PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Instance segmentation

Robert Haase

With material from

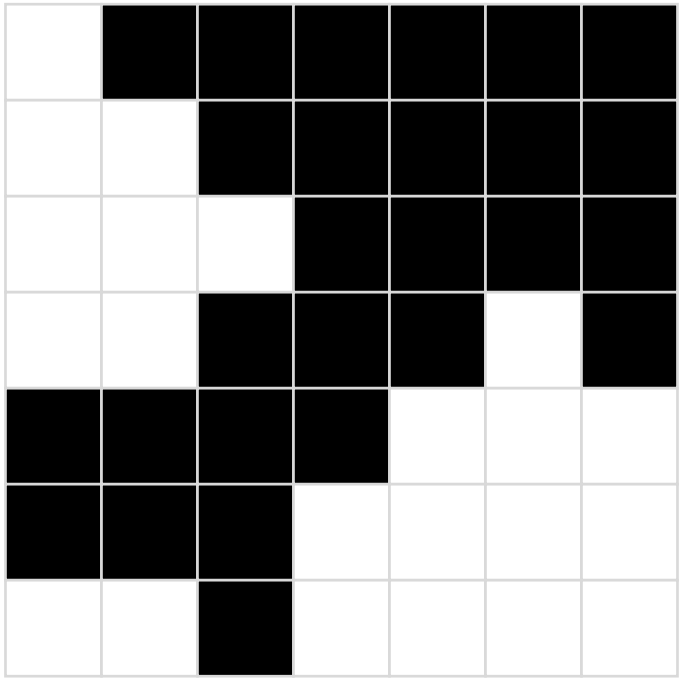
Benoit Lombardot, Scientific Computing Facility, MPI CBG

October 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.



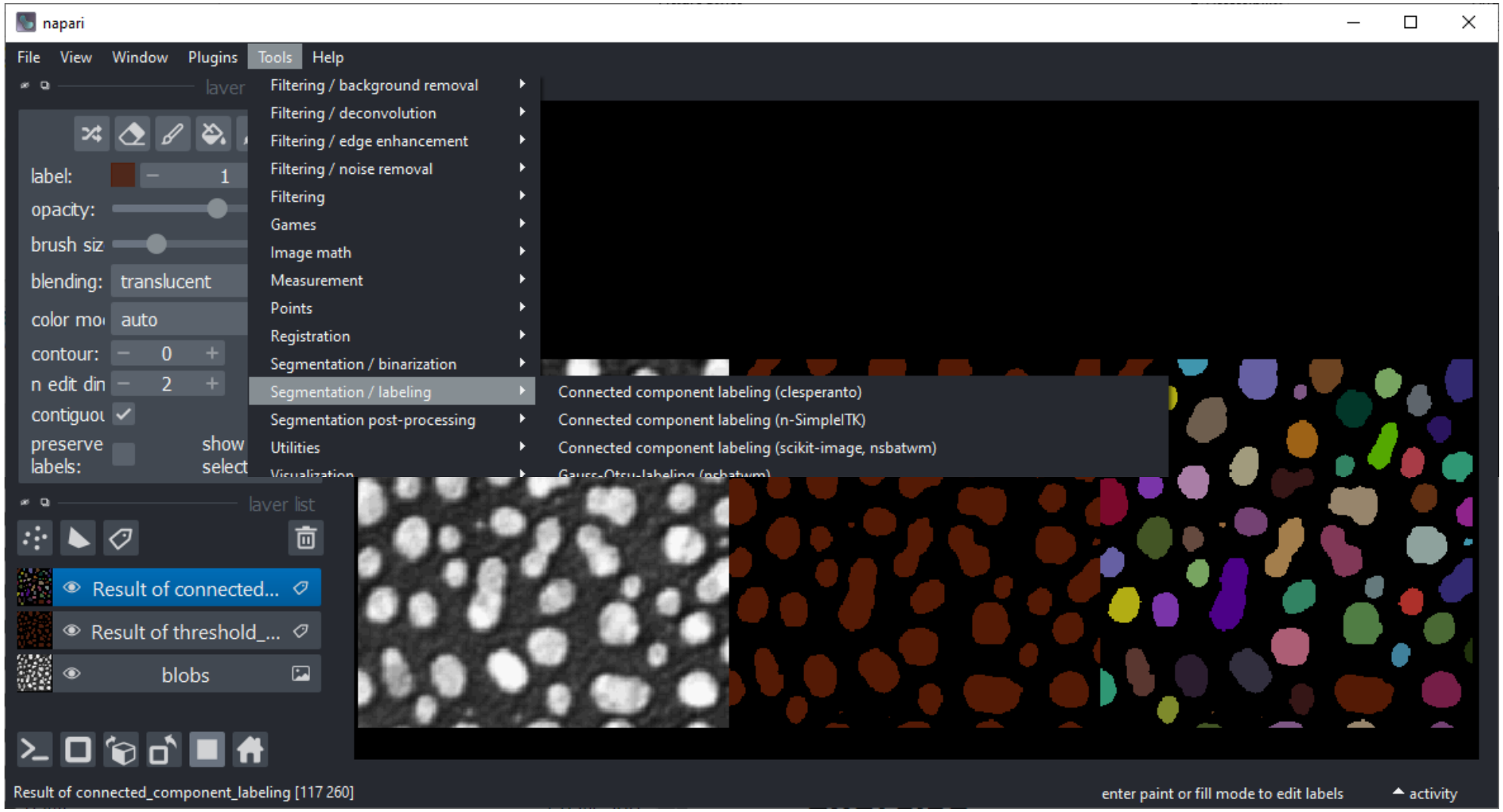
CCA

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

NOTE:
The maximum intensity of the image can be used to determine how many objects there are in the image.

