PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

PoL Bio-Image Analysis Symposium 2023
Training School Early Career Track

Introduction to Napari

Marcelo Leomil Zoccoler

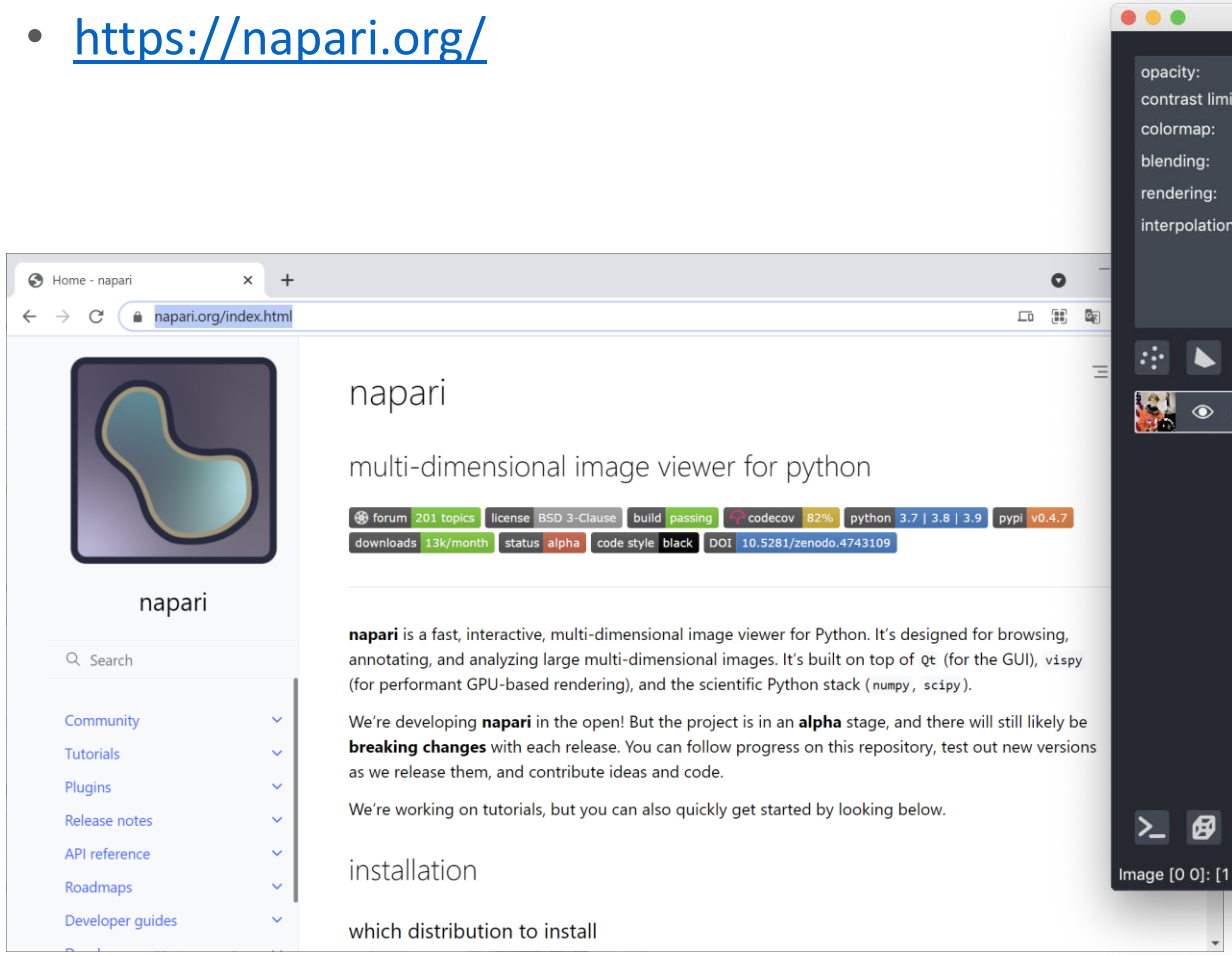
With material from:

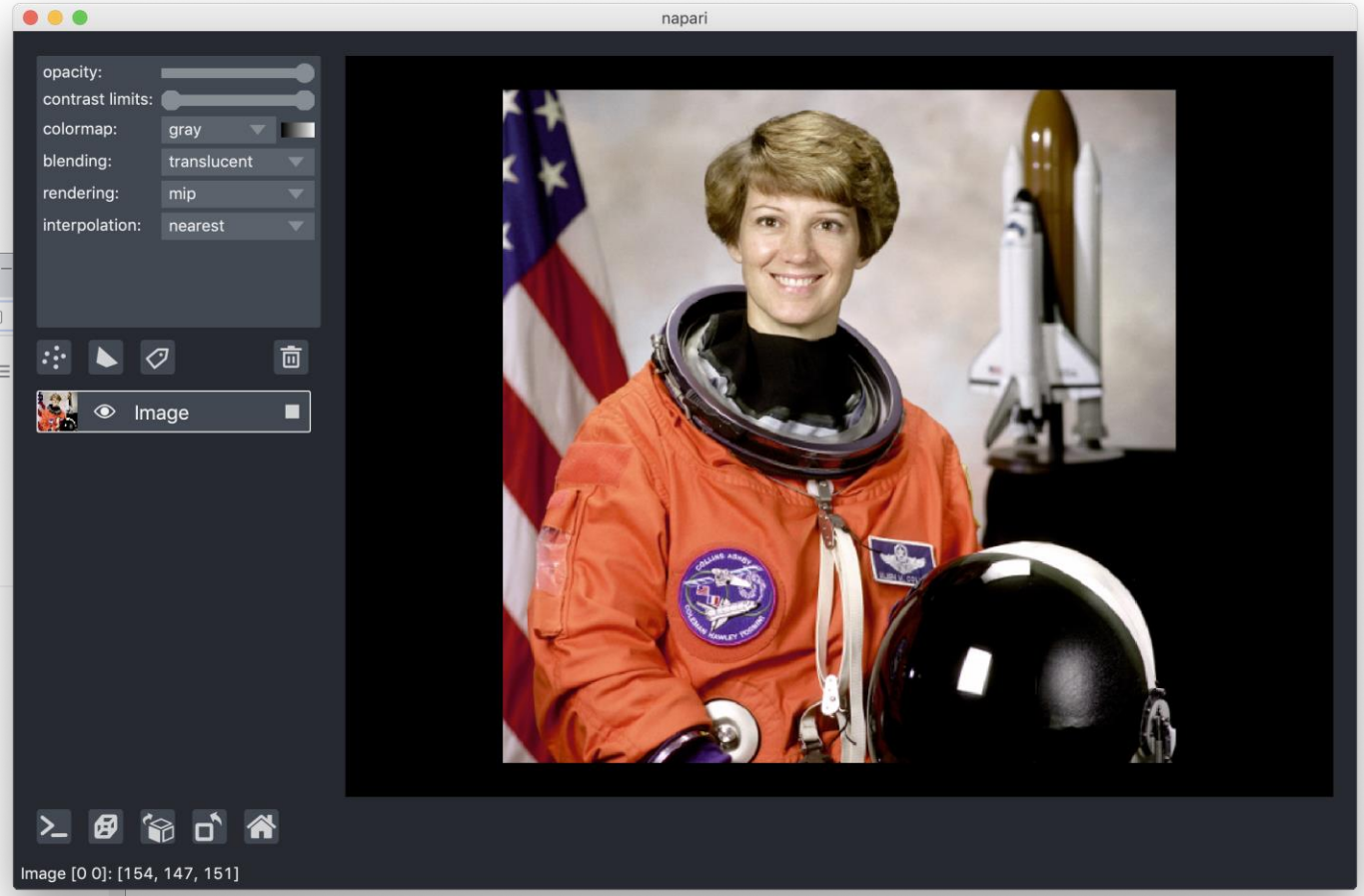
Robert Haase, PoL; Guillaume Witz, Universität Bern

