PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Instance segmentation

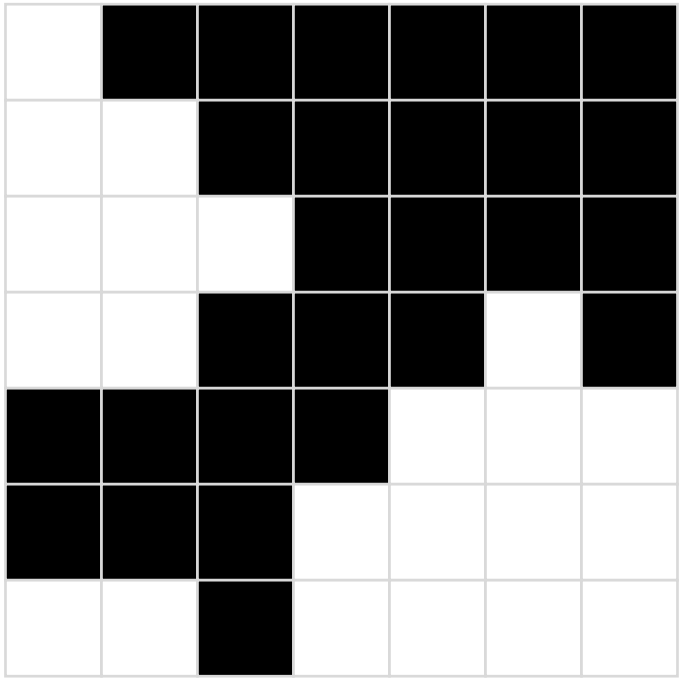
Robert Haase

December 2022

@haesleinhuepf

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 a labelled map corresponds to the number of objects.

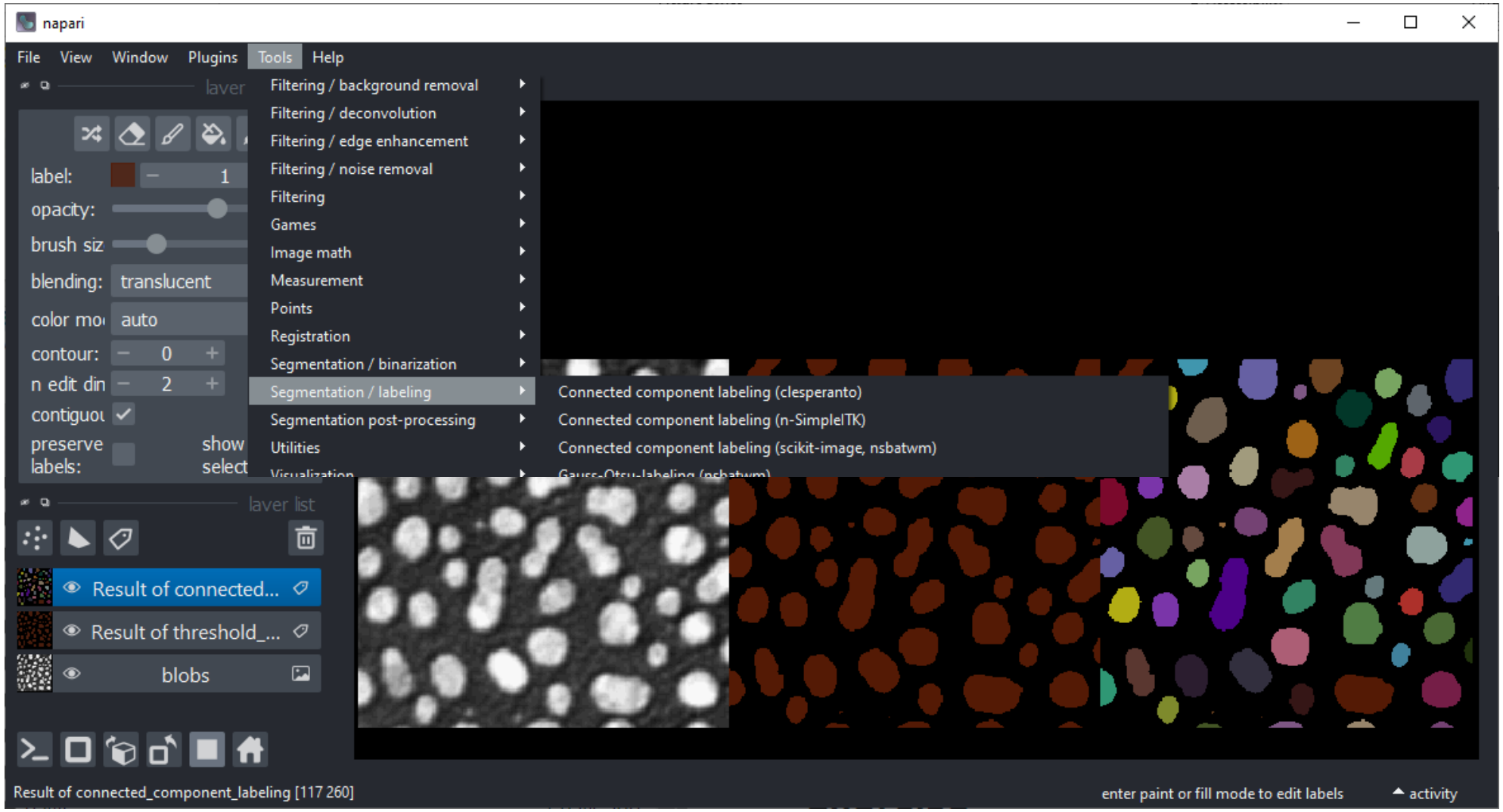


CCA

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

Connected component labelling

- In napari: Tools > Segmentation / labeling menu

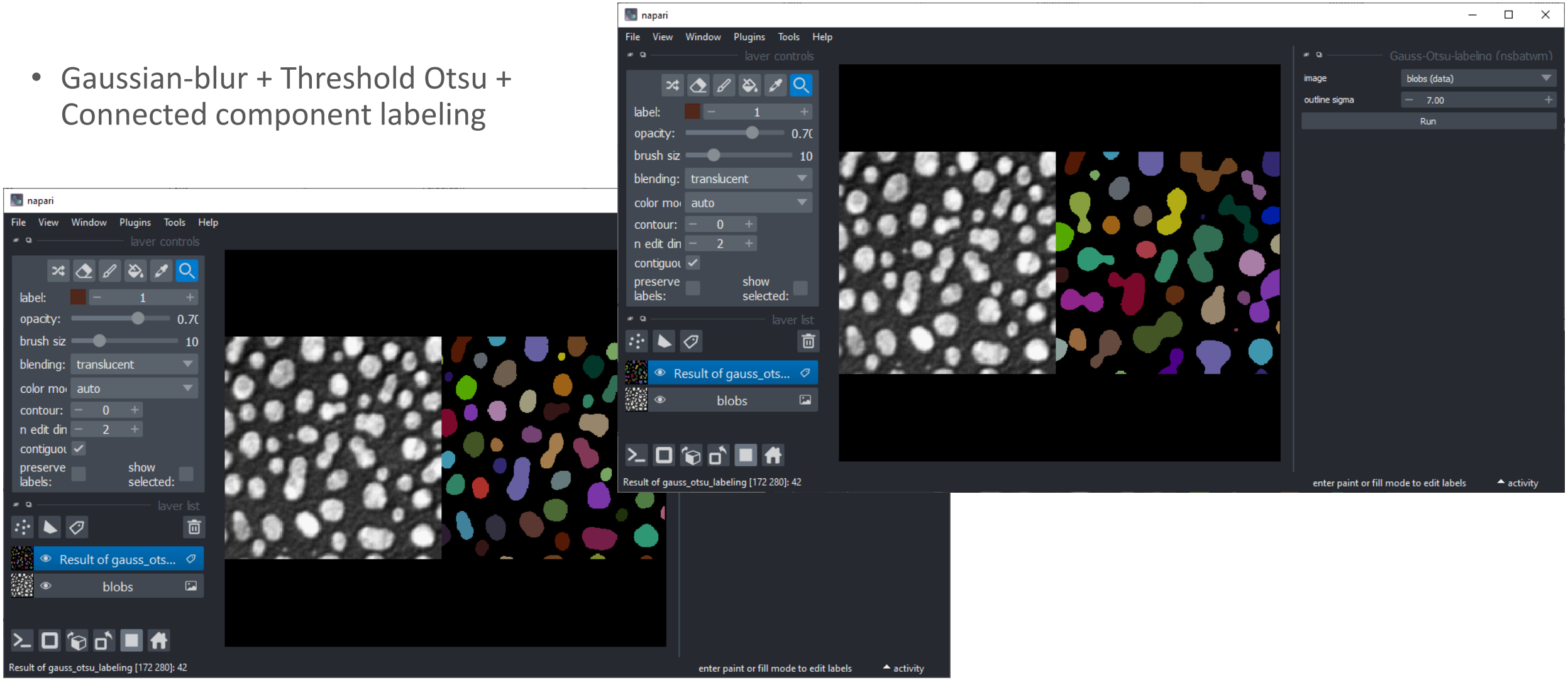


Short-cuts: Gauss-Otsu-Labeling

PoL Physics of Life TU Dresden

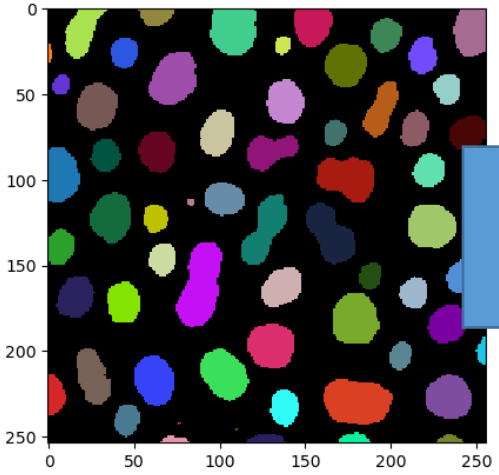
- In napari: Tools > Segmentation / labeling menu