Connected component labelling

- In napari: Tools > Segmentation / labeling menu

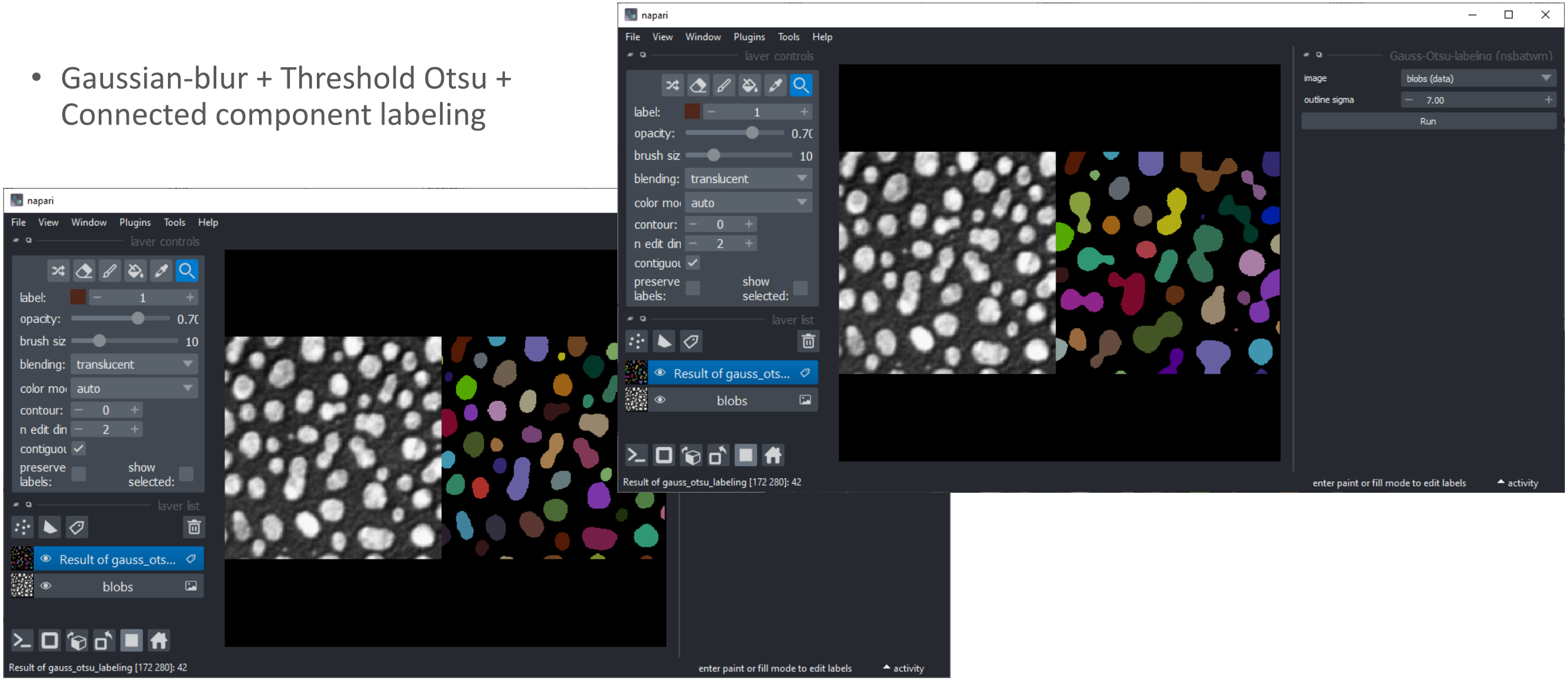


Short-cuts: Gauss-Otsu-Labeling

PoL Physics of Life TU Dresden

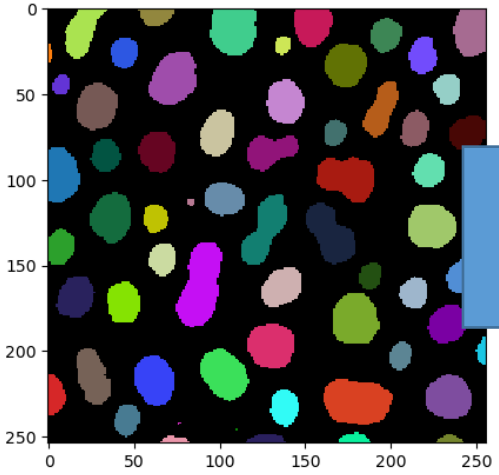
- In napari: Tools > Segmentation / labeling menu

- Gaussian-blur + Threshold Otsu + Connected component labeling

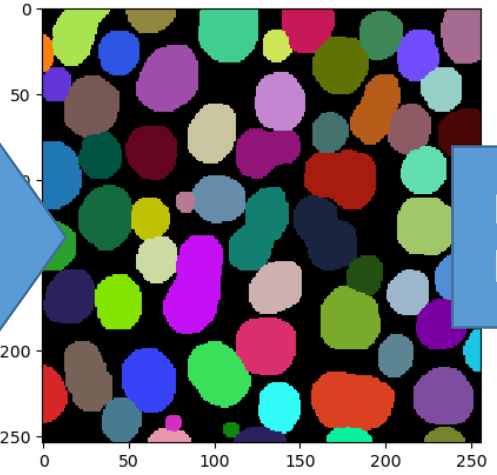


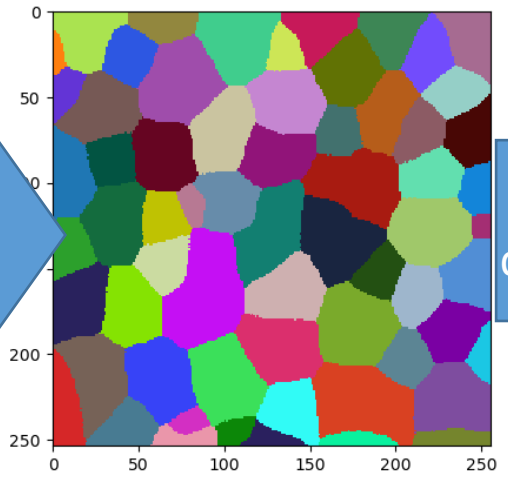
Voronoi-Tesselation

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

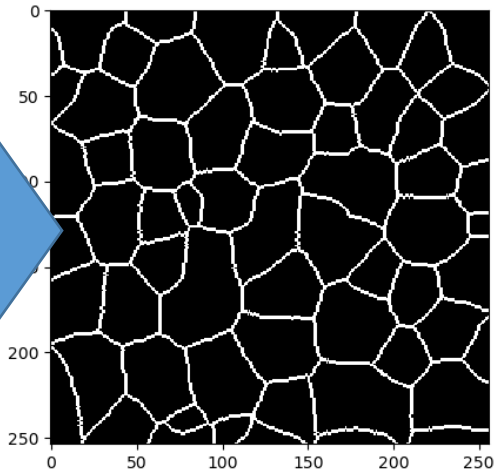


Label-image



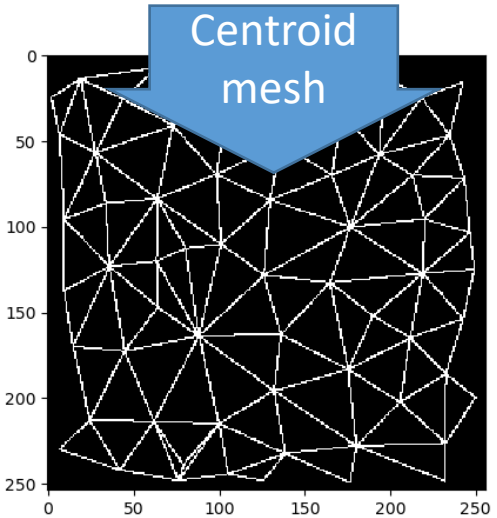


Voronoi-label-image

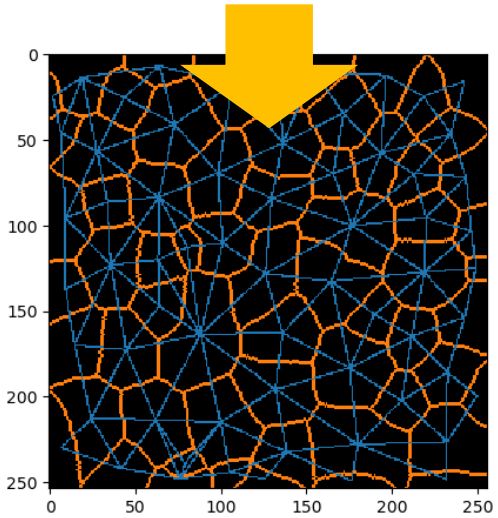


Label-edge-image

???
Check out the VORONIO labelling
Does it use an object as a seed?
Can an abitrary object generate the same result with this method.



