size	4.8 kB
min	0.0
max	1.0

Pitfalls in Python

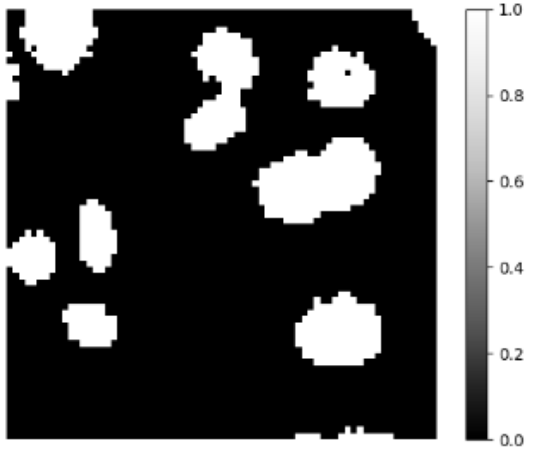
- Some threshold functions return numbers, others return images.

```
threshold = filters.threshold_otsu(image)

threshold
```

```
77
```

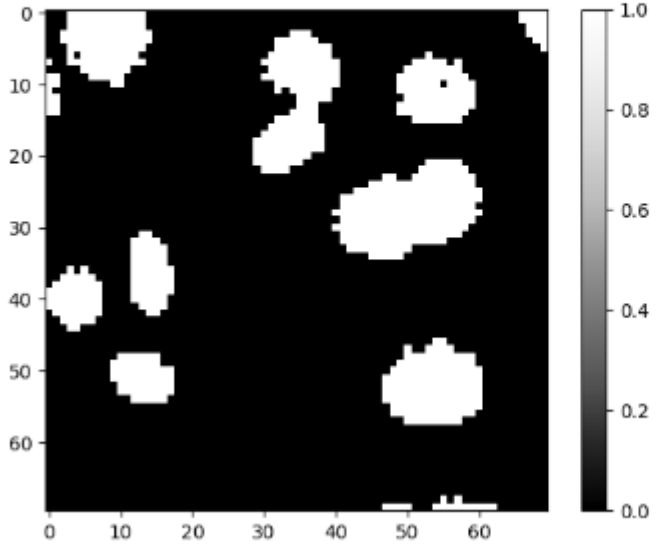
```
binary_image = image > threshold

stackview.insight(binary_image)
```

```
c:\structure\code\stackview\stackview\_static_view.py:101: RuntimeWarning: Converting input from bool to <class 'numpy.uint8'> for compatibility.
  h, _ = np.histogram(self.obj, bins=num_bins)
```



shape	(70, 70)
dtype	bool
size	4.8 kB
min	False
max	True

```
binary_image = cle.threshold_otsu(image)
binary_image
```



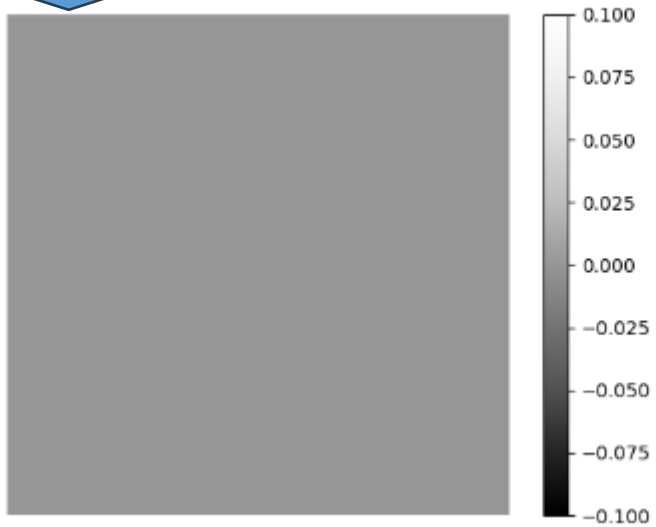
cle._ image

shape	(70, 70)
dtype	uint8
size	4.8 kB
min	0.0
max	1.0

Pitfalls in Python

```python
image = imread("data/mitosis_mod.tif")

# determine a threshold
threshold = filters.threshold_otsu(image)

# blur the image to make segmentation outlines smoother
blurred_image = filters.gaussian(image, sigma=2)

# threshold the image
binary_image = blurred_image > threshold

stackview.insight(binary_image)
```

Quiz: Can anyone spot the issue?



shape	(70, 70)
dtype	bool
size	4.8 kB
min	False
max	False

Background: gaussian() normalizes the intensity range. Mathematical nonsense, but default unfortunately.

- set `preserve_range=True` or
- Determine the threshold on the same image where you also apply the threshold.

```python
image.min(), image.max(), image.dtype
```

```
(8, 255, dtype('uint8'))
```

```python
blurred_image.min(), blurred_image.max(), blurred_image.dtype
```

```
(0.06732543055888668, 0.7405626105467442, dtype('float64'))
```

Pitfalls in Python

- 3D image processing may require GPU-acceleration
- In case there are errors like PLATFORM_NOT_FOUND or BUILD_PROGRAM_FAILURE: https://github.com/clEsperanto/pyclesperanto_prototype/?tab=readme-ov-file#troubleshooting-graphics-cards-drivers

ScaDS.AI
DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS AND ARTIFICIAL INTELLIGENCE

Interactive image processing using Napari

Robert Haase

Funded by

Bundesministerium für Bildung und Forschung

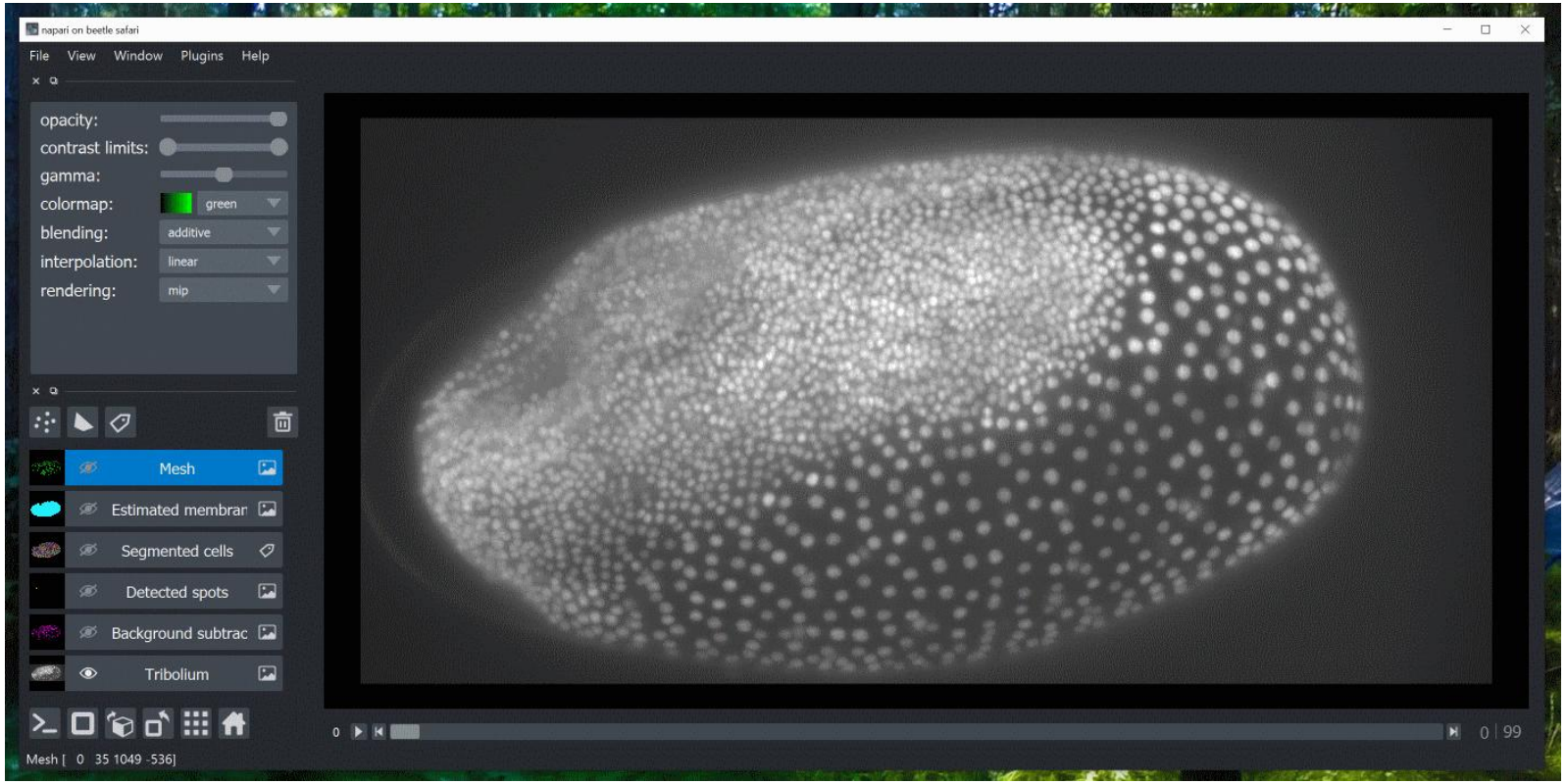
SACHSEN

Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Chan Zuckerberg Initiative

Napari

- A viewer for n-dimensional image data written in Python

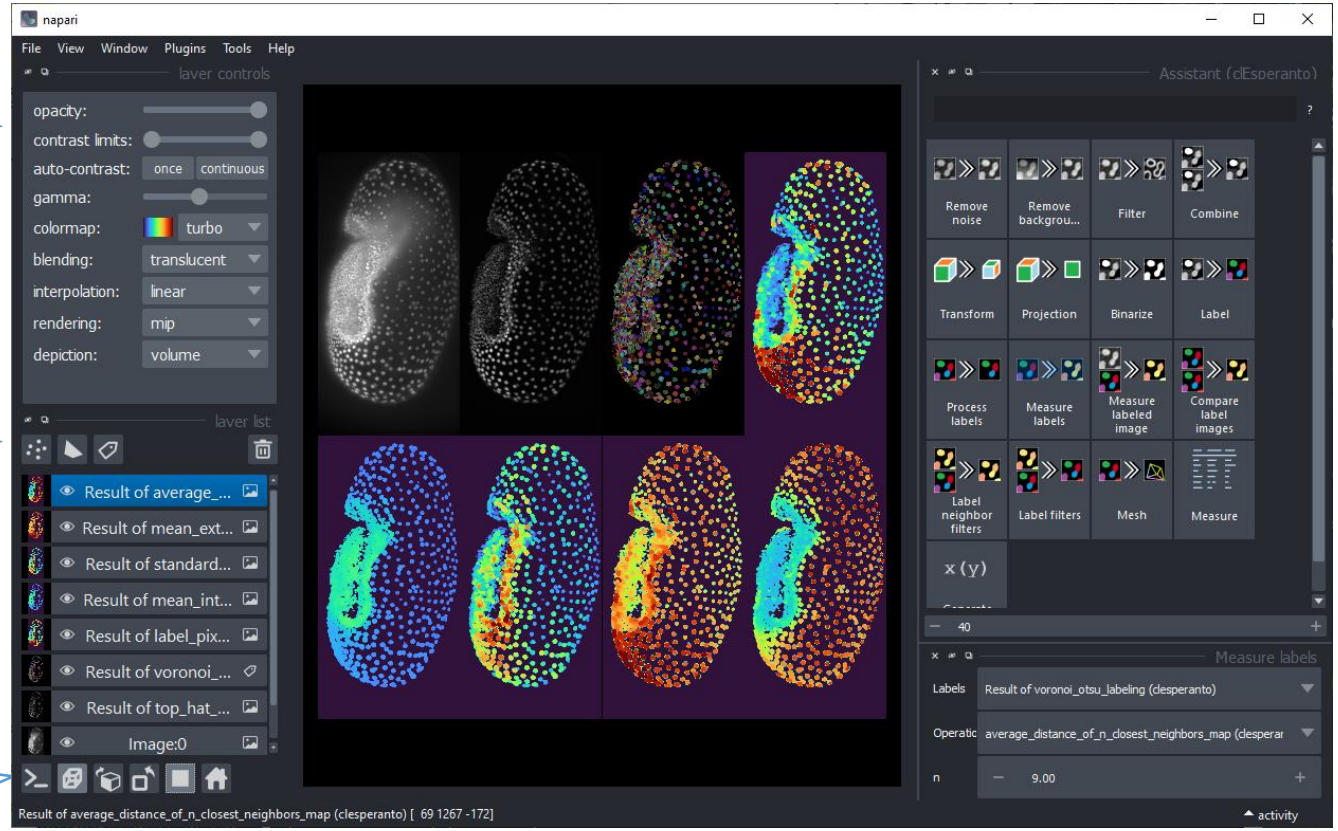


Napari – Graphical User Interface

View configuration / tools

Layers

Viewer controls

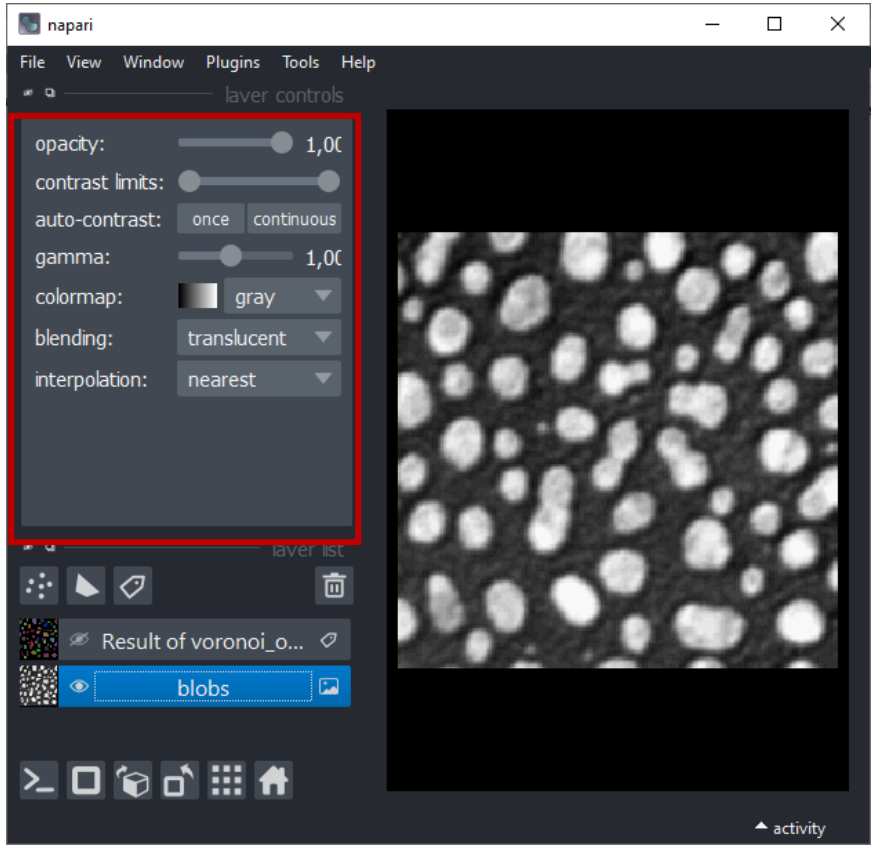


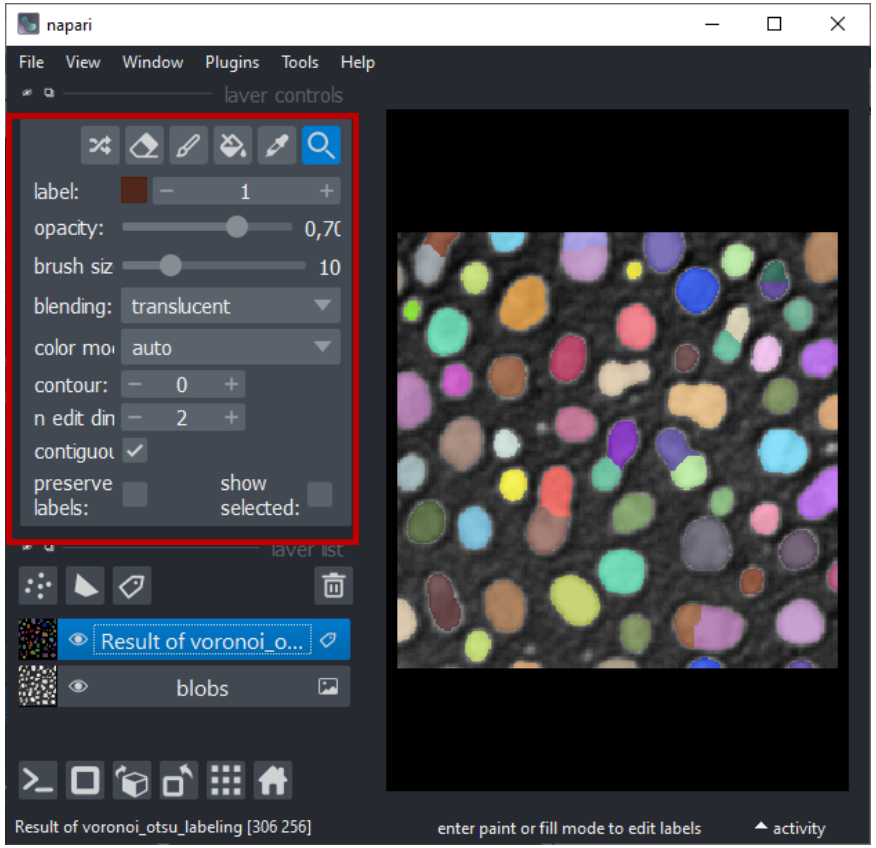
Dock widgets (custom plugins)

