

# Handling image data in Napari

Robert Haase

With material from

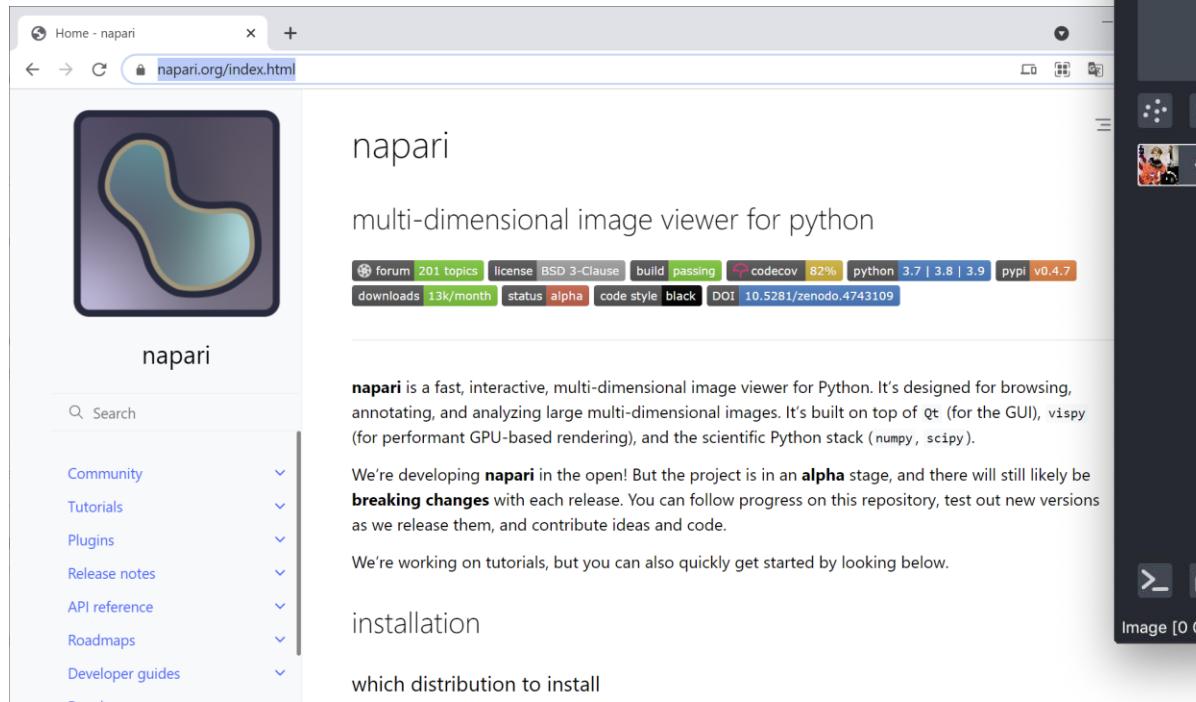
Marcelo Leomil Zoccoler, Physics of Life, TU Dresden

December 2022

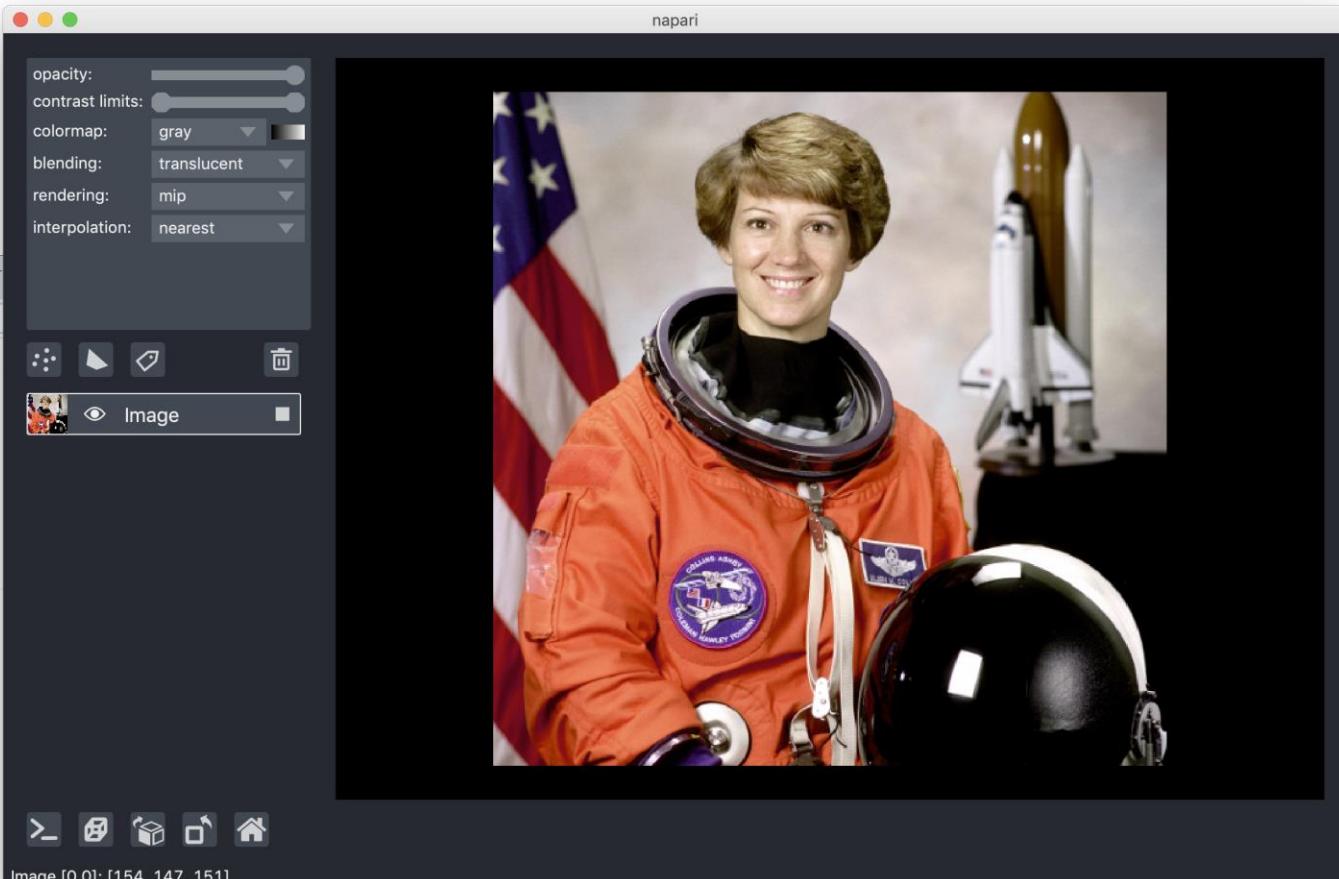
These slides are licensed CC-BY 4.0 unless mentioned otherwise

# Napari: 3D viewer for Python

- Multi-dimensional image
- <https://napari.org/>



The screenshot shows the official website for napari. At the top, there's a navigation bar with links for Home, Forum, License, Build, Codecov, Python versions, Downloads, Status, Code Style, and DOI. Below the navigation is a search bar and a sidebar with links for Community, Tutorials, Plugins, Release notes, API reference, Roadmaps, and Developer guides. The main content area features a large image of a green, blob-like multi-dimensional image, the text "napari multi-dimensional image viewer for python", and a brief description of what napari is and its development status.