Touching neighbor mesh

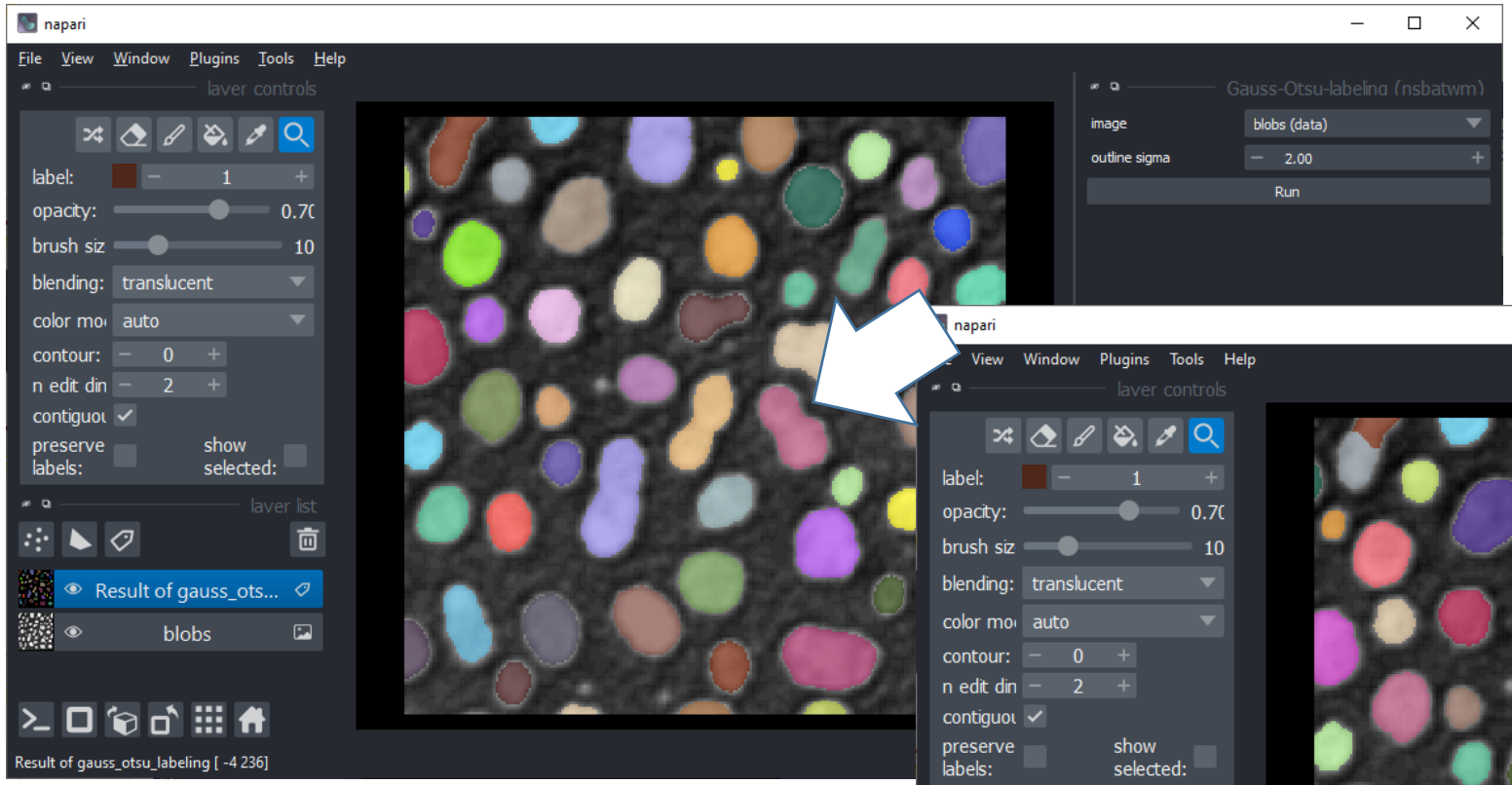


Voronoi-Tesselation
Delauney-Triangulation

Short-cuts: Voronoi-Otsu-Labeling

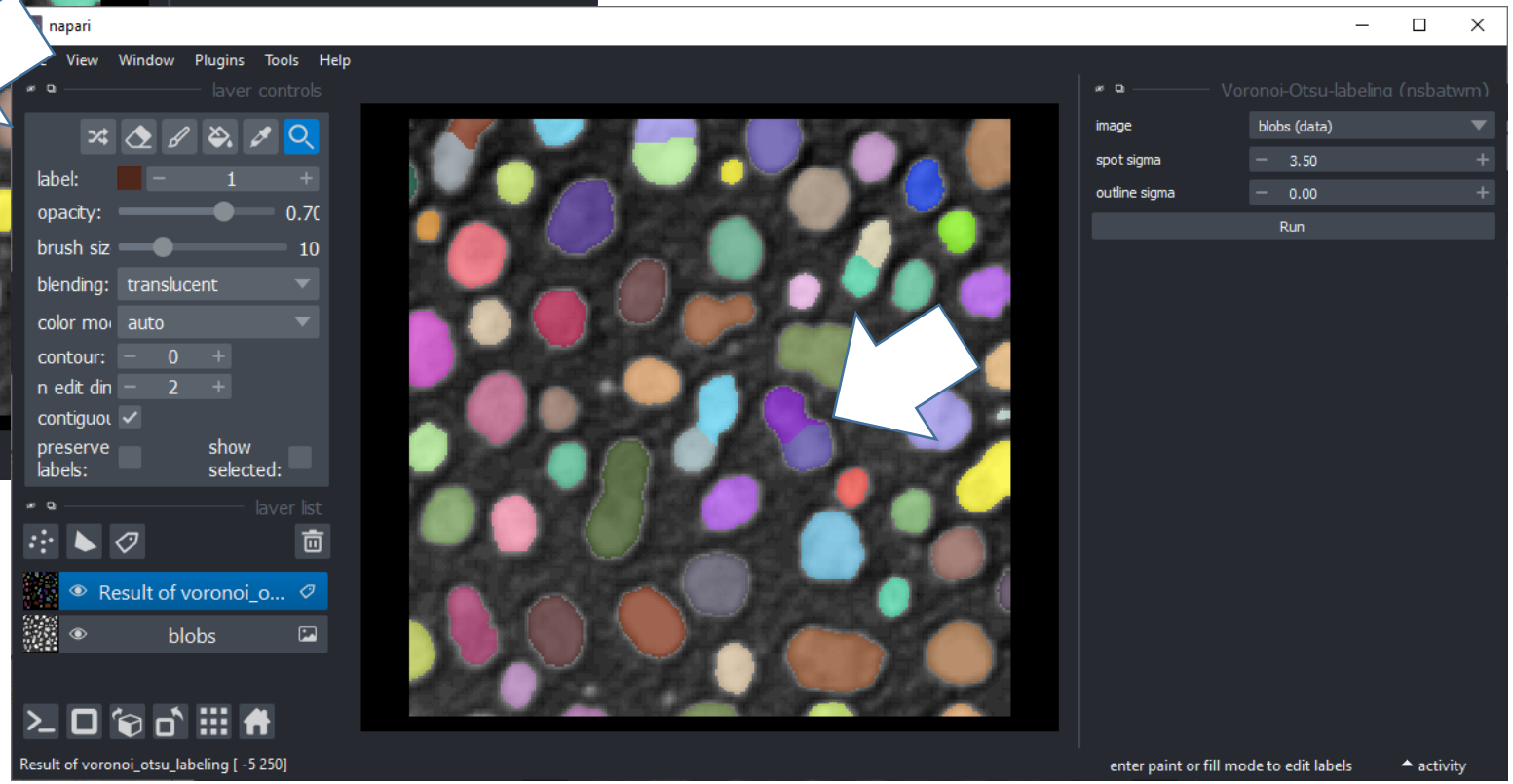
PoL Physics of Life TU Dresden

- In napari: Tools > Segmentation / labeling menu



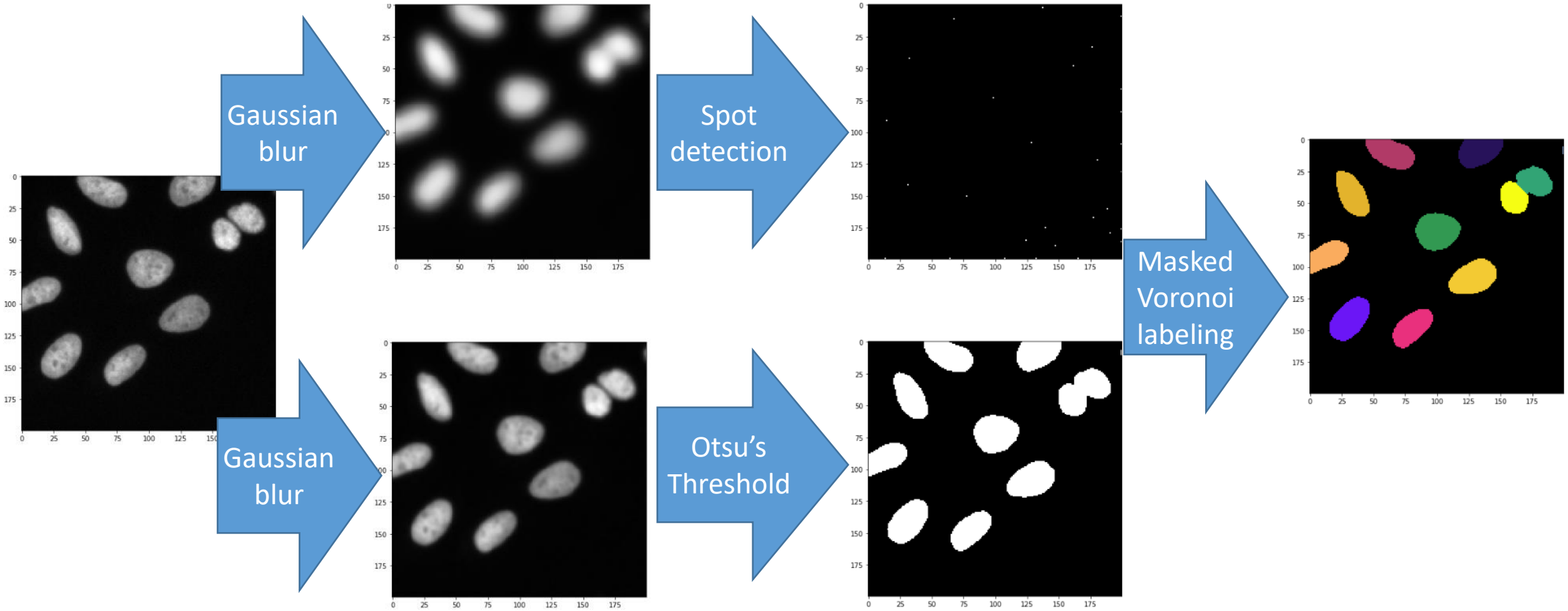