- Gaussian-blur + Threshold Otsu + Connected component labeling



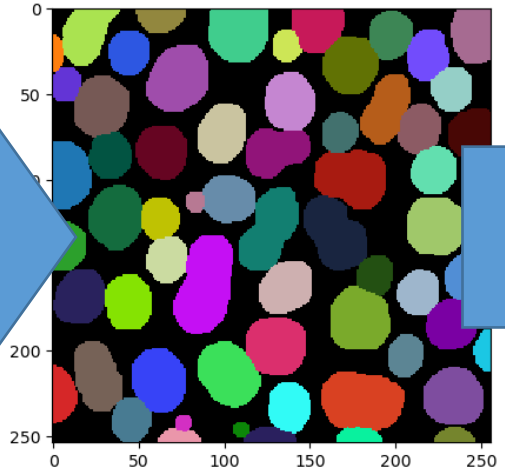
Voronoi-Tesselation

- In napari-menu: Tools > Segmentation post-processing

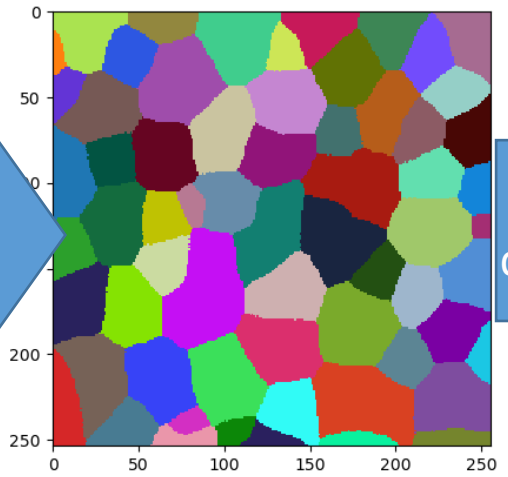


Label-image

Dilate labels

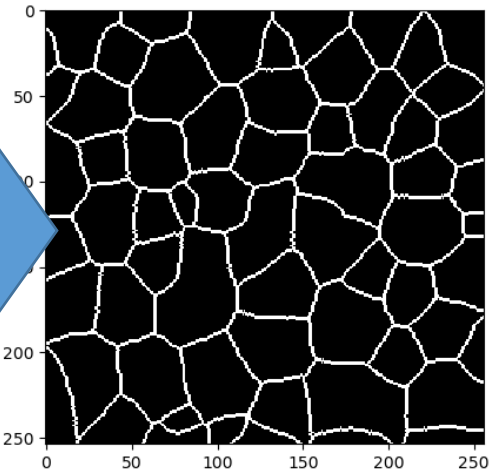


Dilate labels



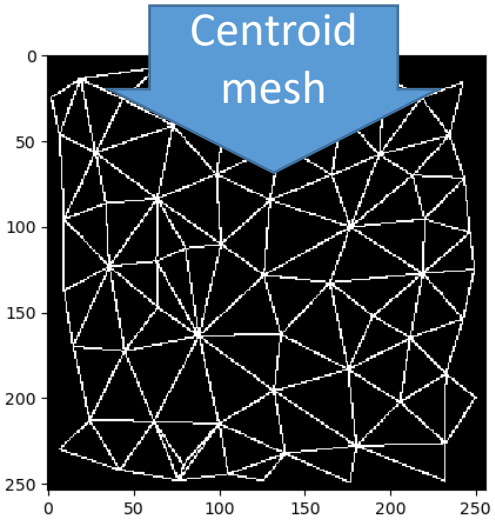
Voronoi-label-image

Edge detection

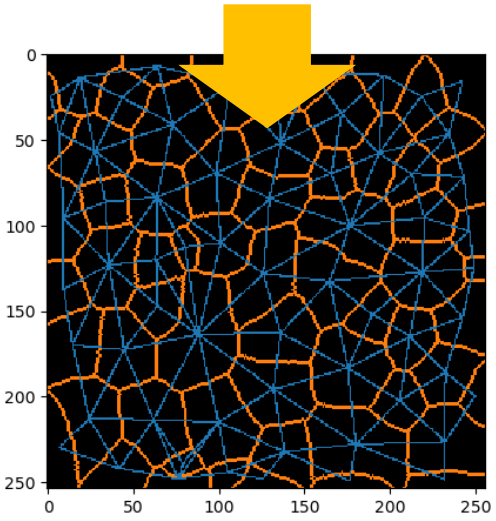


Label-edge-image

Centroid mesh



Touching neighbor mesh

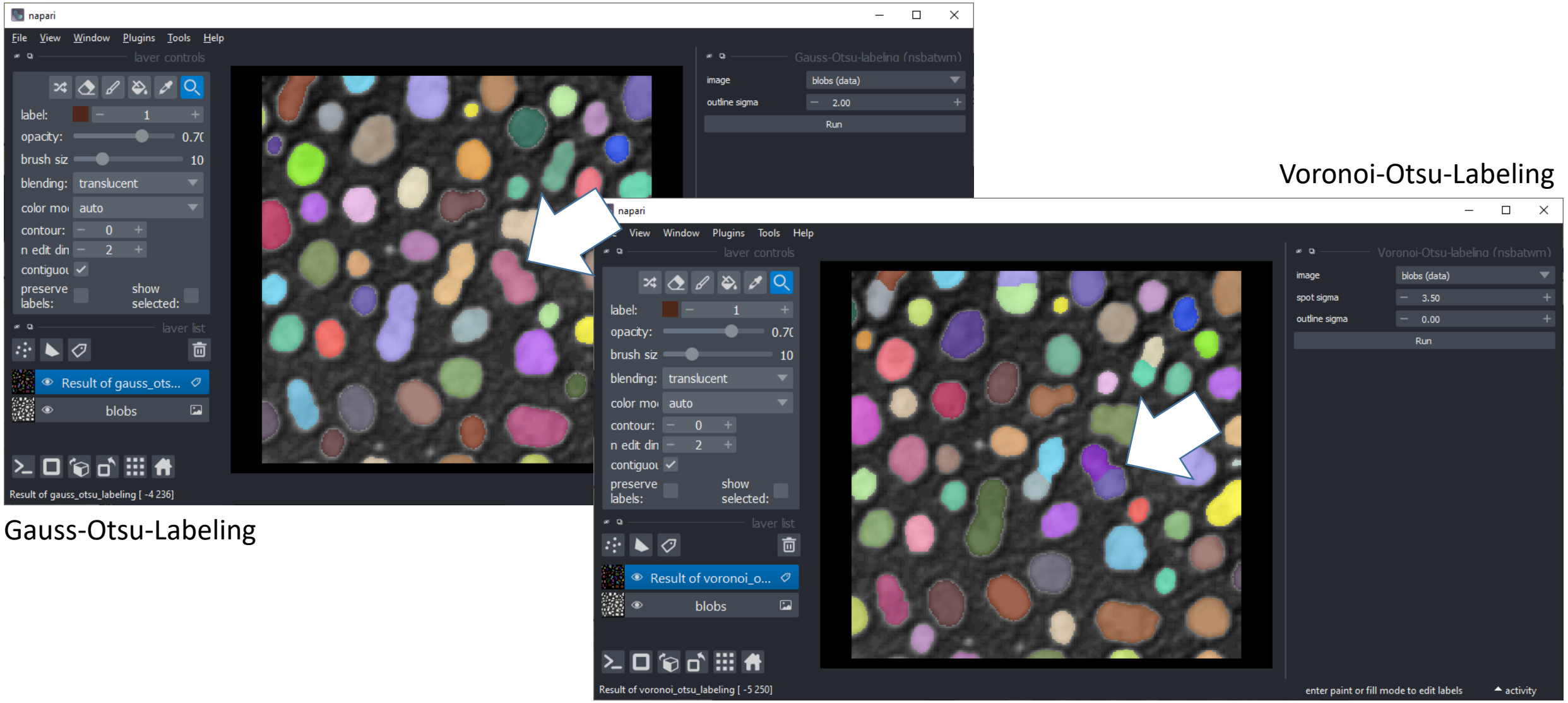


Voronoi-Tesselation

Delauney-Triangulation

Short-cuts: Voronoi-Otsu-Labeling

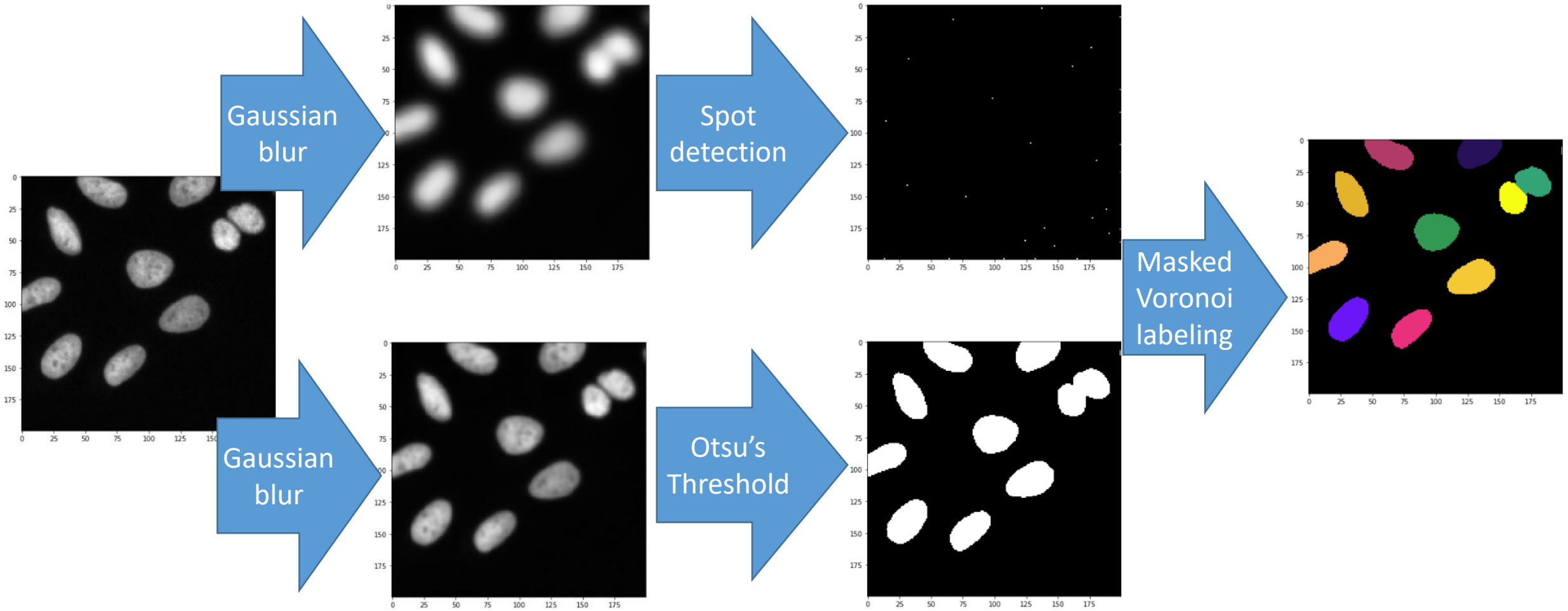
- In napari: Tools > Segmentation / labeling menu



Gauss-Otsu-Labeling

Voronoi-Otsu-Labeling

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



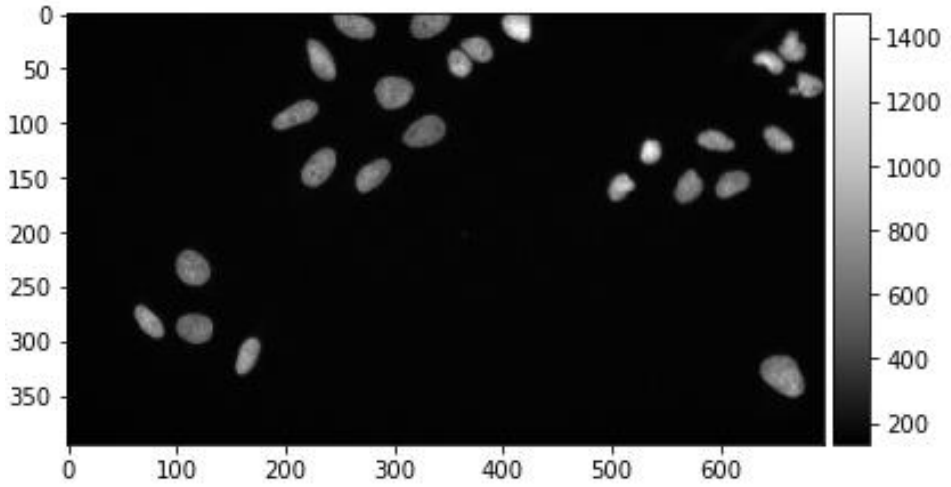
https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/20_image_segmentation/11_voronoi_otsu_labeling.html