Function widgets (custom plugins)

Napari – Graphical User Interface

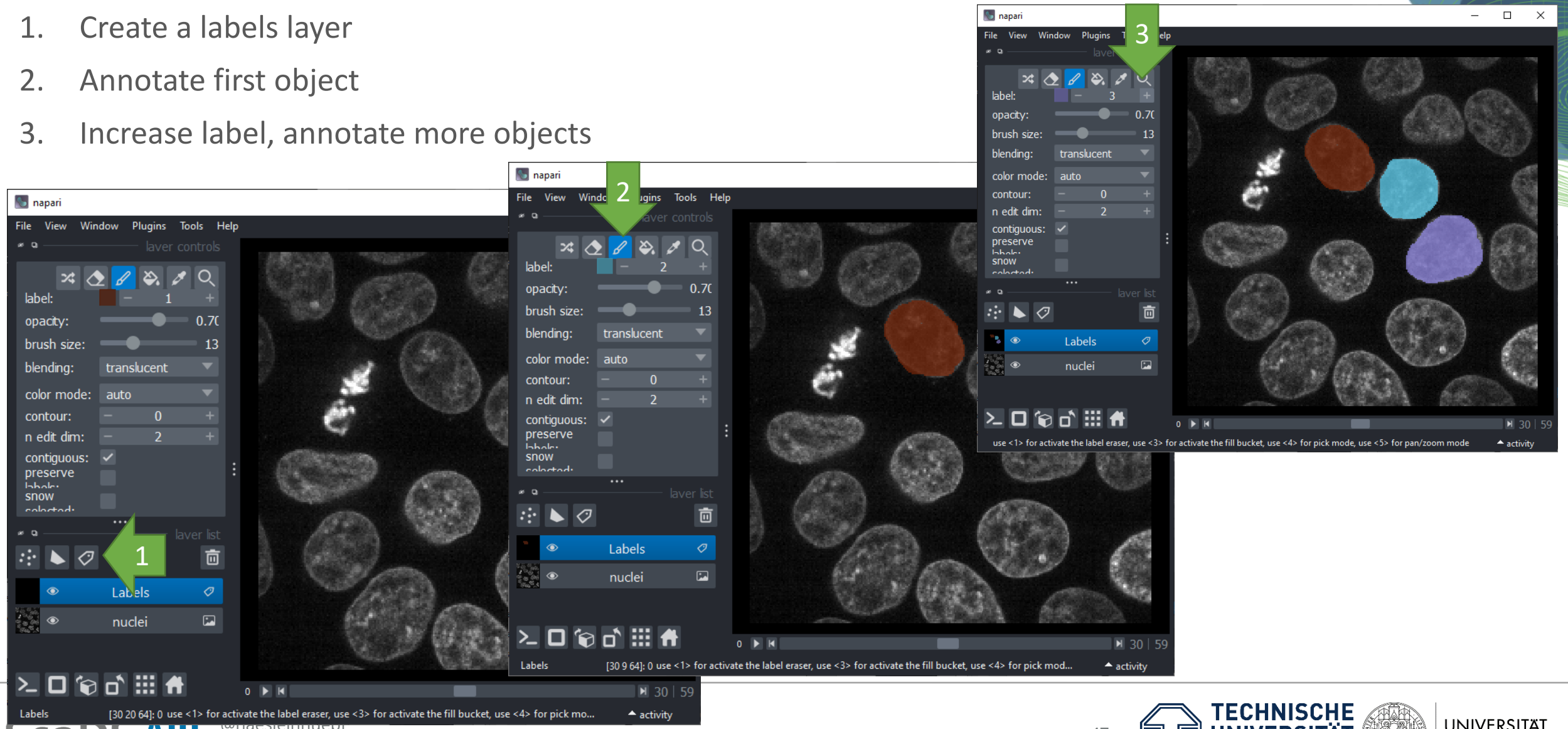
- Context / data type dependent tools





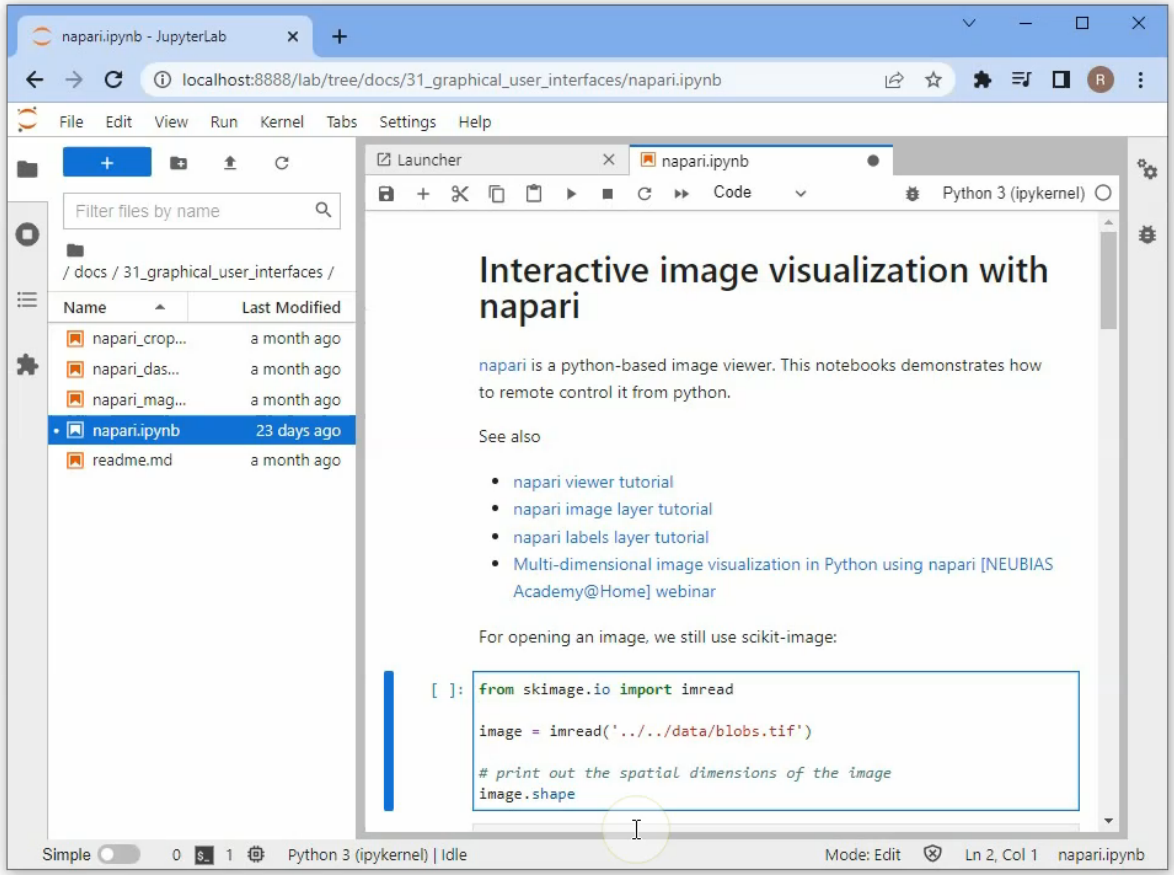
Manual annotation

1. Create a labels layer
2. Annotate first object
3. Increase label, annotate more objects



Napari – Python Scripting

- Mixing interactivity and reproducibility



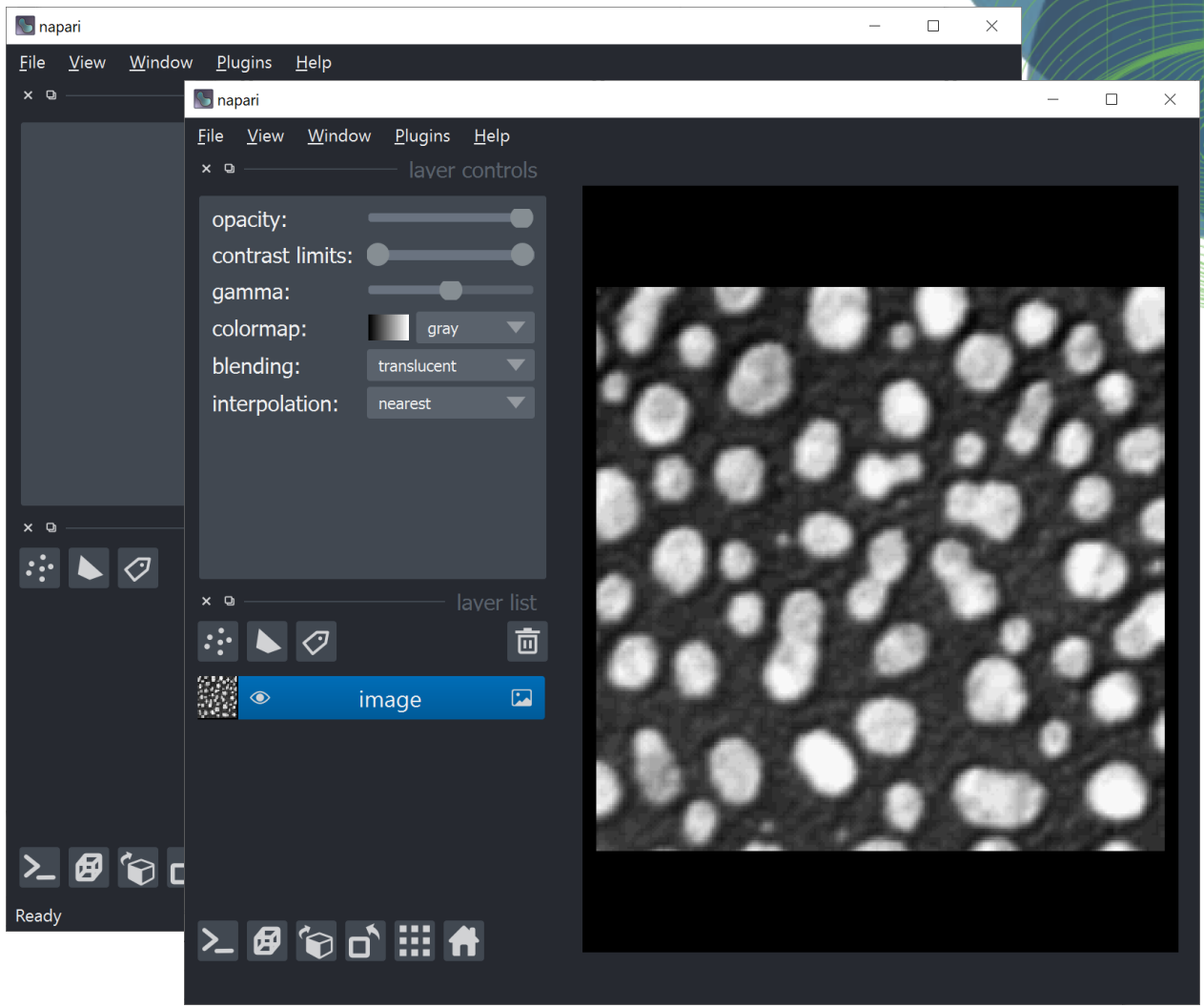
Napari – Python Scripting

- Initialization

```
import napari

# Create an empty viewer
viewer = napari.Viewer()
```

- Adding images

```
viewer.add_image(image)
```

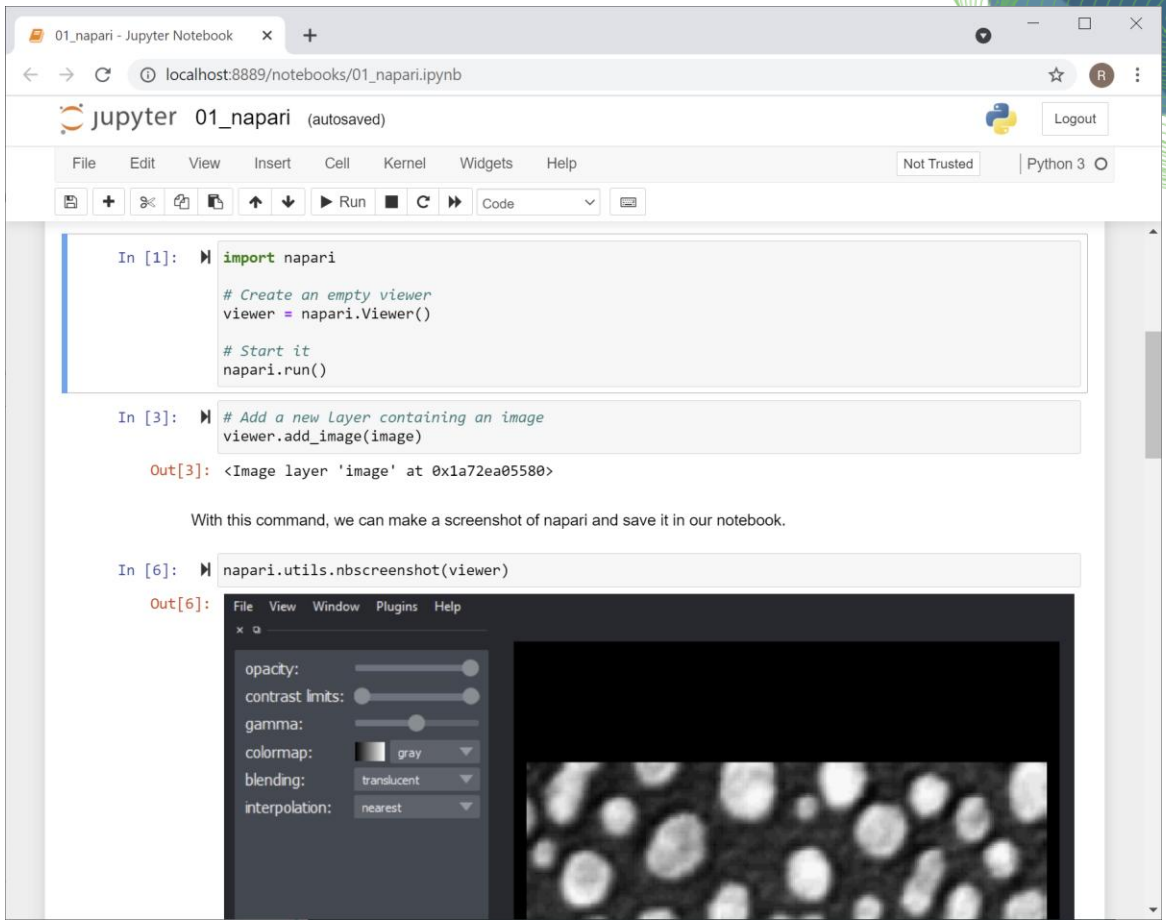


Napari – Python Scripting

- Make screenshots from napari and put them in your jupter notebook

```
napari.utils.nbscreenshot(viewer)
```