@zoccolermarcelo

Napari: 3D viewer for Python

- Multi-dimensional image viewer in python
- https://napari.org/





https://napari.org/

Napari: 3D viewer for Python

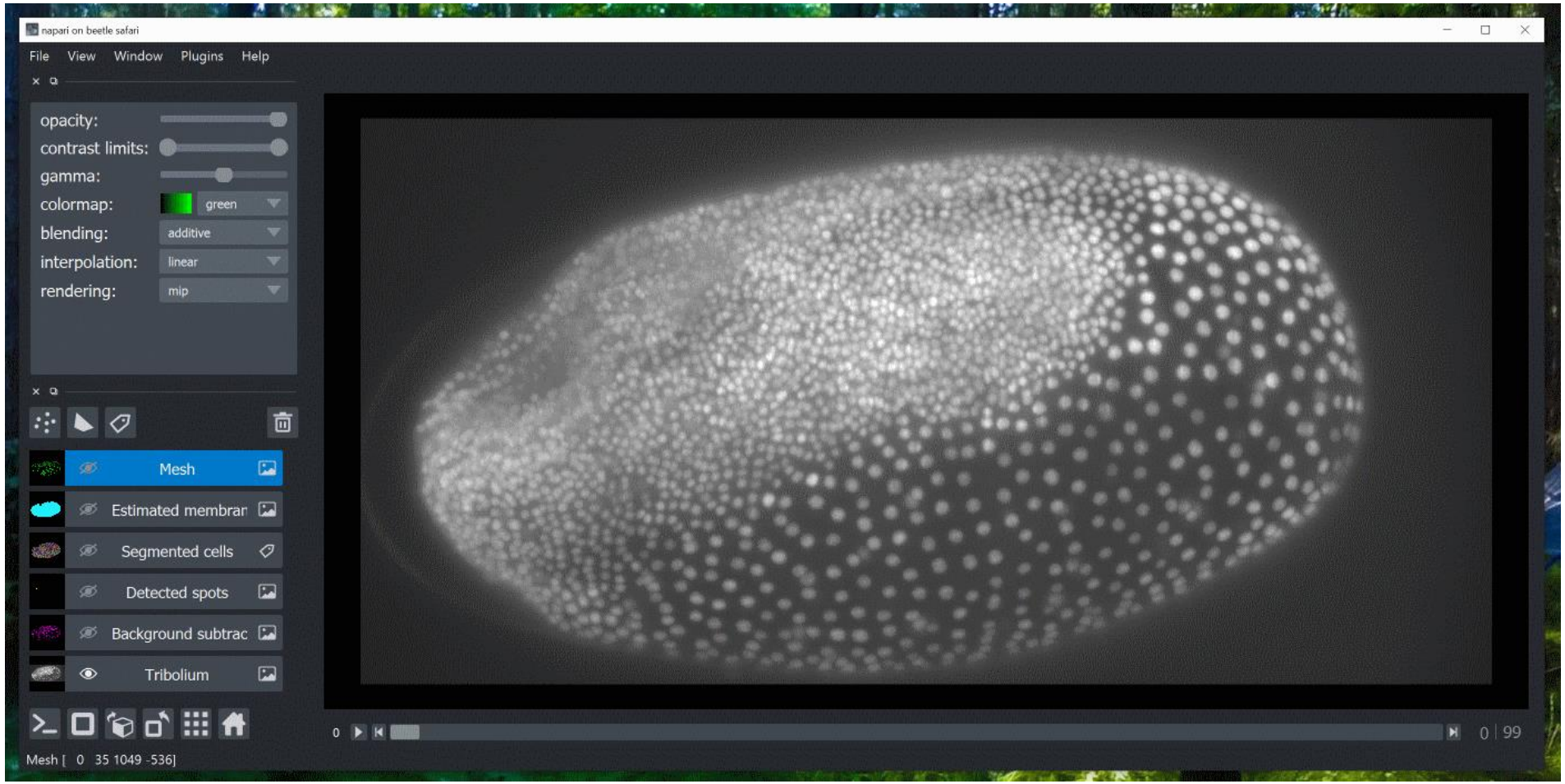


Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

https://napari.org/

@zoccolermarcelo

Napari user interface

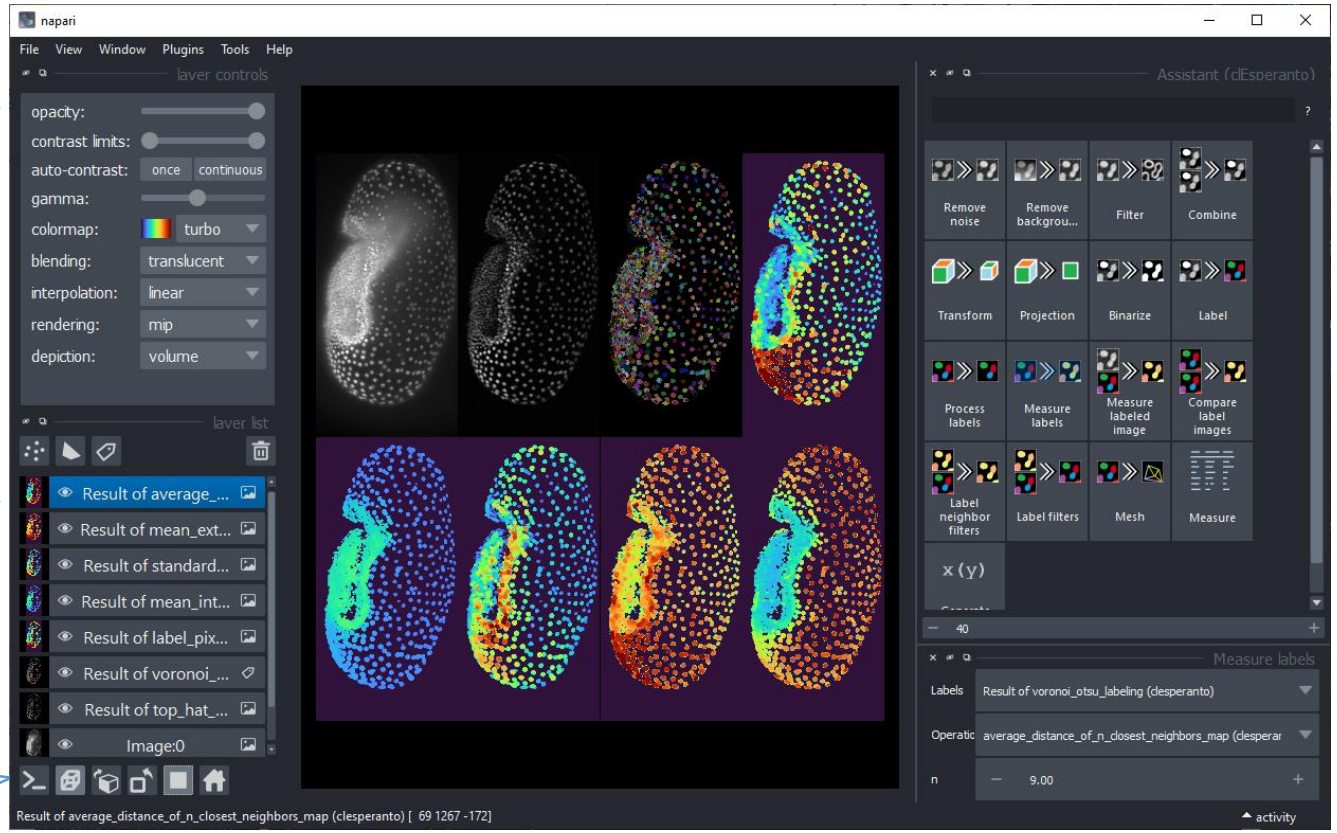
View configuration / tools

```
layer.opacity = 0.5
```

Layers

```
layer.visible = False
```

Viewer controls



Dock widgets (custom plugins)

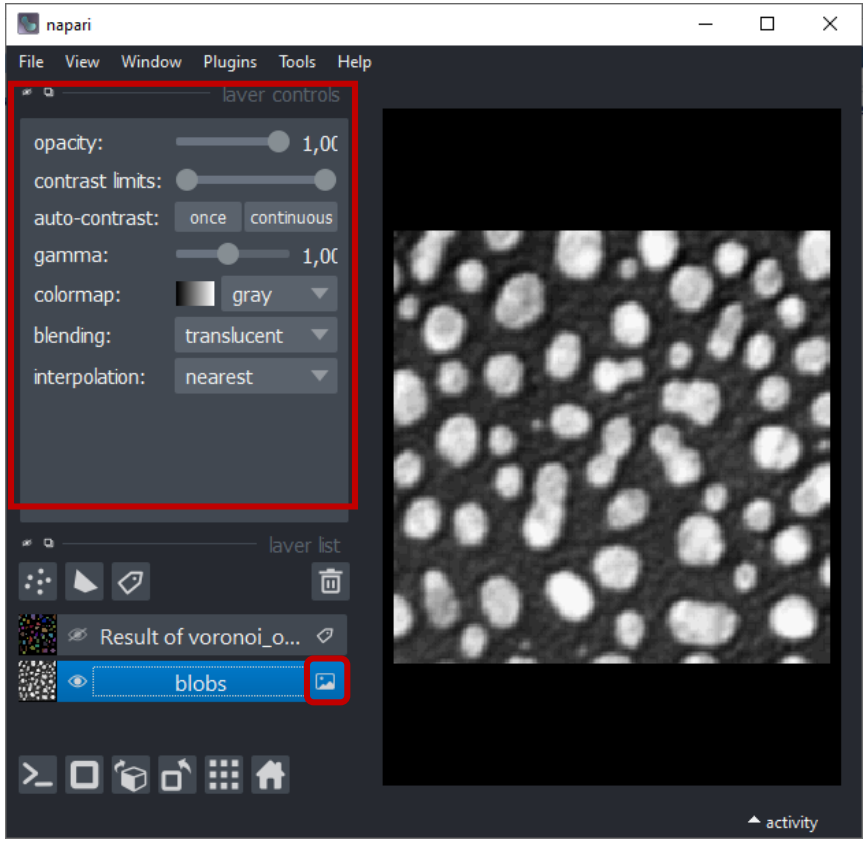
Function widgets (custom plugins)