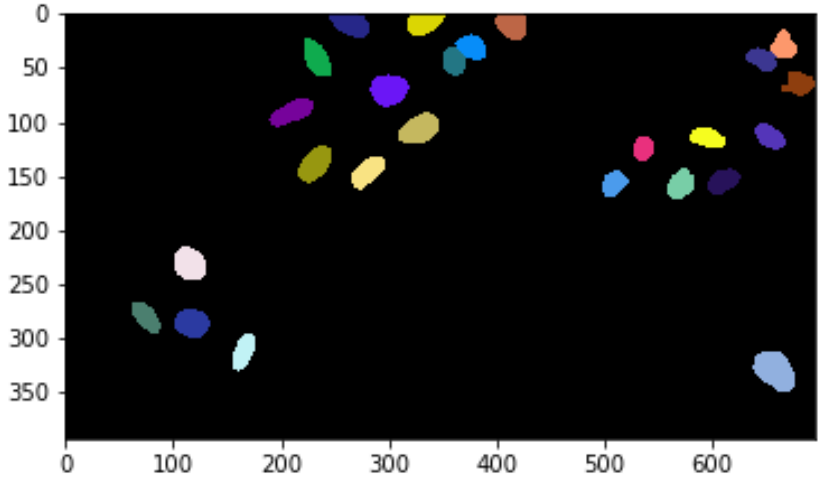
Short-cuts: Voronoi-Otsu-Labeling

- 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

RTFM

PoL Physics of Life TU Dresden

- Some [segmentation] algorithms have prerequisites…

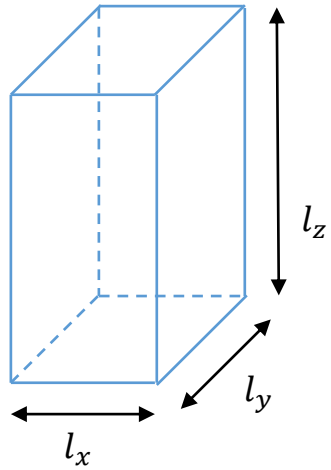
```
[1]: import pyclesperanto_prototype as cle

[ ]: 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
```

Volumetric Data

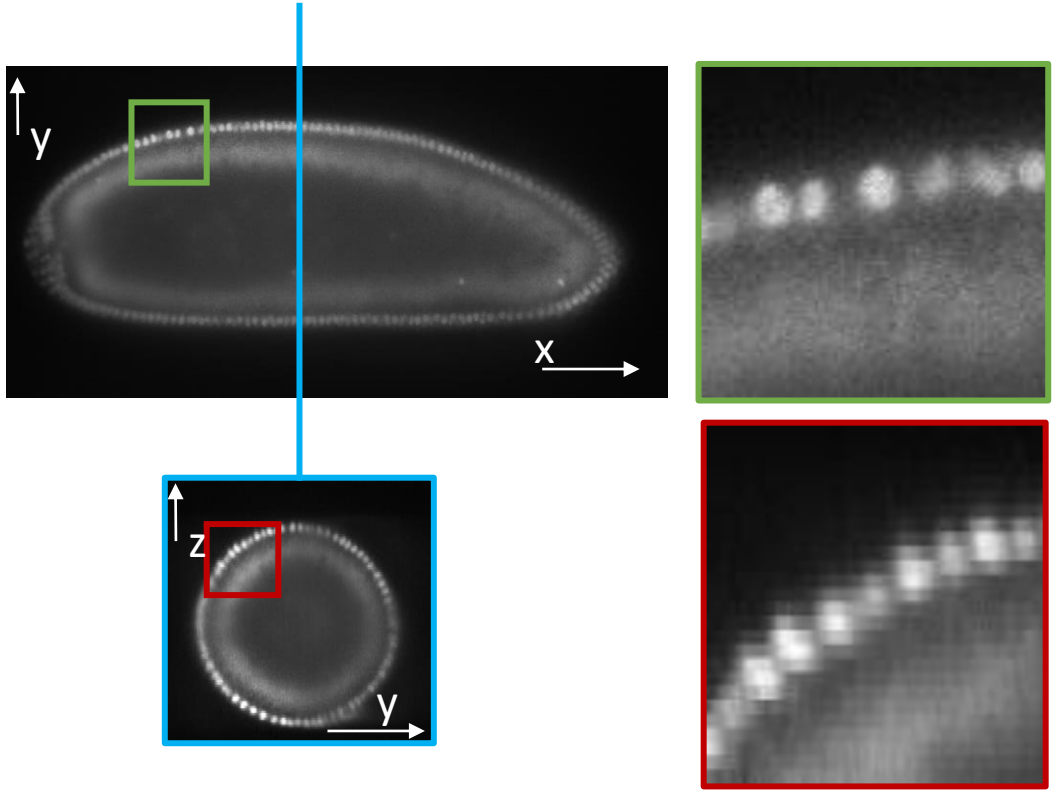
- Voxel: “Volume element”, in microscopy *usually* anisotropic

- An-iso-tropy, from Greek:
  - ἀν- (not)
  - ἴσος *isos (equal)*
  - τρόπος *tropos (rotation, direction)*
- *Not the same in all directions*
- Usually in 3D image processing:



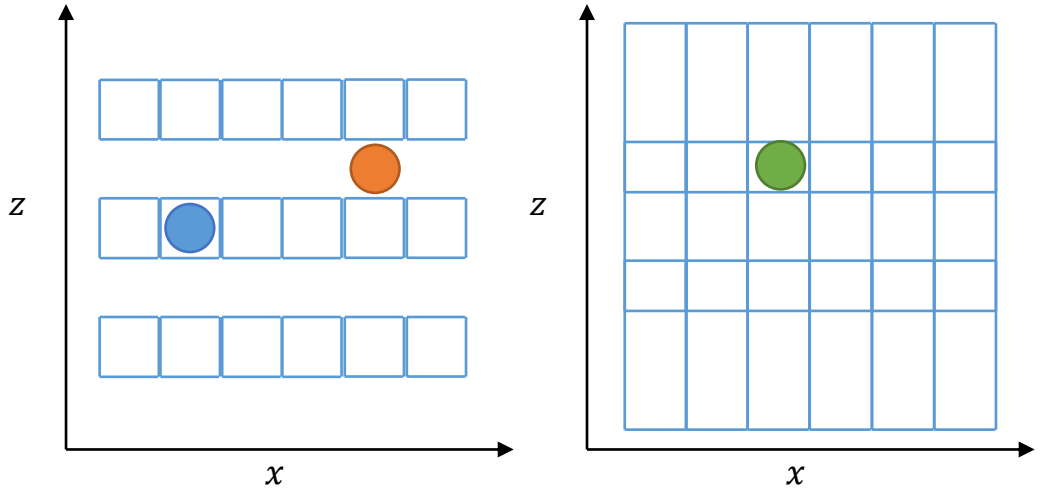
$$l_x = l_y \neq l_z$$

- Image analysts *love* to have isotropic voxels, but microscopes usually have a lower resolution in z than in x and y.