Place your viewer here



Napari – Python Scripting

- Removing layers

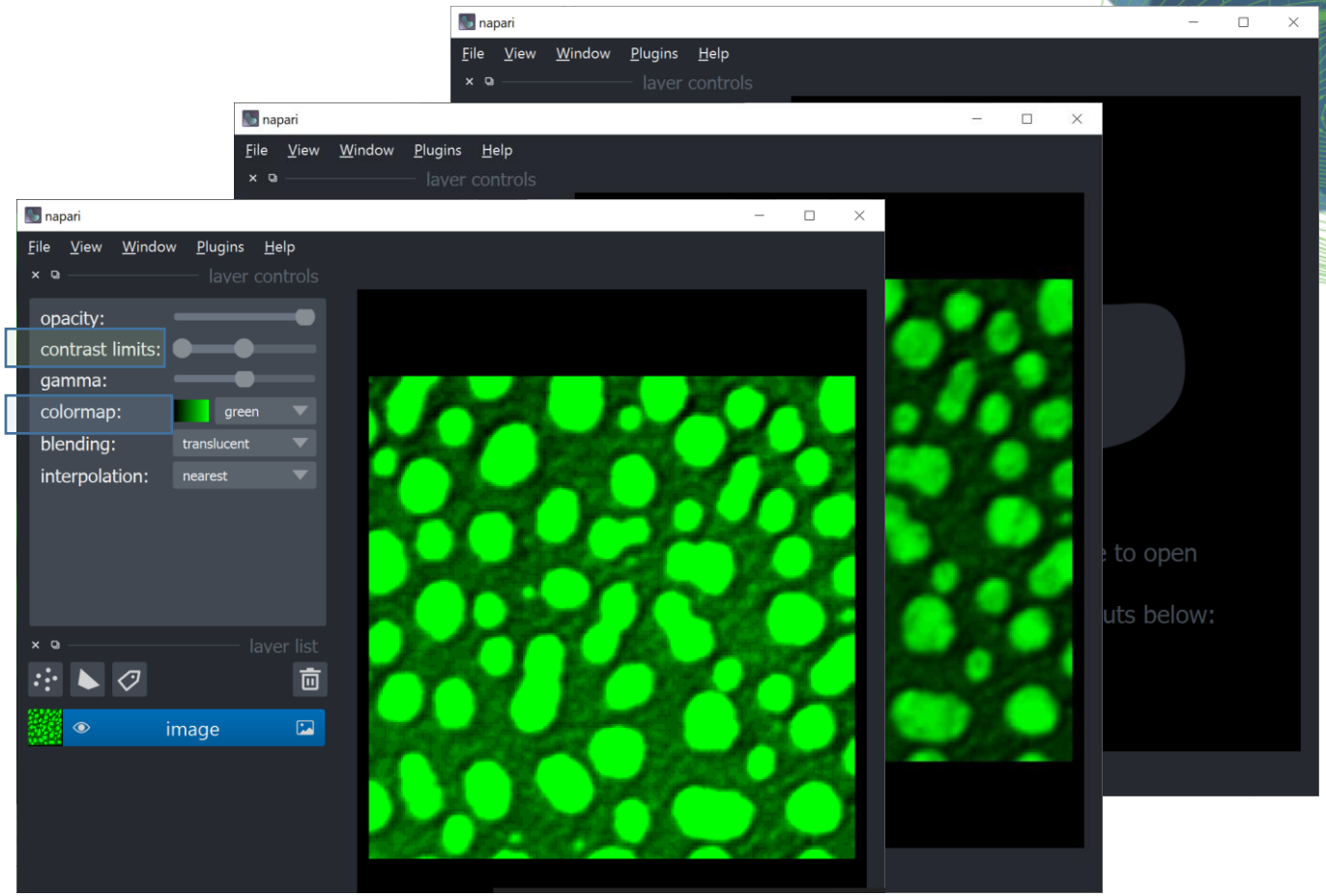
```
for l in viewer.layers:
    viewer.layers.remove(l)
```

- Modify visualization while adding layers

```
viewer.add_image(image,
                 colormap='green’)
```

- Modify layers after adding

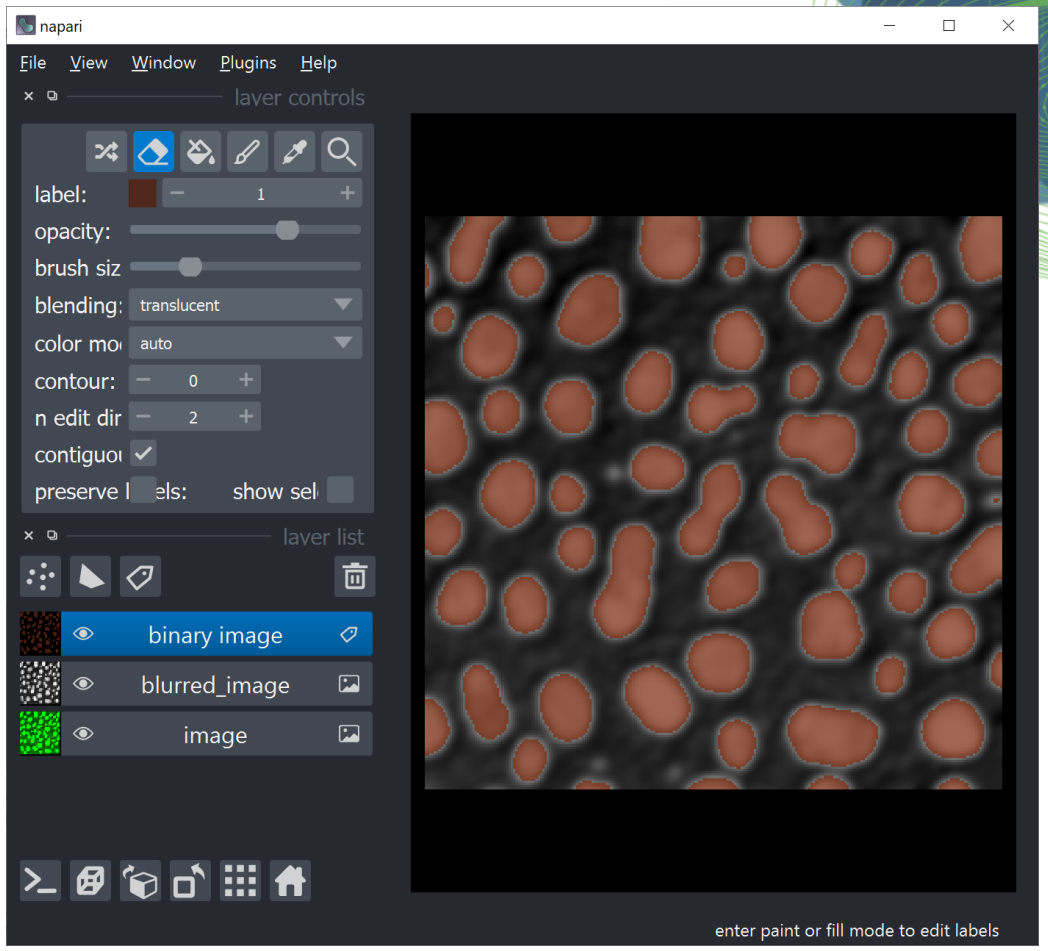
```
layer = viewer.add_image(image)
layer.colormap = 'green’
layer.contrast_limits = (0, 128)
```



Napari – Python Scripting

- Binary images and label images visualized as label layers

```
from skimage.filters import threshold_otsu
threshold = threshold_otsu(blurred_image)
binary_image = blurred_image > threshold

# Add a new labels layer containing an image
viewer.add_labels(binary_image)
```



The Napari Assistant

- Tools > Utilities > Assistant (na)

Viewer controls

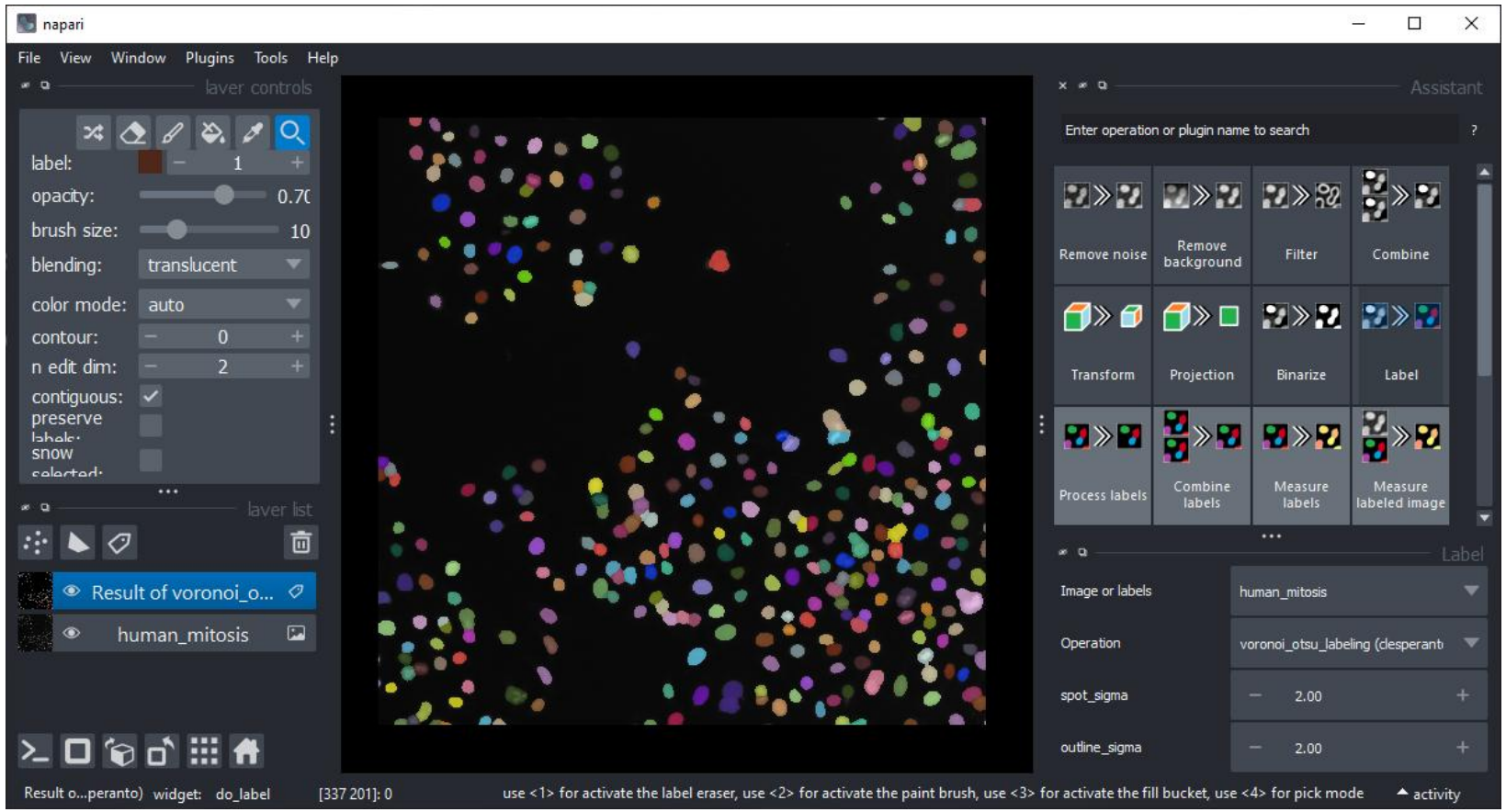
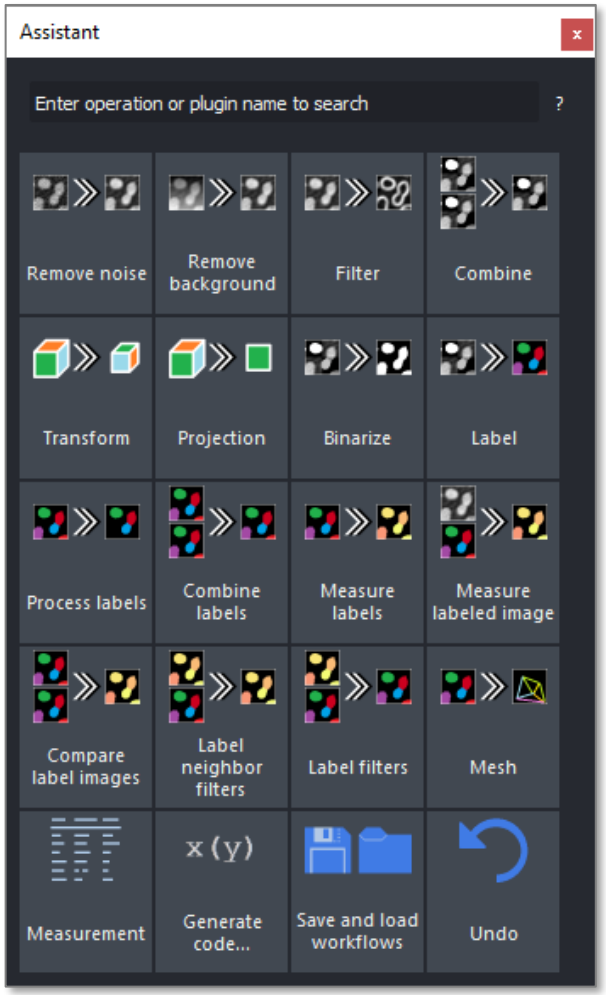
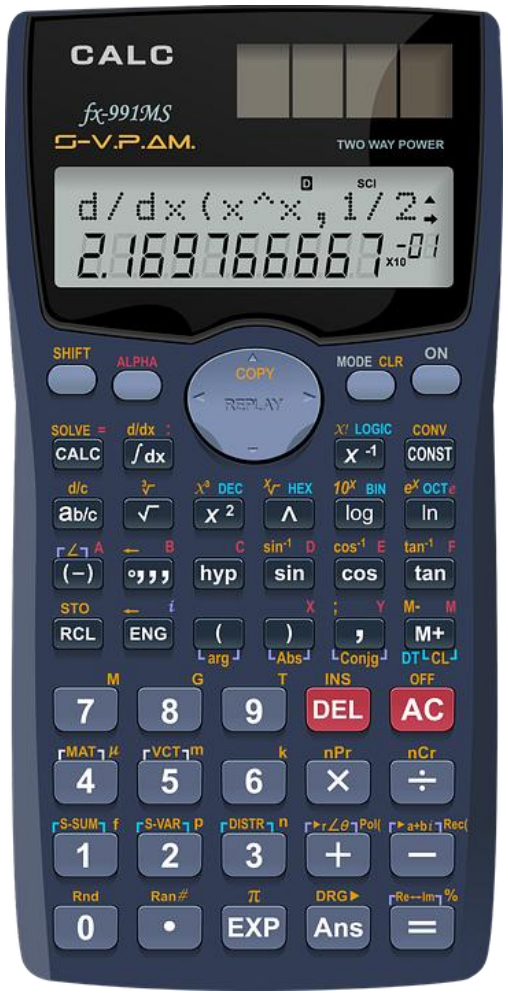