<https://napari.org/>

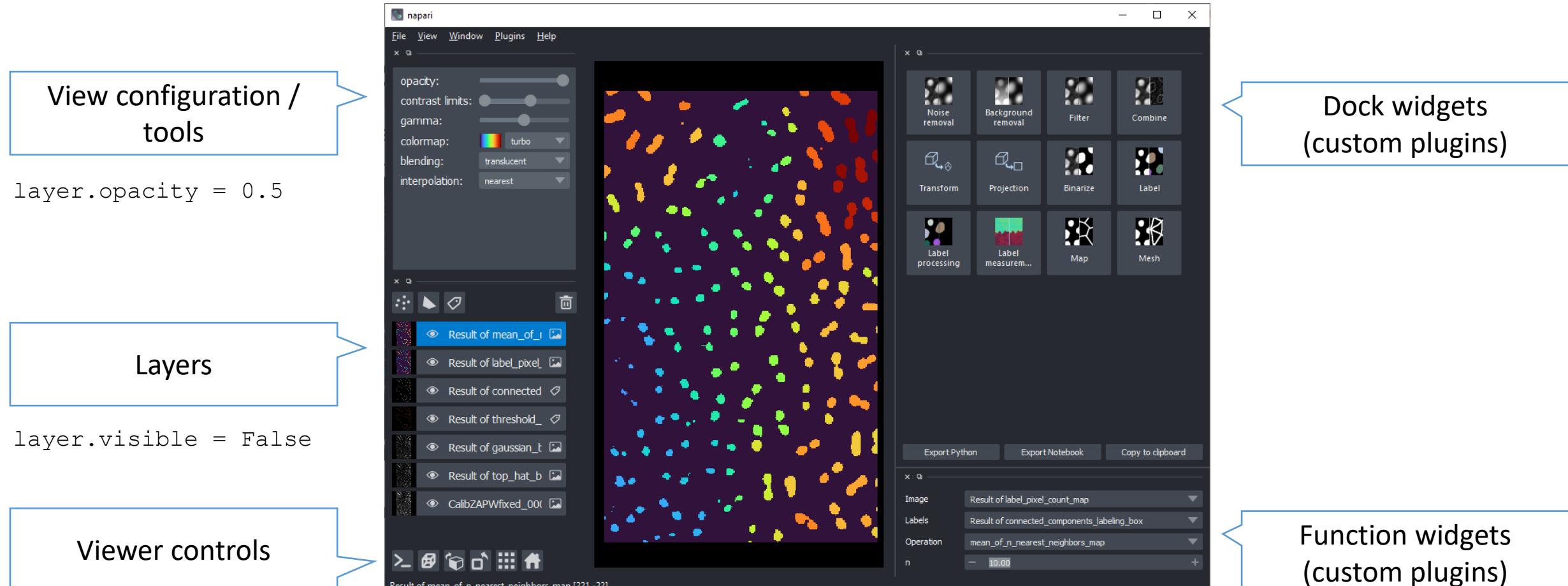
December 2022

# Napari: 3D viewer for Python



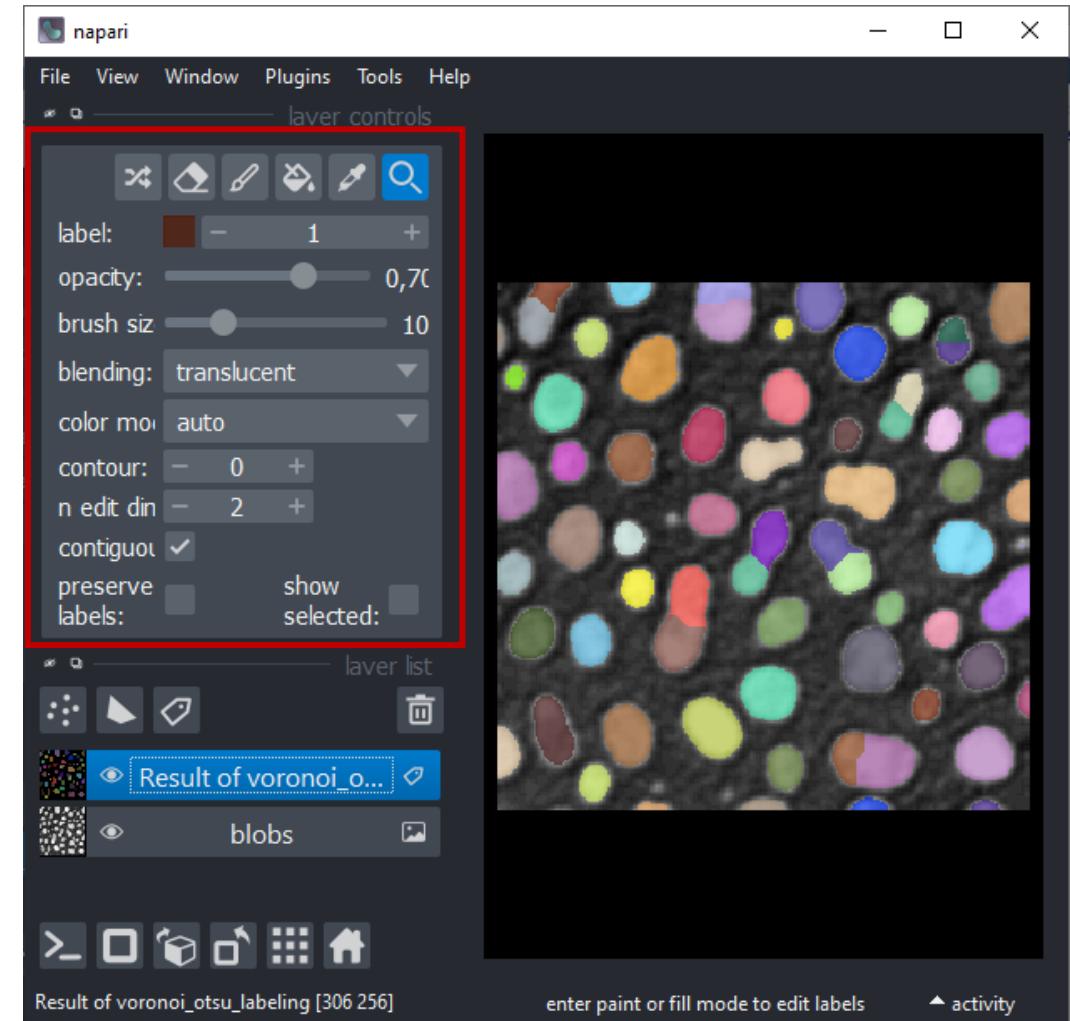
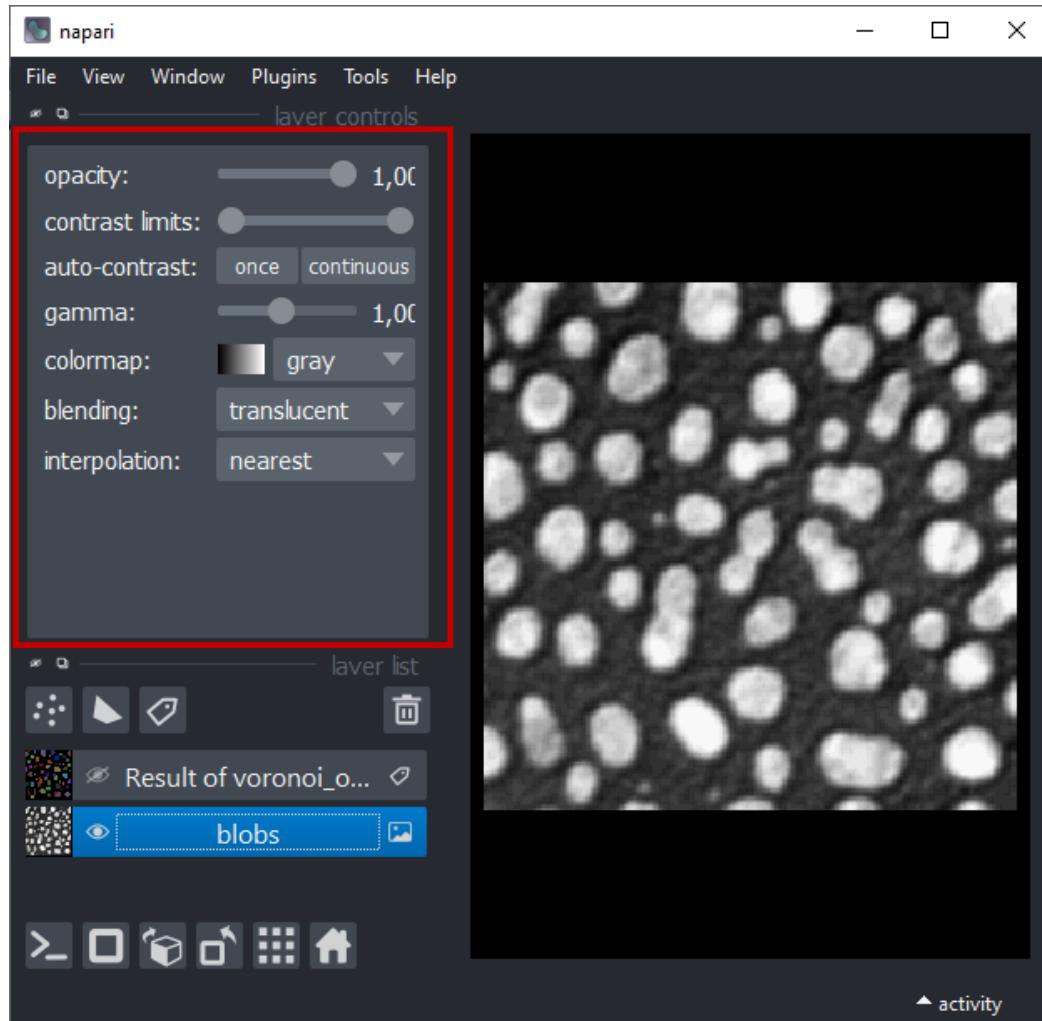
<https://napari.org/>

# Napari user interface



# Visualizing image segmentation

Different layers have different configurations



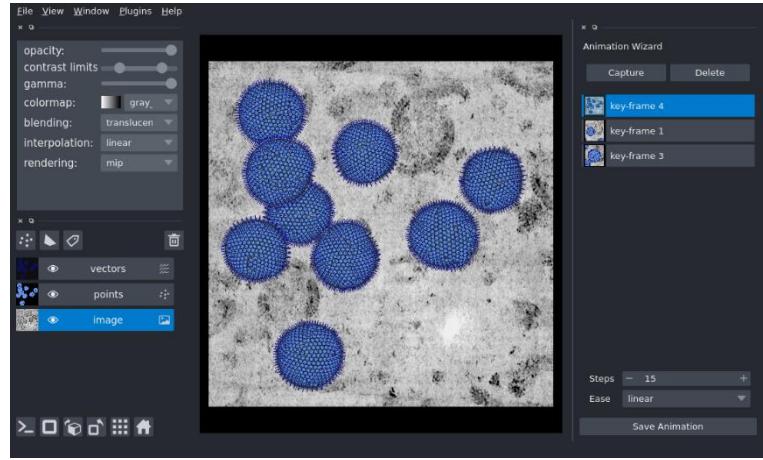
# napari plugins

## clusters-plotter



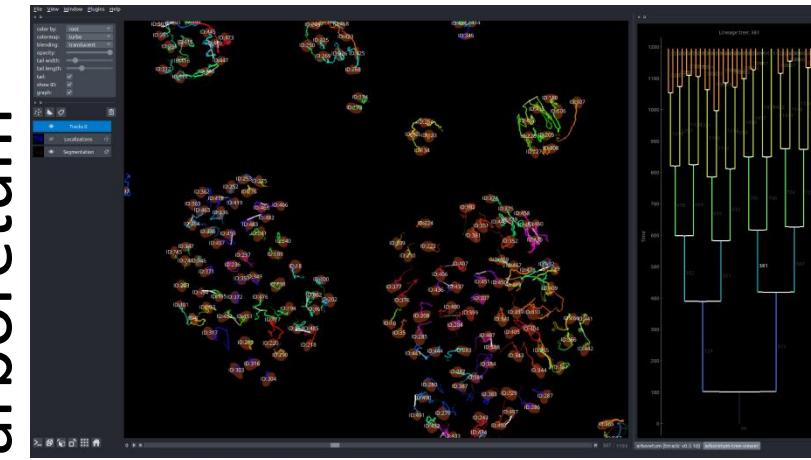
<https://github.com/BiAPoL/napari-clusters-plotter>

## animation



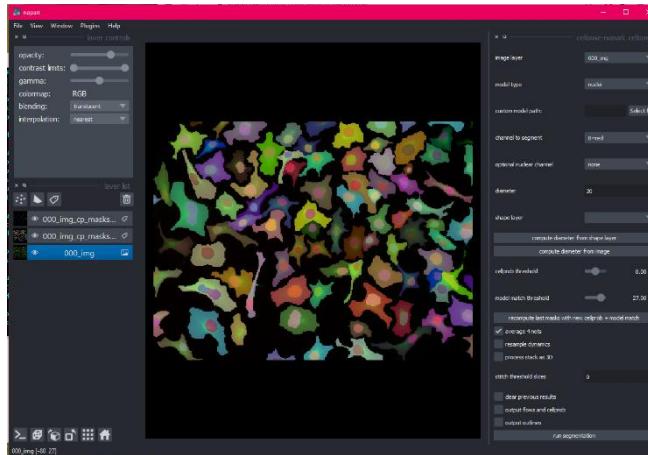
<https://github.com/napari/napari-animation>

## arboretum



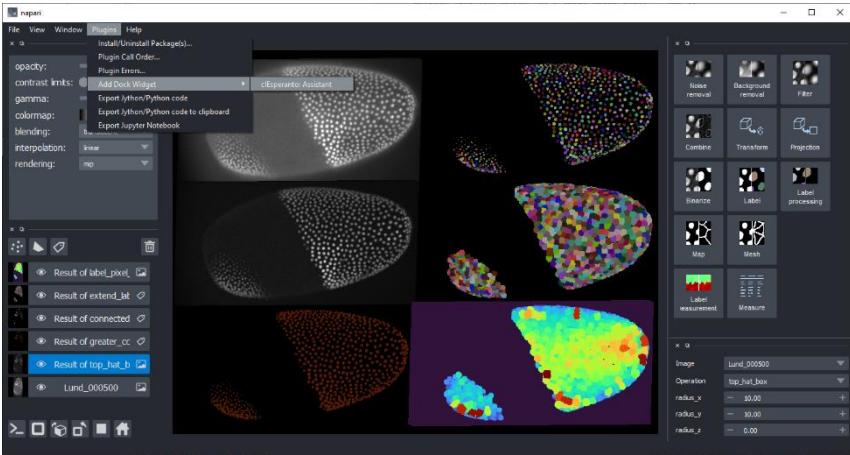
<https://github.com/quantumjot/arboretum>