https://napari.org/tutorials/fundamentals/viewer.html

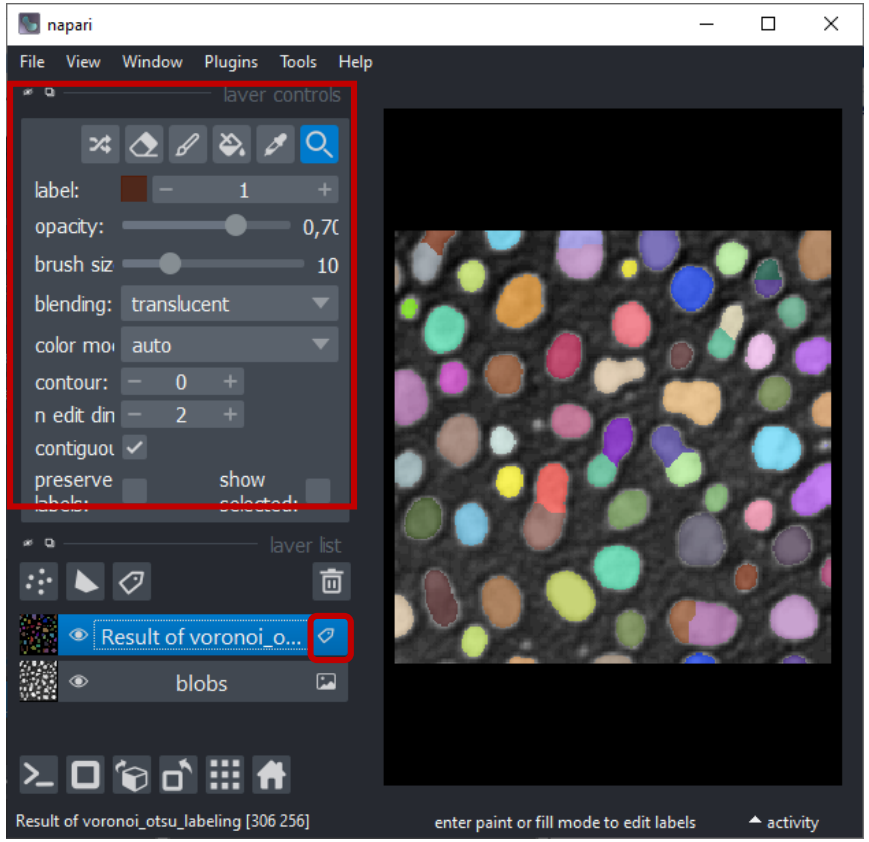
Layer types

- Different layers have different tools and options

Image Layer



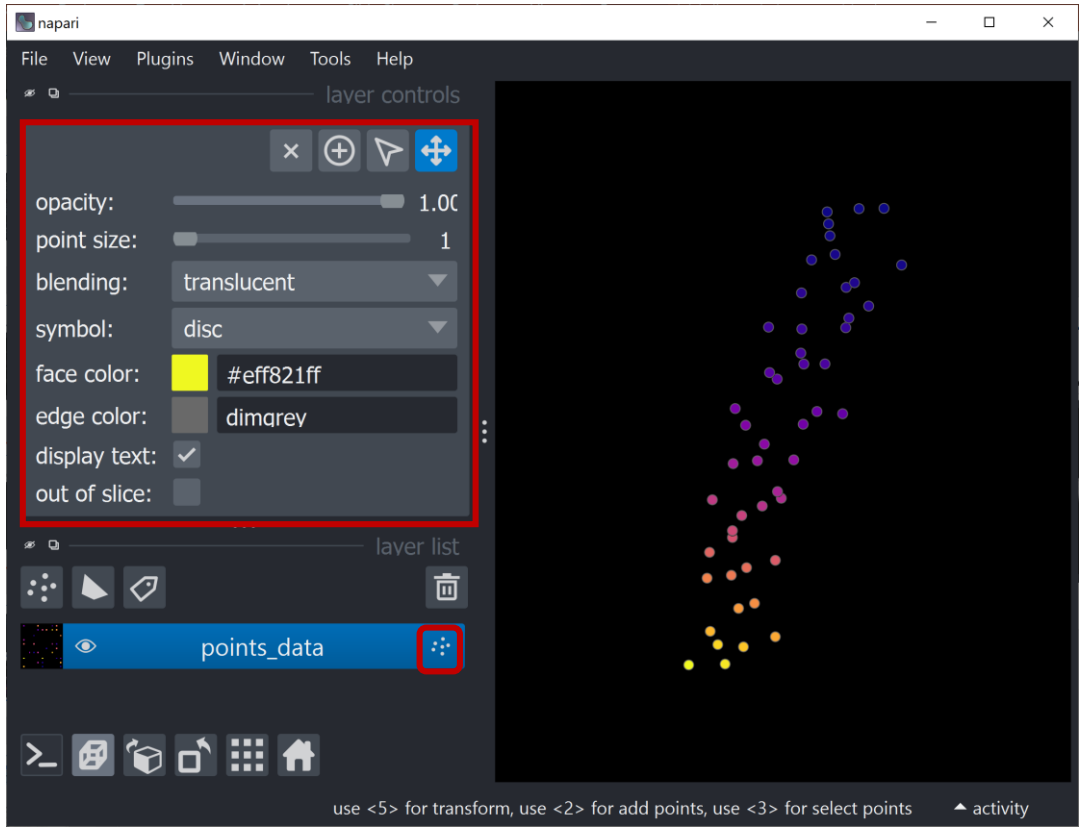
Labels Layer



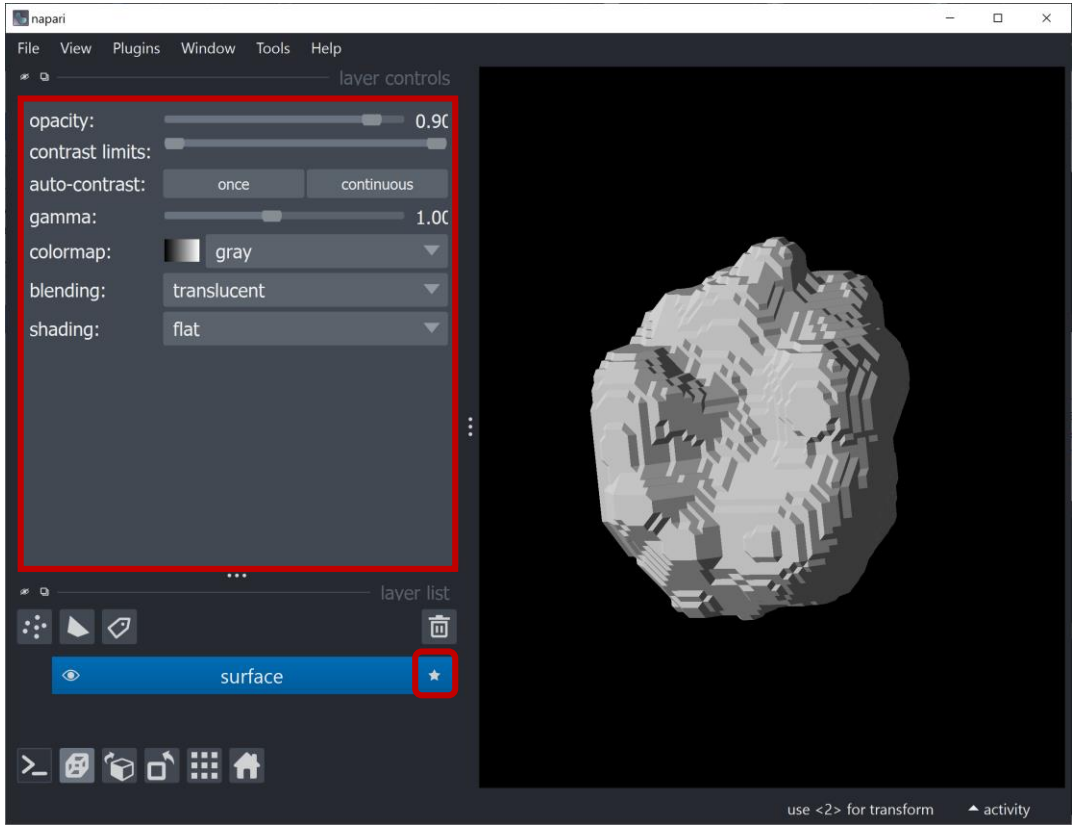
Layer types

- Different layers have different tools and options

Points Layer



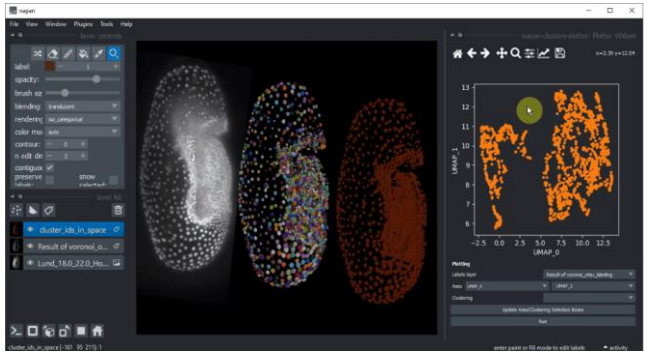
Surface Layer



napari plugins

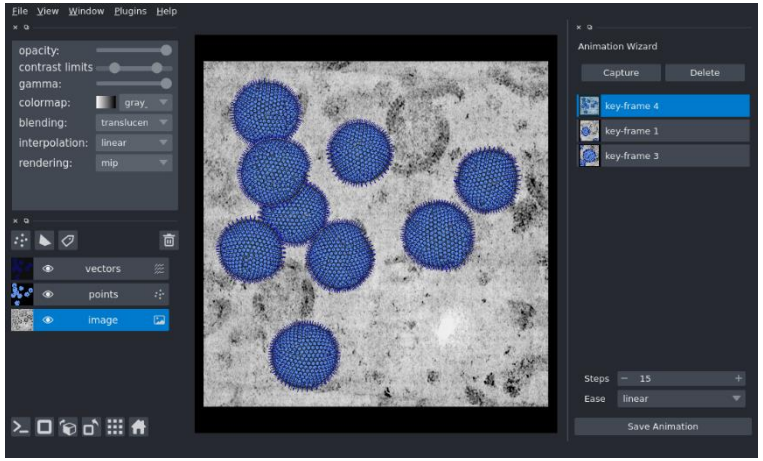
PoL Physics of Life TU Dresden

clusters-plotter



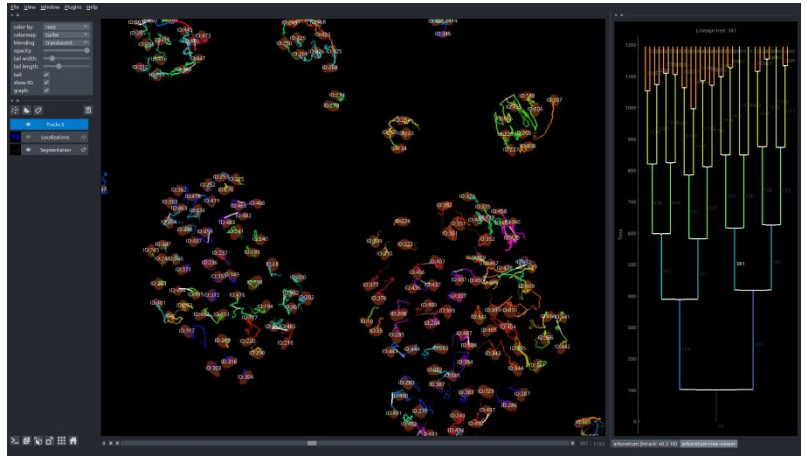
https://github.com/BiAPoL/napari-clusters-plotter