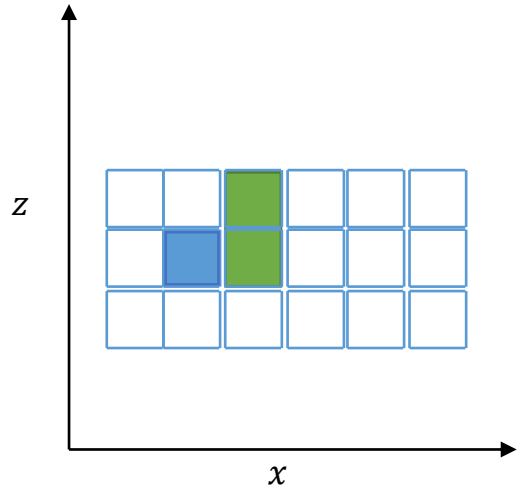


Slice distance versus slice thickness

What you may have measured using imaging:



What you see when processing 3D images:

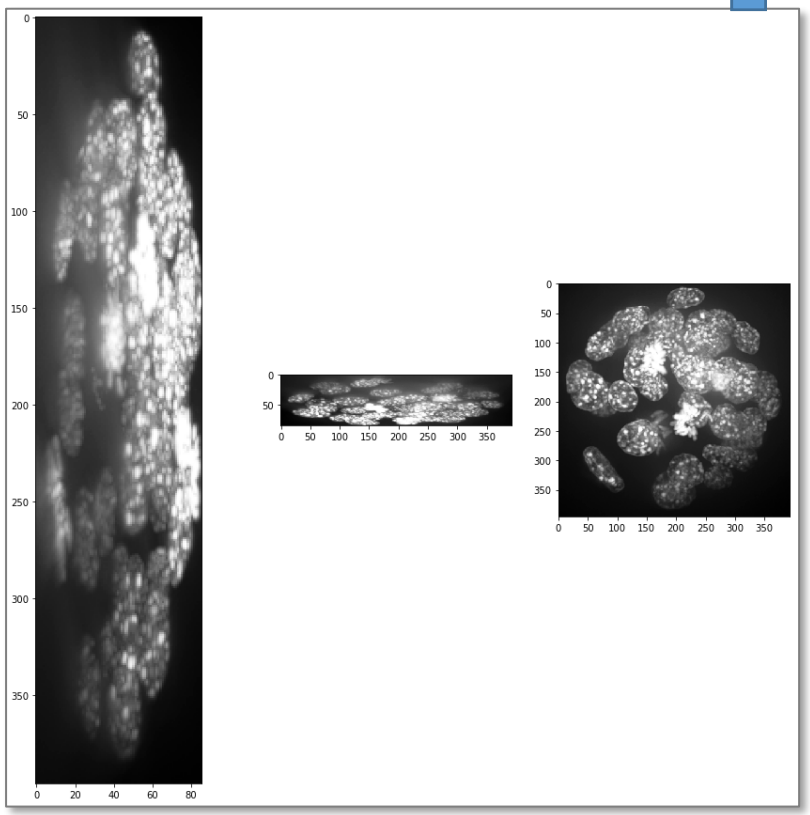


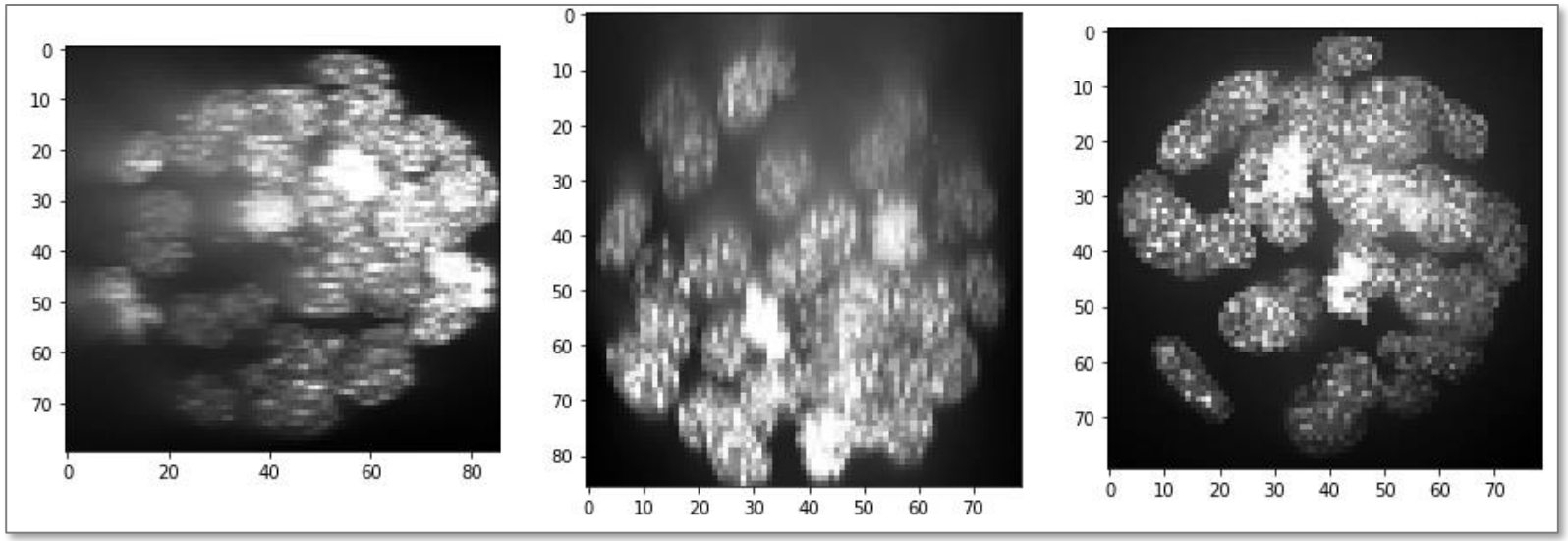
- Slice distance and slice thickness may be different, but
- many image processing tools ignore that!

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)
```

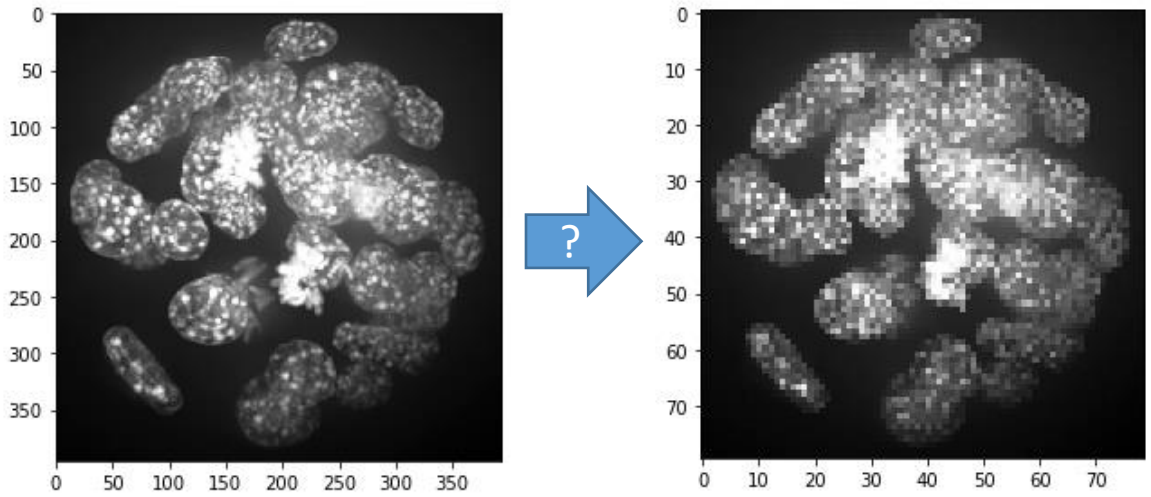




Reslicing / scaling / sampling

- When calling this code, the voxel size of our dataset changes. What is the new isotropic 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)
```



