



Quantitative Bio-image Analysis with Python and Napari

Marcelo Leomil Zoccoler, Robert Haase

With materials from Johannes Müller, TU Dresden and
Benoit Lombardot, MPI CBG

The team



Marcelo Leomil Zoccoler

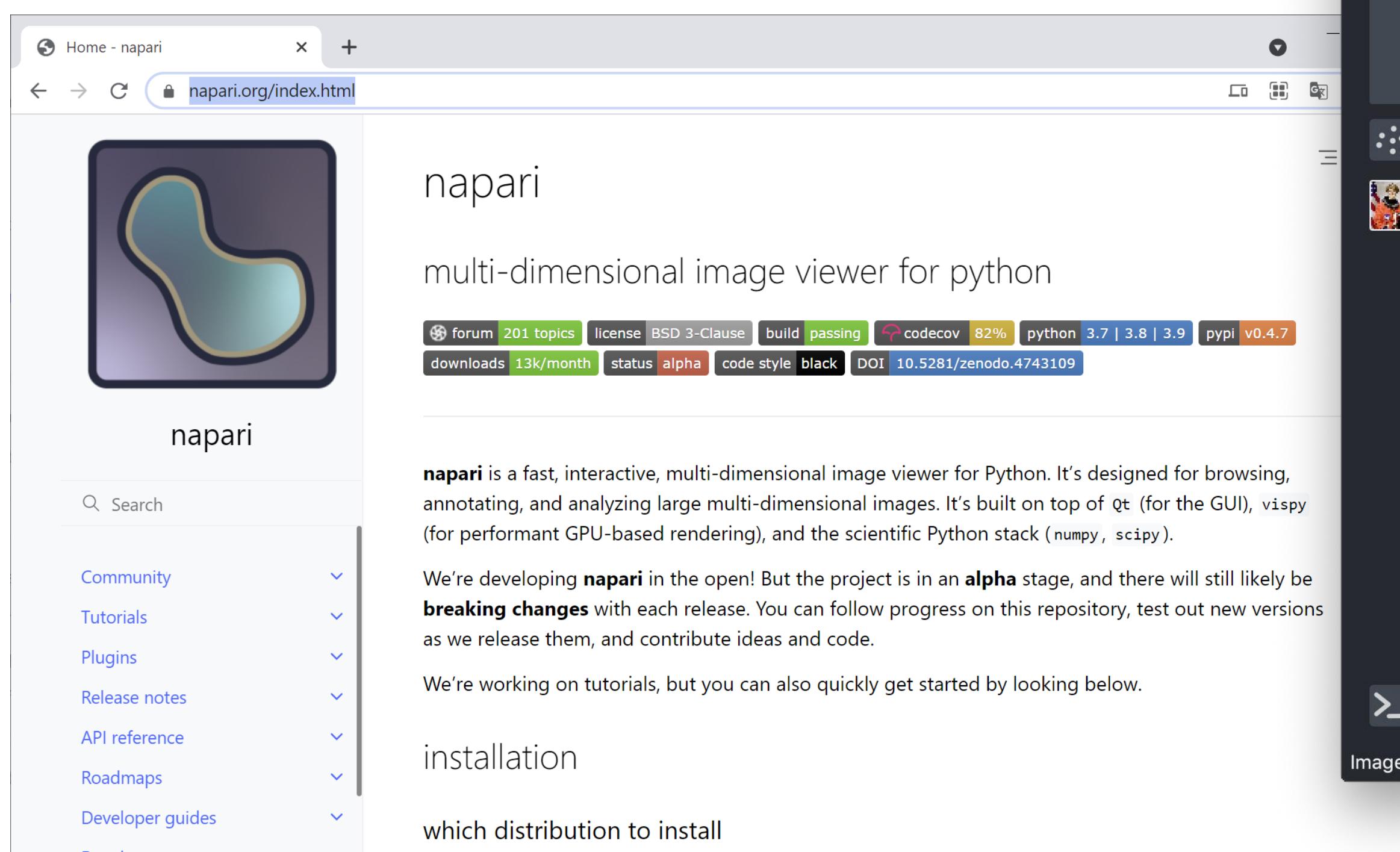


Robert Haase

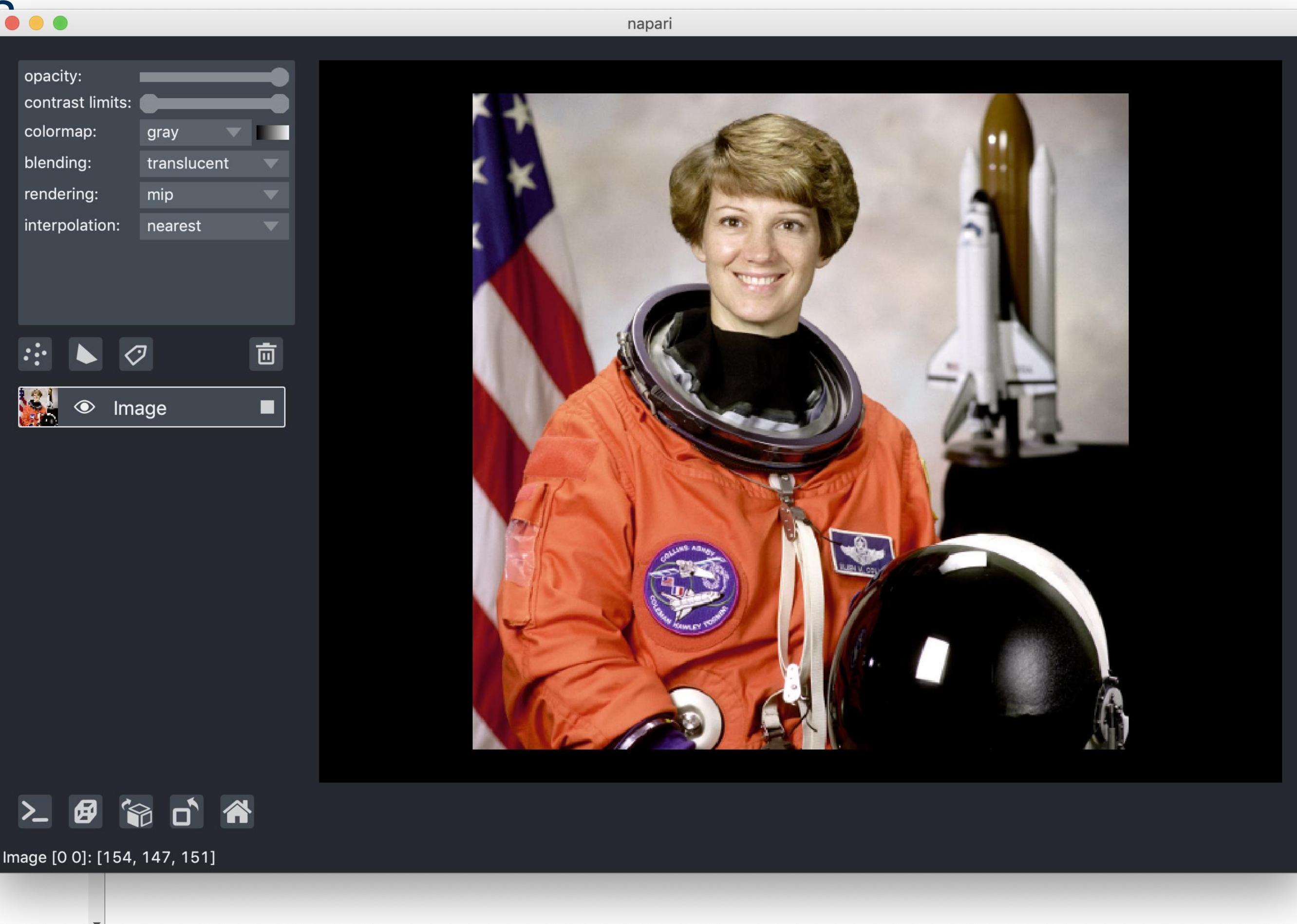
Napari: 3D viewer for Python

Multi-dimensional image viewer in python

<https://napari.org/>



The screenshot shows the official website for napari at <https://napari.org/>. The page features a large logo icon on the left, followed by the word "napari" and a subtitle "multi-dimensional image viewer for python". Below this, there's a sidebar with links to "Community", "Tutorials", "Plugins", "Release notes", "API reference", "Roadmaps", and "Developer guides". The main content area contains a brief description of what napari is, a note about it being in an alpha stage, and links for "installation" and "which distribution to install". At the bottom, there's a footer with social media links for GitHub, Twitter, and LinkedIn.