animation



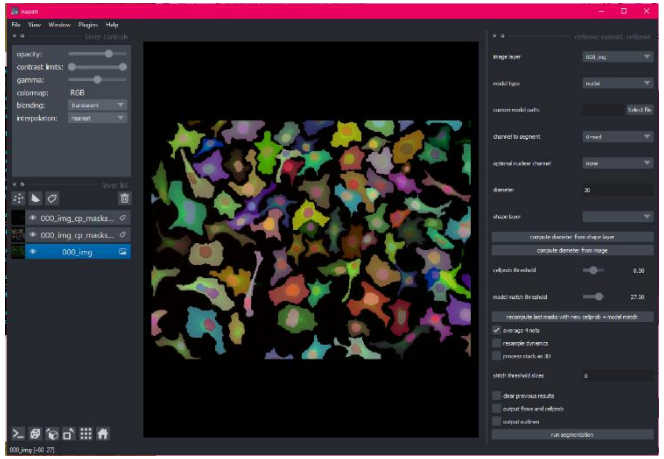
https://github.com/napari/napari-animation

arboretum



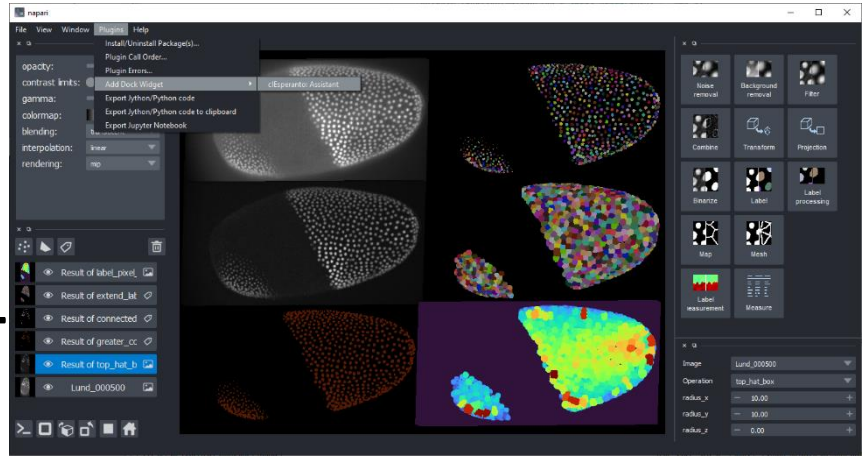
https://github.com/quantumjot/arboretum

cellpose



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

clesperanto



https://github.com/clEsperanto/napari_pyclesperanto_assistant

In development: https://github.com/topics/napari-plugin
Released: https://pypi.org/search/?q=&o=&c=Framework+%3A%3A+napari

@zoccolermarcelo

Correspondence | Published: 18 August 2023

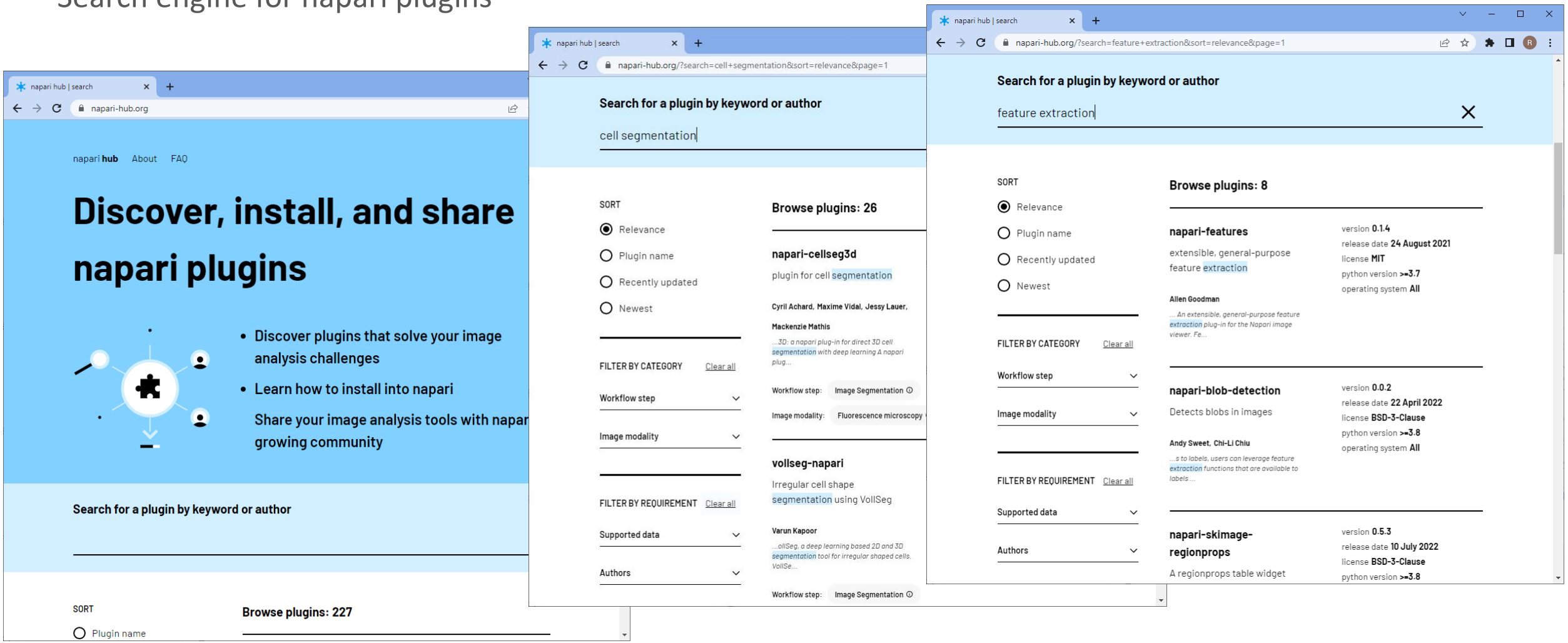
napari-imagej: ImageJ ecosystem access from napari