Image Processing

The Napari Assistant

- A pocket-calculator-like interface to build image analysis workflows

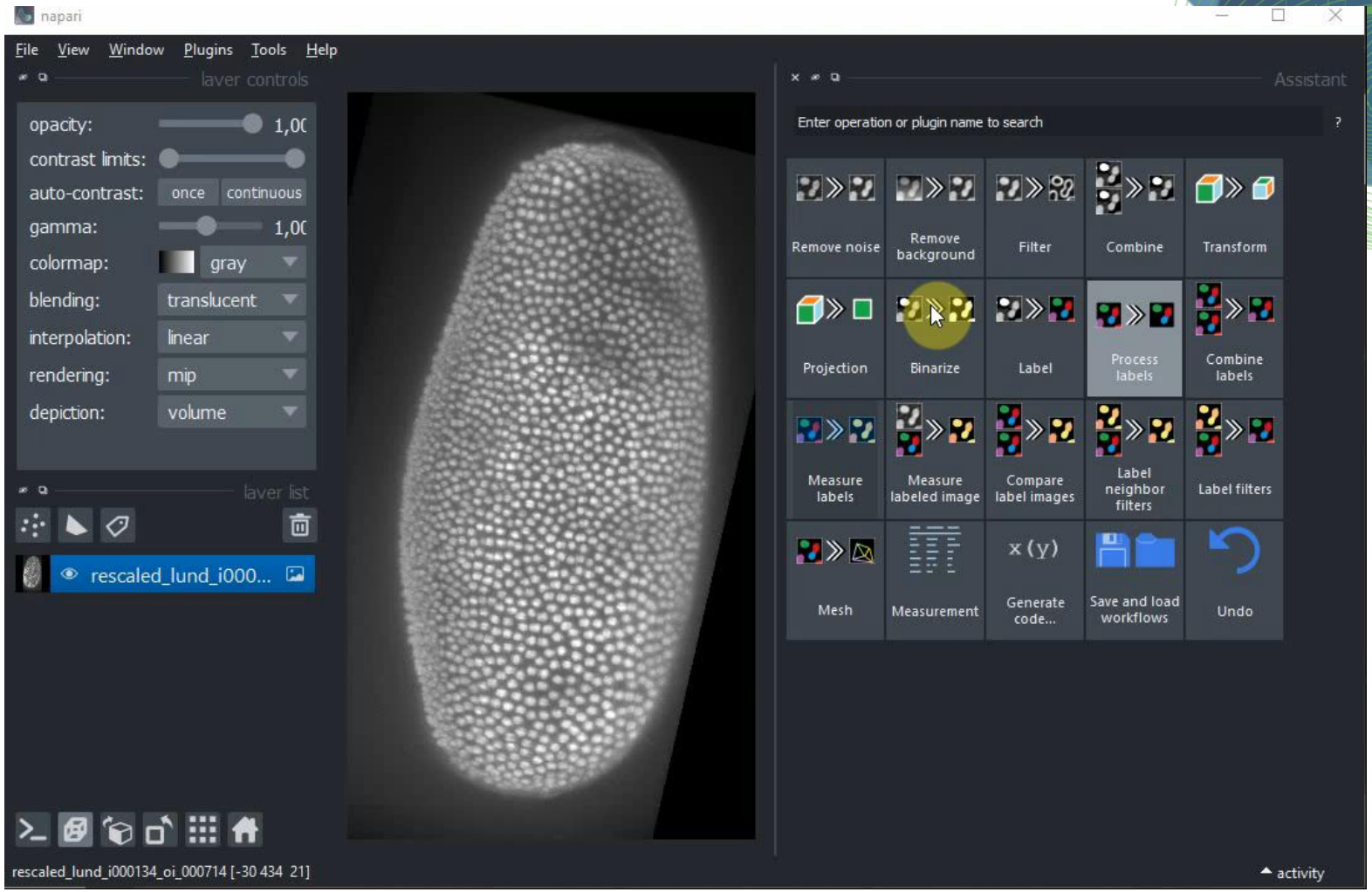




The Napari Assistant

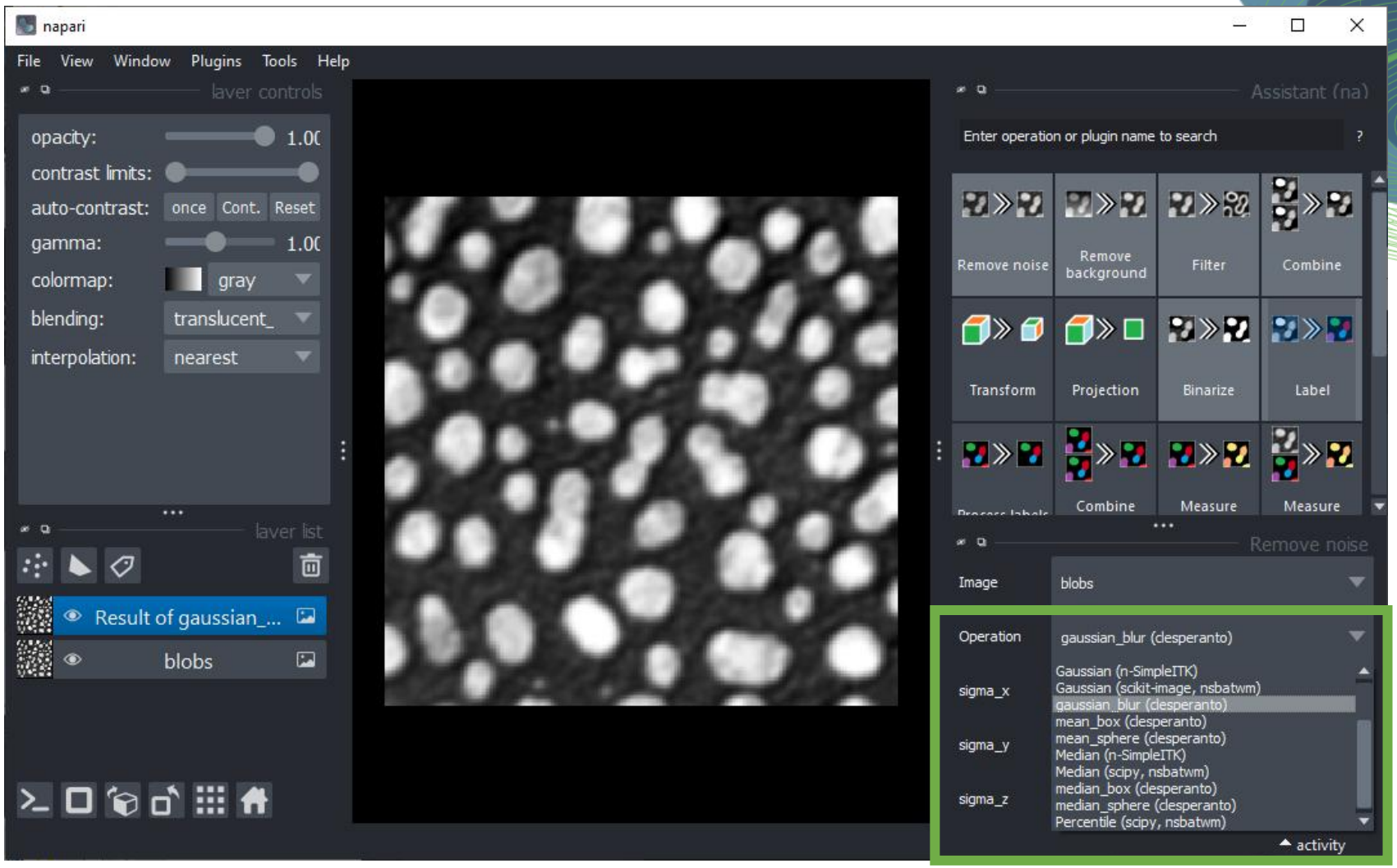
- Classical image processing operations + advanced tools
- Saving&loading supported
- Undo [redo]
- Hints for next steps
- …