## cellpose



<https://cellpose-napari.readthedocs.io/en/latest/>

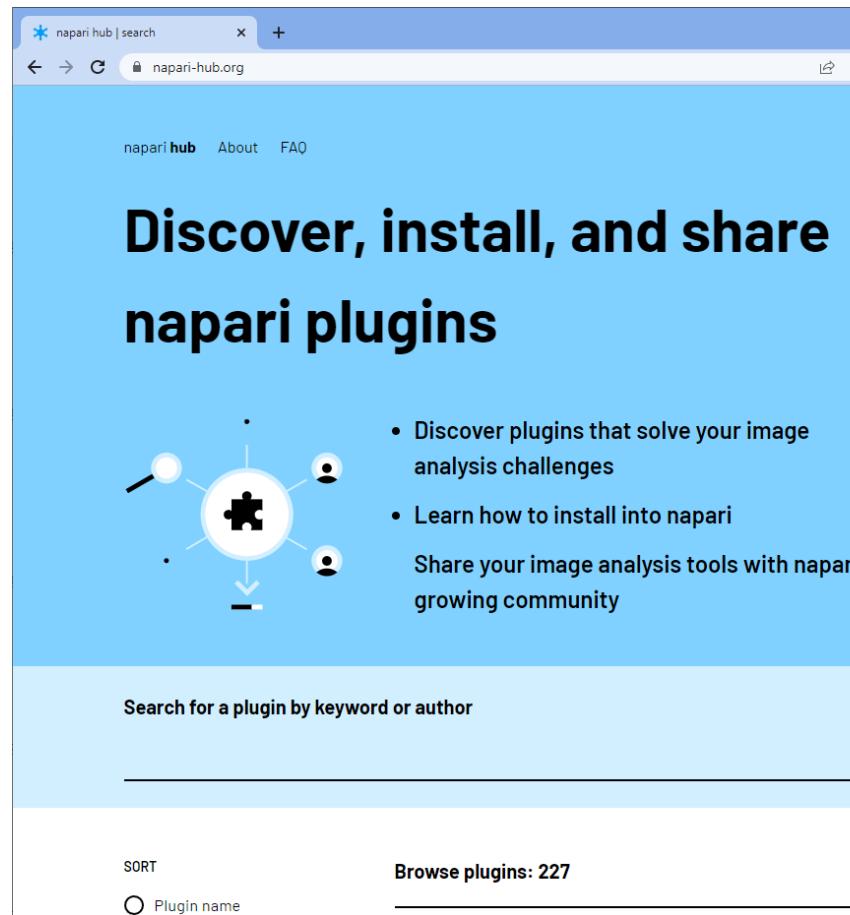
## clesperanto



[https://github.com/c1Esperanto/napari\\_pyclesperanto\\_assistant](https://github.com/c1Esperanto/napari_pyclesperanto_assistant)

# The Napari Hub

- Search engine for napari plugins



napari hub | search napari-hub.org

Discover, install, and share napari plugins

Discover plugins that solve your image analysis challenges

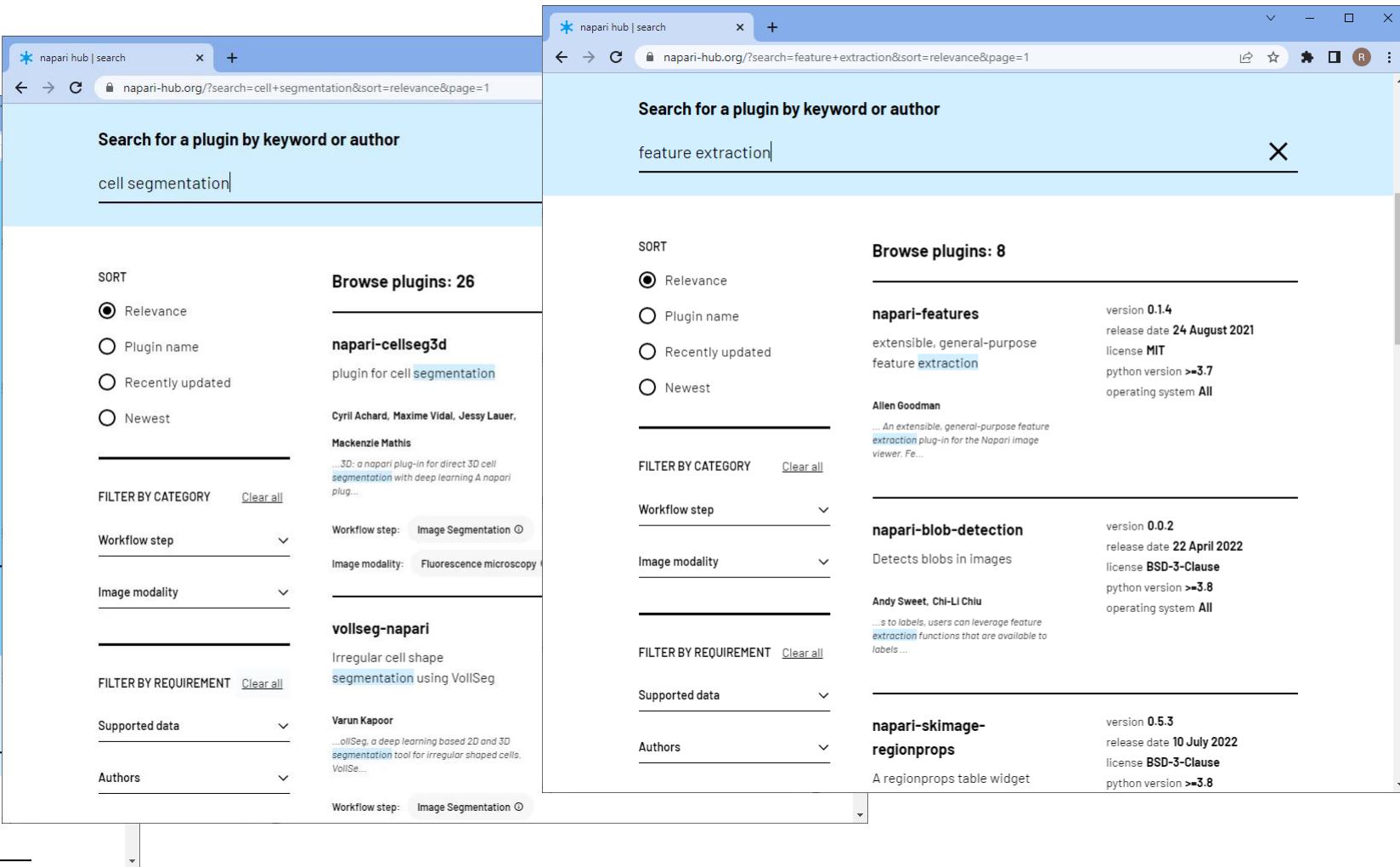
Learn how to install into napari

Share your image analysis tools with napari's growing community

Search for a plugin by keyword or author

SORT  Plugin name

Browse plugins: 227



napari hub | search napari-hub.org/?search=cell+segmentation&sort=relevance&page=1

Search for a plugin by keyword or author

cell segmentation

SORT  Relevance  Plugin name  Recently updated  Newest

Browse plugins: 26

**napari-cellseg3d**  
plugin for cell segmentation  
Cyril Achard, Maxime Vidal, Jessy Lauer, Mackenzie Mathis  
...3D: a napari plug-in for direct 3D cell segmentation with deep learning A napari plug...

**vollseg-napari**  
Irregular cell shape segmentation using VollSeg  
Varun Kapoor  
...llSeg: a deep learning based 2D and 3D segmentation tool for irregular shaped cells. VollSe...

FILTER BY CATEGORY [Clear all](#)

Workflow step: Image Segmentation

Image modality: Fluorescence microscopy

FILTER BY REQUIREMENT [Clear all](#)

Supported data

Authors

Workflow step: Image Segmentation

napari-hub | search napari-hub.org/?search=feature+extraction&sort=relevance&page=1

Search for a plugin by keyword or author

feature extraction

SORT  Relevance  Plugin name  Recently updated  Newest

Browse plugins: 8

**napari-features**  
version 0.1.4  
release date 24 August 2021  
license MIT  
python version >=3.7  
operating system All  
extensible, general-purpose feature extraction

**Allen Goodman**  
An extensible, general-purpose feature extraction plug-in for the Napari image viewer. Fe...

**napari-blob-detection**  
version 0.0.2  
release date 22 April 2022  
license BSD-3-Clause  
python version >=3.8  
operating system All  
Detects blobs in images

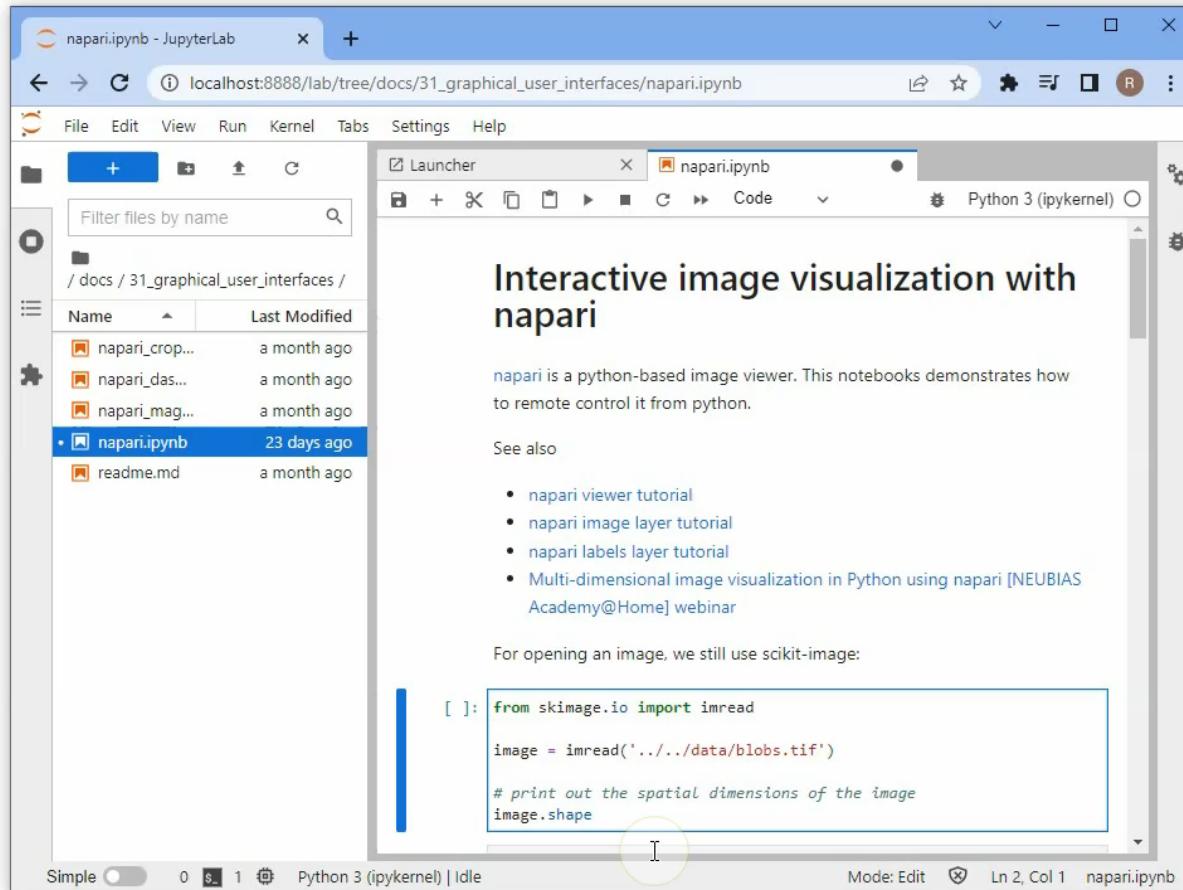
**Andy Sweet, Chi-Li Chiu**  
...s to labels, users can leverage feature extraction functions that are available to labels ...