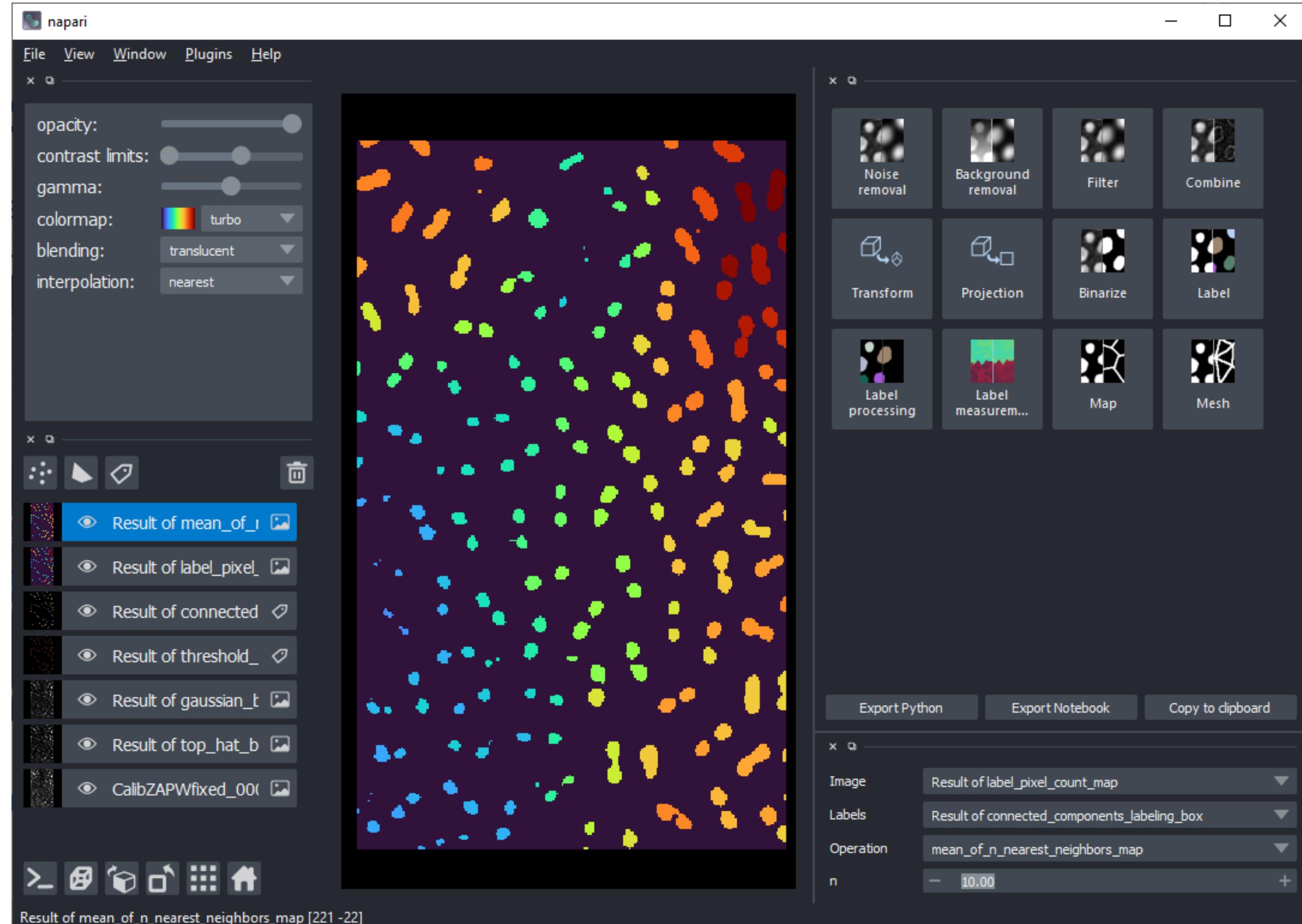
<https://napari.org/>

Napari: 3D viewer for Python



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

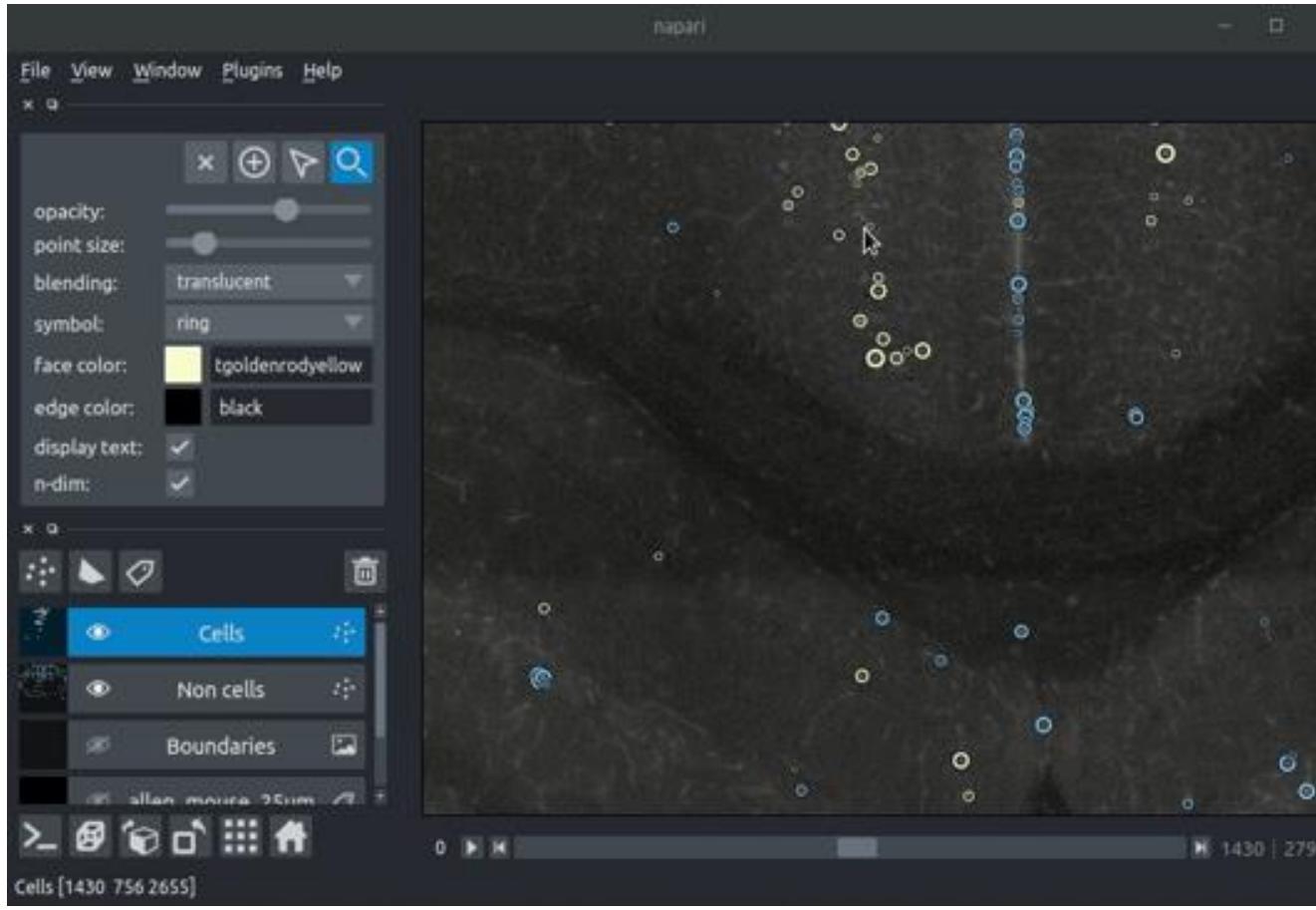
Napari user interface