Gauss-Otsu-Labeling

Voronoi-Otsu-Labeling



Voronoi-Otsu-Labeling

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



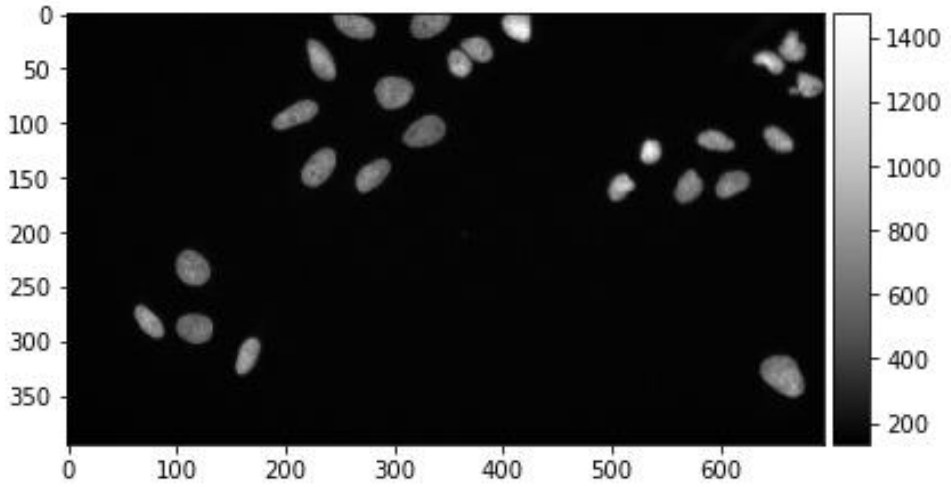
https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/20_image_segmentation/11_voronoi_otsu_labeling.html