Gabriel J. Selzer, Curtis T. Rueden, Mark C. Hiner, Edward L. Evans III, Kyle I. S. Harrington & Kevin W. Eliceiri

Nature Methods (2023) | Cite this article

The Napari Hub

- The plugin you are looking for may be near you!
- Search engine for napari plugins



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

Napari and Jupyter Notebooks

Napari from jupyter notebooks

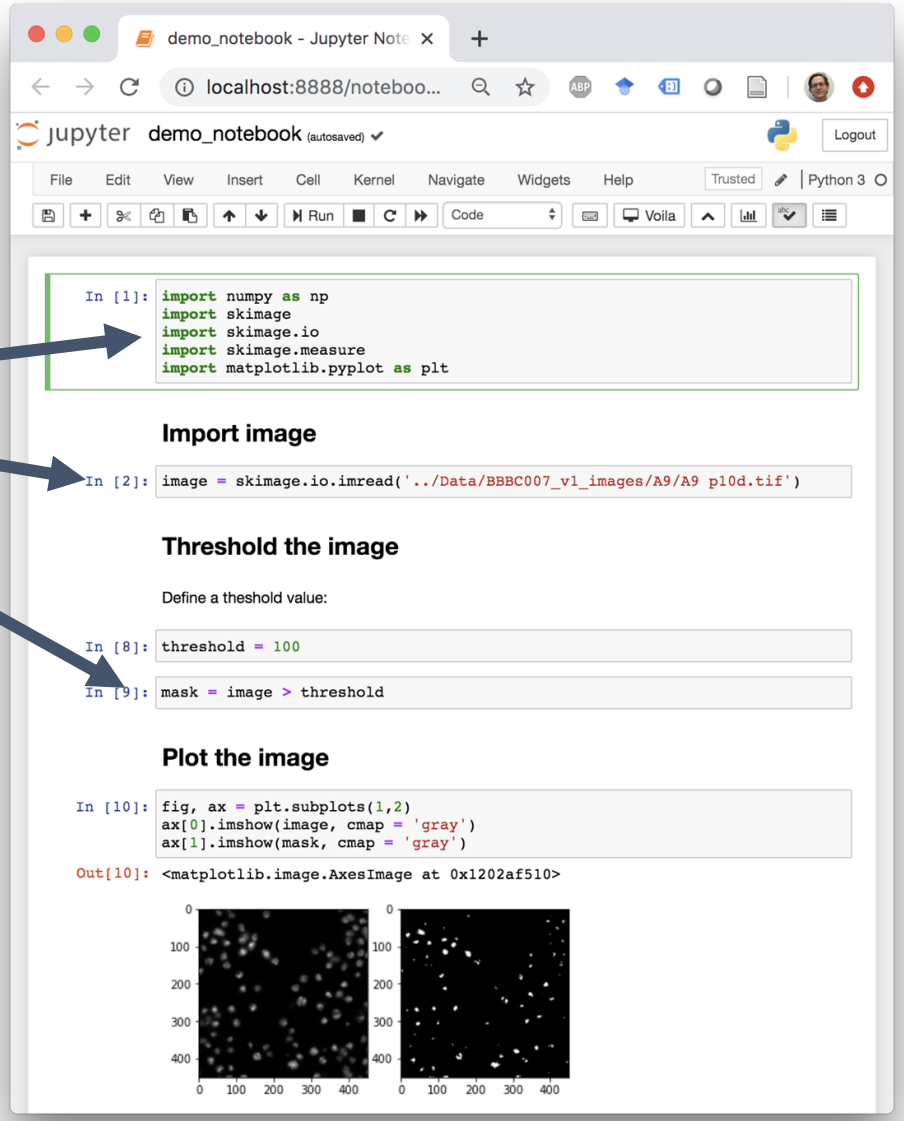
PoL Physics of Life TU Dresden

What is a jupyter notebook?

A text file (easily sent around)

Rendered by Jupyter in the browser

Split into sections called cells



Napari from jupyter notebooks

PoL
Physics of Life
TU Dresden

What is a jupyter notebook?

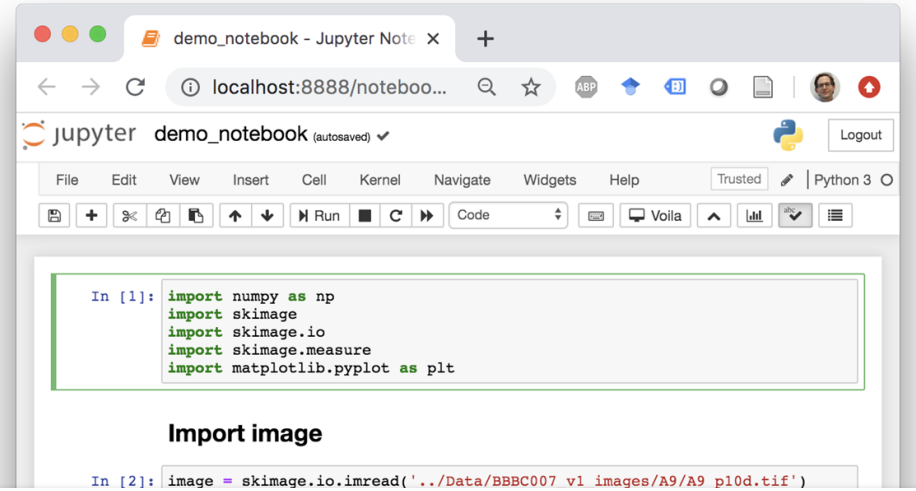
A text file (easily sent around)

Rendered by Jupyter in the browser

Split into sections called cells

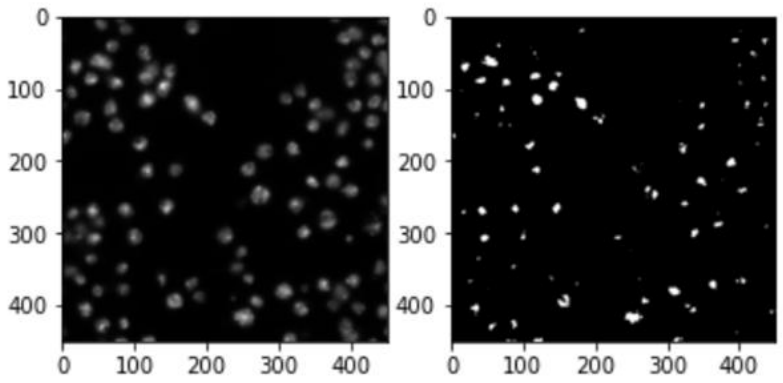
Cells can contain:

- Code
- Formatted text
- Rich output




```
In [2]: imag
```

```
In [10]: fig, ax = plt.subplots(1,2)
         ax[0].imshow(image, cmap = 'gray')
         ax[1].imshow(mask, cmap = 'gray')
Out[10]: <matplotlib.image.AxesImage at 0x1202af510>
```



Why using notebooks?