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

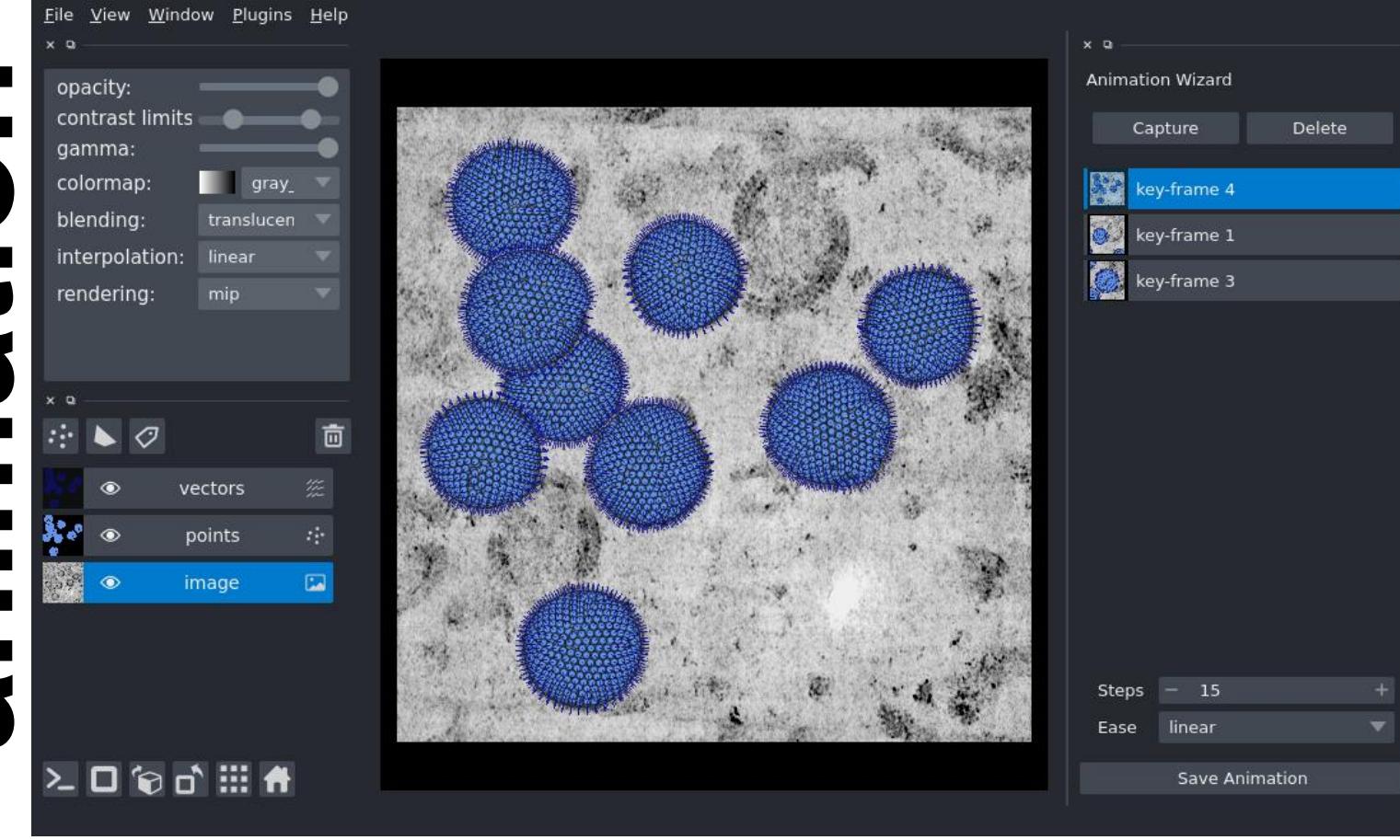
The era of napari plugins has just begun

cellfinder



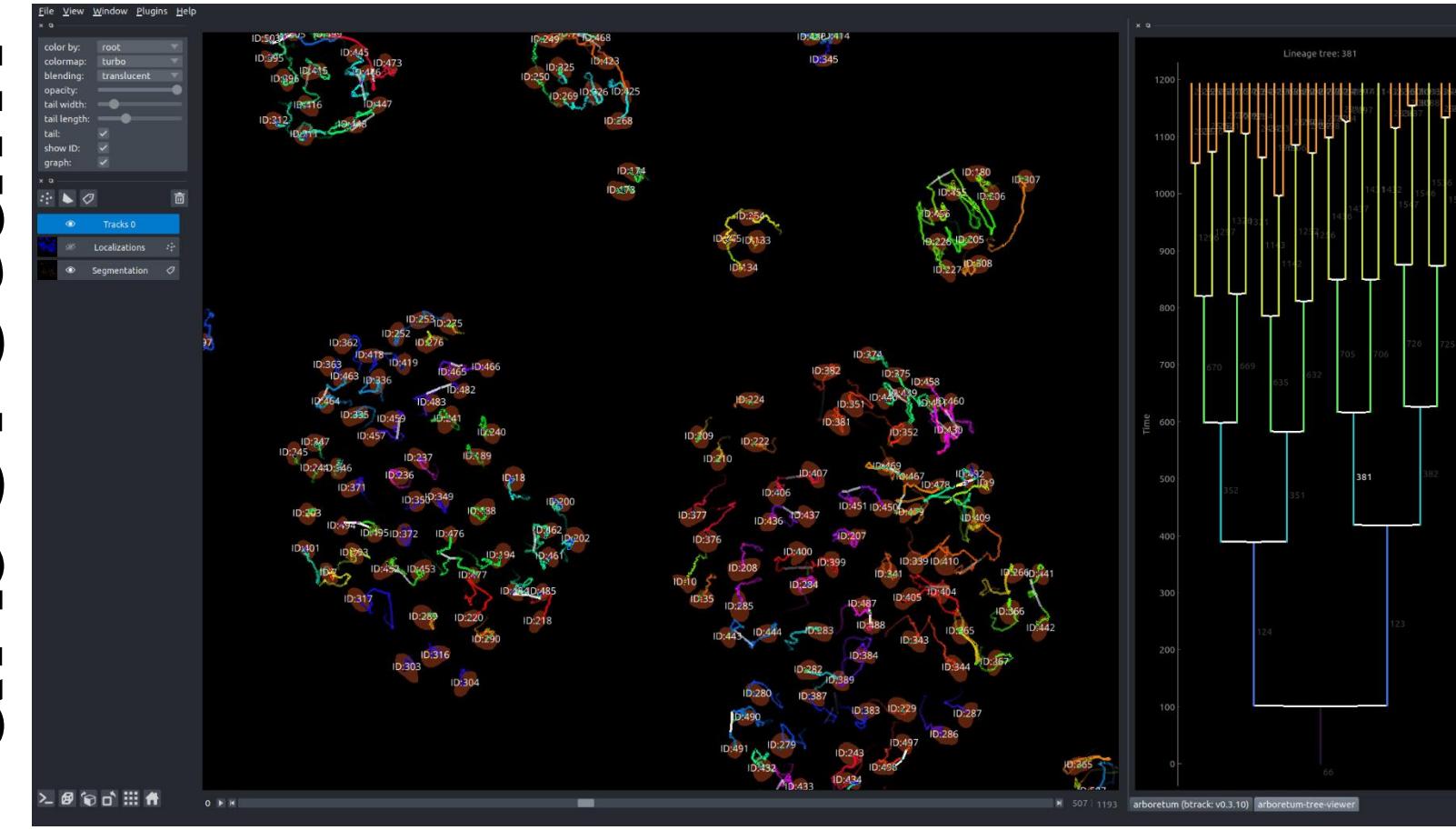
<https://github.com/brainglobe/napari-cellfinder>