**napari-skimage-regionprops**  
version 0.5.3  
release date 10 July 2022  
license BSD-3-Clause  
python version >=3.8  
A regionprops table widget

<https://www.napari-hub.org/>

# Using Napari from Python Code

- A great mix of interactivity and reproducibility



# Scripting napari

- Initialization

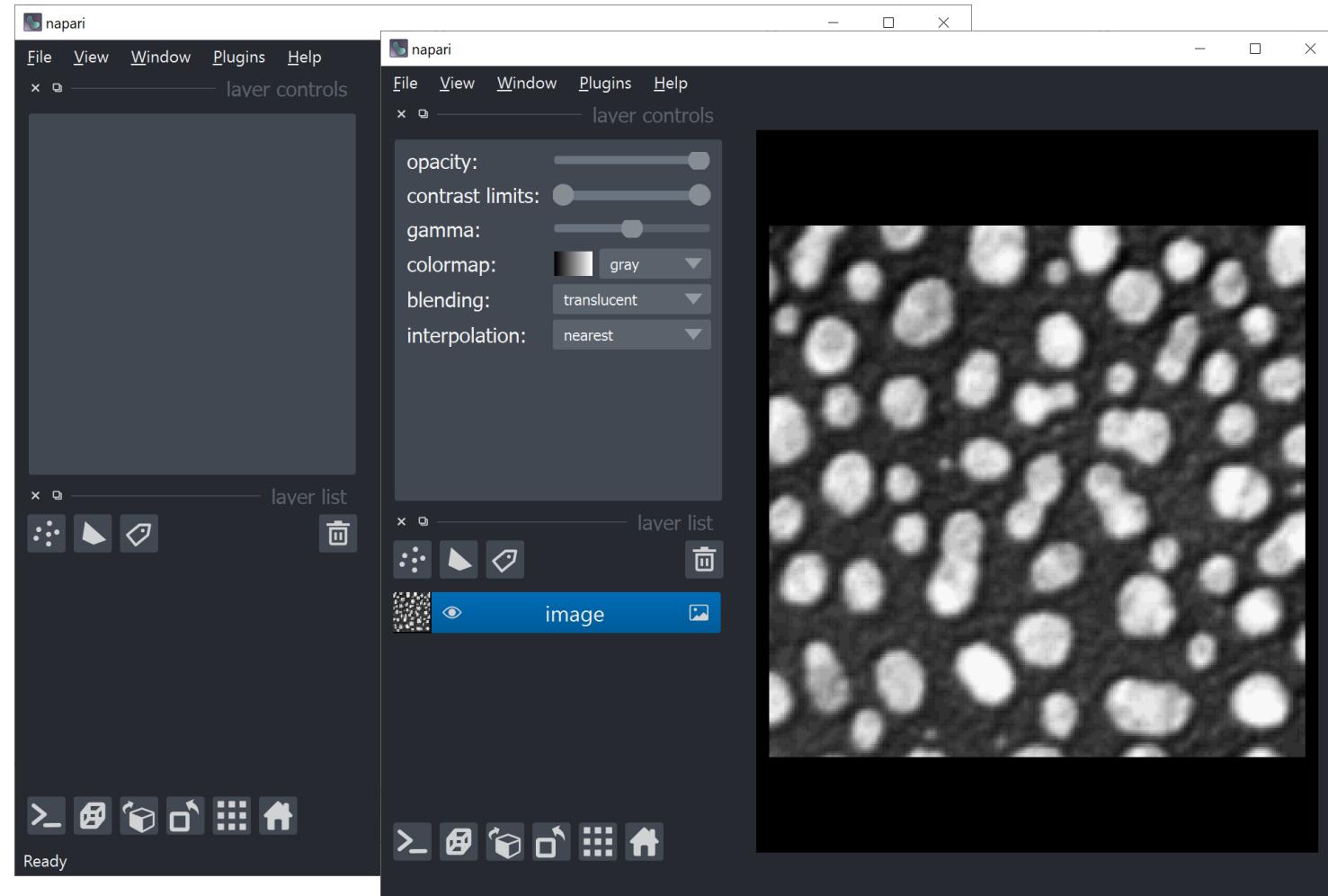
```
import napari
```

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

```
# Start it
napari.run()
```

- Adding images

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

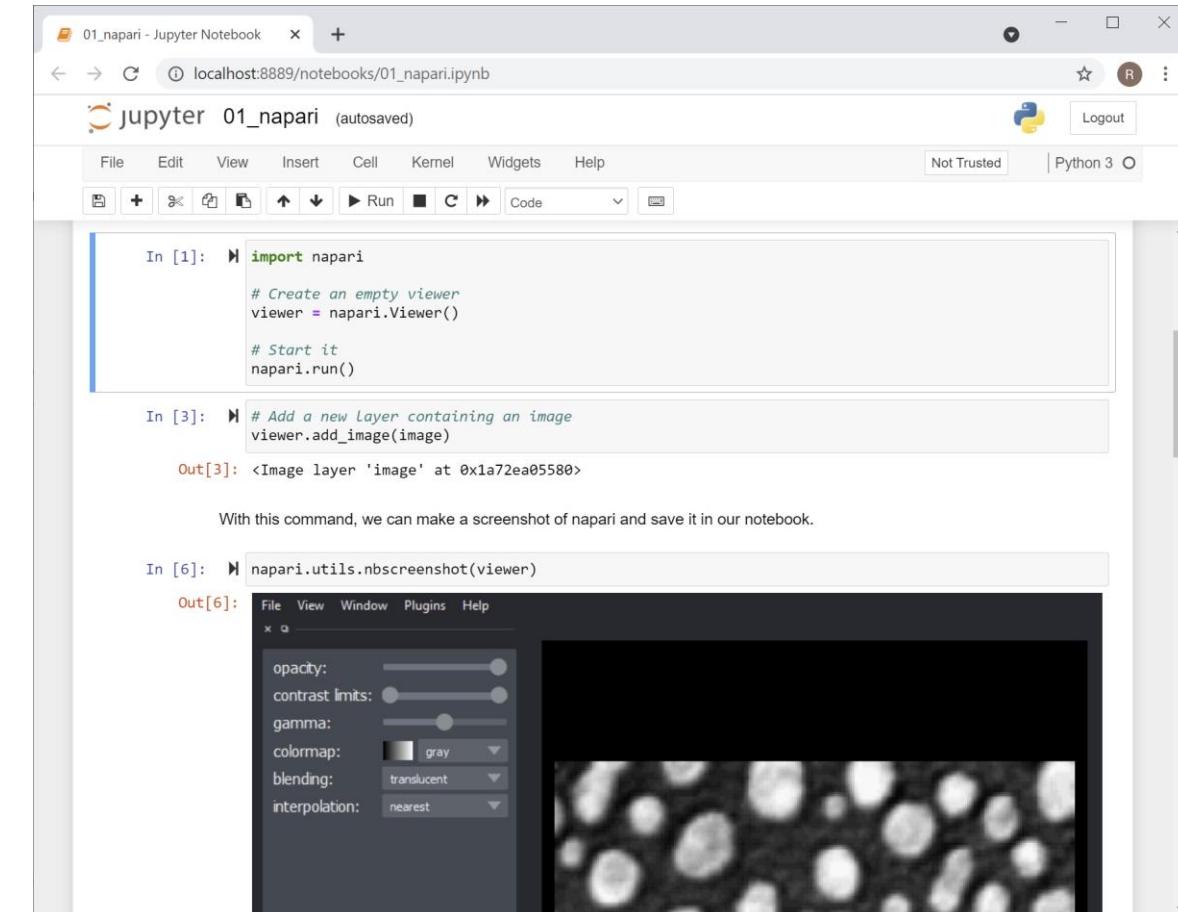


# Scripting napari in notebooks

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

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

Place your viewer here



# Working with layers

- Removing layers

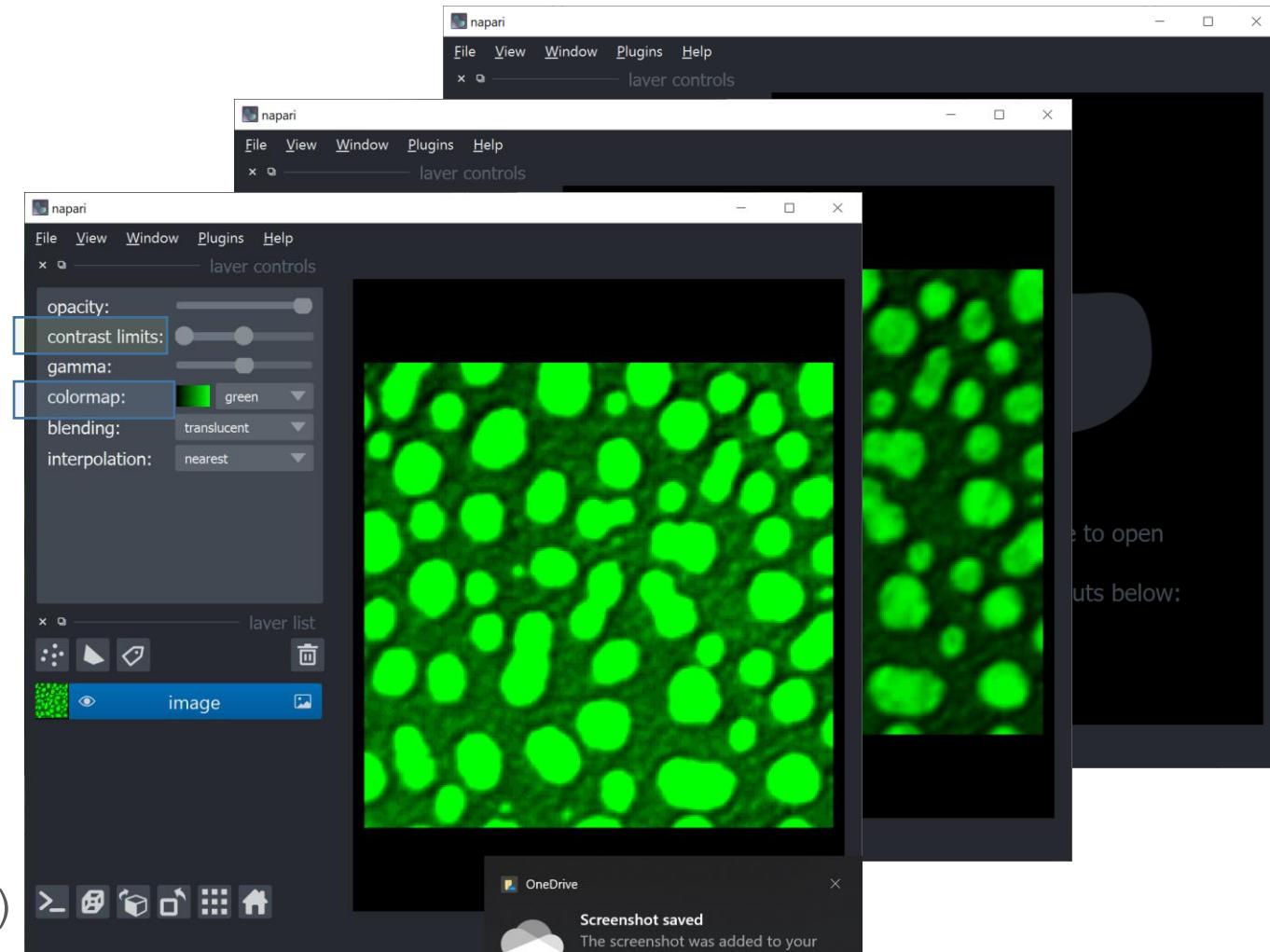
```
for l in viewer.layers:  
    viewer.layers.remove(l)
```