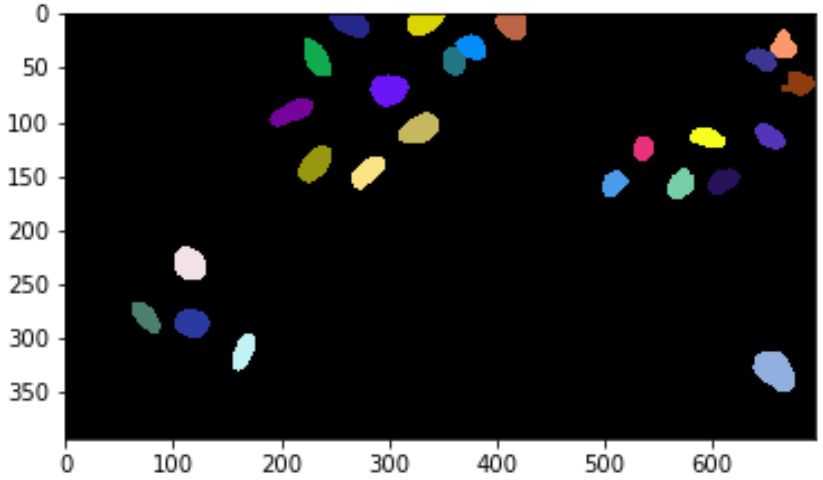
Short-cuts: Voronoi-Otsu-Labeling

- 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

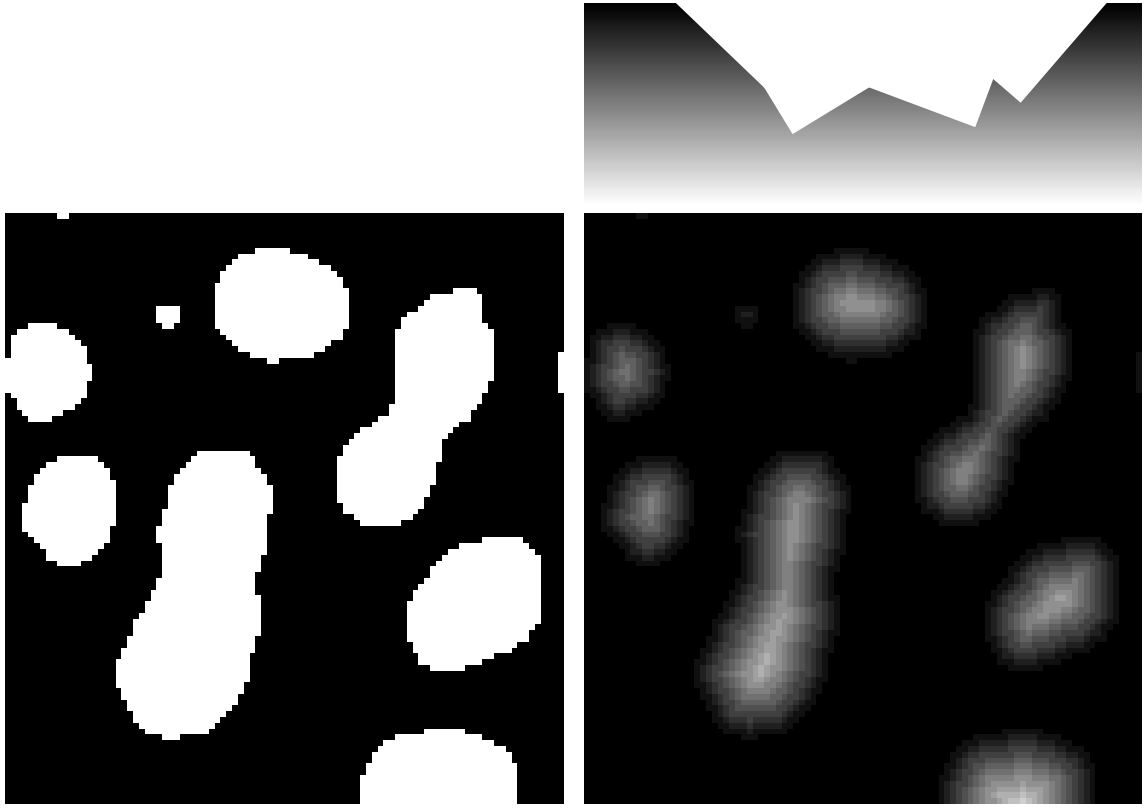
shape	(395, 695)
dtype	int32
size	1.0 MB
min	0
max	25

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

https://www.youtube.com/watch?v=evgRgDfVuEc

Watershed

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



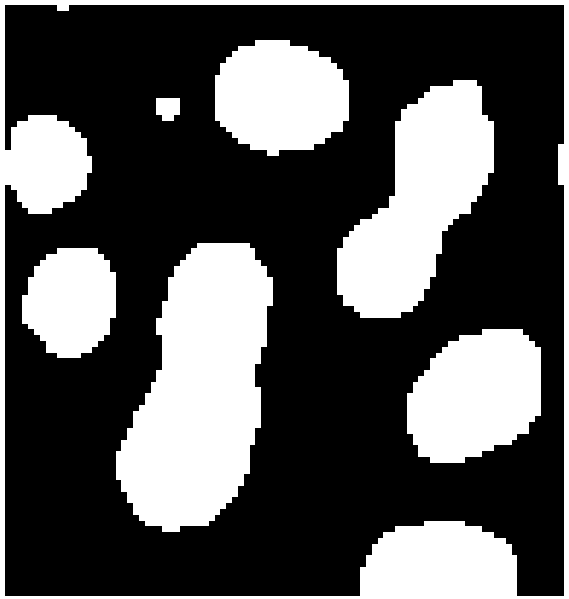
Binary segmentation

Distance map

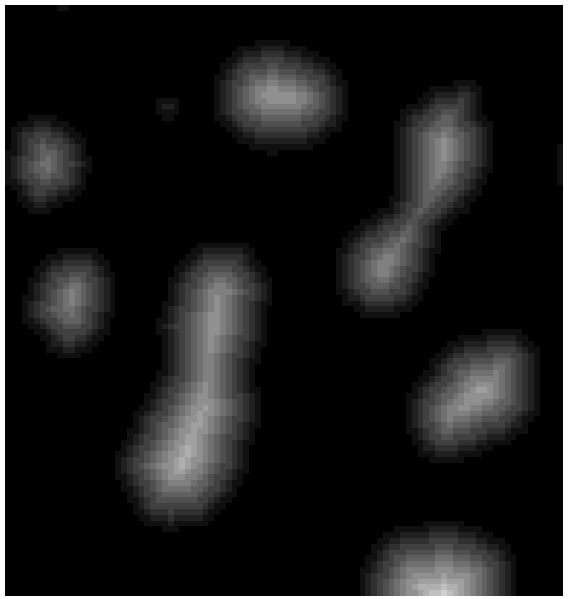
NOTE:
Distance map is used to determine how far the black background of an object is from the neighbouring object. It's principle is used in implementing watershed algorithm.

Watershed

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



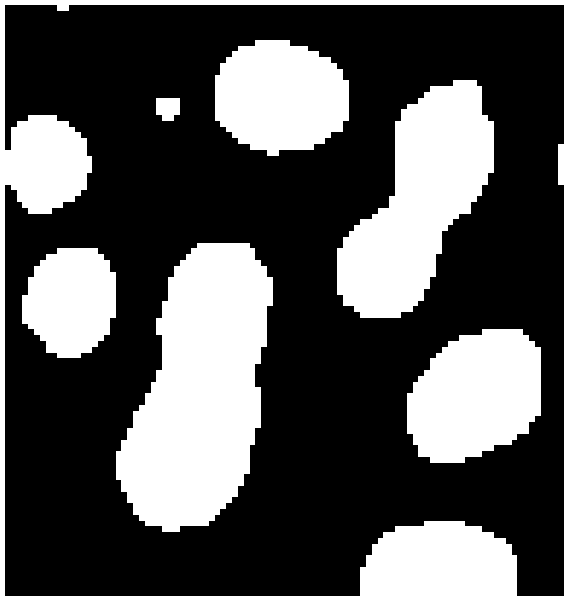
Binary segmentation



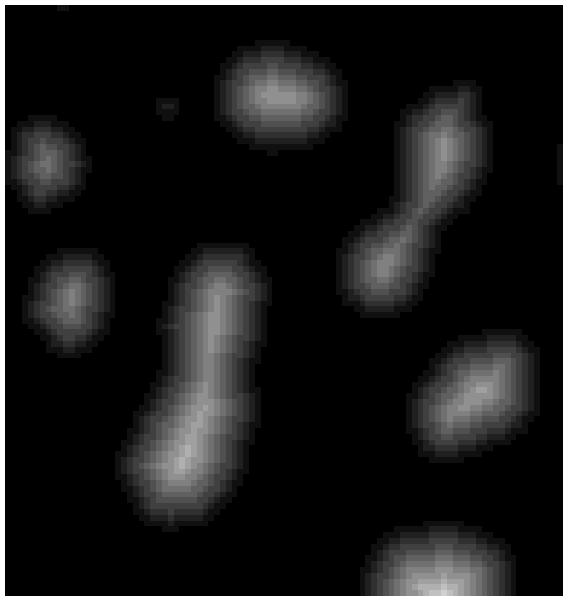
Distance map

Watershed

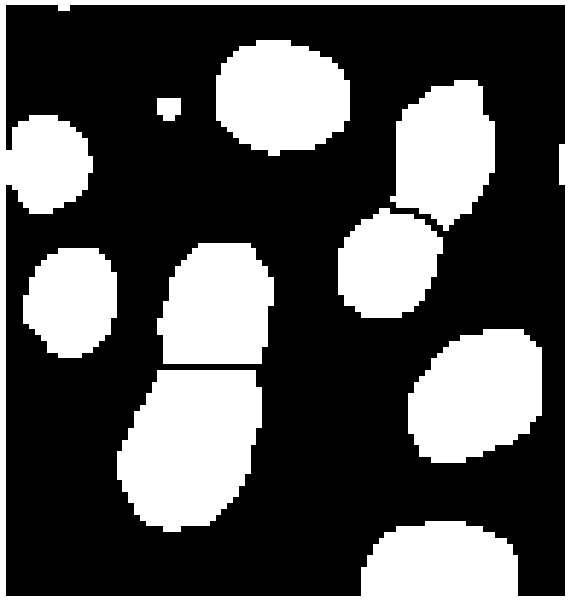
- The watershed algorithm for binary images allows cutting one object into tow where it’s reasonable.
- The watersheds are made from binary images. The algorithm does not take the original image into account!