animation

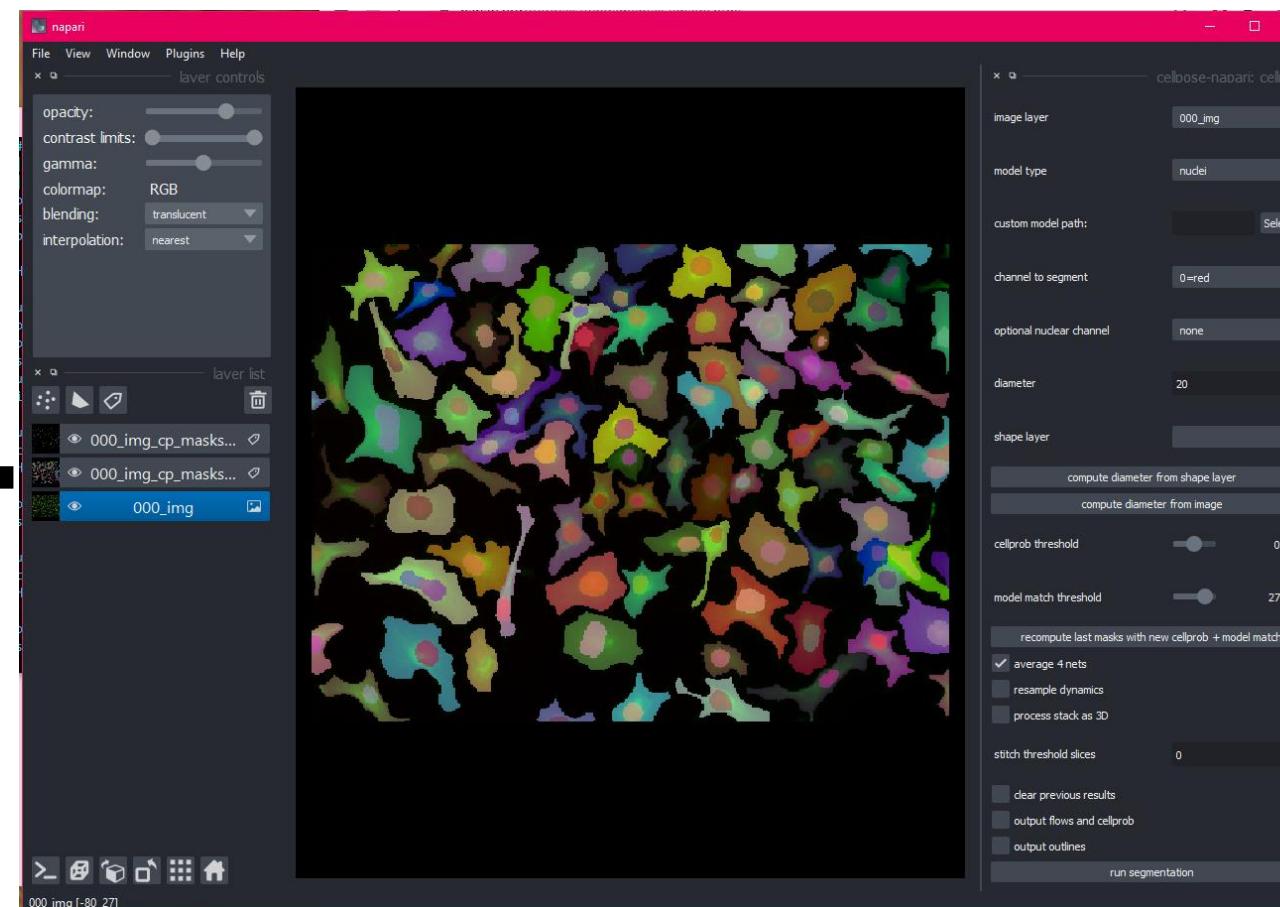


<https://github.com/napari/napari-animation> <https://github.com/quantumjot/arboretum>