Ryan Savill ()
@RyanSavill4



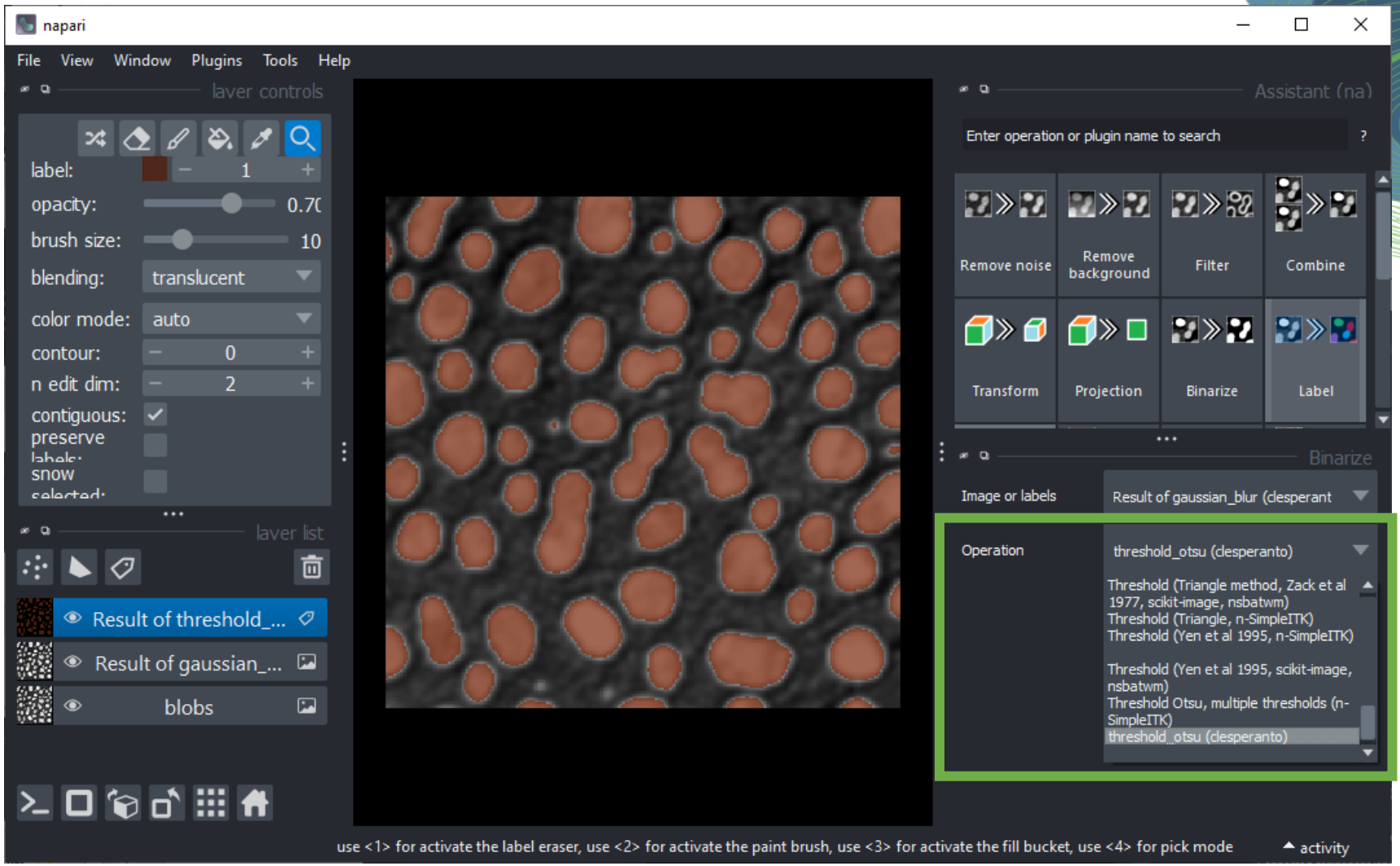
Workflow building

- Try different algorithms, e.g. for removing noise
- Find them in the pulldown



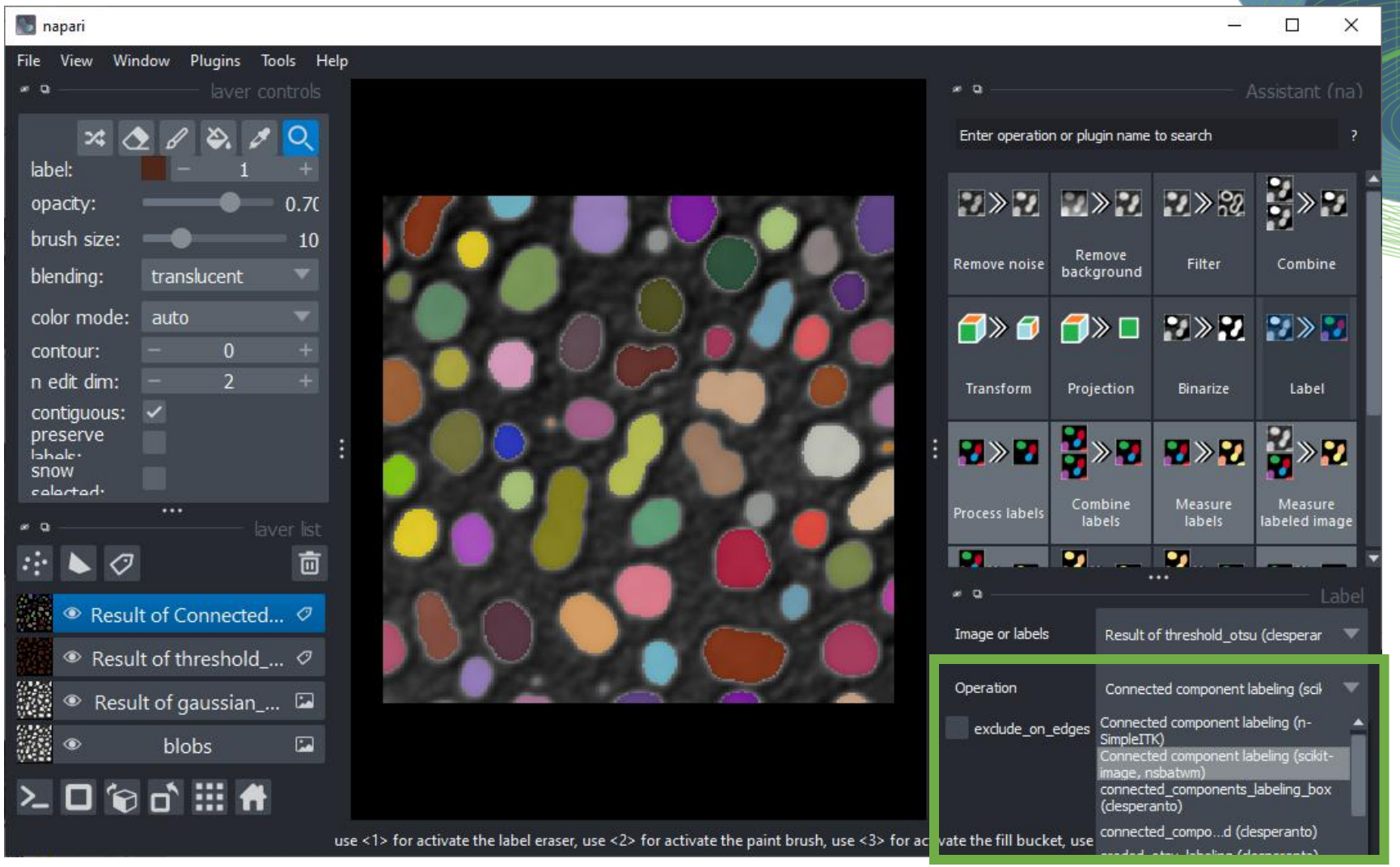
Workflow building

- Try different binarization algorithms



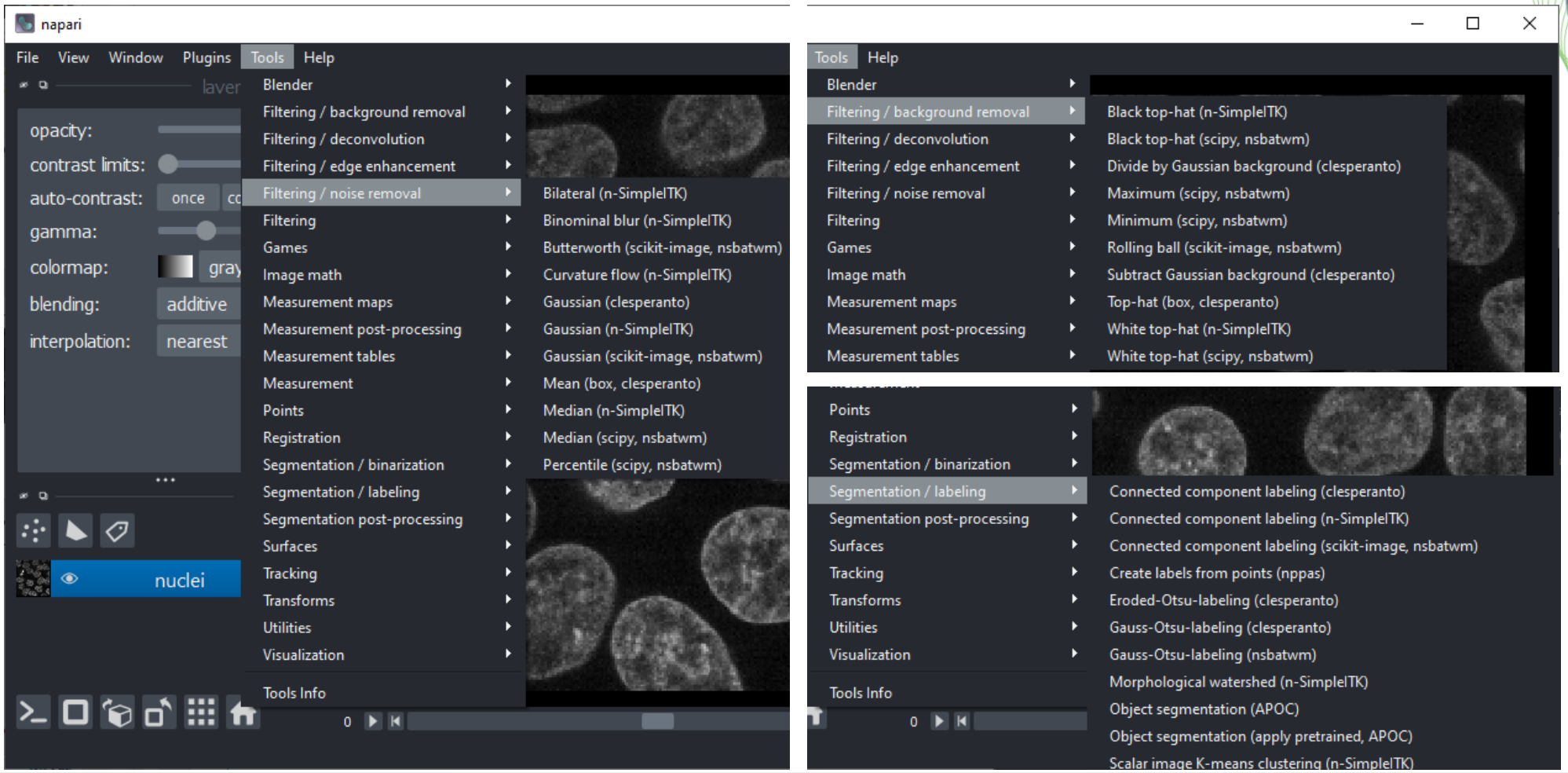
Workflow building

- Multiple labeling algorithms available



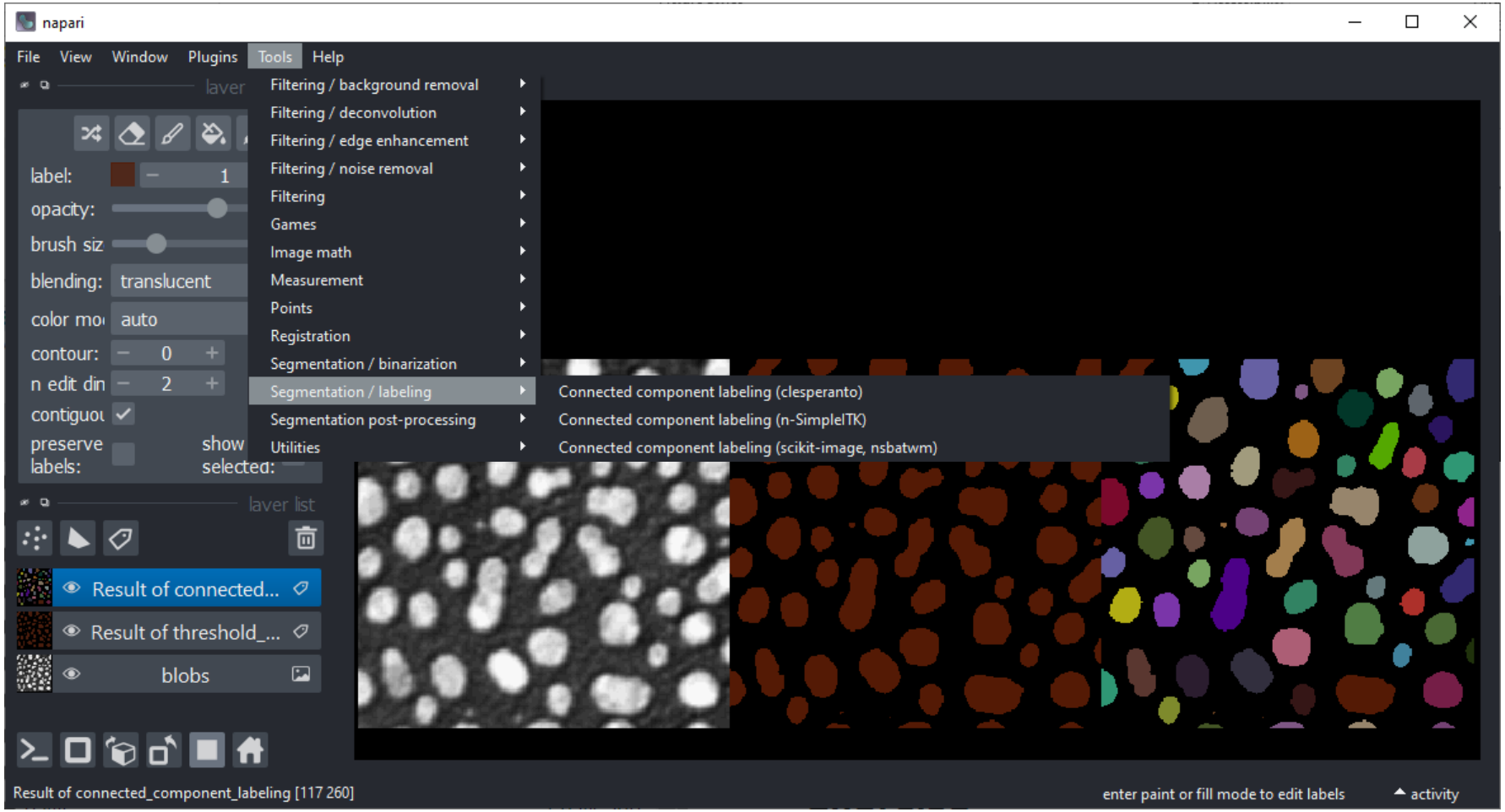
The Tools menu

- Organized in categories



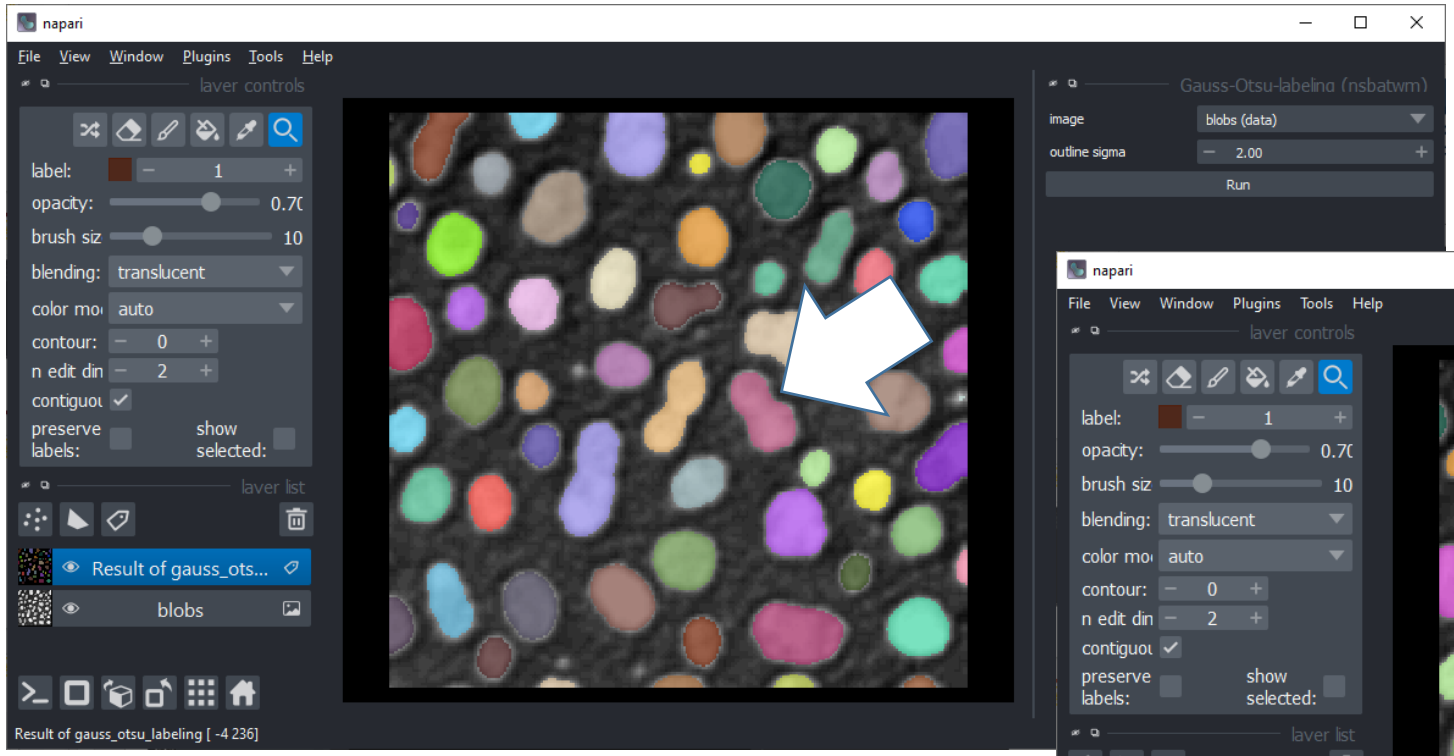
Workflow building

Also check out the Tools > Segmentation / labeling menu



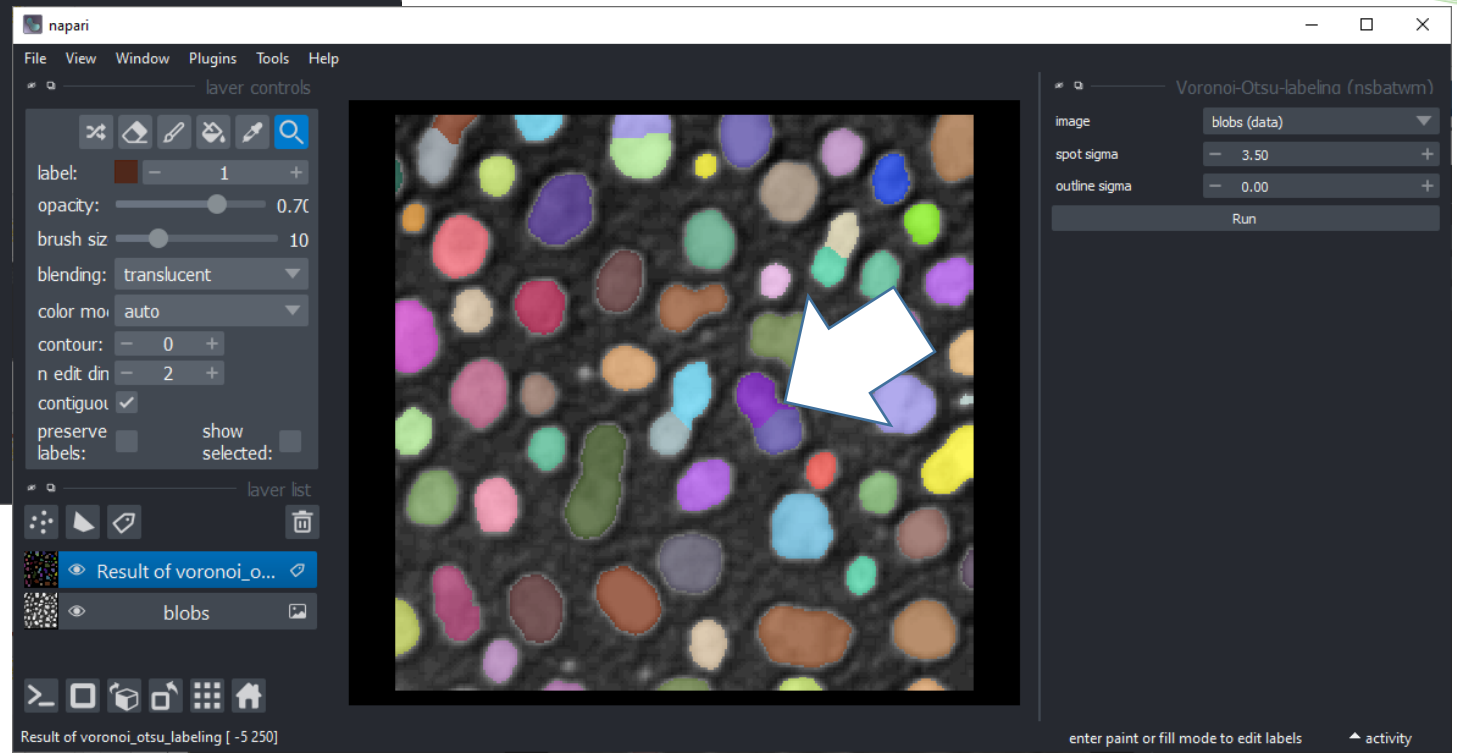