PoL Physics of Life TU Dresden

- Keep all the benefits from using code:

  - Batch processing

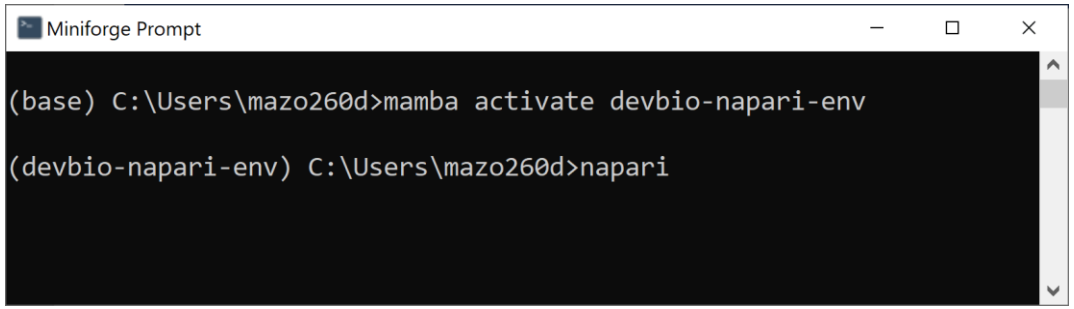
  - Running python functions/tools still unavailable as plugins

Why using notebooks with napari?

- Easy data interaction and visualization with napari:
    - Great for visualizing 3D (and more) data
    - Each processing step result can be displayed as a separate layer
- Data annotation

Opening napari

- From command line:

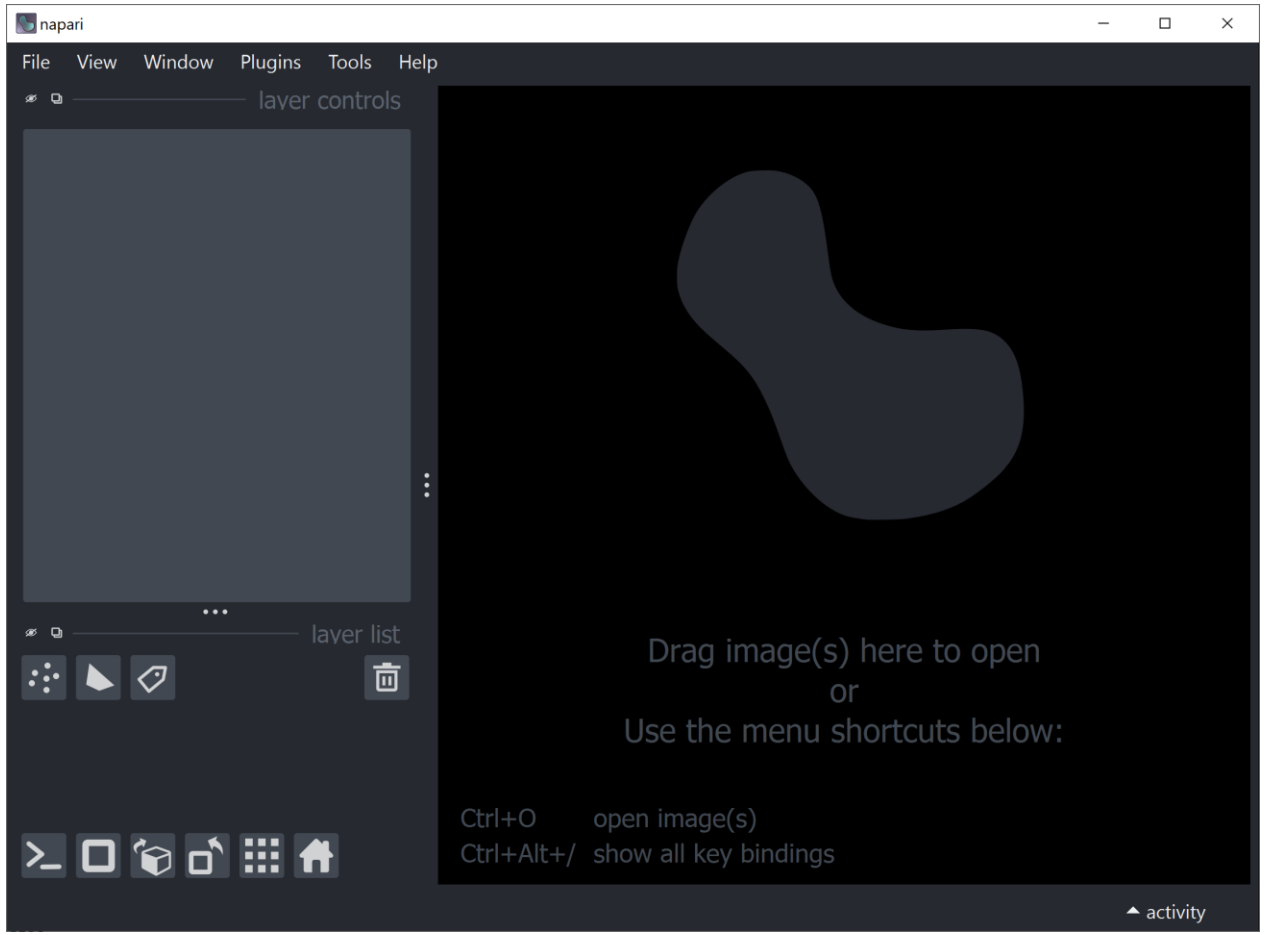


- From code (jupyter notebook):

```
[1]: import napari

[*]: viewer = napari.Viewer()

[ ]:
```



Opening Images in napari

- Drag and drop

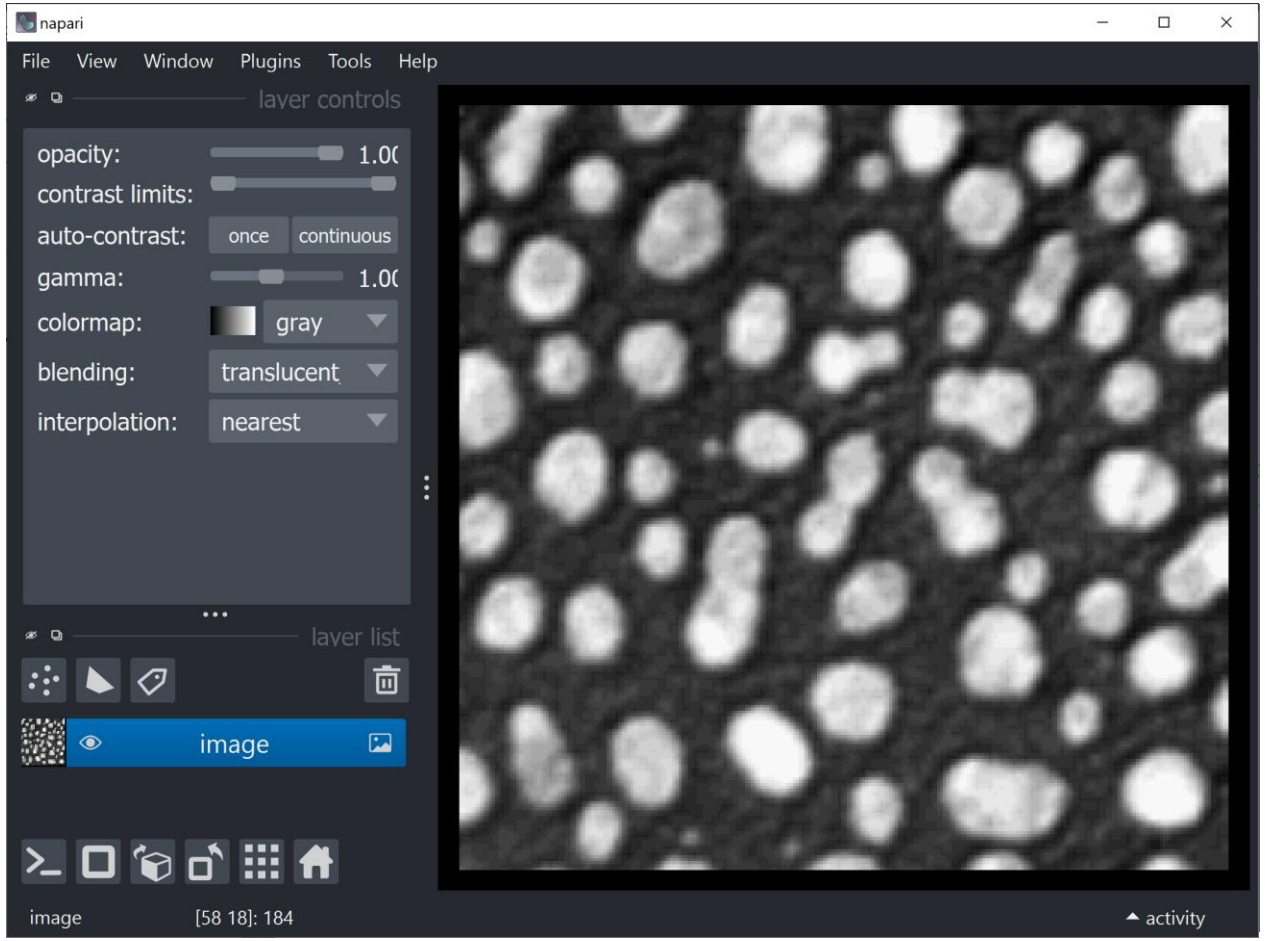
- File > Open File…

- From code:
  - Open an image with an “imread” function

```
image = imread(“image.tif”)
```