arboretum

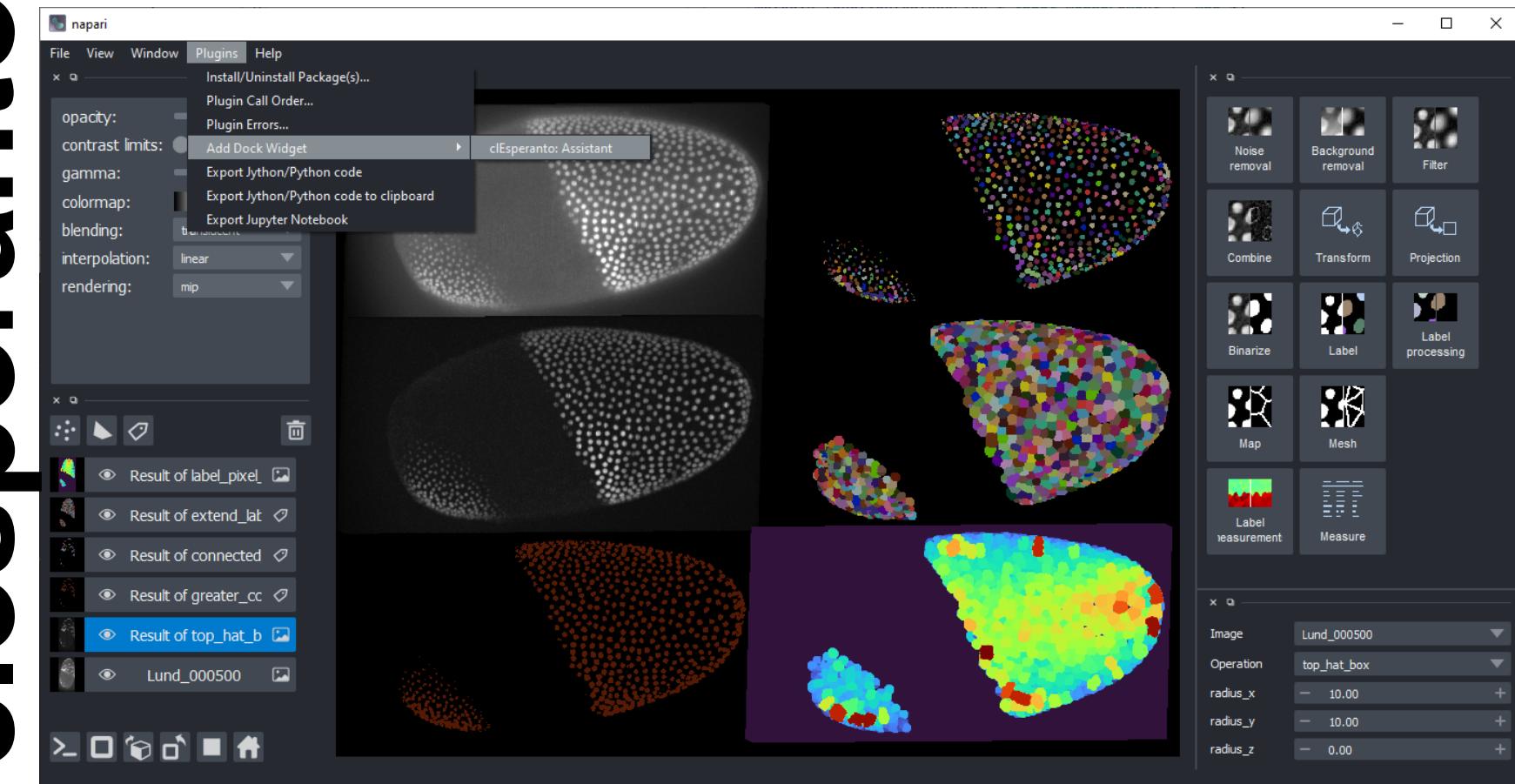


cellpose



<https://cellpose-napari.readthedocs.io/en/latest/> https://github.com/cI Esperanto/napari_pyclesperanto https://github.com/cI Esperanto/napari_pyclesperanto_assistant