0.1 µm

1 µm

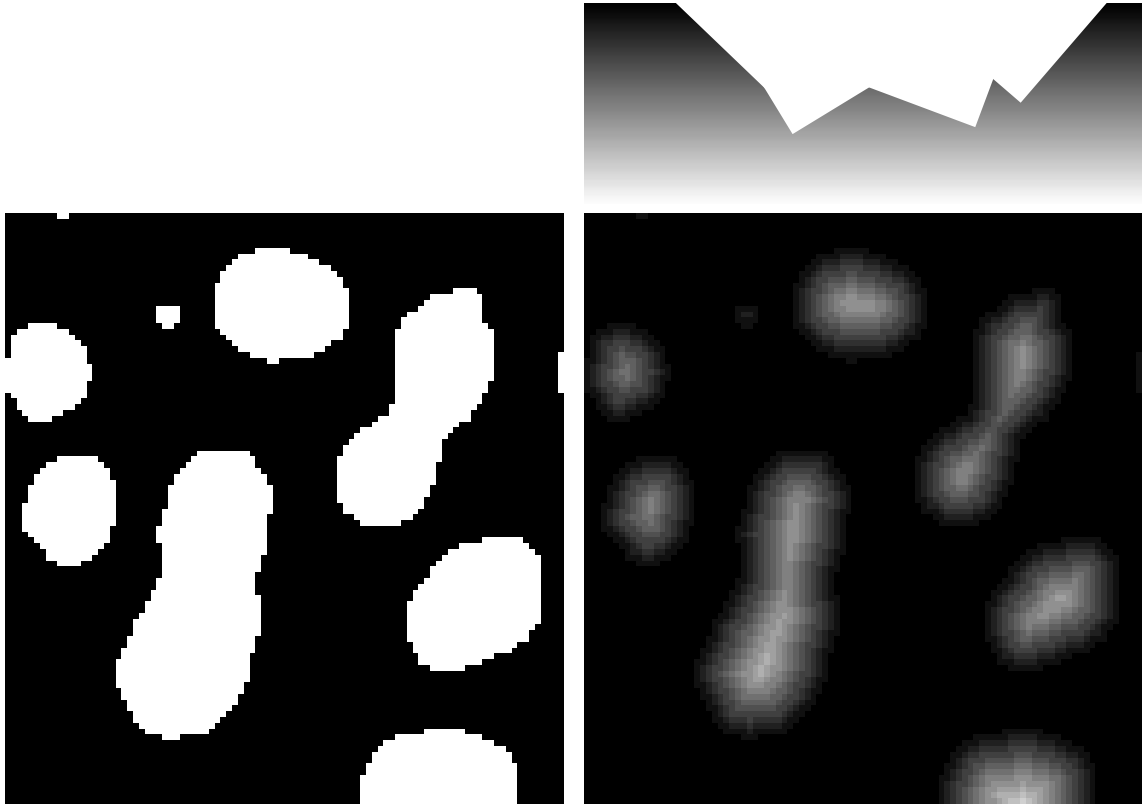
Same as original in Z

Same as original in X and Y

Hint: To answer this question you do *not* need to know the original voxel size.

Watershed

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

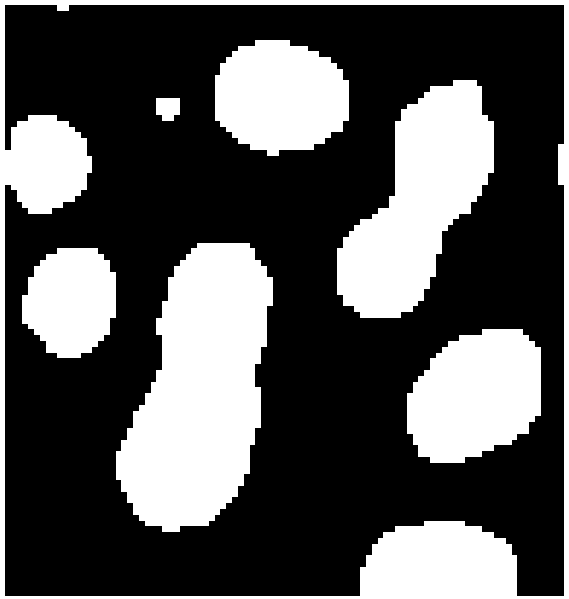


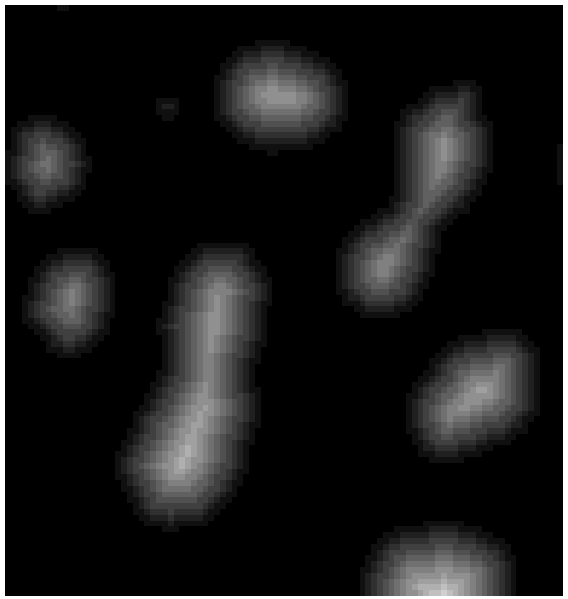
Binary segmentation

Distance map

Watershed

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



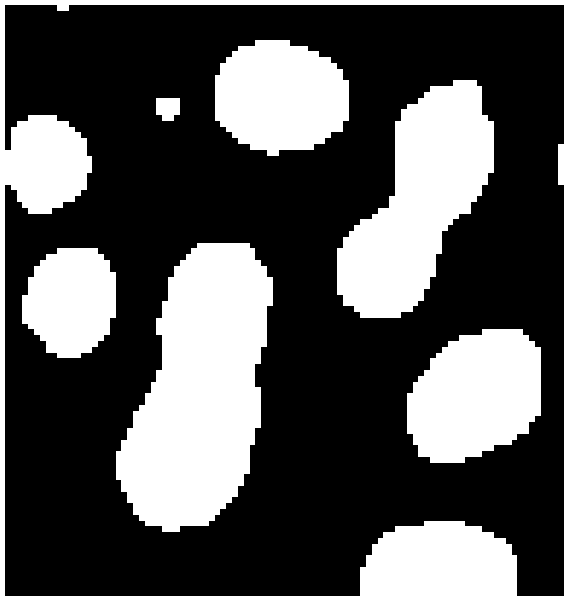


Binary segmentation

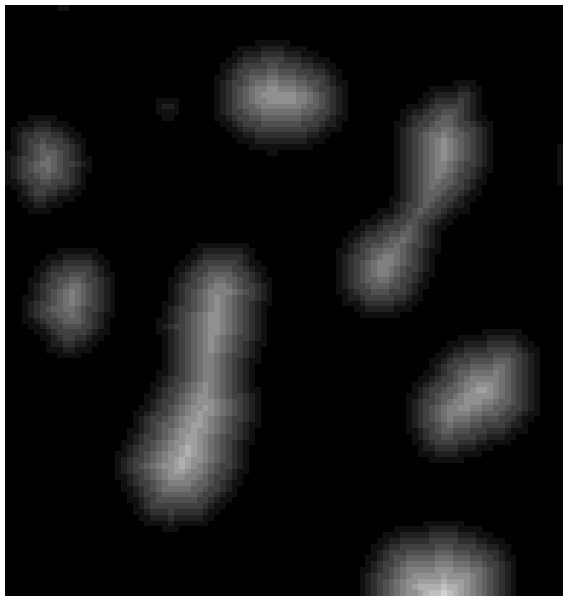
Distance map

Watershed

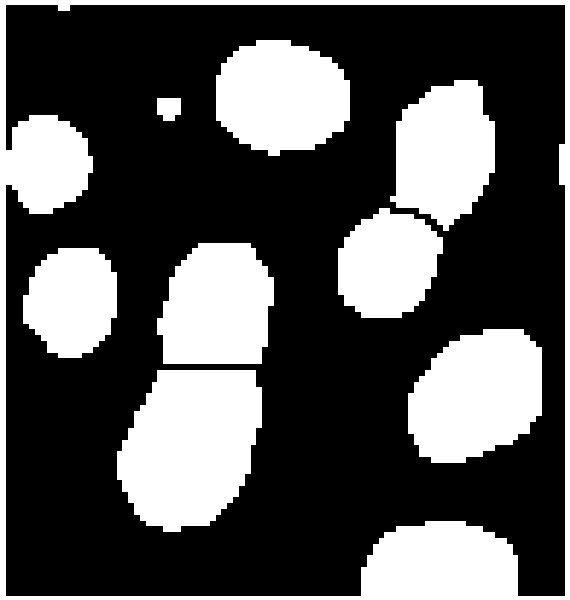
- The watershed algorithm for binary images allows cutting one object into tow where it’s reasonable.
- The watersheds are made from binary images. The algorithm does not take the original image into account!