  - Load the image to the viewer

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



Scripting napari in notebooks

- Add layers to napari to visualize intermediate processing results on top of each other or side by side.
- Change layer visualization within napari…

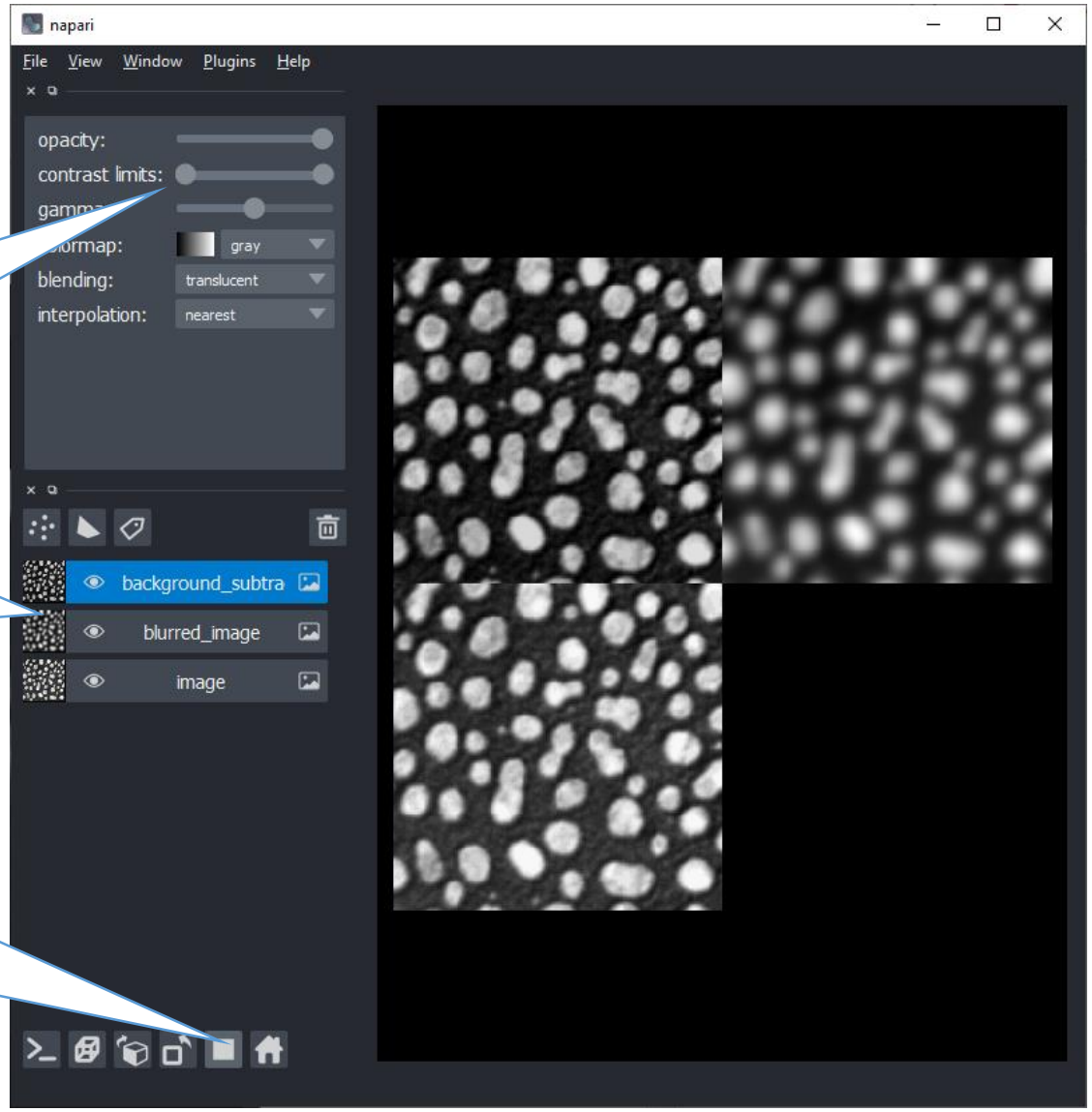
  … or via code in a jupyter notebook:

```
viewer.layers[0].contrast_limits
```

```
[0, 255]
```

1. Access the viewer
2. Access the layers
3. Choose a layer (by index or name)
4. “Press TAB” and check out available properties

```
viewer.layers[0].contrast_limits = [30,170]
```