- 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)
```



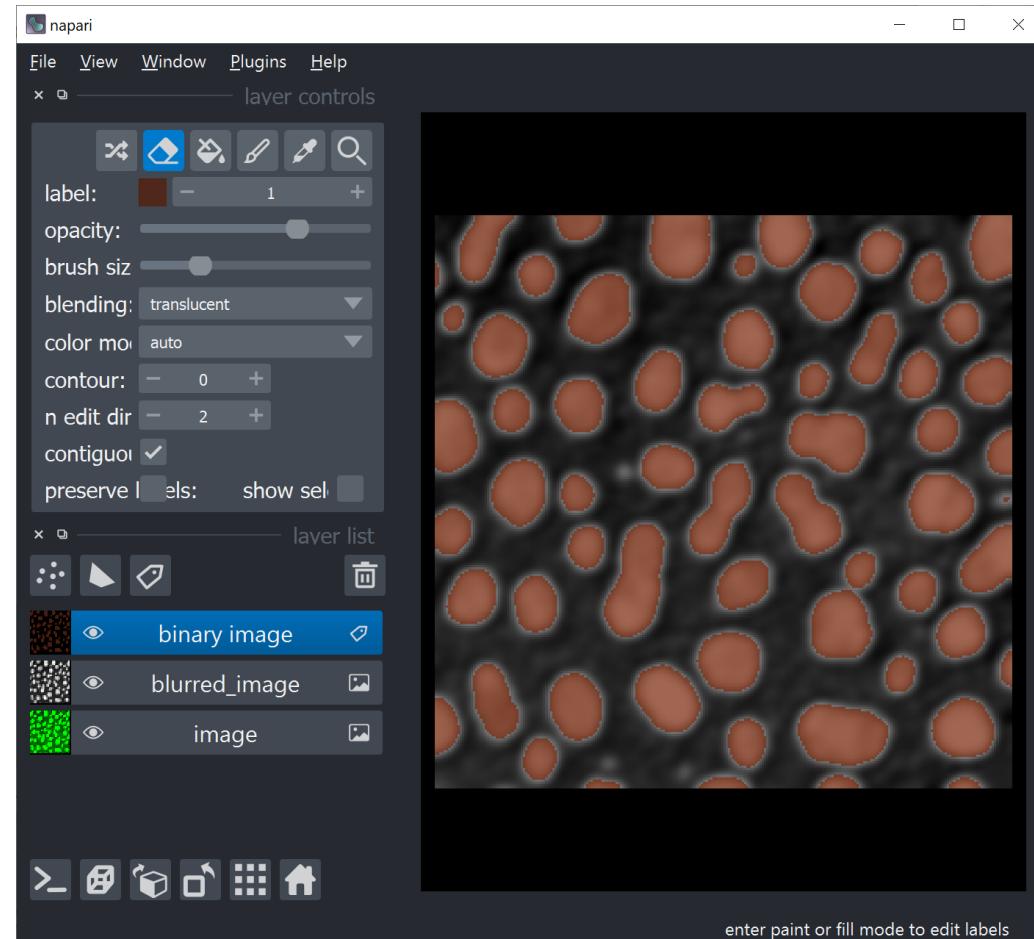
# Visualizing image segmentation

- 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,
                  name="binary image")
```

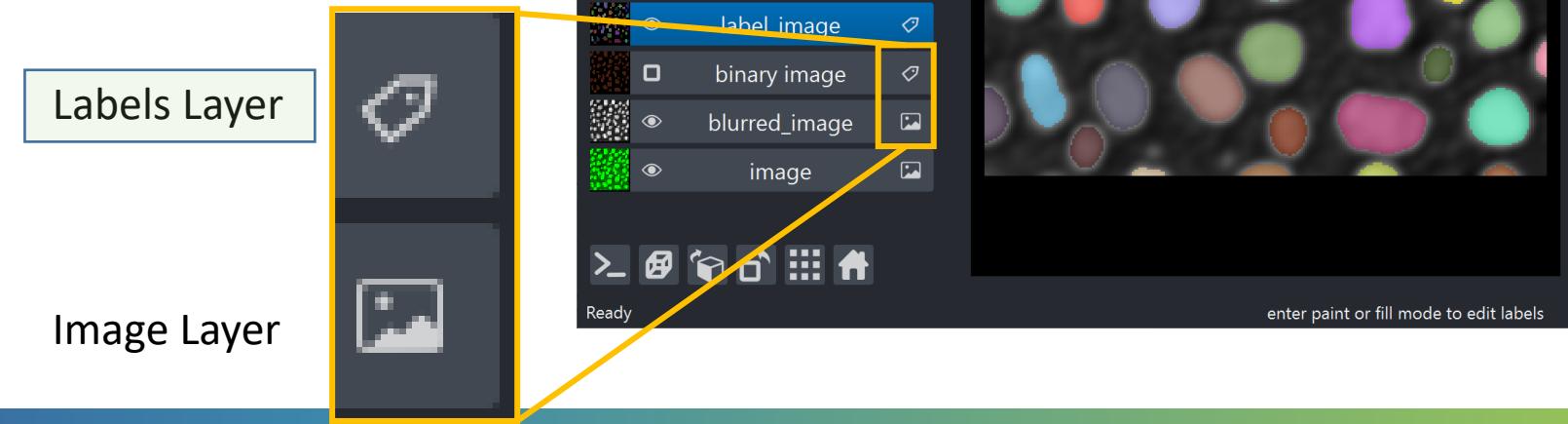
Name your layers to keep track  
of what they contain



# Visualizing image segmentation

- Binary images and label images visualized as label layers

```
from skimage.measure import label  
  
label_image = label(binary_image)
```



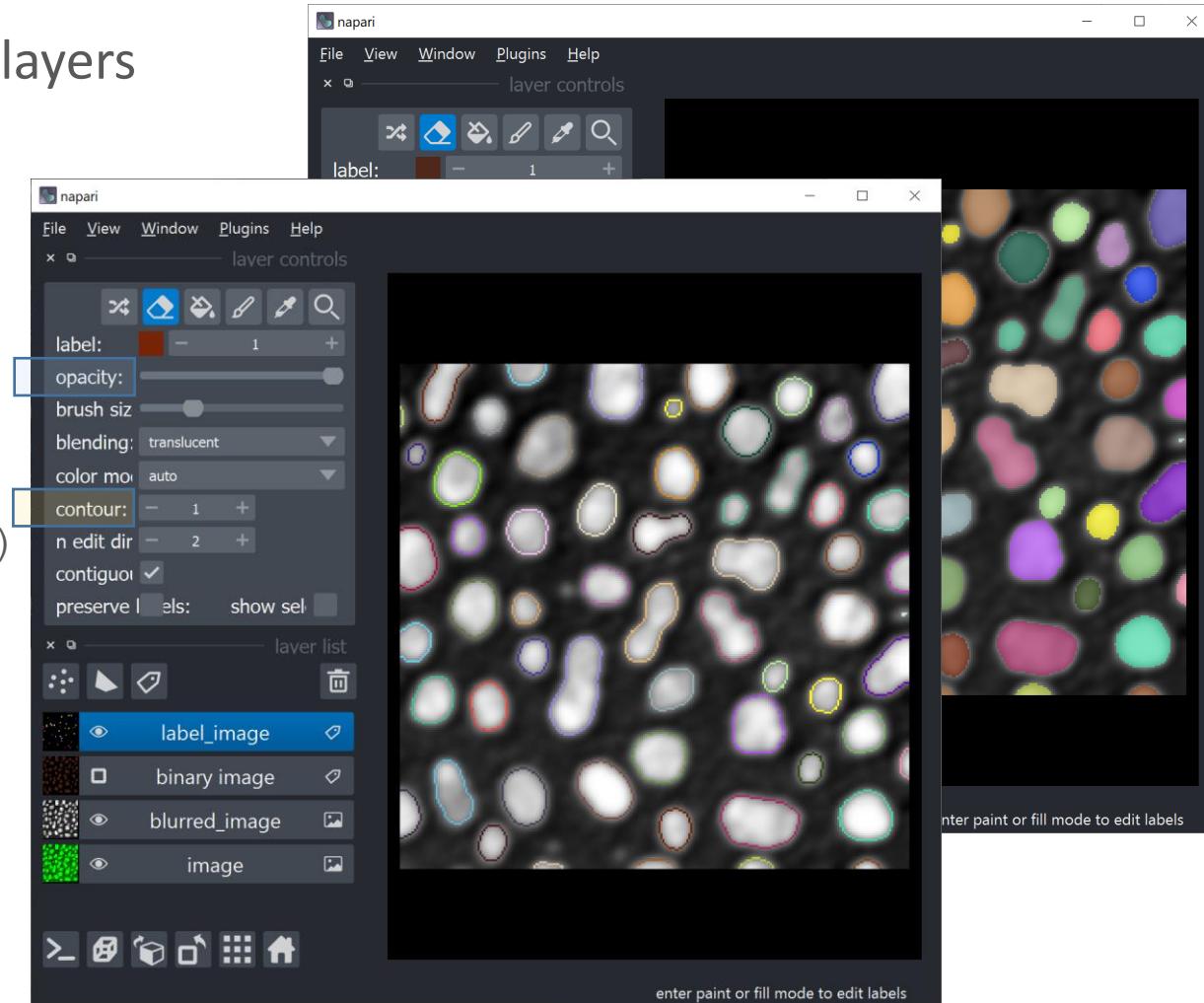
# Visualizing image segmentation

- Binary images and label images visualized as label layers

```
from skimage.measure import label  
  
label_image = label(binary_image)  
  
# add labels to viewer  
  
label_layer = viewer.add_labels(label_image)
```