Short-cuts: Voronoi-Otsu-Labeling

Also check out the Tools > Segmentation / labeling menu



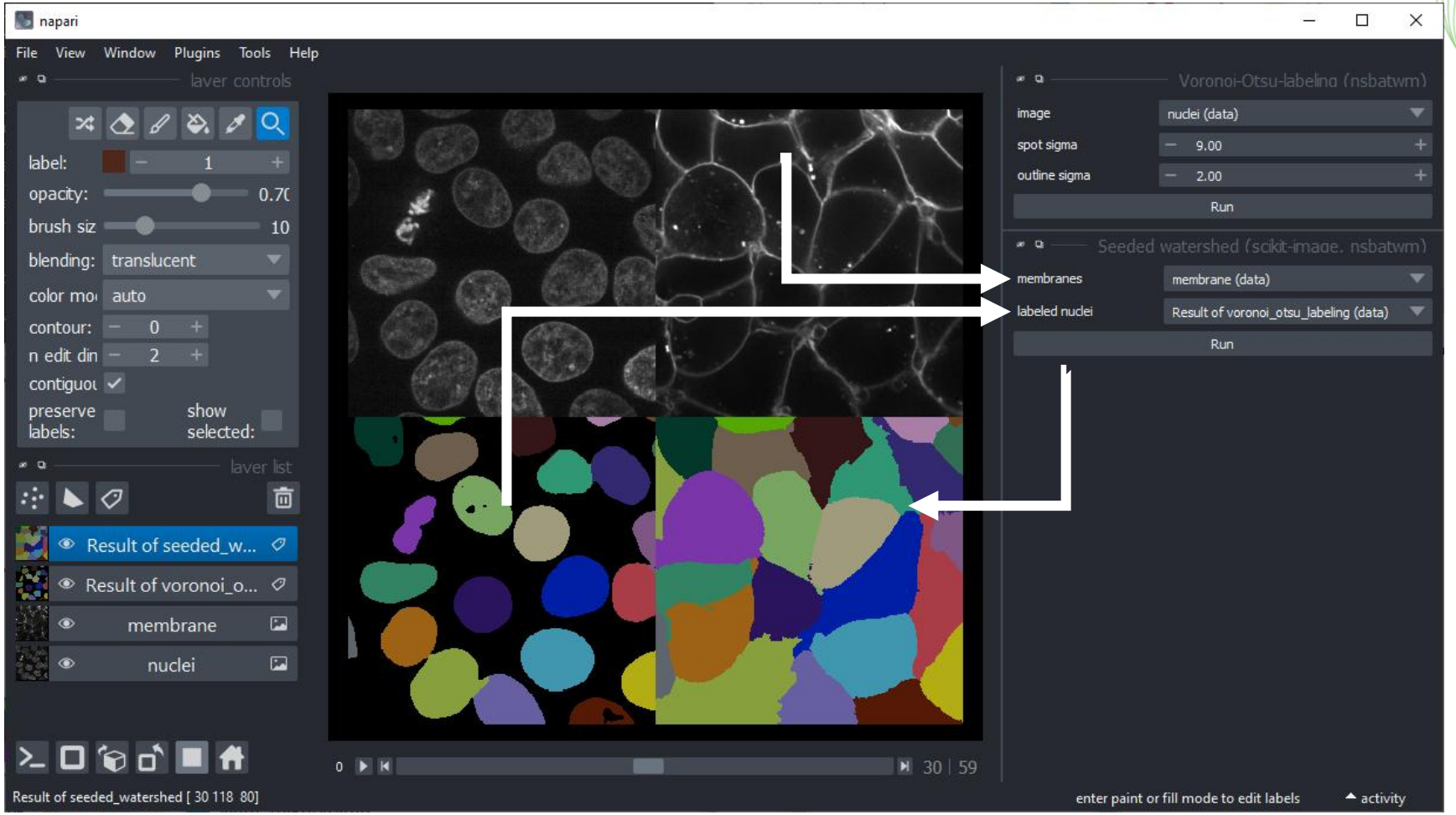
Gauss-Otsu-Labeling

Voronoi-Otsu-Labeling



Watershed

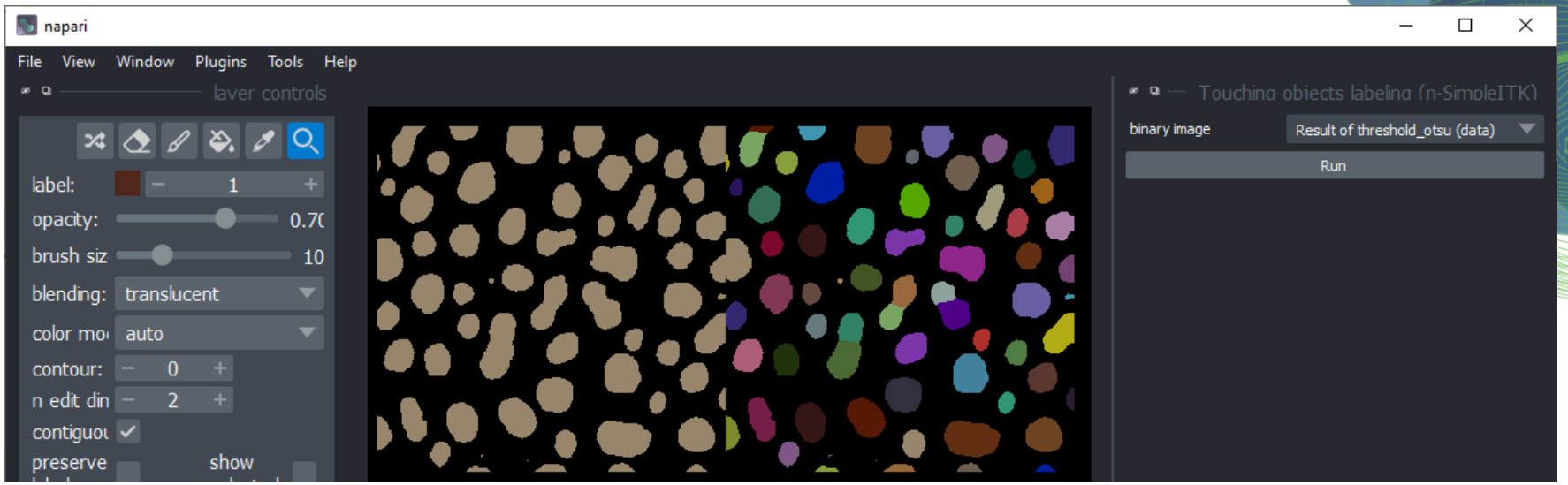
Also check out the Tools > Segmentation / labeling menu



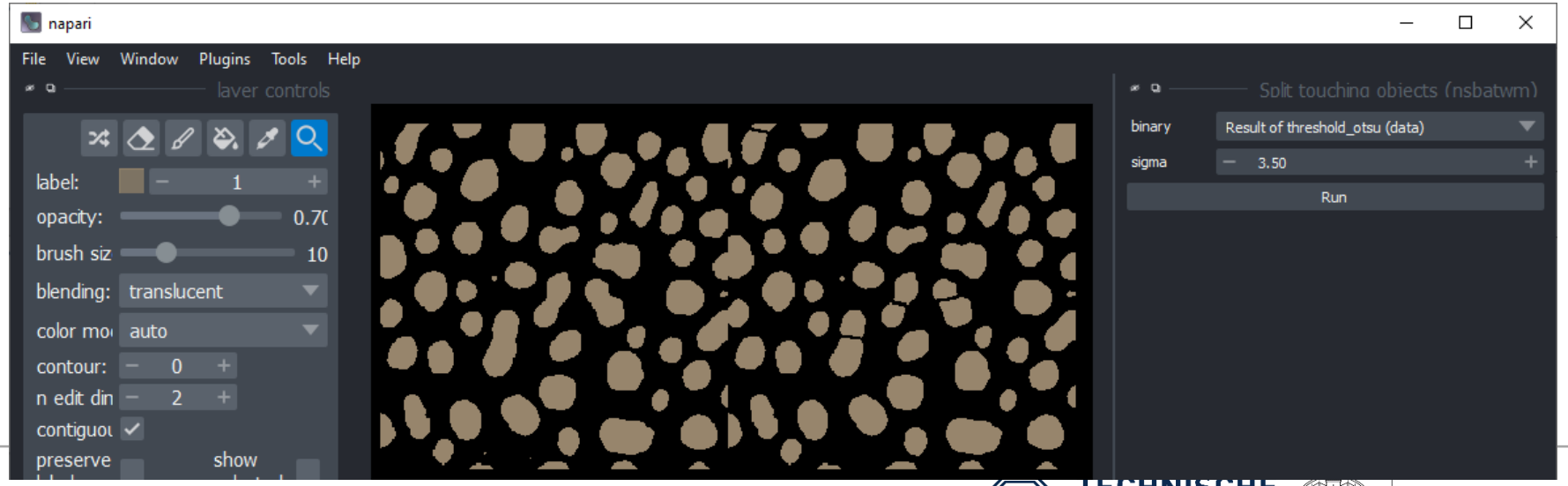
Watershed

- From binary images

Tools > Segmentation / labeling >
Label touching objects

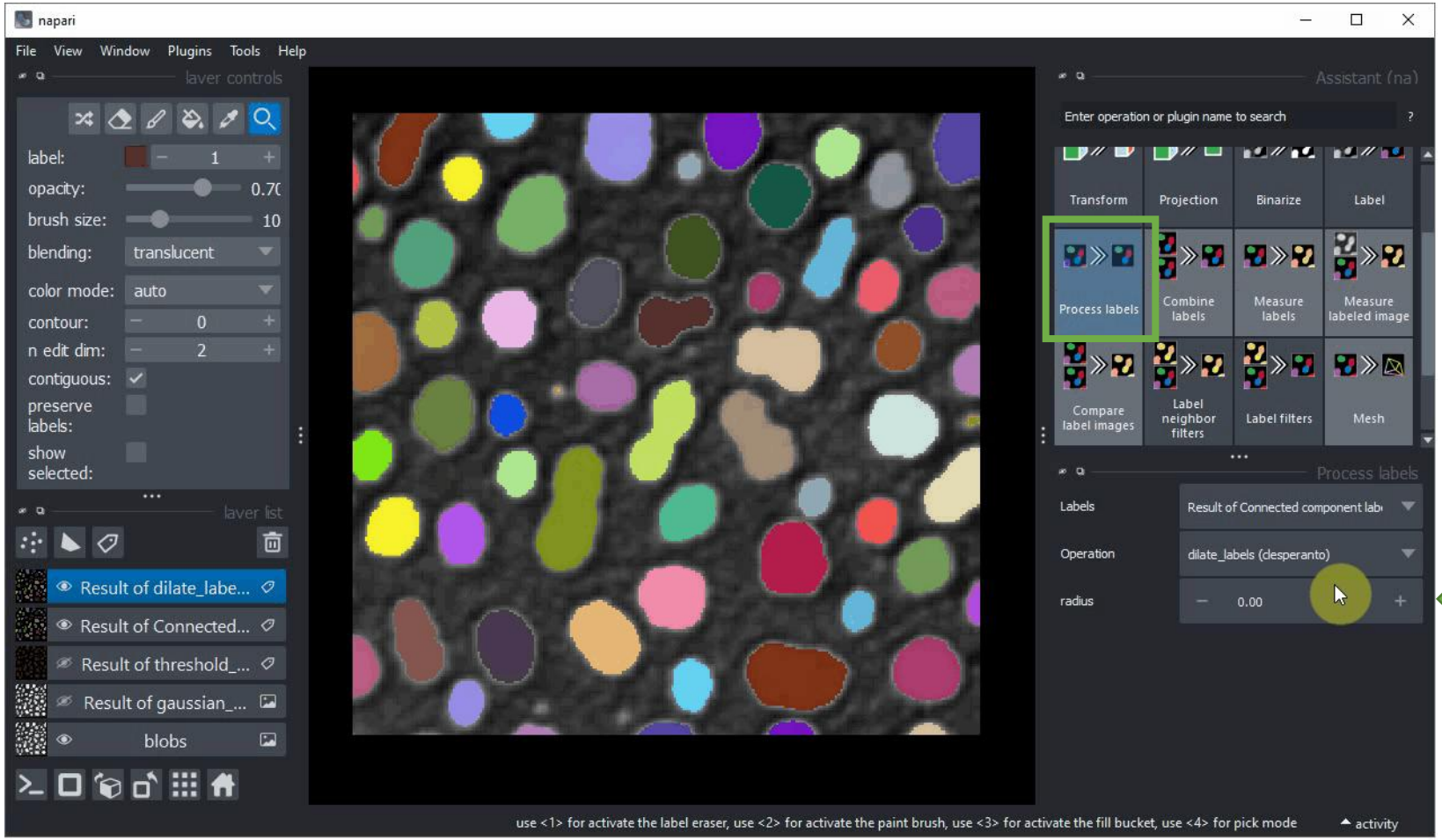


Tools > Segmentation post-processing >
Split touching objects
(Similar to ImageJ's Watershed)



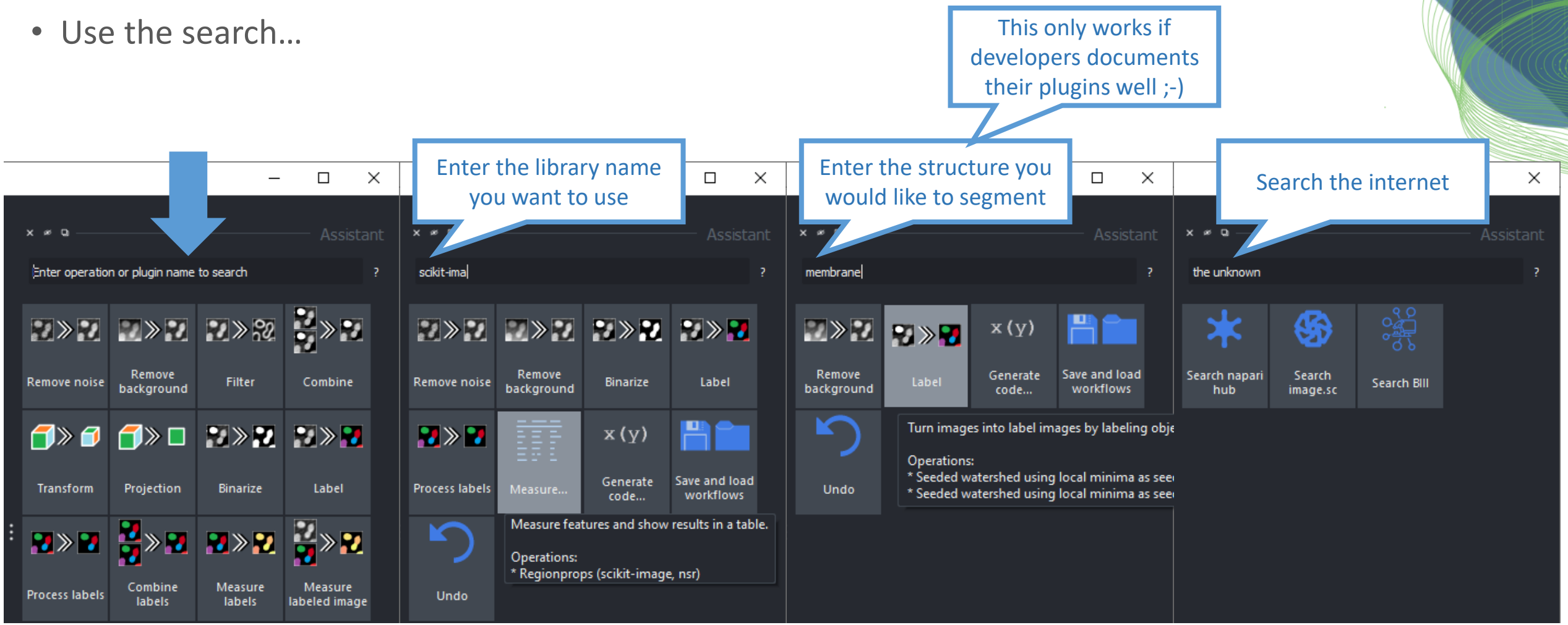
Label erosion, dilation, opening, closing, …