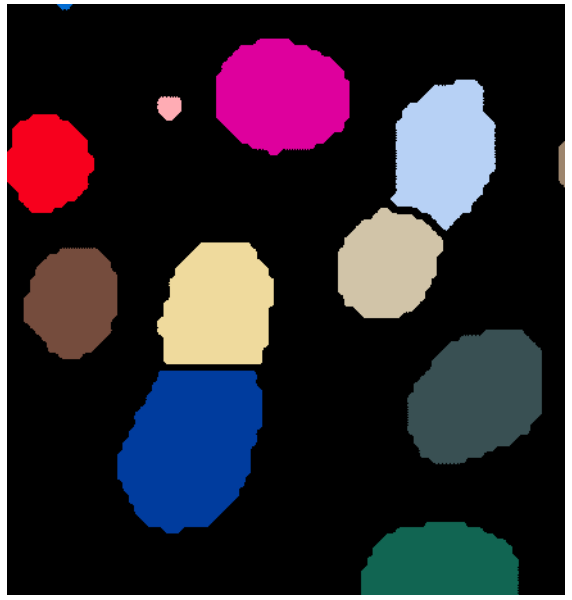
Binary segmentation



Distance map



Binary watershed

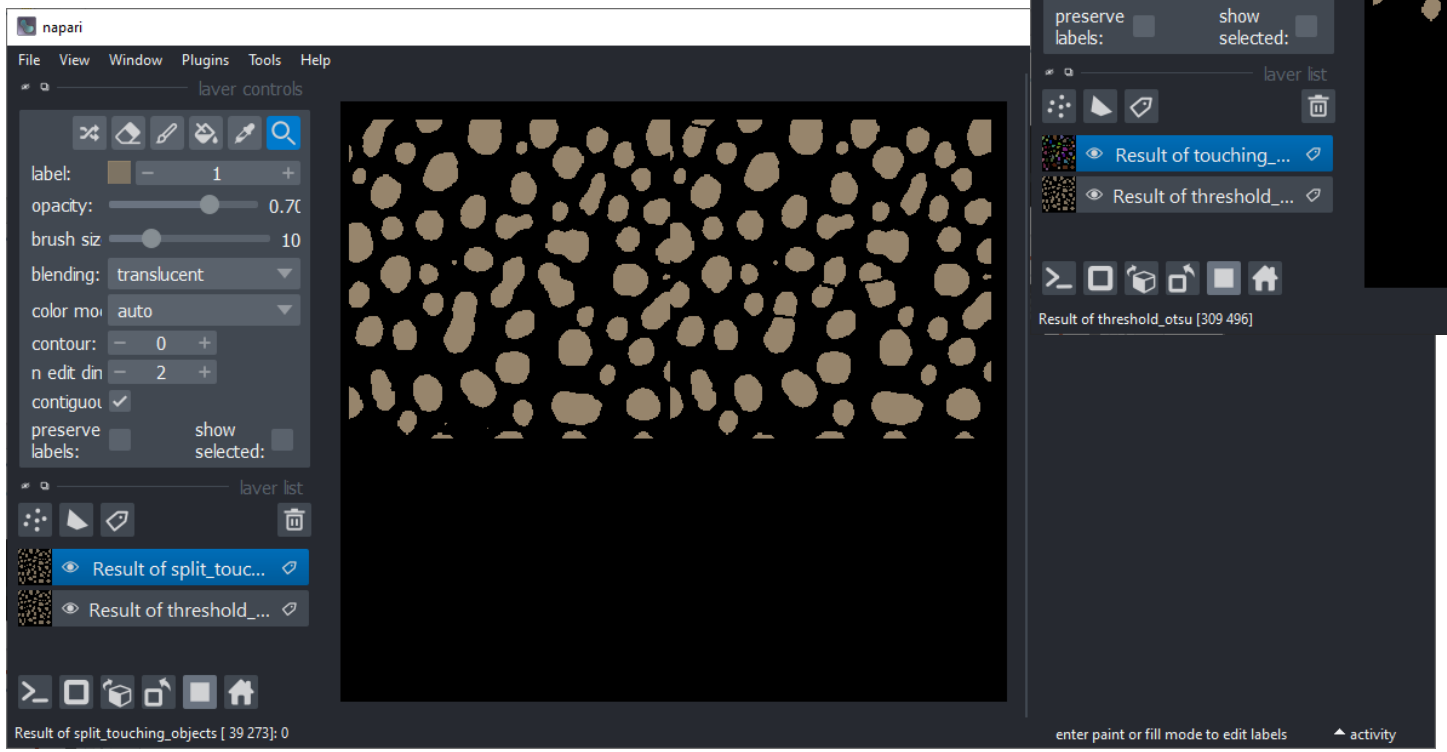


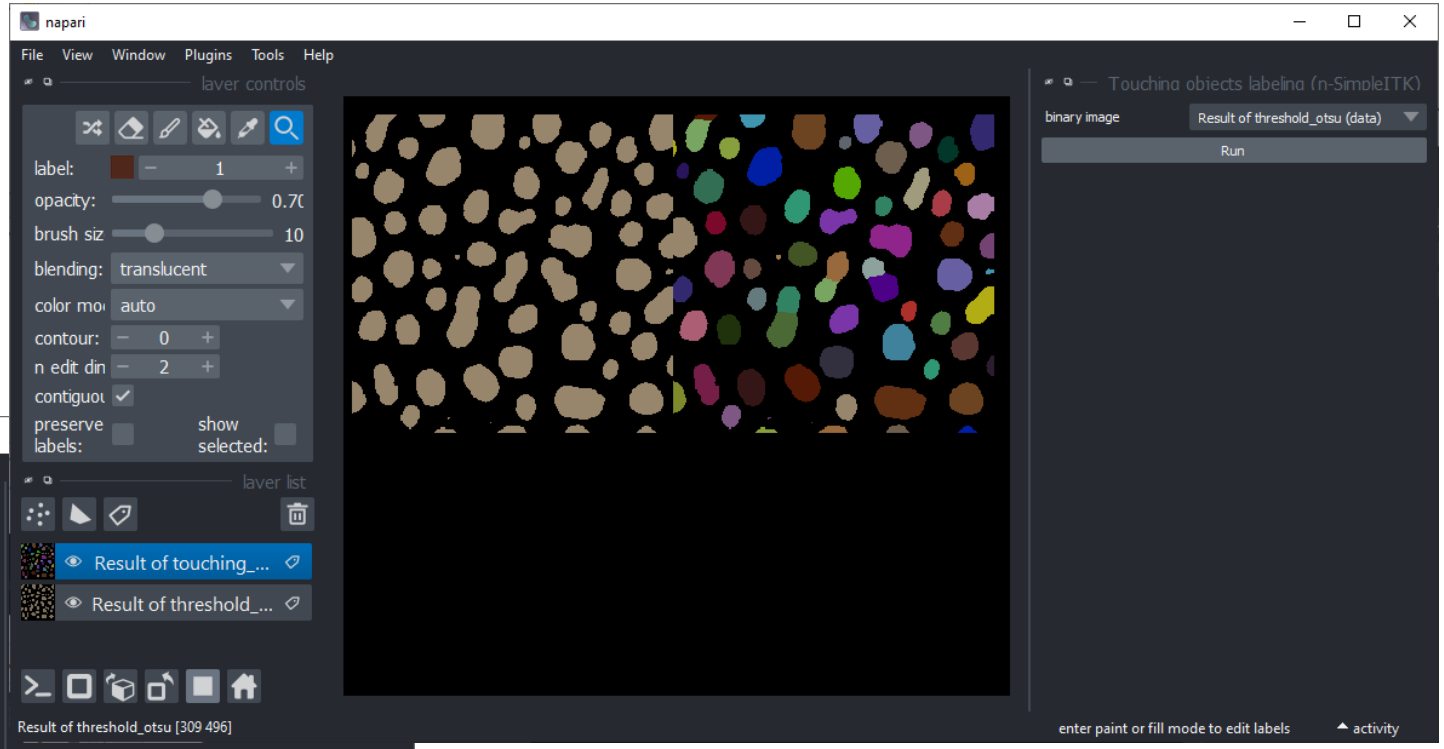
Labeled watershed

Watershed

- In Napari