- Visualize contours instead of the overlay

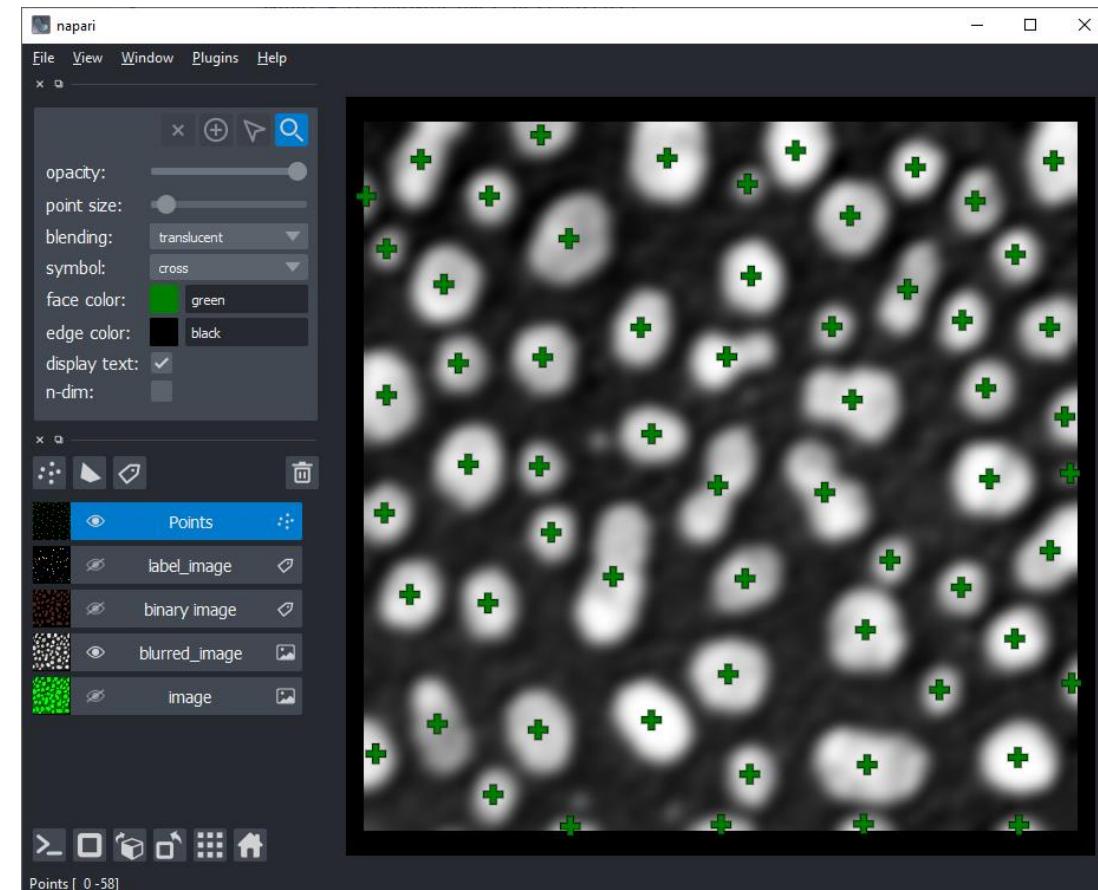
```
label_layer.contour = 1  
  
label_layer.opacity = 1
```



# Points layers

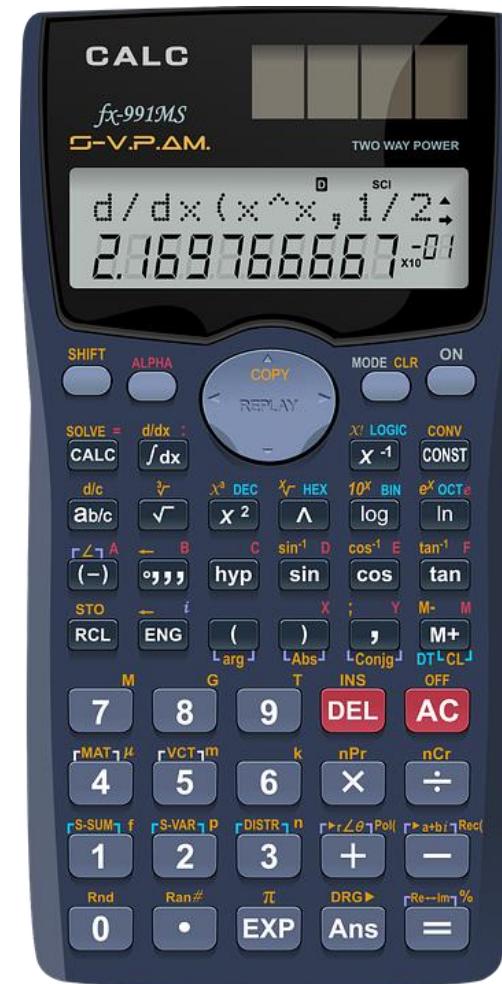
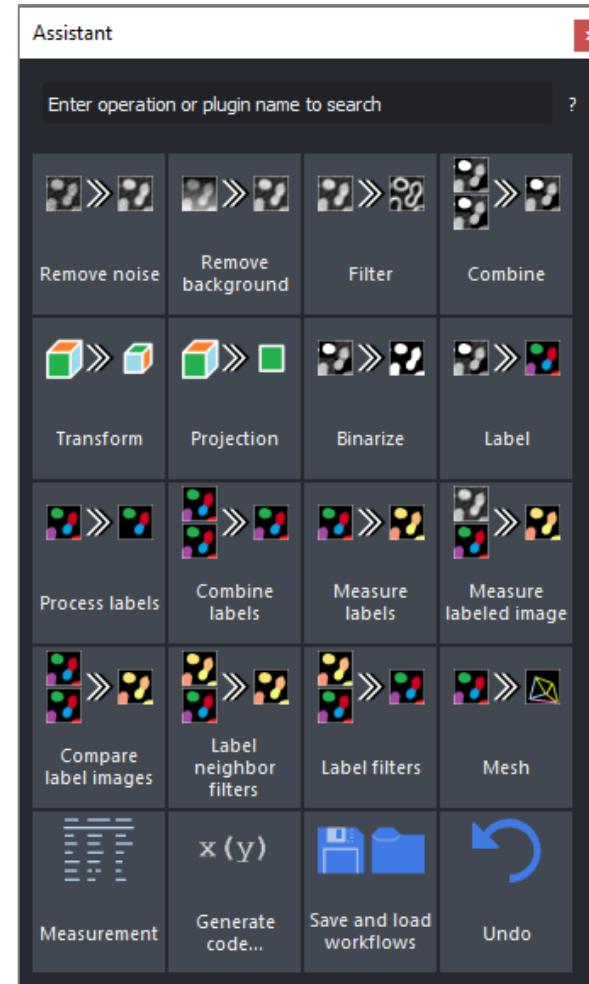
- There is also other layer types
  - Shapes
  - Points
  - Surfaces
  - Tracks
  - Vectors

```
from skimage.measure import regionprops
statistics = regionprops(label_image)
points = [s.centroid for s in statistics]
# add points to viewer
label_layer = viewer.add_points(points, face_color='green', symbol='cross', size=5)
```

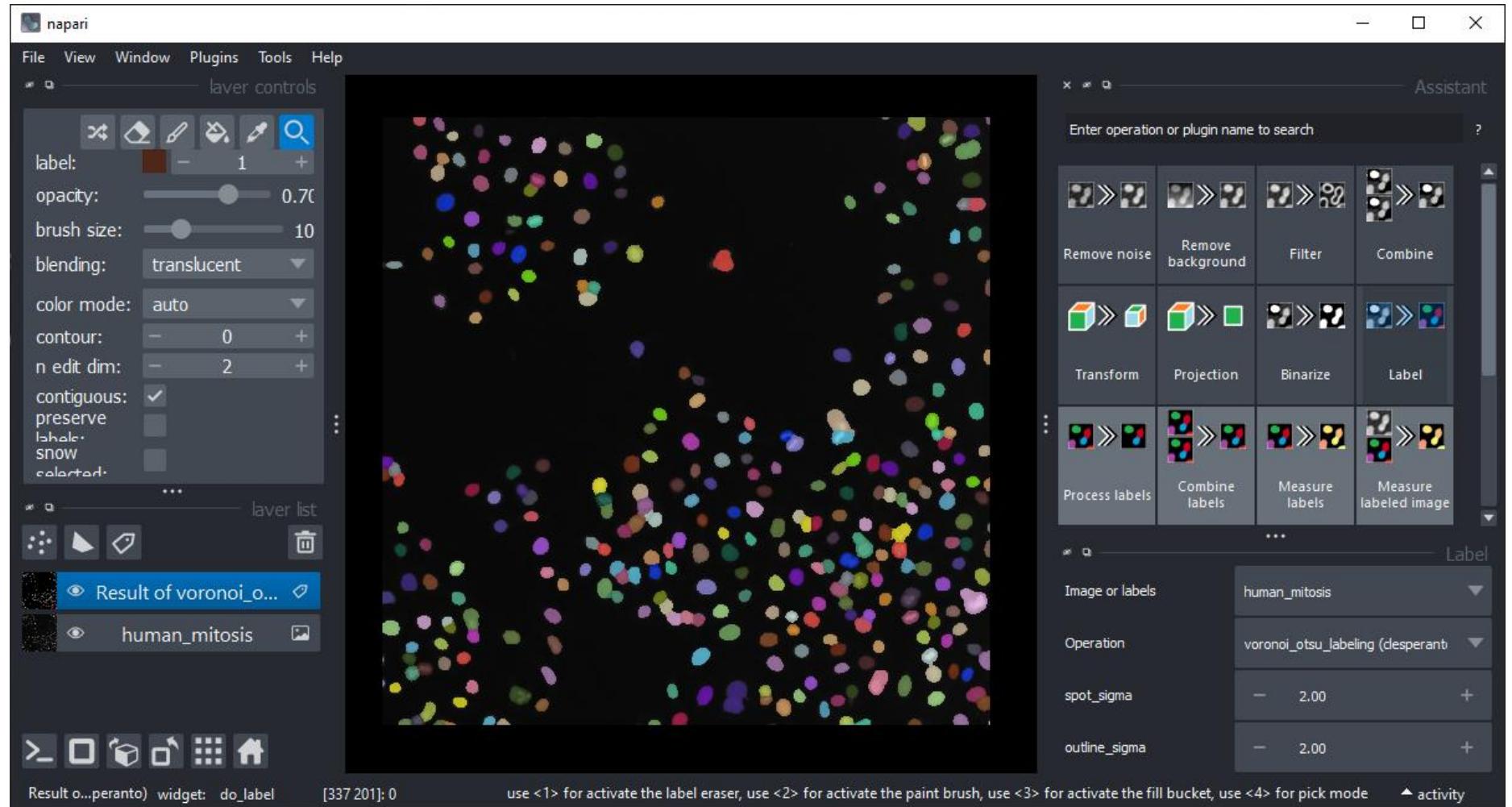


# The Napari Assistant

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



# The Napari Assistant



Viewer  
controls  
(Napari core)

Image  
Analysis  
(Napari Assistant)