- In Napari Assistant: Process labels

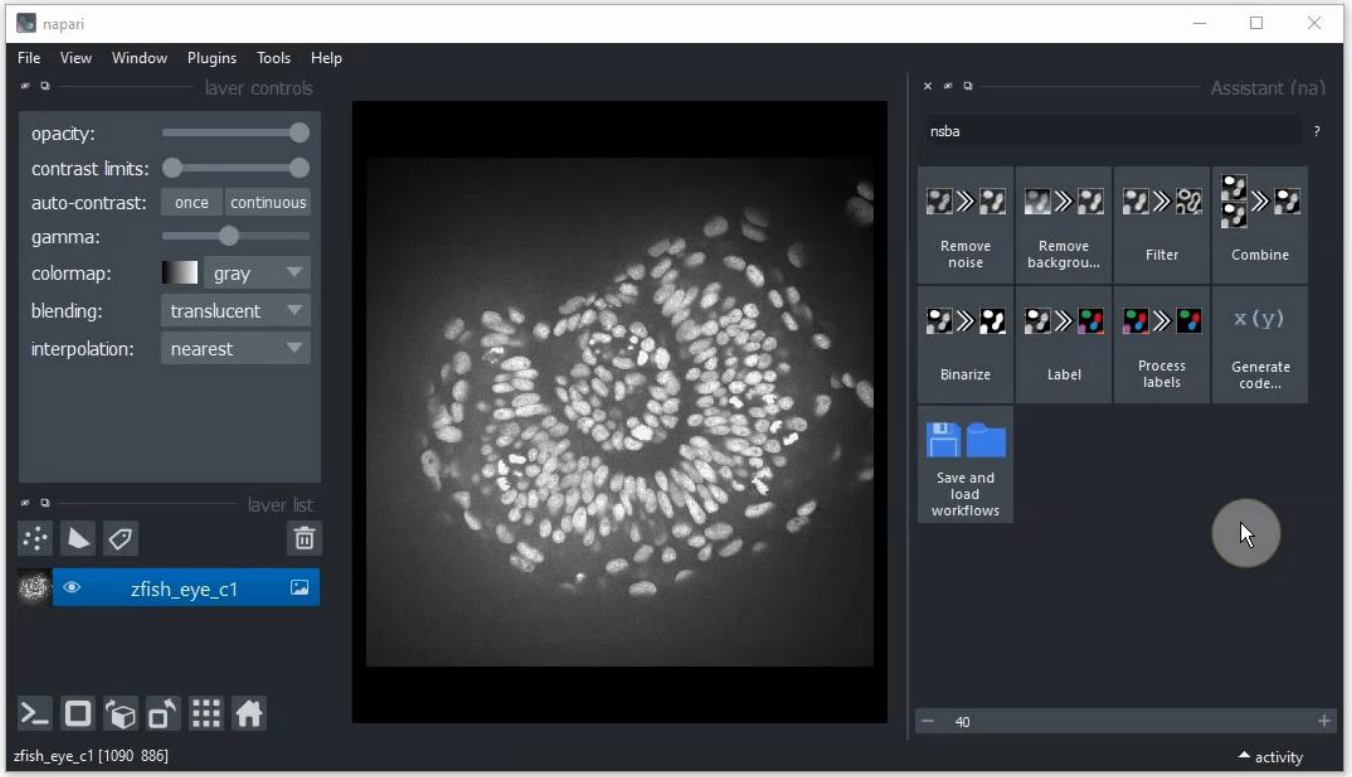


Browse operations

- Use the search…



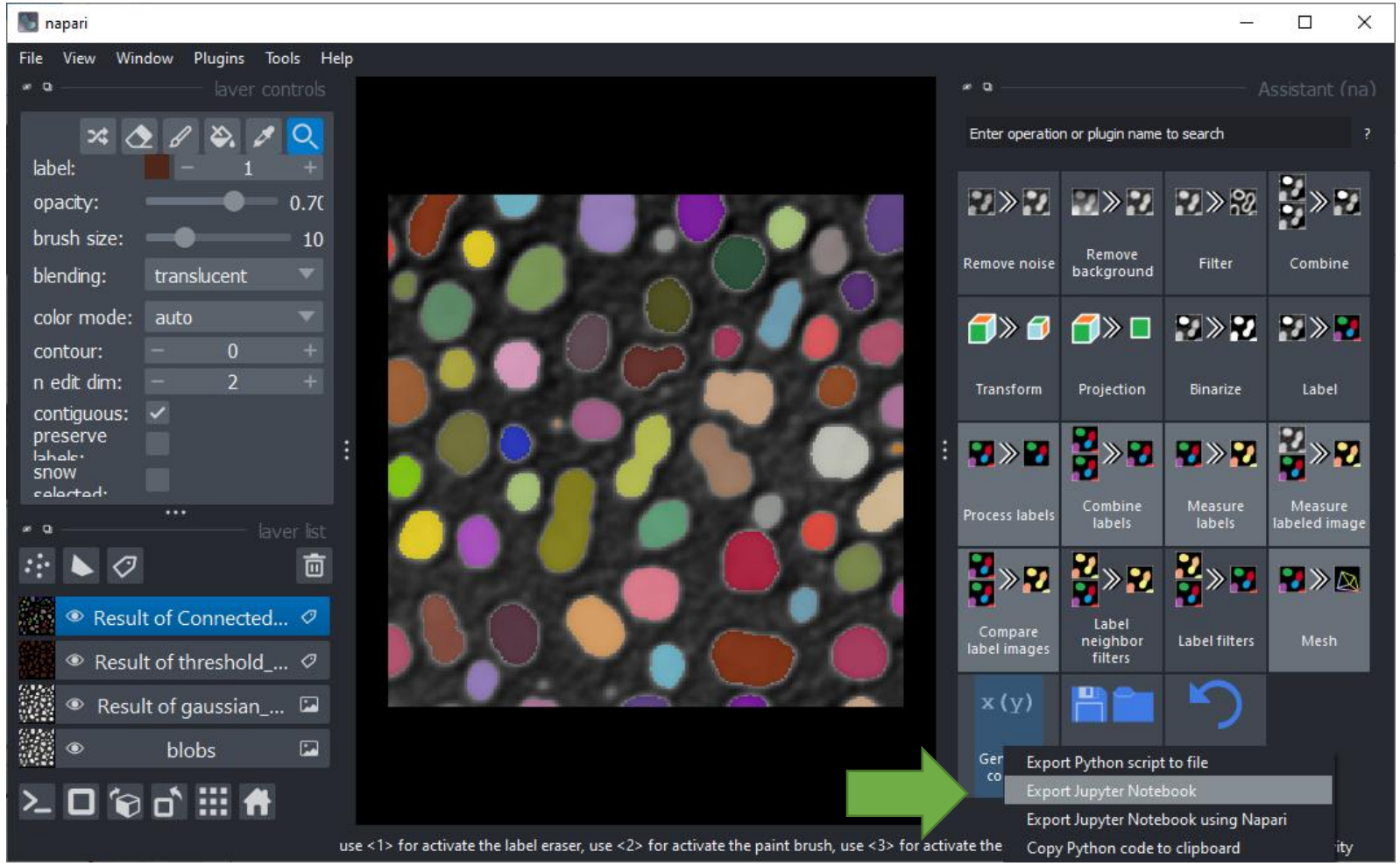
Export code to Jupyter Notebooks

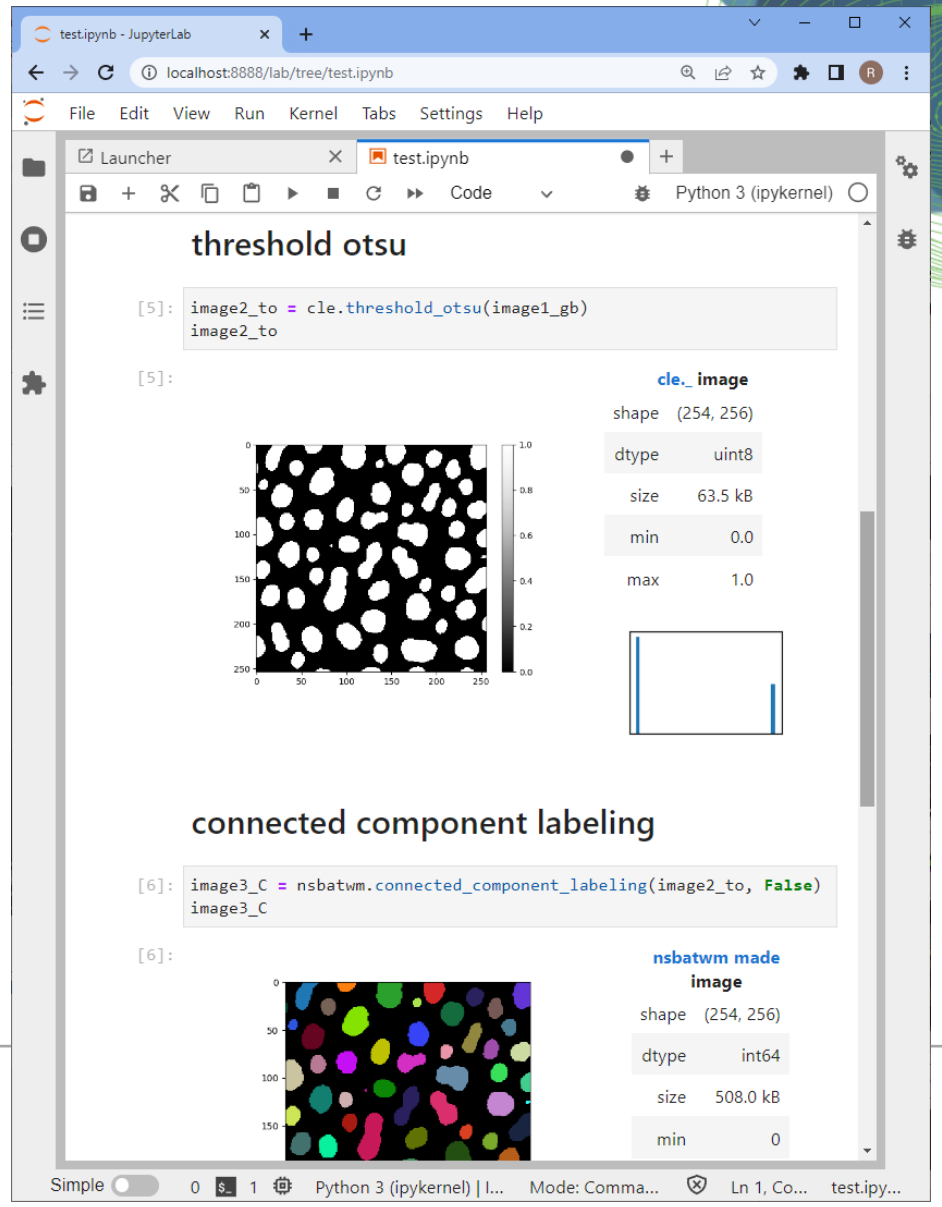


https://github.com/haesleinhuepf/napari-assistant

Image data source: Mauricio Rocha Martins, Norden lab, MPI CBG (now at IGC Oeiras)

Export code to Jupyter Notebooks





ScaDS.AI DRESDEN LEIPZIG

Robert Haase
@haesleinhuepf
BIDS Lecture 4/14
April 23rd 2024

https://github.com/haesleinhuepf/napari-assistant#code-generation

ScaDS.AI
DRESDEN LEIPZIG
CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Exercises

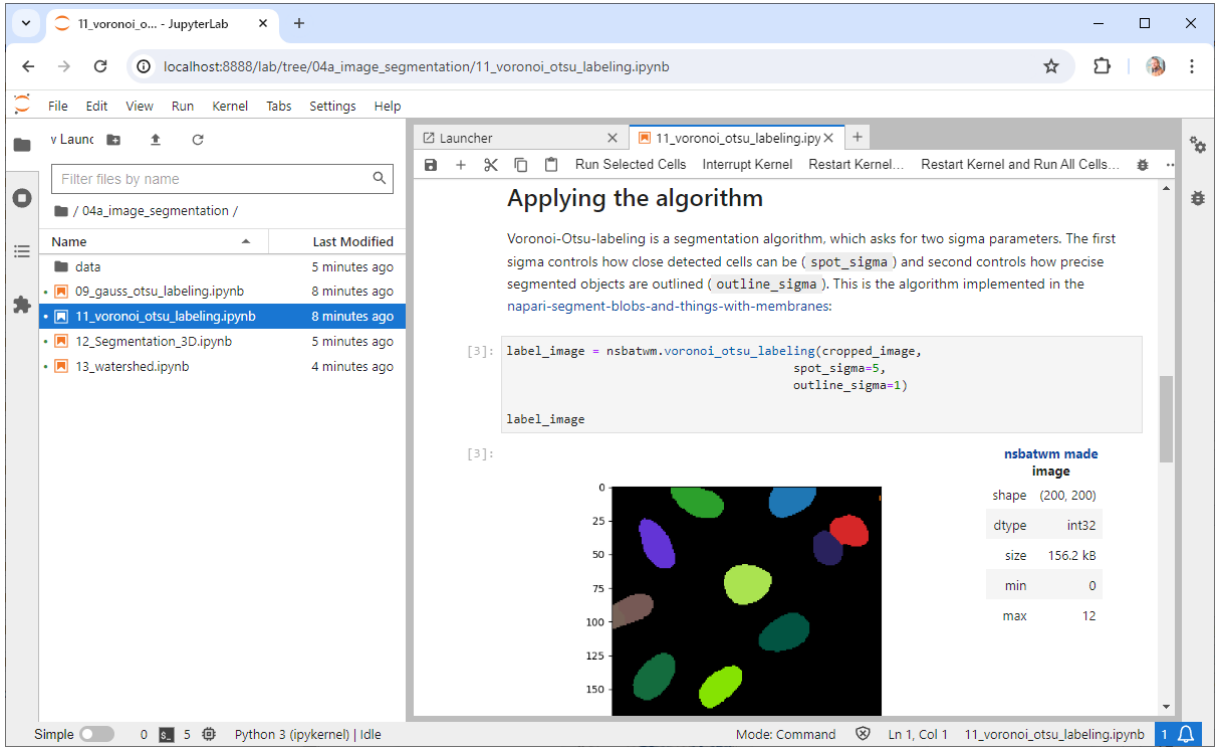
Robert Haase

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

SACHSEN
Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages.
Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Image segmentation exercises

- Try out segmentation algorithms and apply them to other datasets



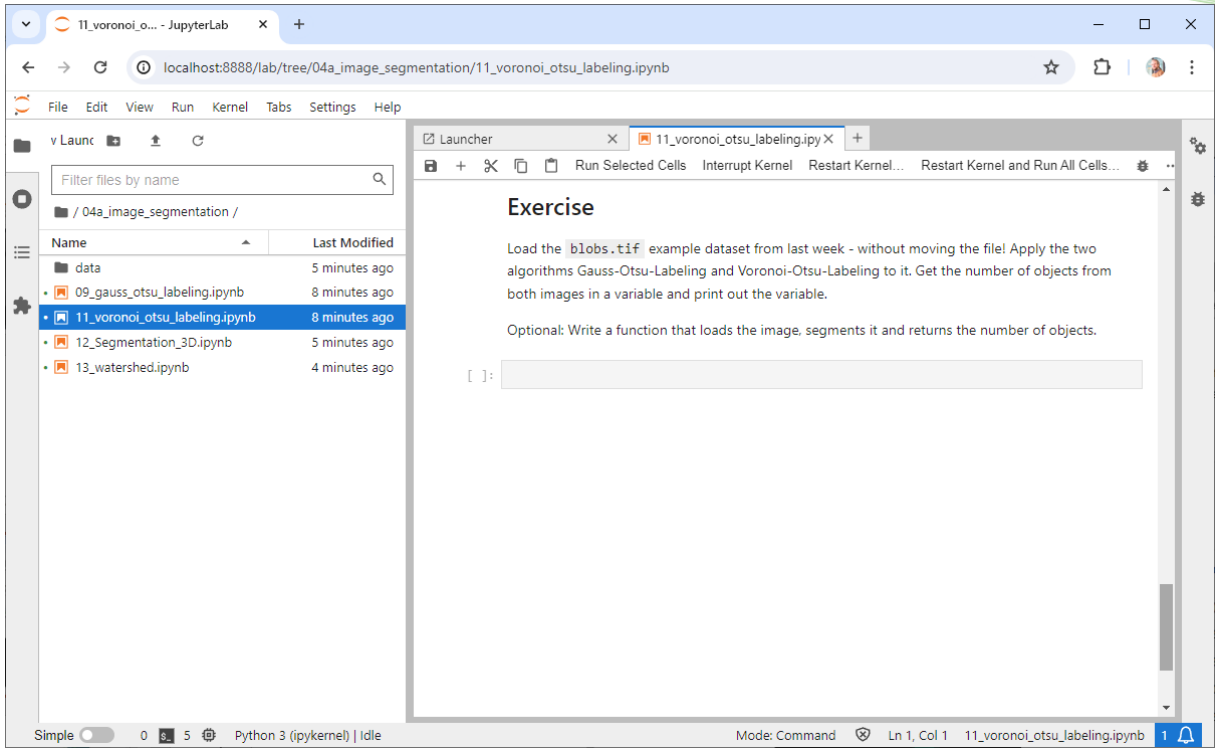
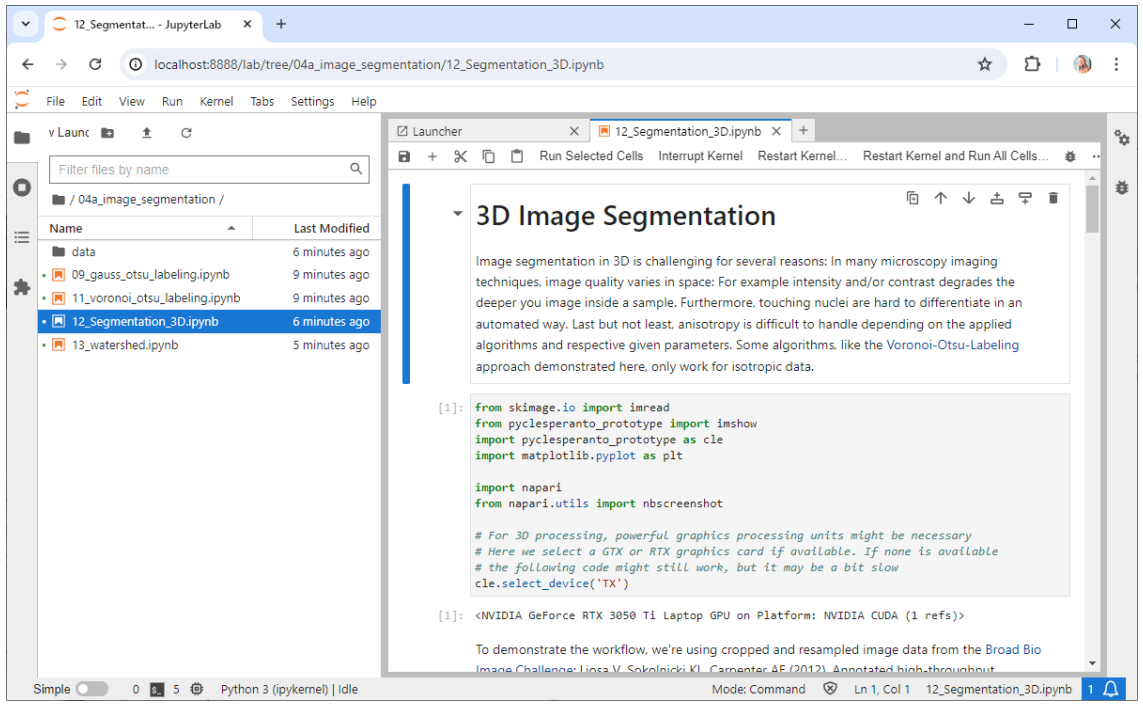
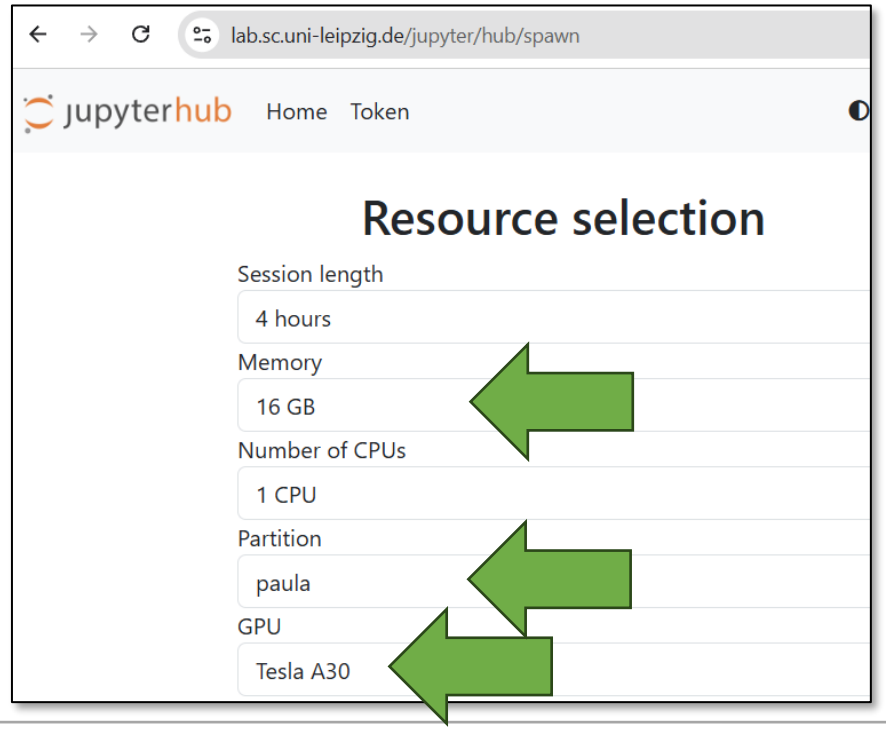