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

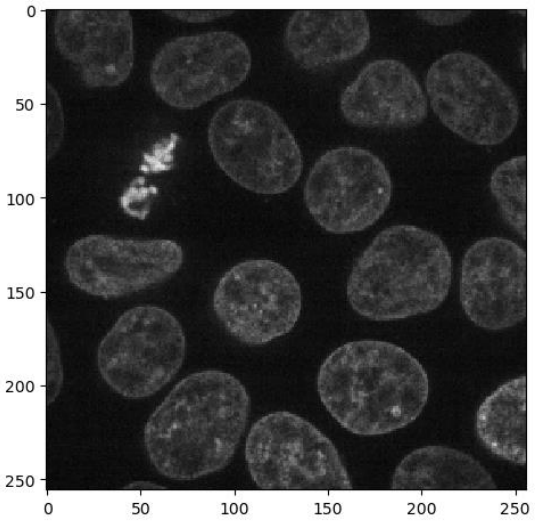


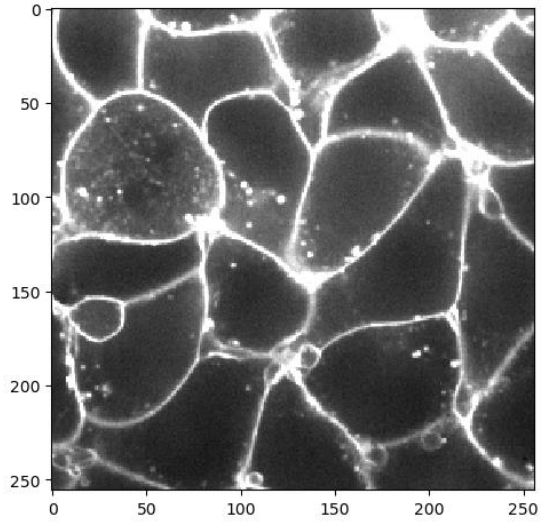


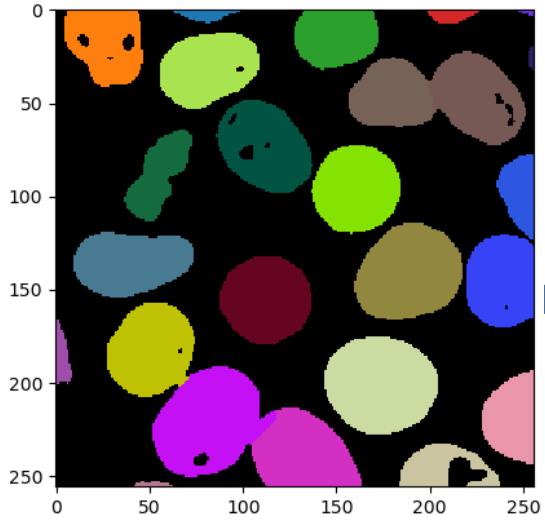
Results directly in a label image:
Tools > Segmentation / labeling >
Label touching objects