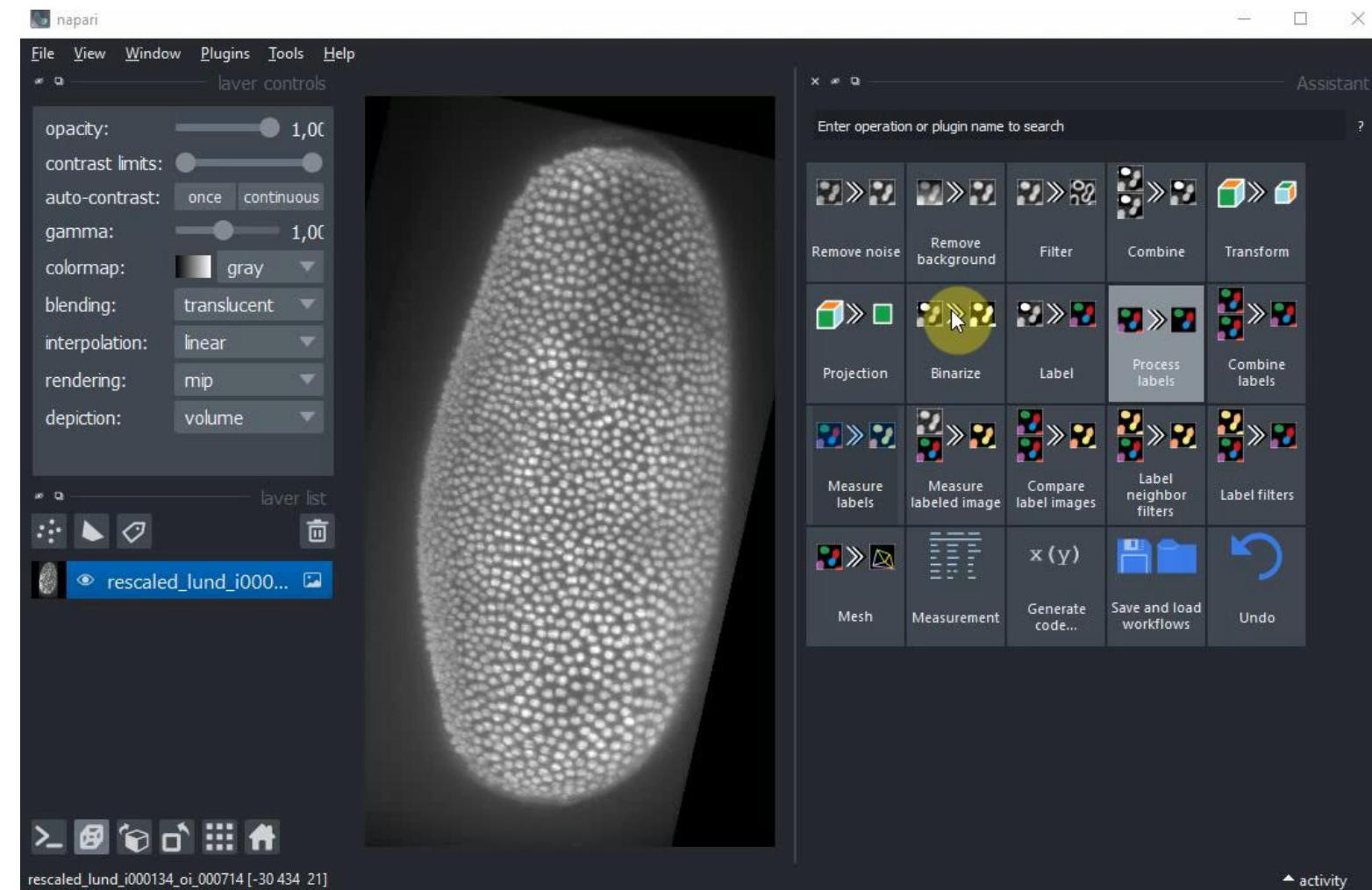
# The Napari Assistant

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

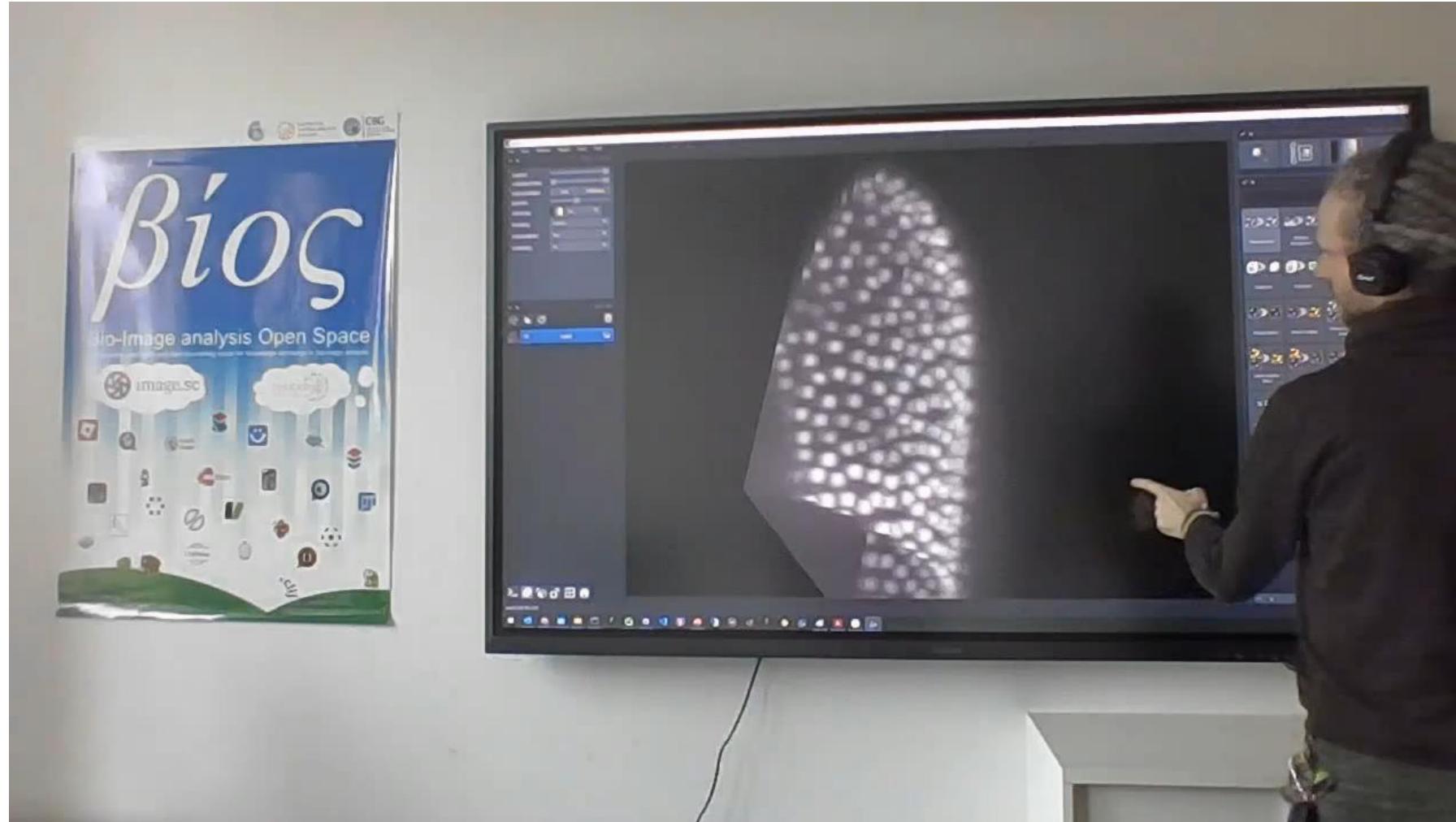
Big thanks to:



Ryan Savill  
(now at MPI-CBG) @RyanSavill4

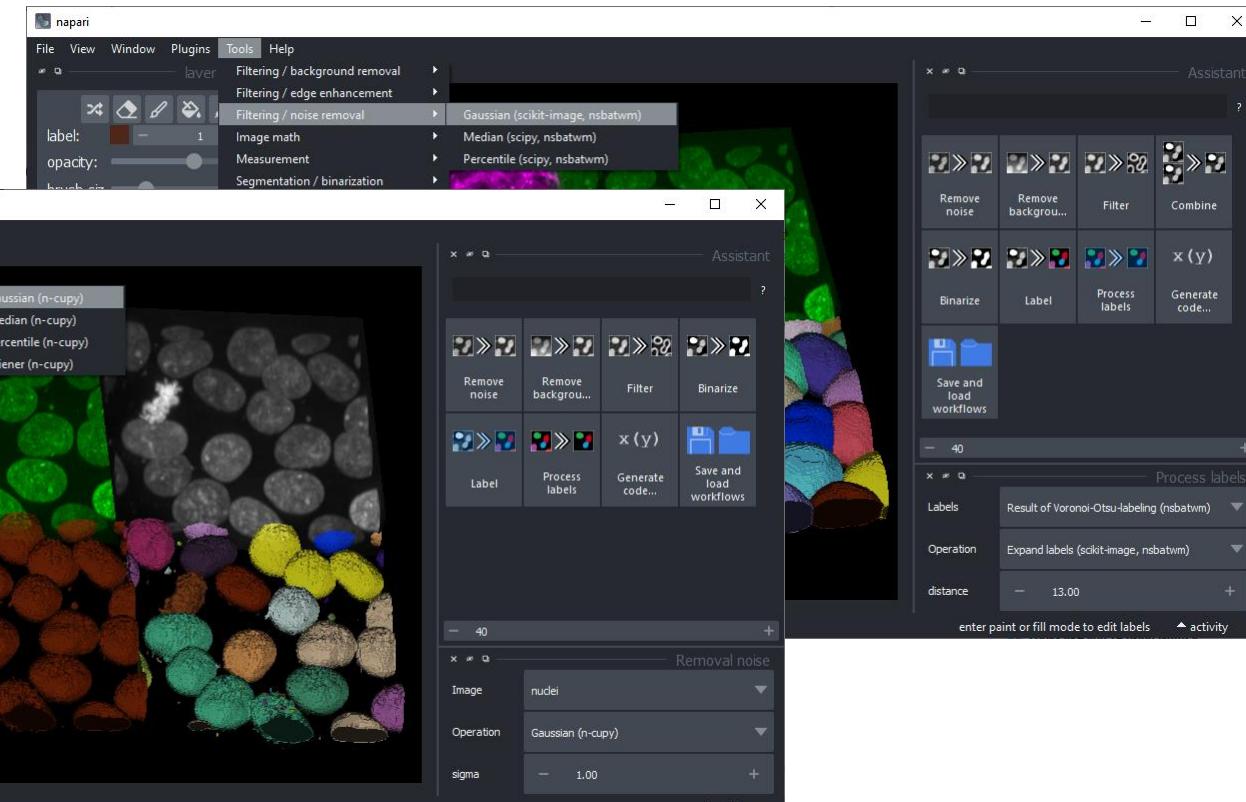
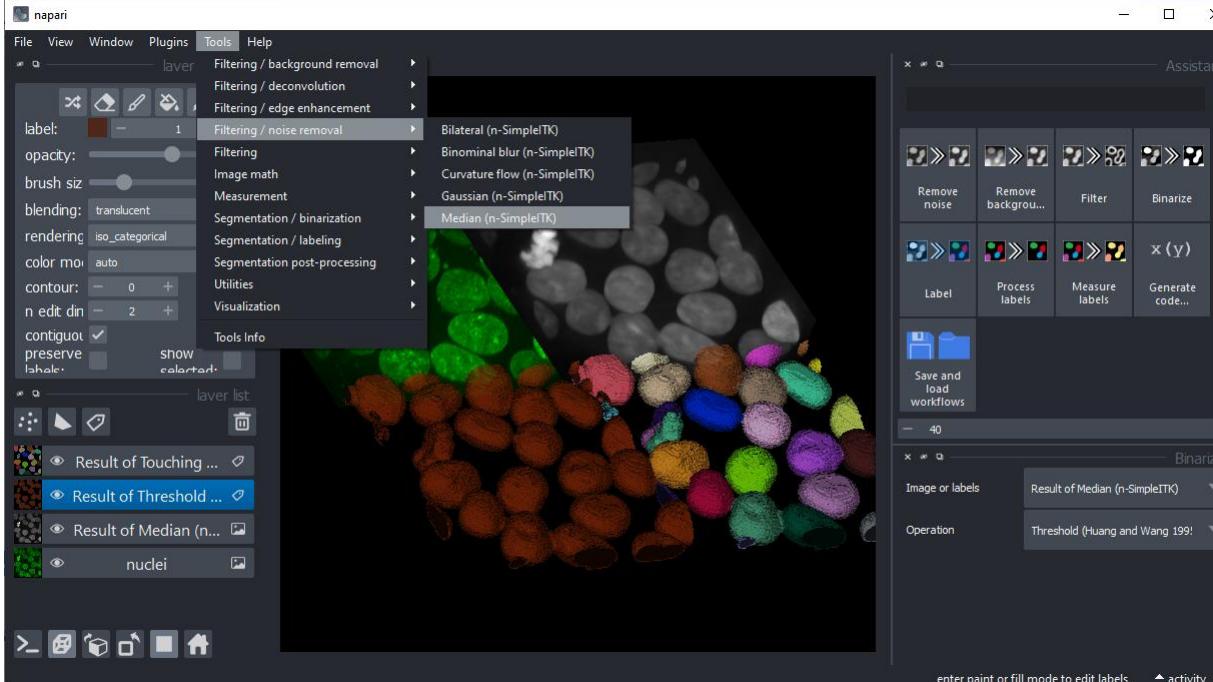