clesperanto



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

Napari-animation

Making animations – as easy as it gets

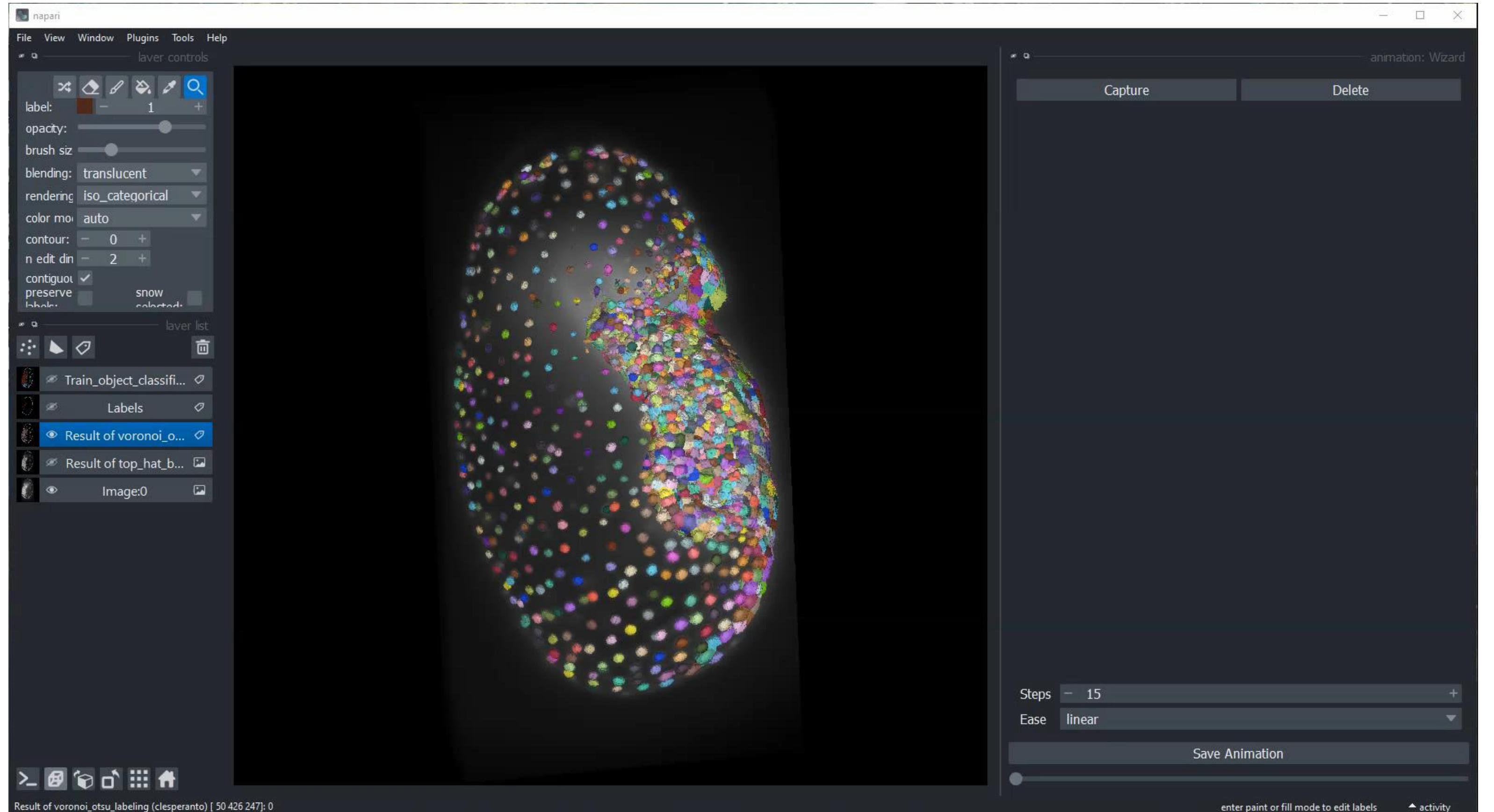


Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

Napari-animation

Making animations – as easy as it gets

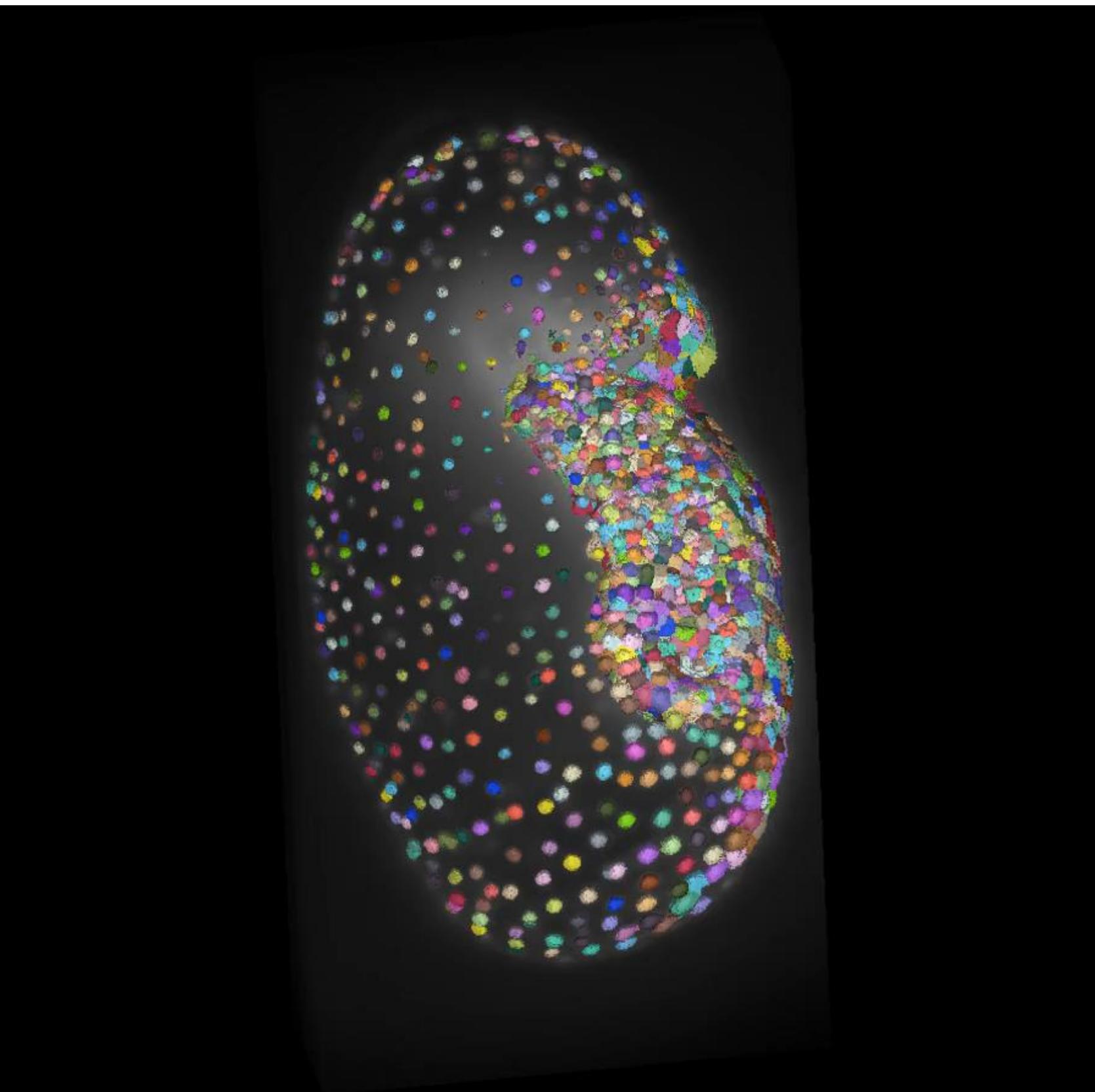