Image segmentation exercises

- 3D-segmentation using GPUs.
- Consider using UL's Jupyter Hub.

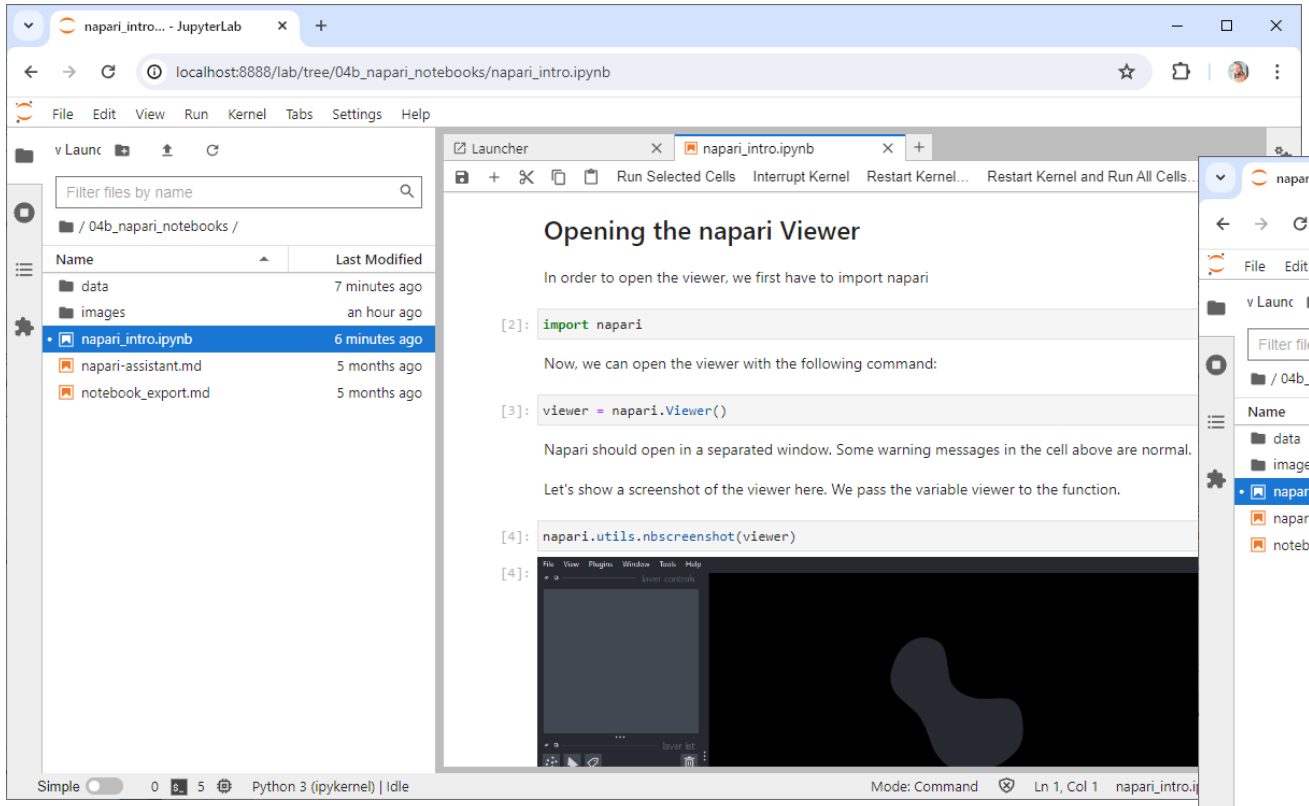


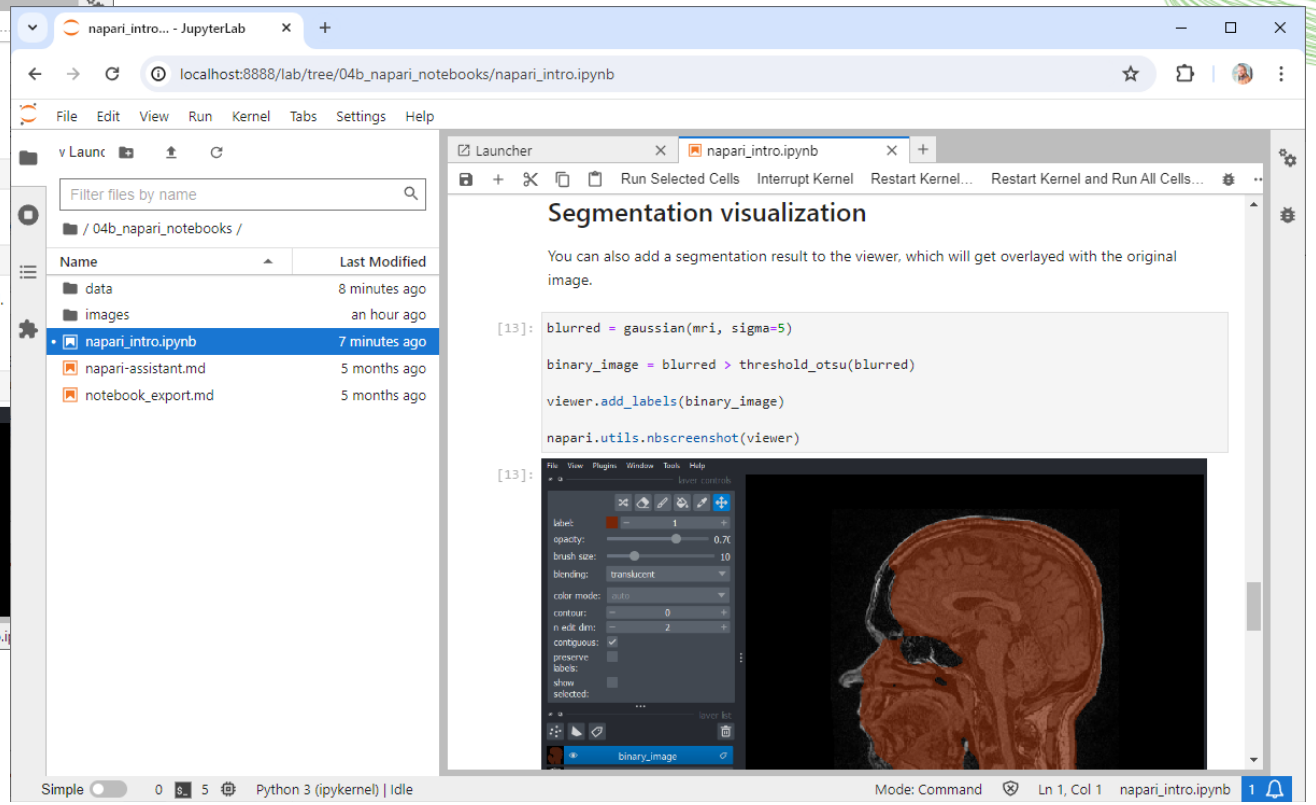


Napari - Exercises

- Start using napari from Python

Napari will not work on Jupyter Hub because it's a Window-based app.

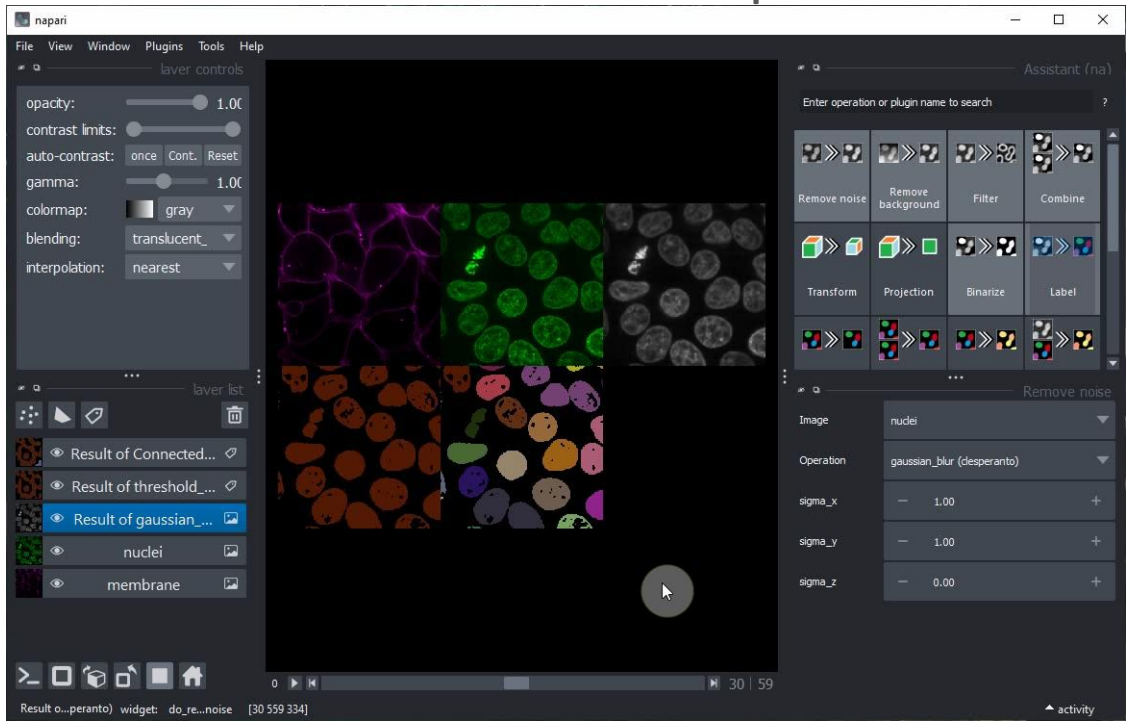


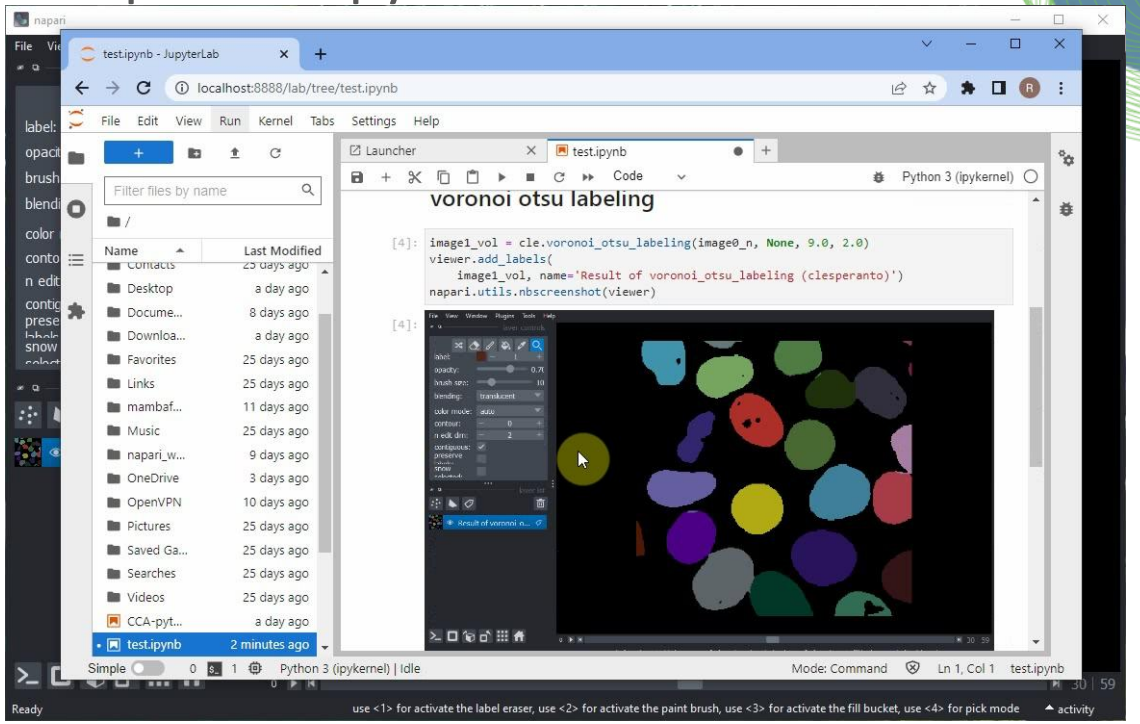


Napari - Exercises

Napari will not work on Jupyter Hub because it's a Window-based app.

- Start napari from the terminal
- Follow the instructions to set up a workflow and export a Jupyter notebook





https://github.com/ScaDS/BIDS-lecture-2025/blob/main/03b_napari_notebooks/napari-assistant.md

https://github.com/ScaDS/BIDS-lecture-2025/blob/main/03b_napari_notebooks/notebook_export.md