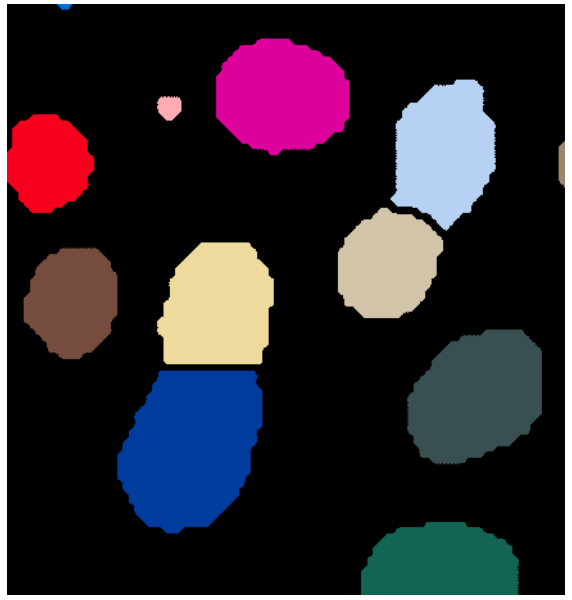
Binary segmentation



Distance map



Binary watershed

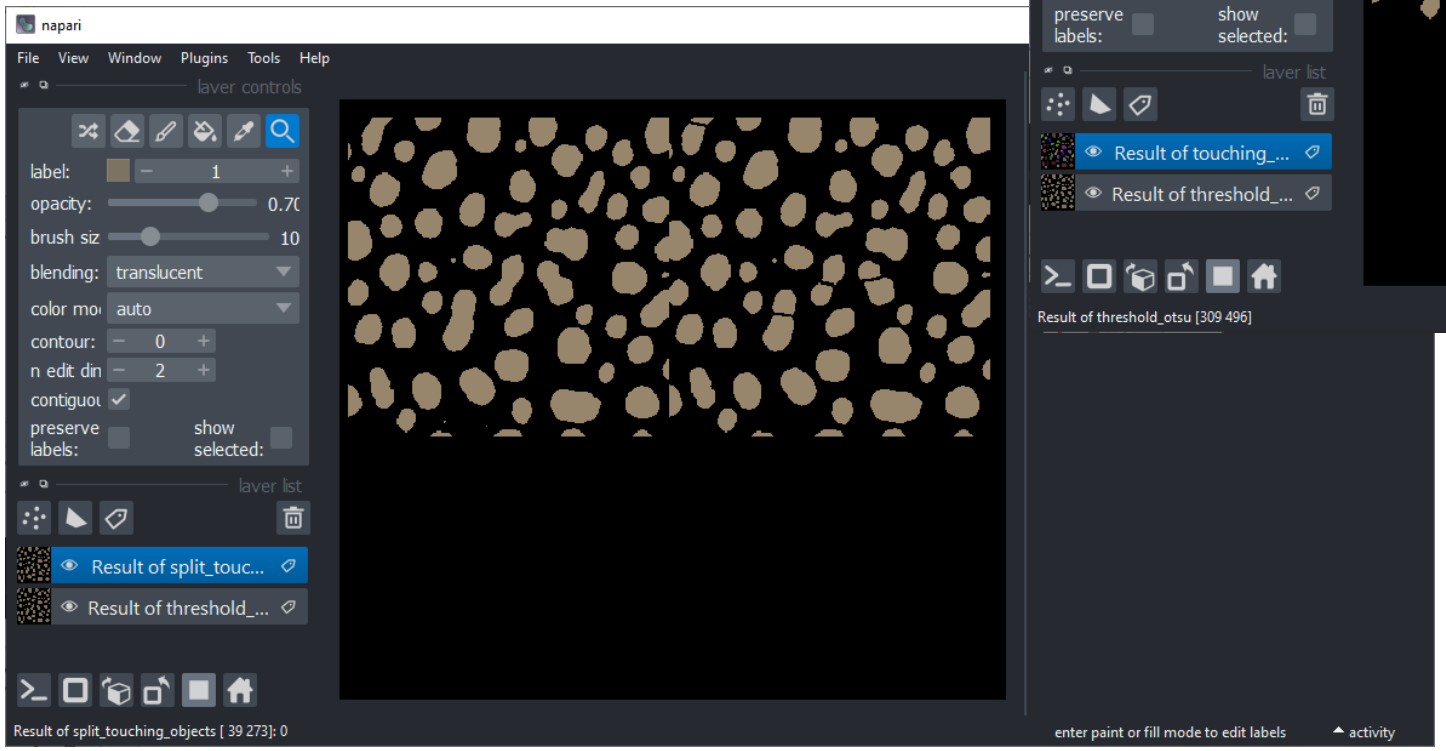


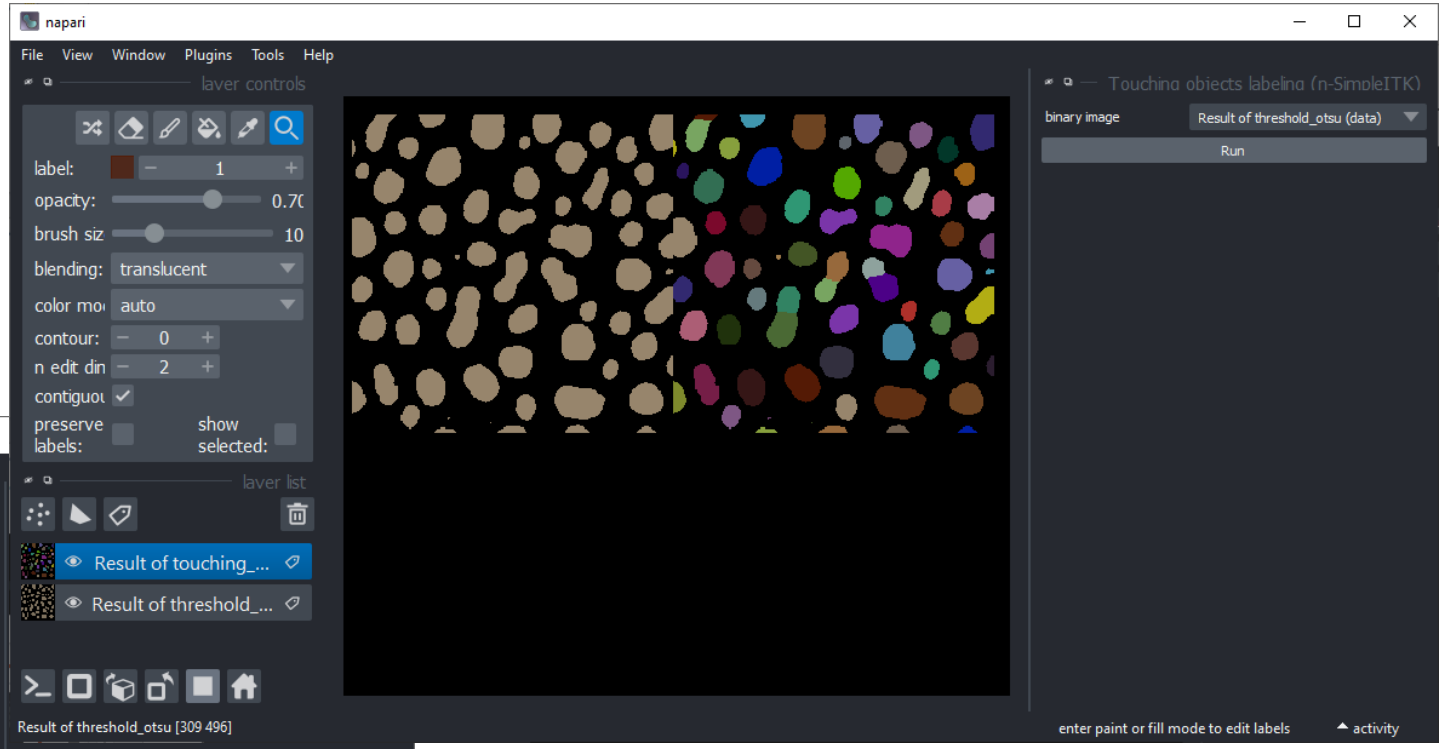
Labeled watershed

Watershed

- In Napari

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

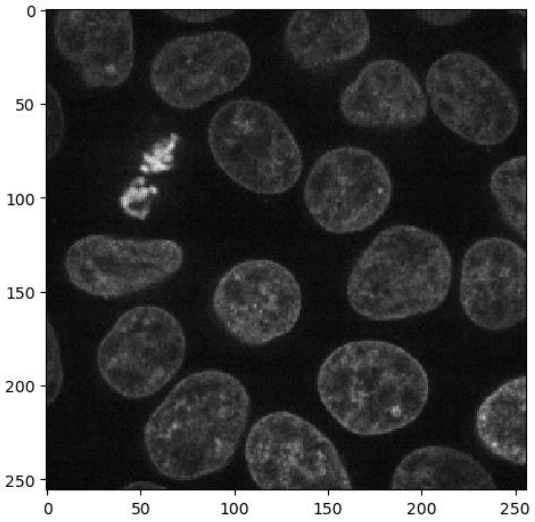


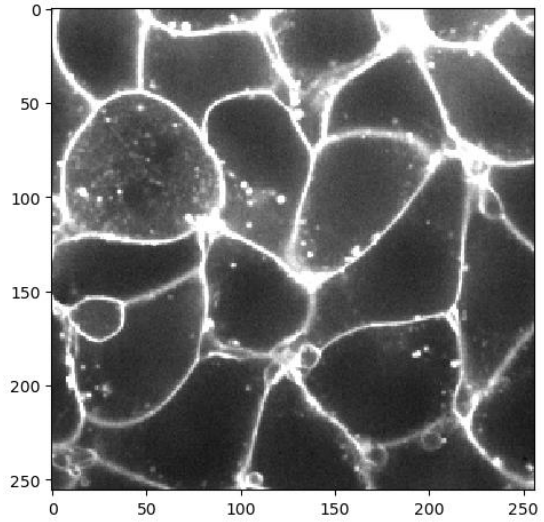


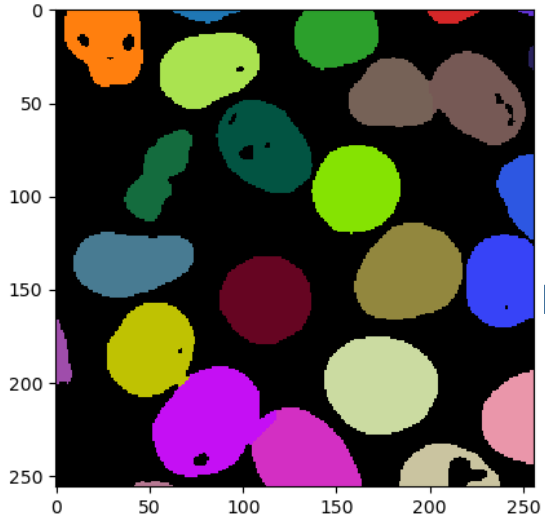
Results directly in a label image:
Tools > Segmentation / labeling >
Label touching objects