Watershed

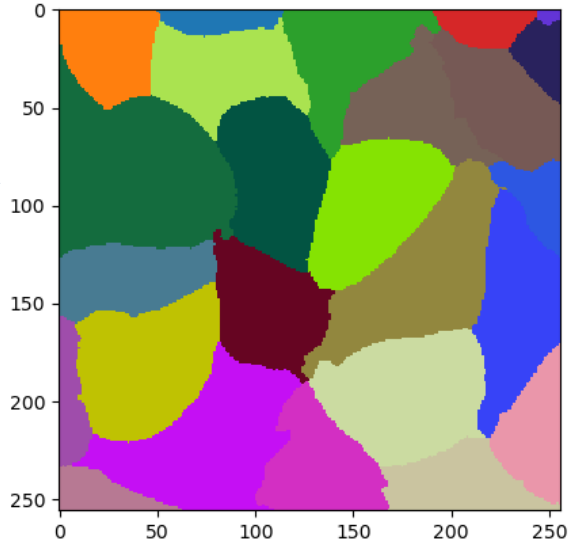
- … in Python practice







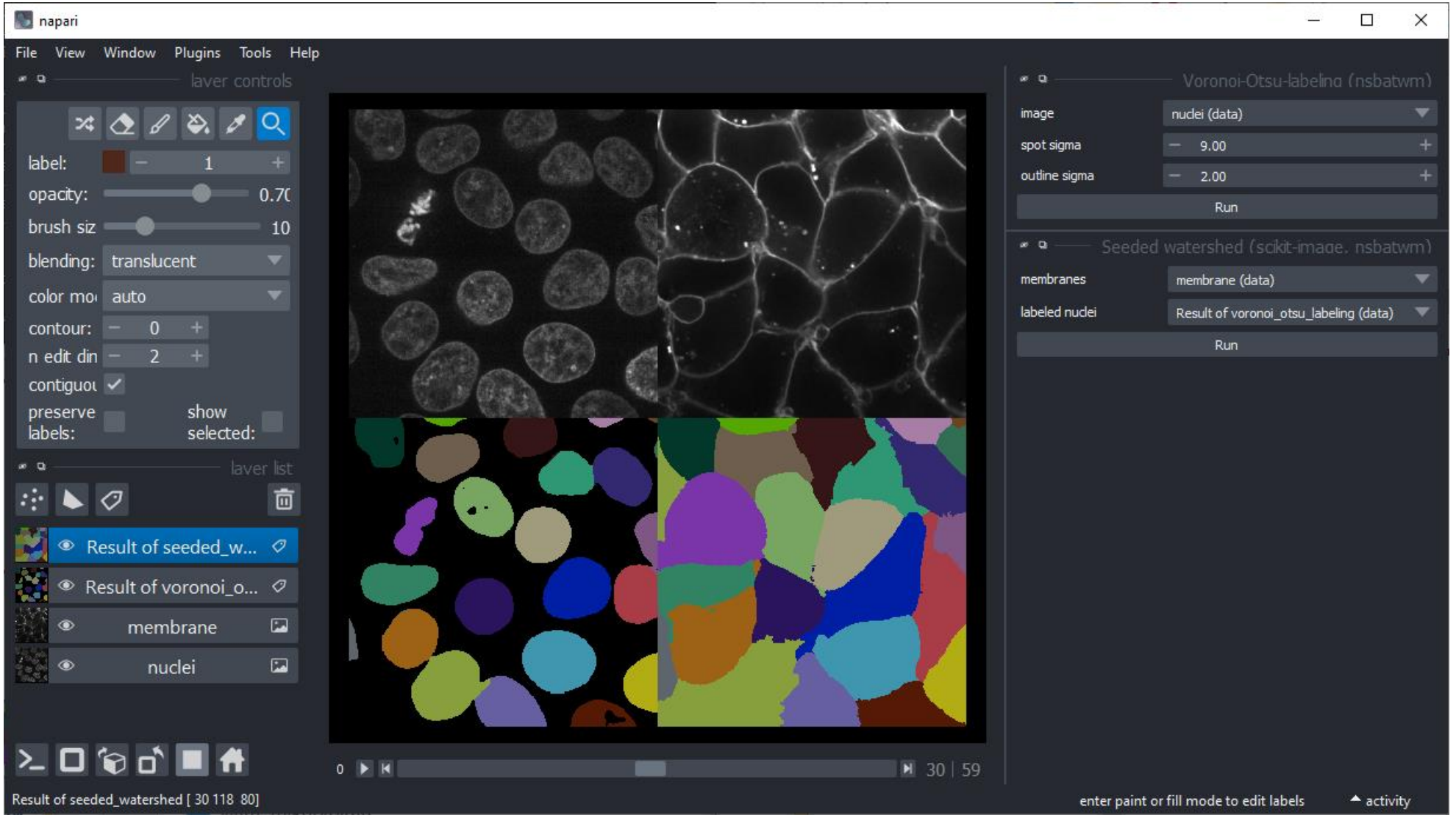

```
labeled_cells = seeded_watershed(membrane_channel, labeled_nuclei)

labeled_cells
```



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

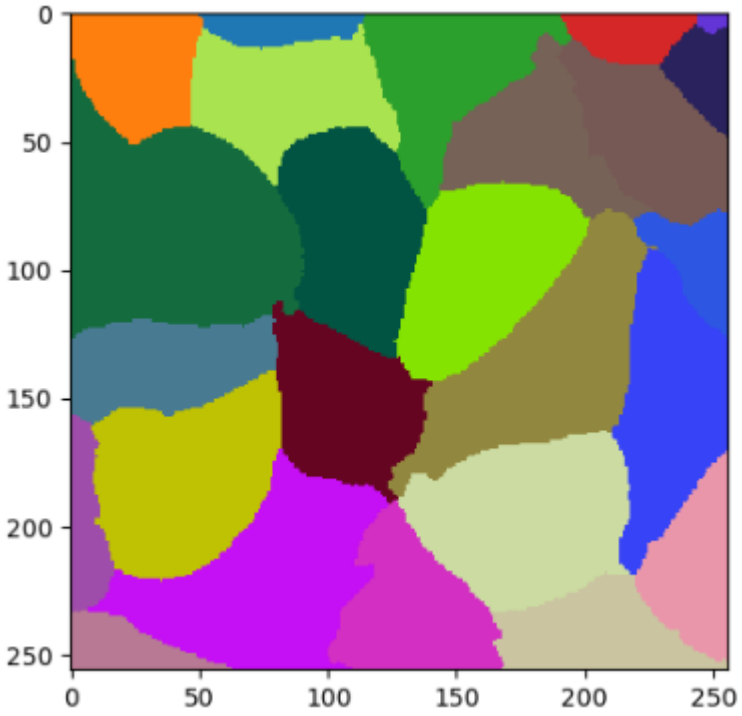
Watershed

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



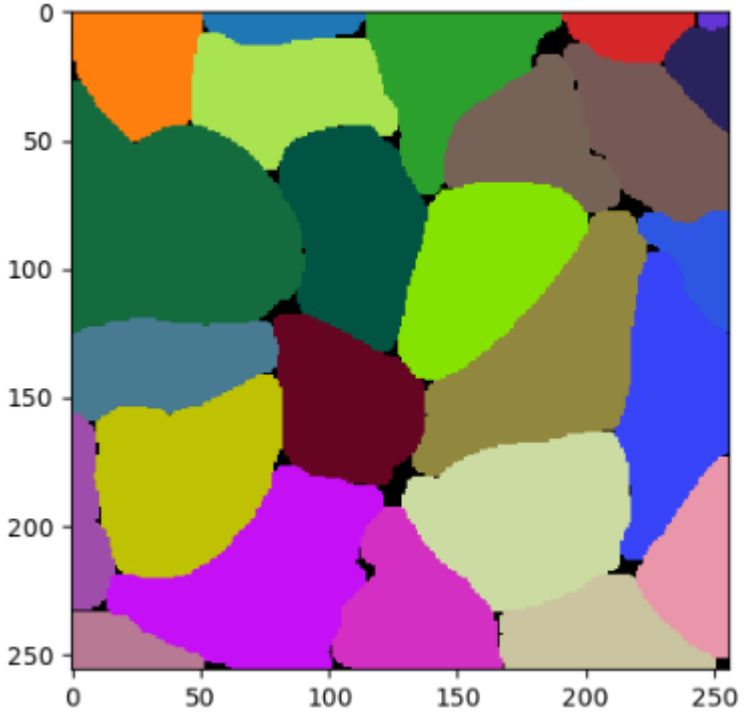
Label post-processing / morphological operations

- … similar to morphological operations on binary images



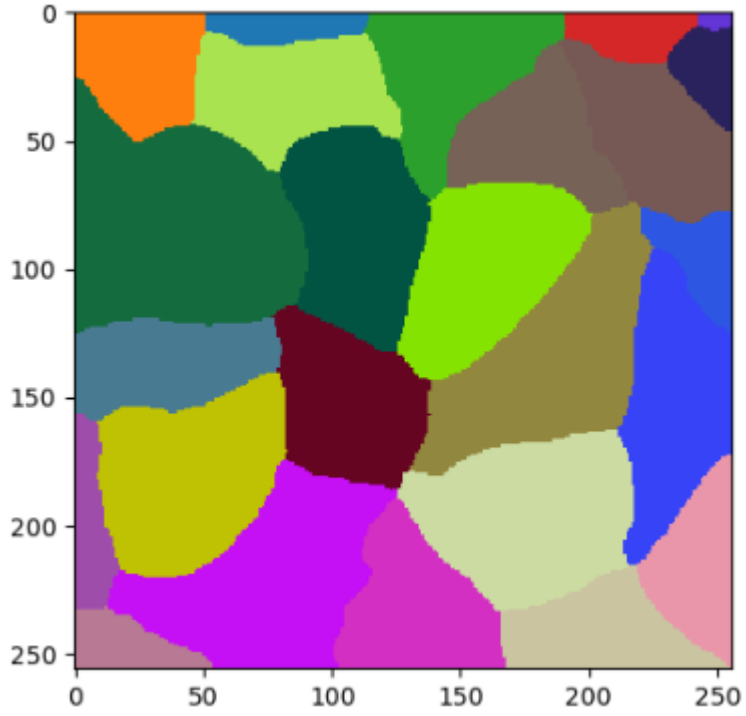
Original

```
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)