# 3D image visualization + interaction



# Napari-Assistant compatible plugins

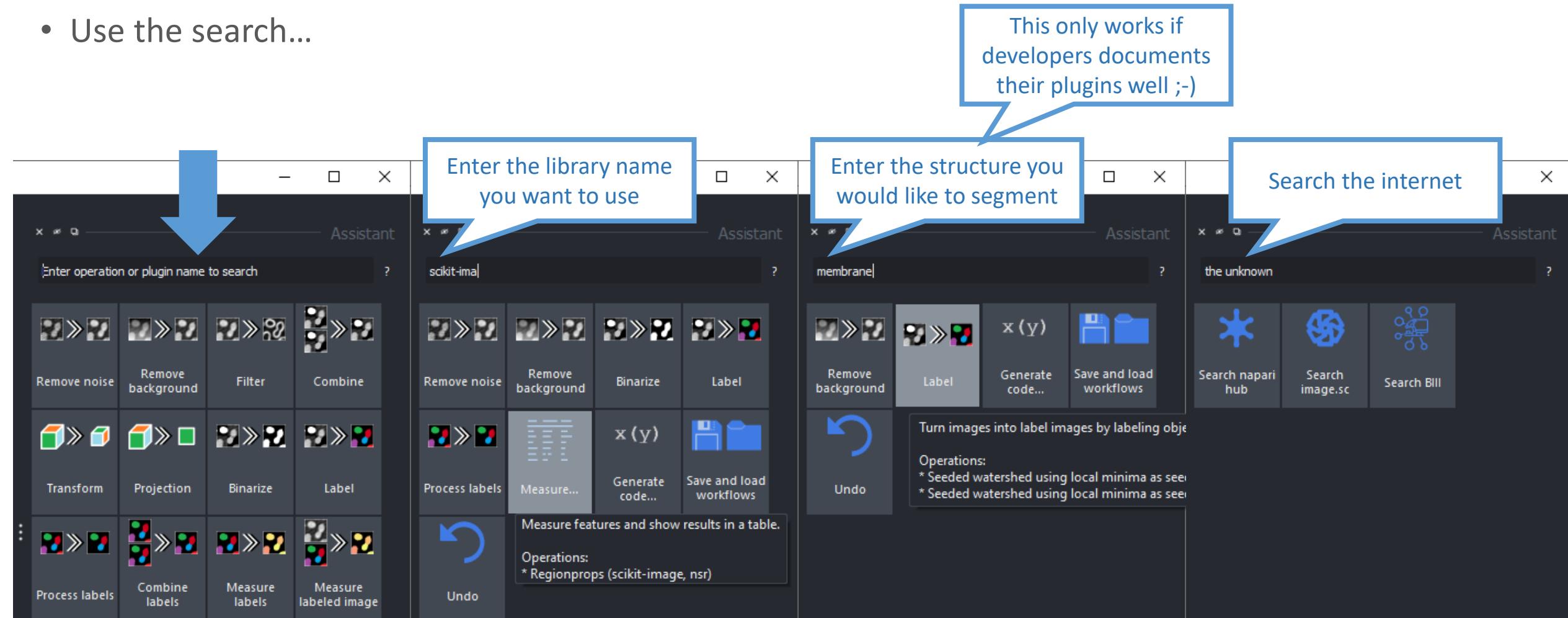
- Classical image-processing algorithms
- Based on scikit-image, scipy, numpy, cupy, clesperanto and SimpleITK



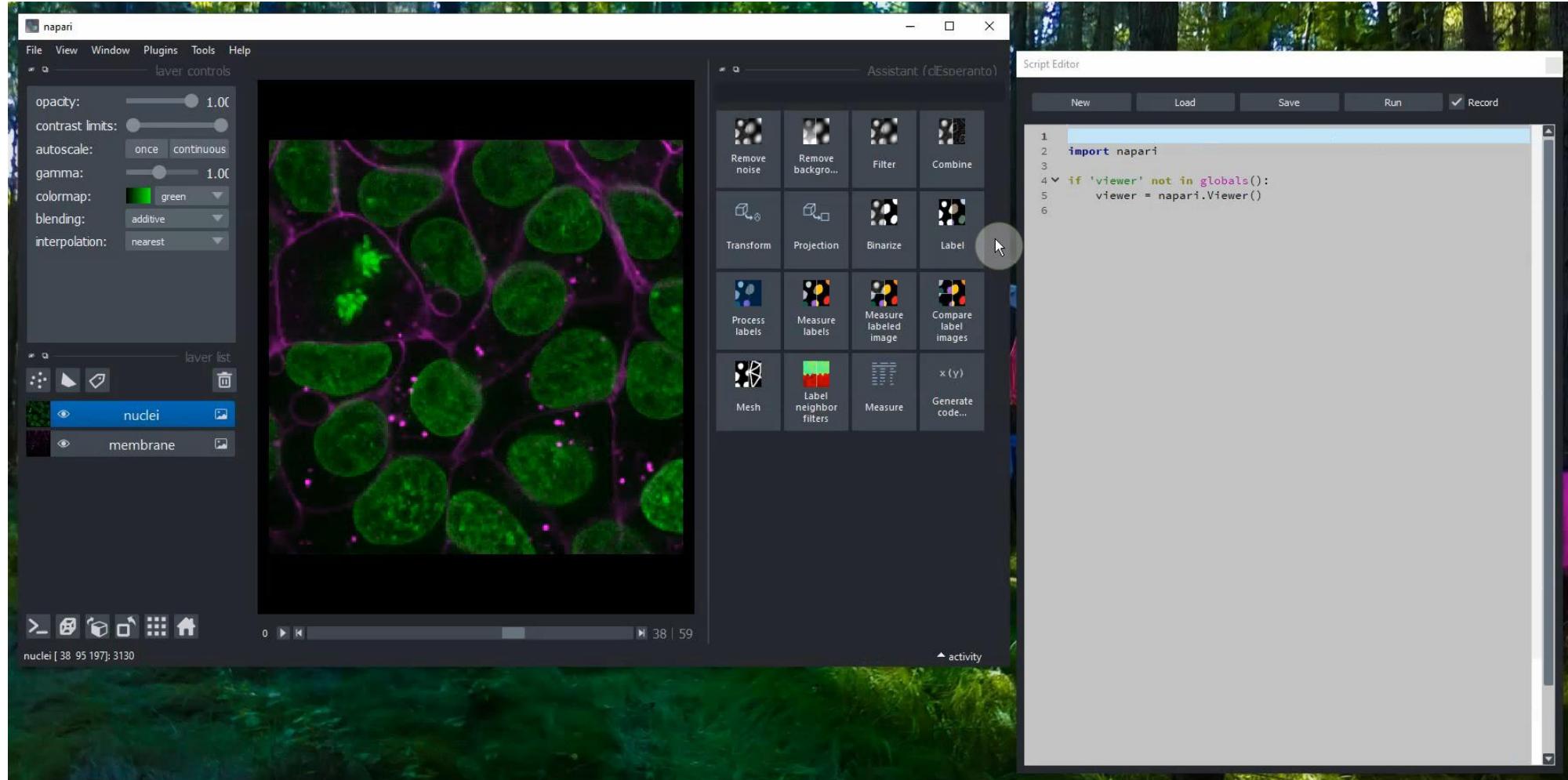
<https://www.napari-hub.org/plugins/napari-segment-blobs-and-things-with-membranes>  
<https://www.napari-hub.org/plugins/napari-cupy-image-processing>  
<https://www.napari-hub.org/plugins/napari-simpleitk-image-processing>

# Browse operations

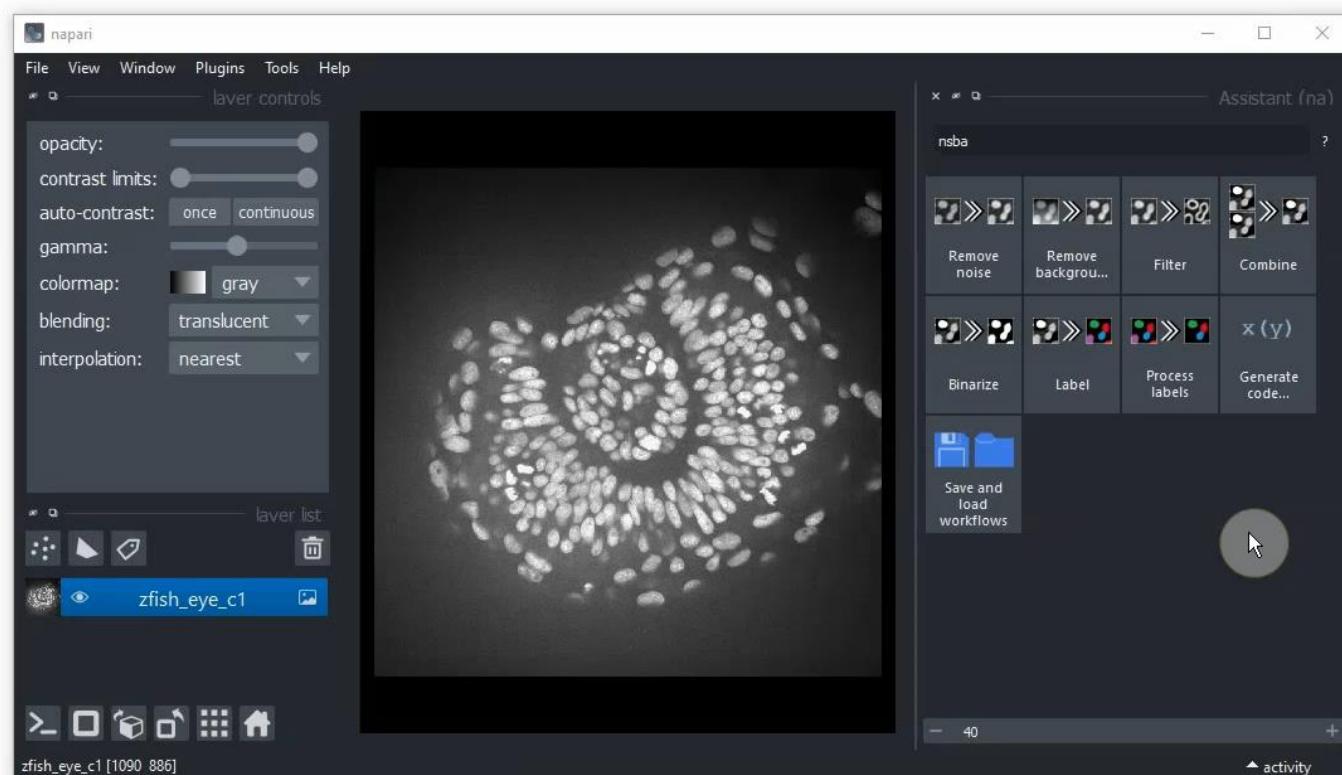
- Use the search...



# Export code to Python



# 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)

# Exercise

- Use the Napari assistant to generate a Jupyter Notebook