Watershed

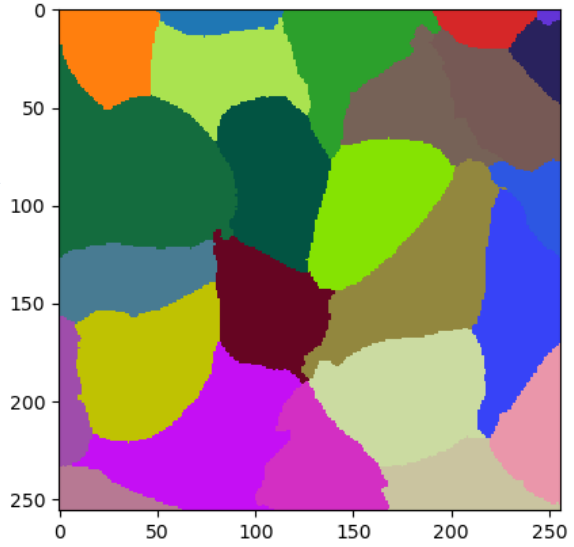
- … in Python practice







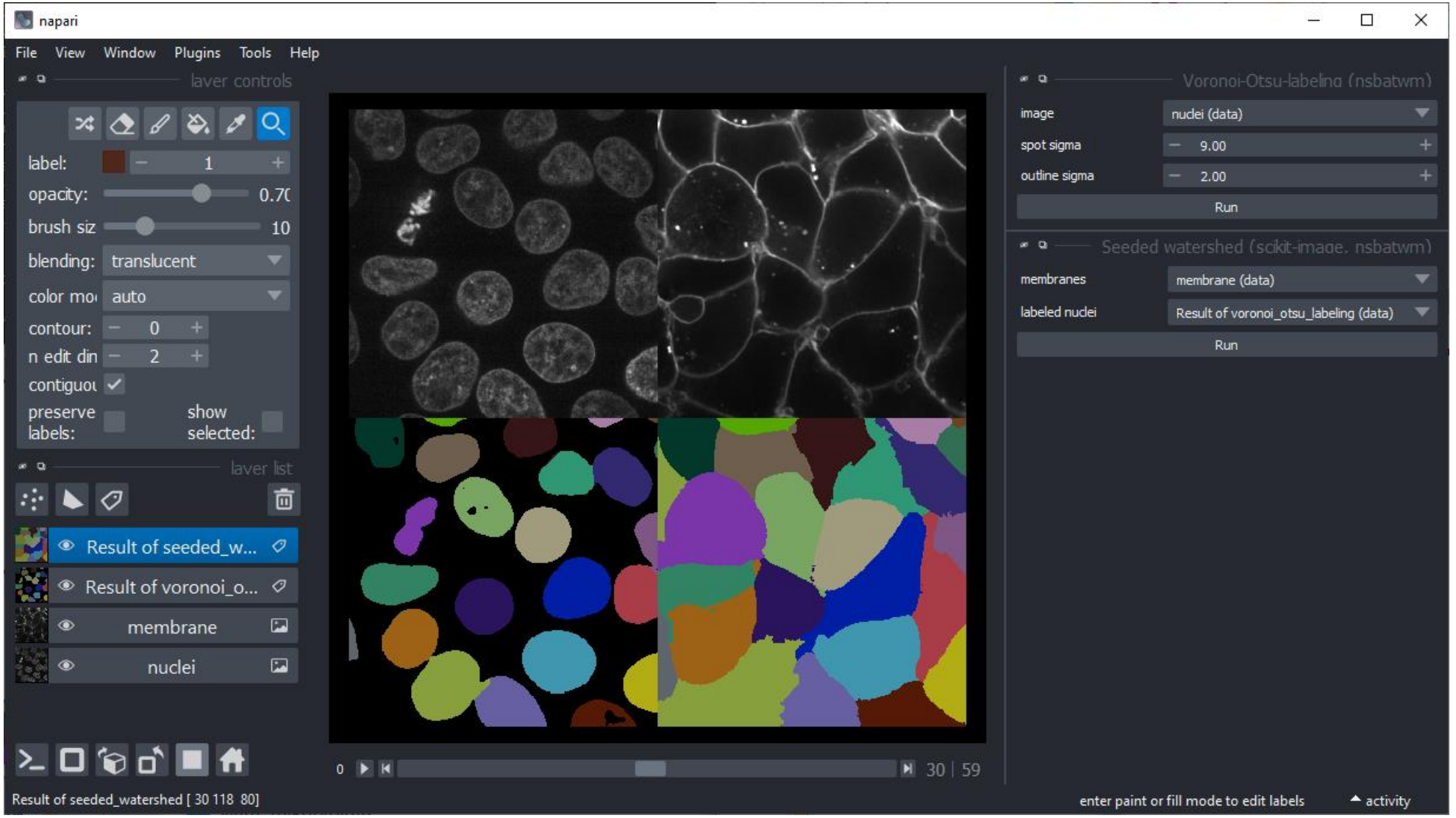

```
labeled_cells = seeded_watershed(membrane_channel, labeled_nuclei)

labeled_cells
```



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

Watershed

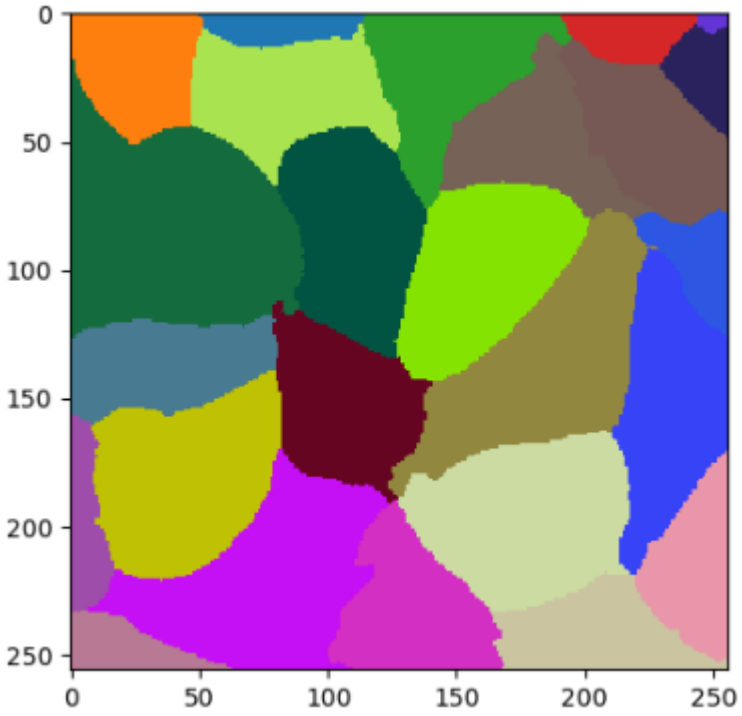
- … in Napari practice: Tools > Segmentation / Labeling menu