Change Brightness and contrast here

Toggle visualization of layers here

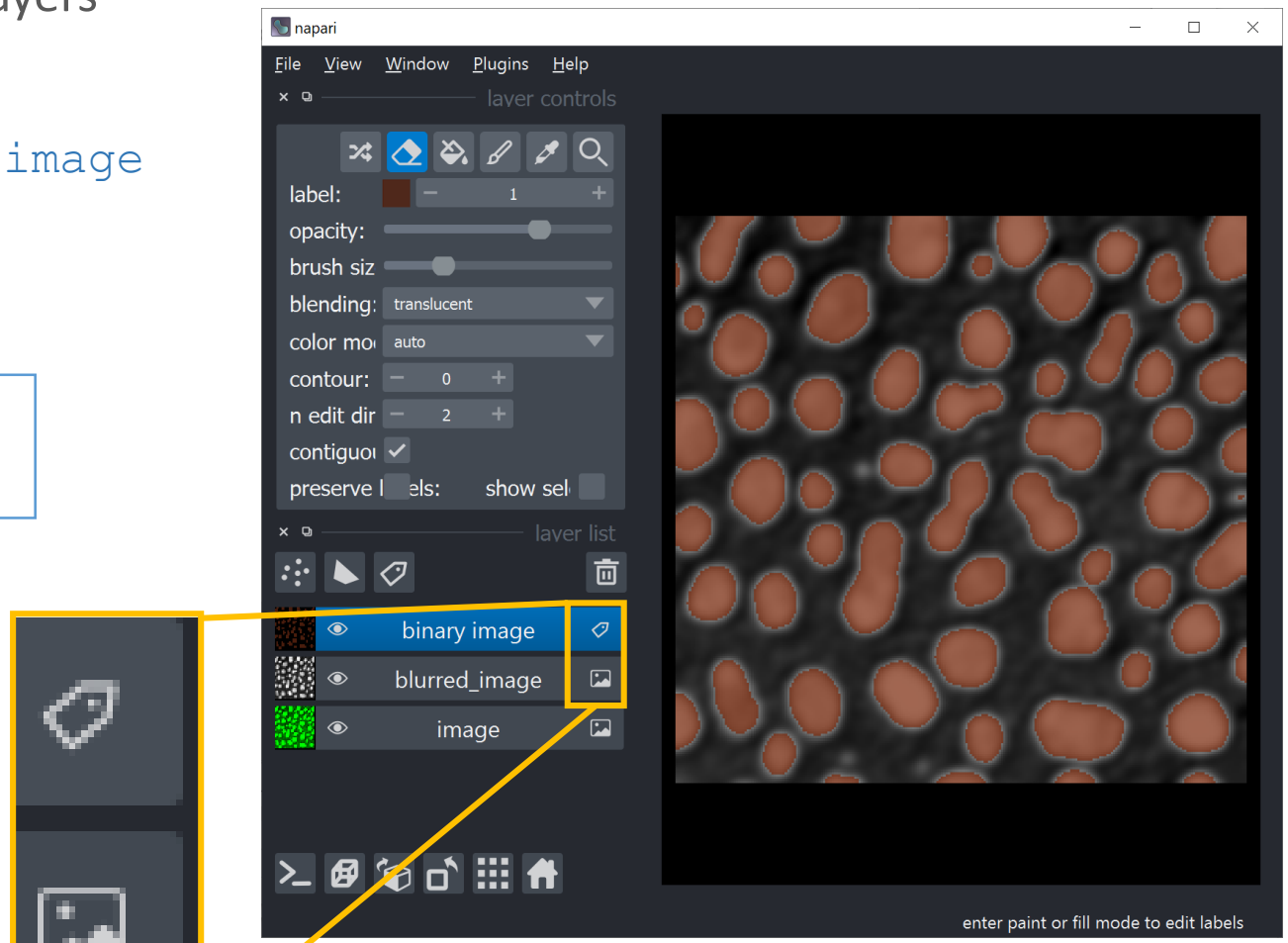
Gallery view

Visualizing image segmentation

- Binary images and label images visualized as label layers

```
# 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



Labels Layer

Image Layer

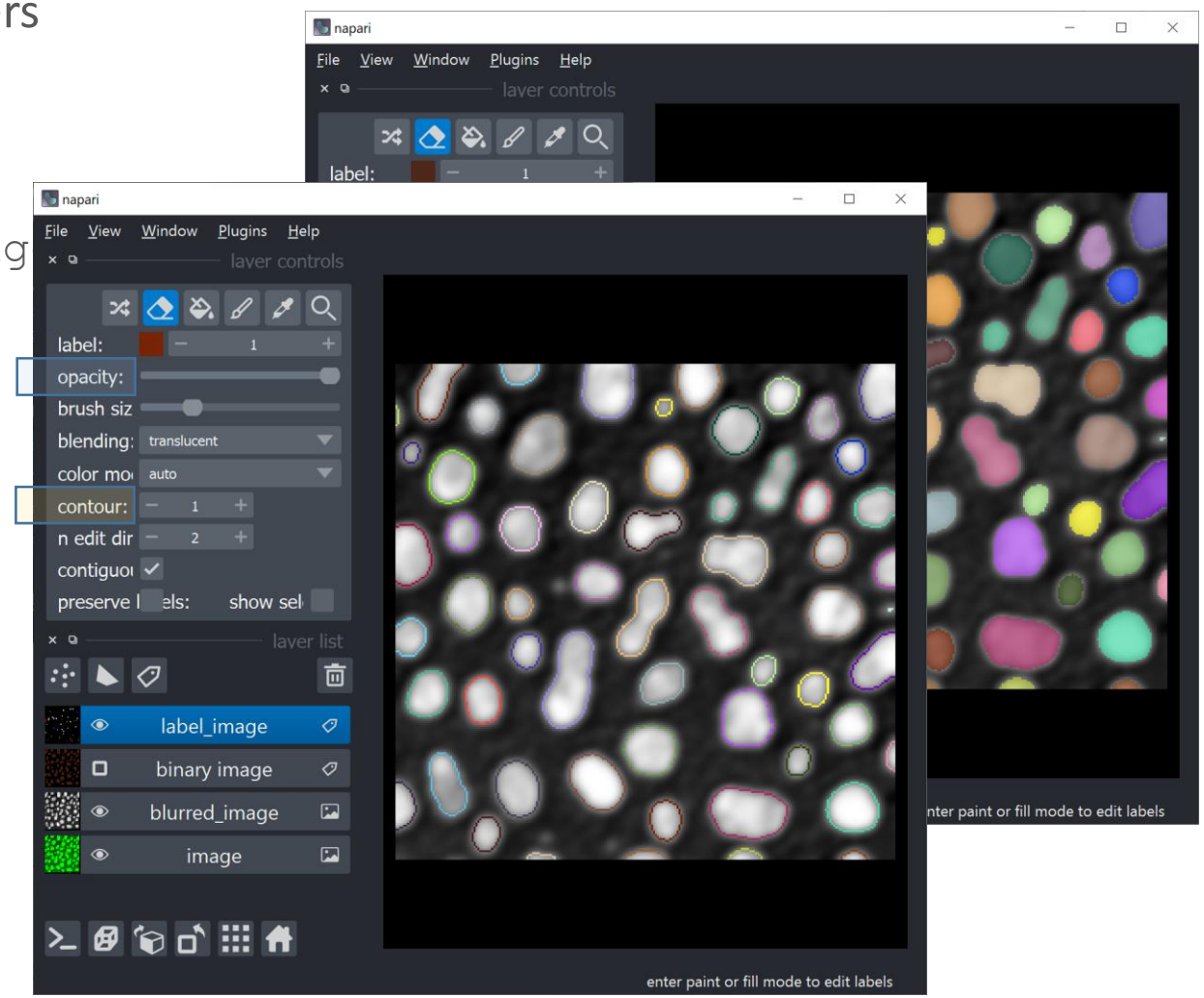
@zoccolermarcelo

Visualizing image segmentation

- Binary images and label images visualized as label layers

- `# add labels to viewer`
- `label_layer = viewer.add_labels(label_imag`

- Visualize contours instead of the overlay
- `label_layer.contour = 1`
- `label_layer.opacity = 1`

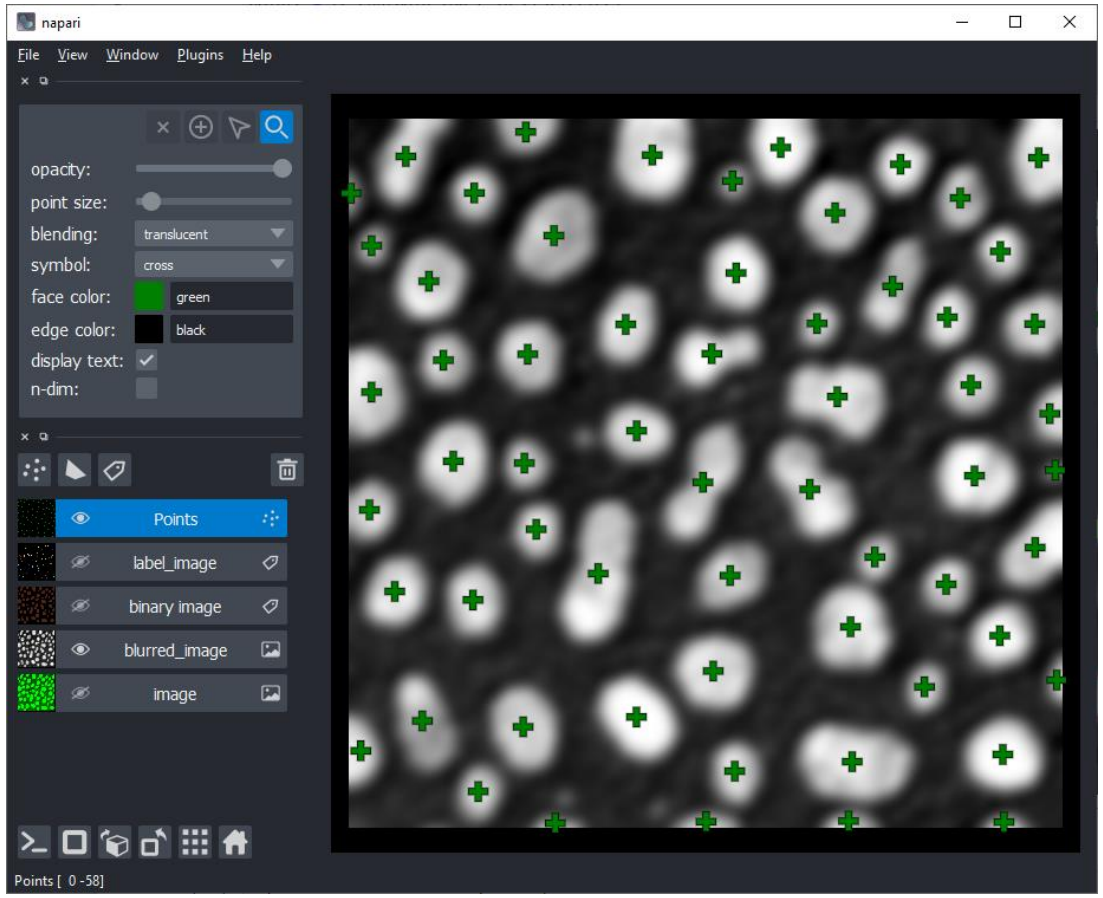


http://nbviewer.jupyter.org/github/BiAPoL/on_the_fly_image_processing_napari/blob/master/01_napari.ipynb

Points layers

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

```
# add points to viewer
label_layer = viewer.add_points(points,
  face_color='green', symbol='cross', size=5)
```



Acknowledgements

PoL Physics of Life TU Dresden

BiAPoL team

- Mara Lampert
- Marcelo Zoccoler
- Johannes Soltwedel
- Maleeha Hassan
- Allyson Ryan
- Till Korten
- Stefan Hahmann
- Somashekhar Kulkarni

Former lab members:

- Ryan George Savill
- Laura Zigutyte

Networks

BioDIP Biopolis Dresden Imaging Platform

CENTER FOR SYSTEMS BIOLOGY DRESDEN

NFDI4BIOIMAGE

neubias network of european bioimage analysts

image.sc

Funding

DFG public

Federal Ministry of Education and Research

Chan Zuckerberg Initiative