Label post-processing / morphological operations

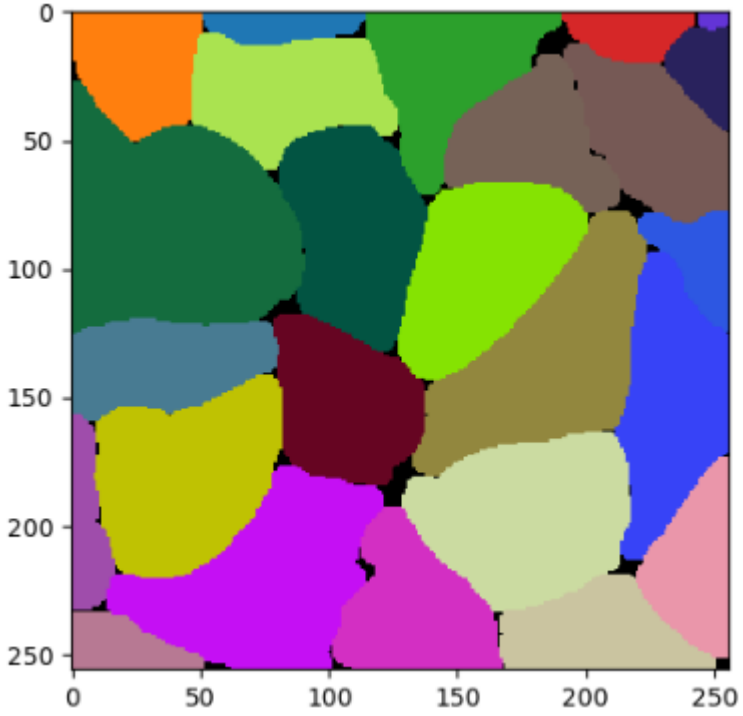
PoL Physics of Life TU Dresden

- … similar to morphological operations on binary images



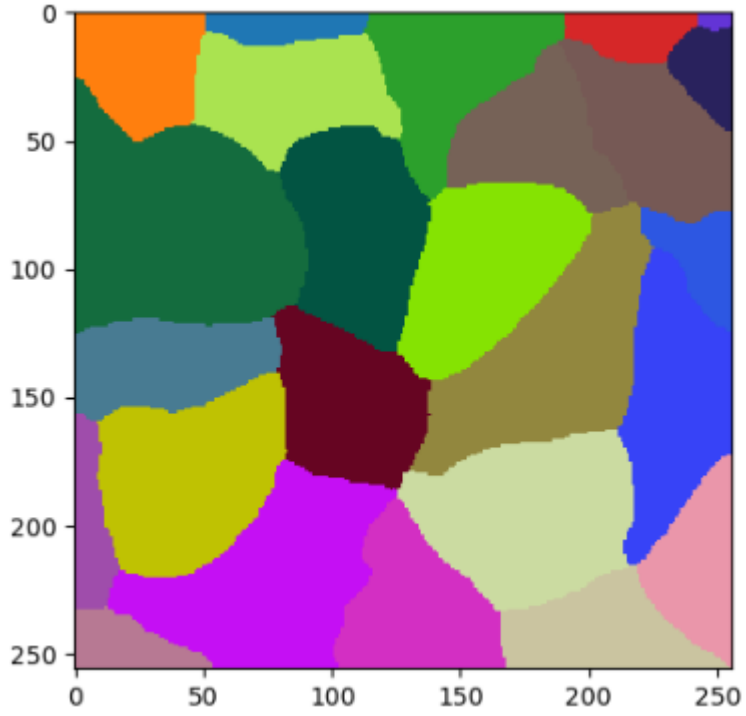
Original

```
cle.opening_labels(label_image, radius=5)
```



Opening Labels

```
cle.smooth_labels(label_image, radius = 5)
```

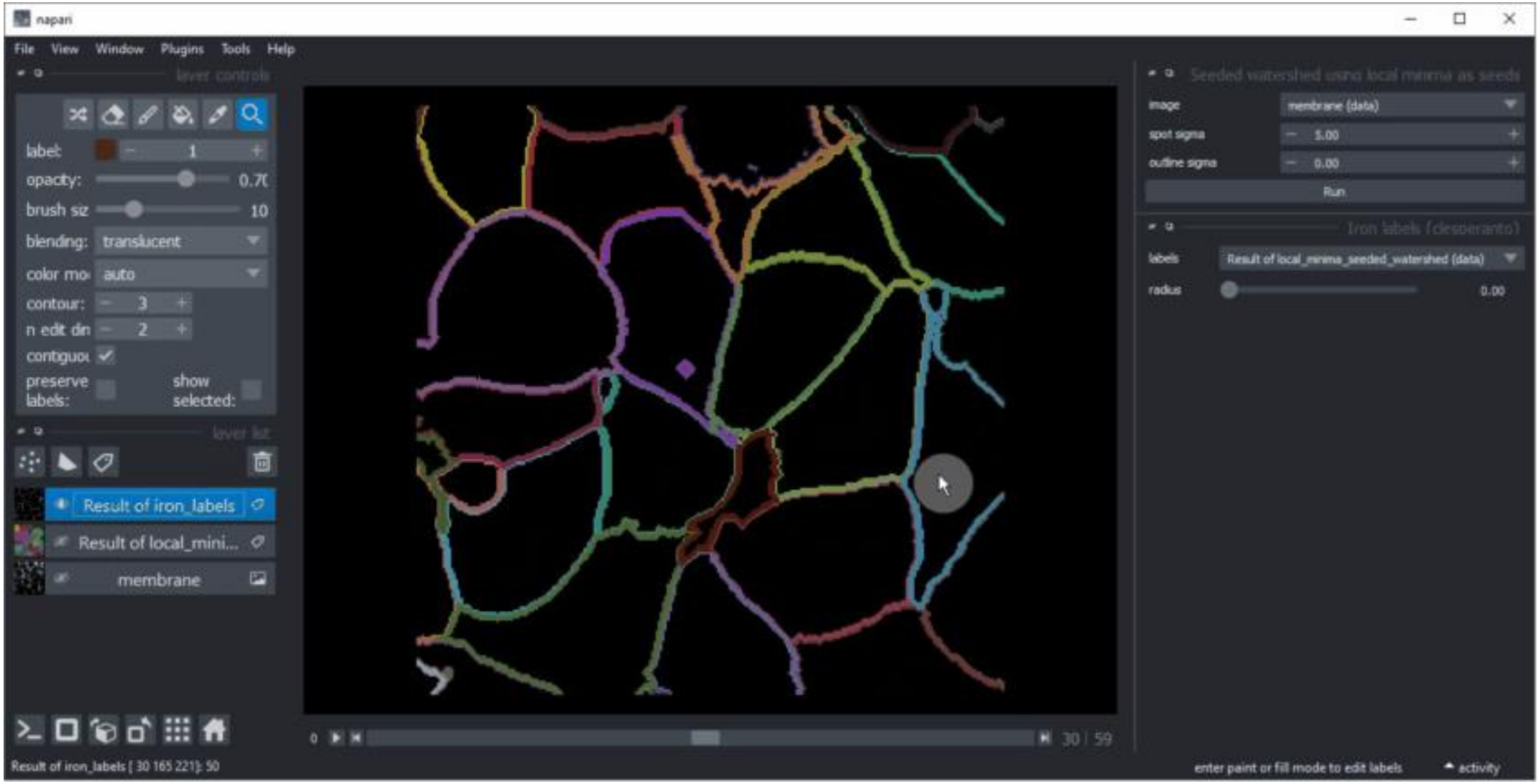


Smoothing Labels

Label post-processing / morphological operations

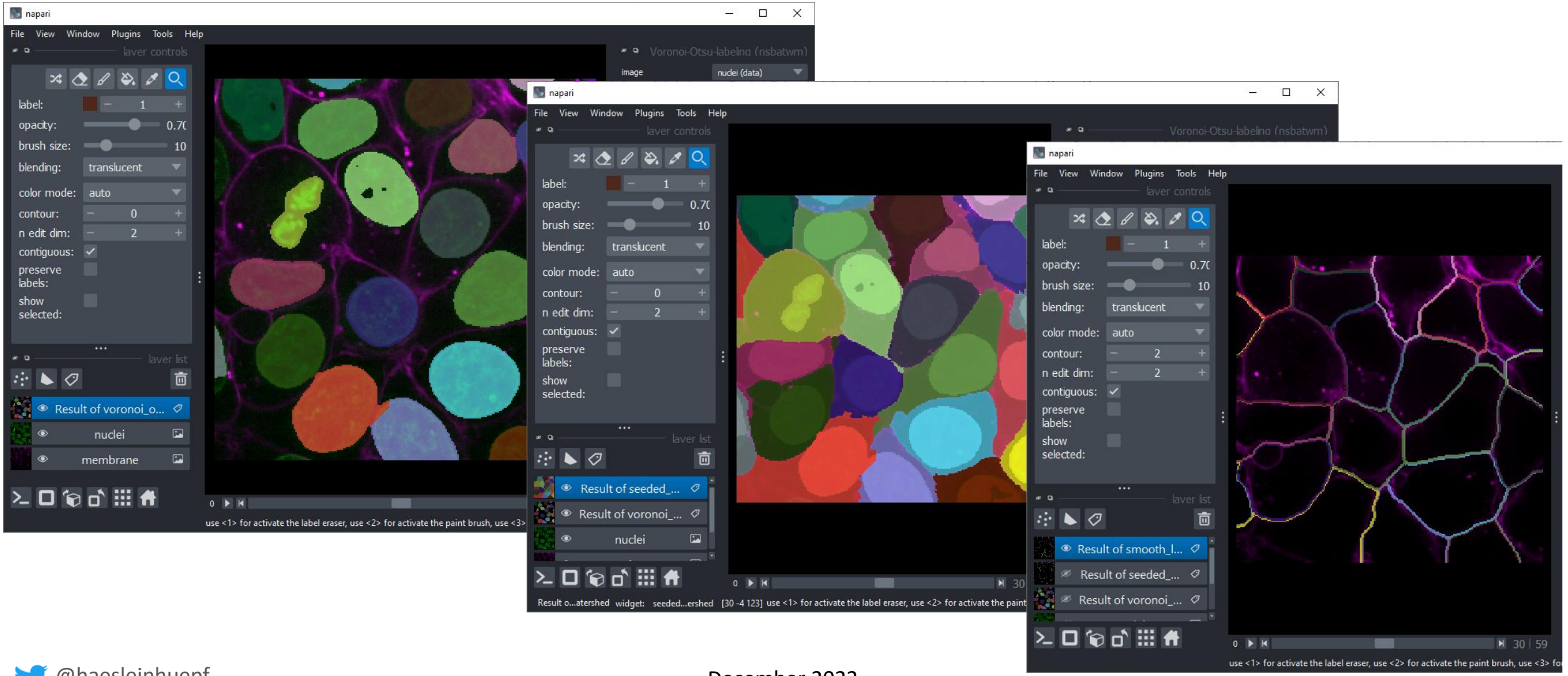
PoL Physics of Life TU Dresden

- In Napari menu Tools > Segmentation post-processing > Smooth labels (clEsperanto)



Exercise

- Segment the nuclei in this dataset and use them as seeds for the seeded watershed algorithm. Smooth the labels of the resulting cell segmentation.



Exercise

- Segment the nuclei in this dataset and use them as seeds for the seeded watershed algorithm. Smooth the labels of the resulting cell segmentation.
- Follow the step-by-step instructions in the Jupyter book