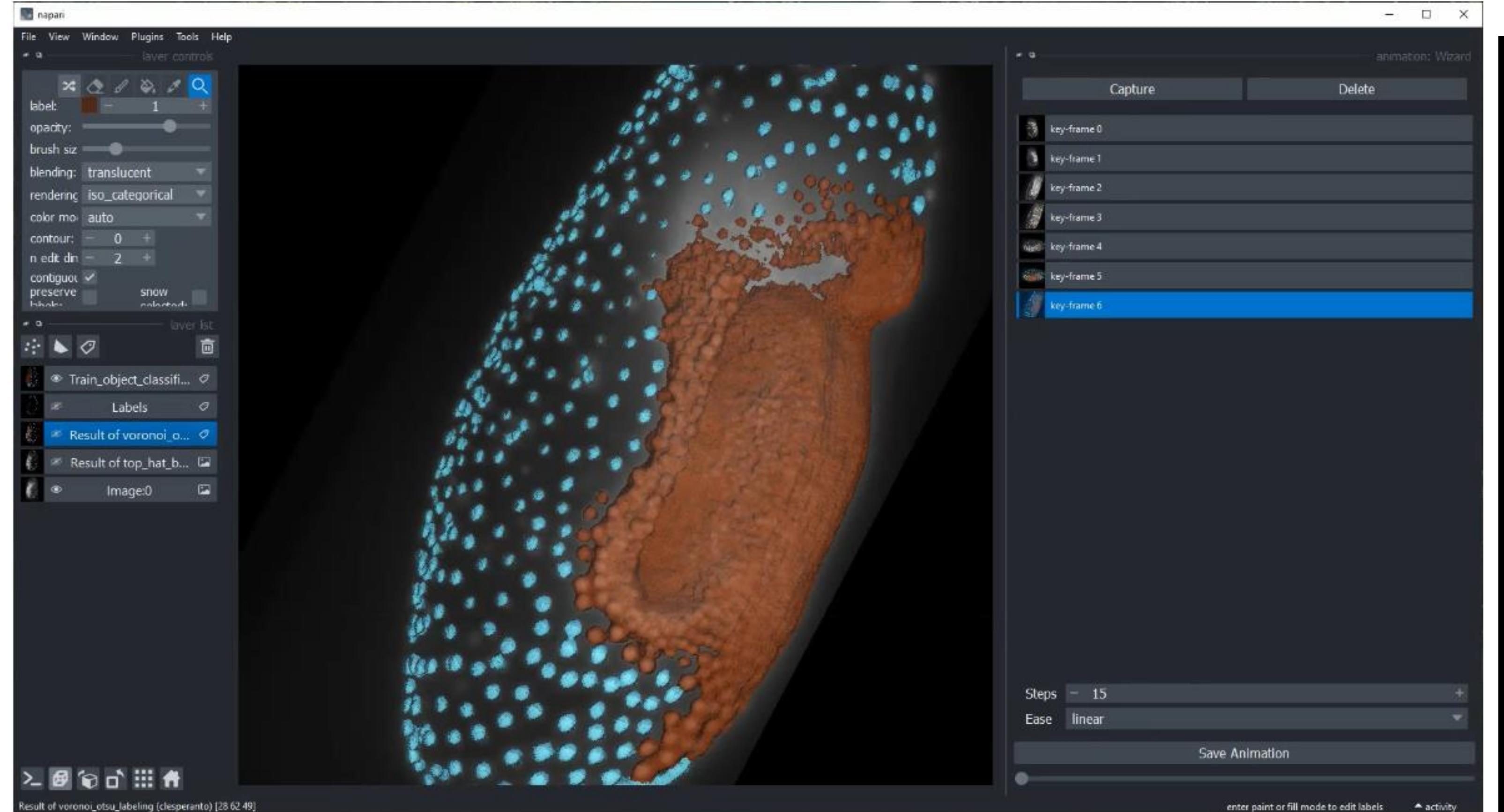
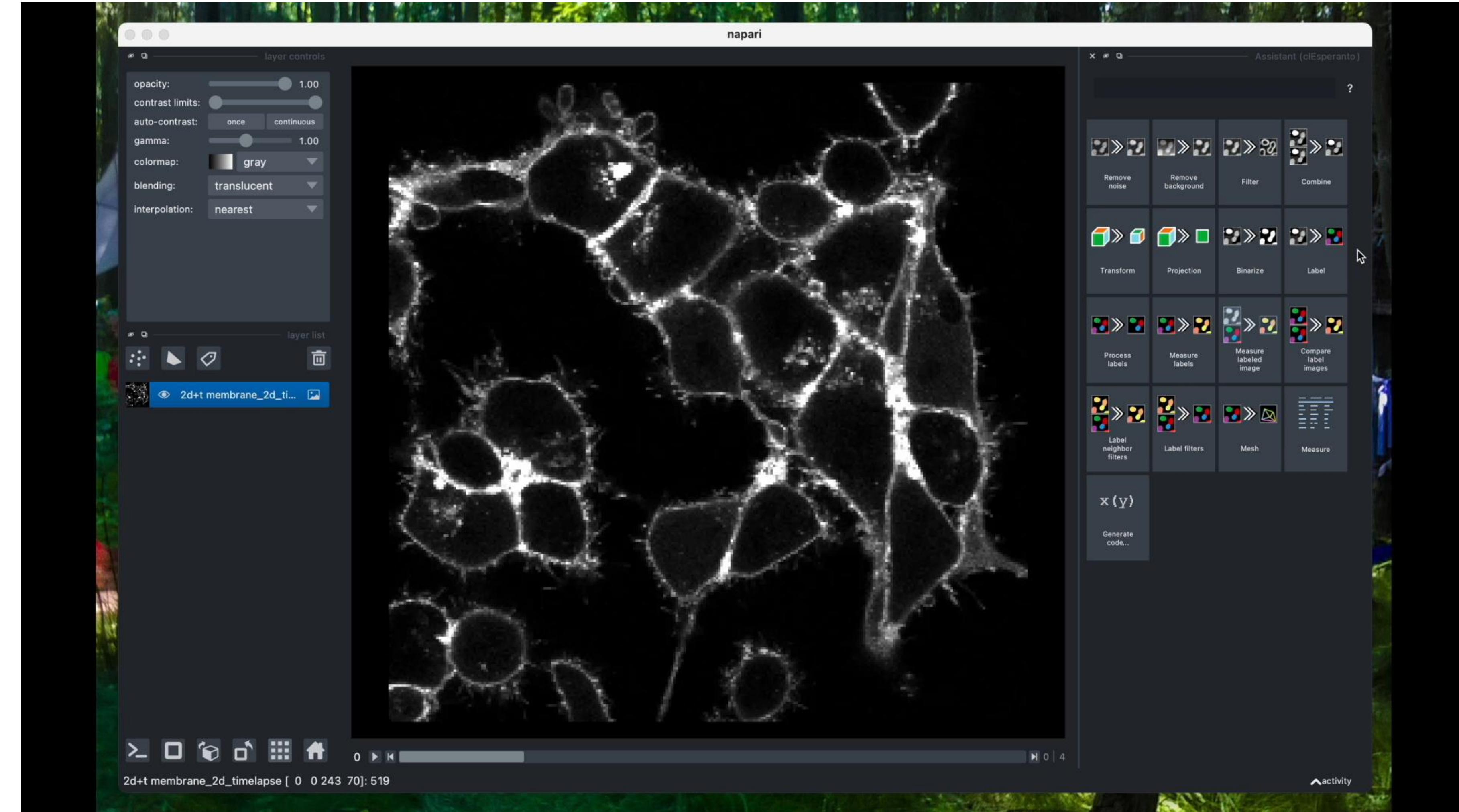


Image data source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD

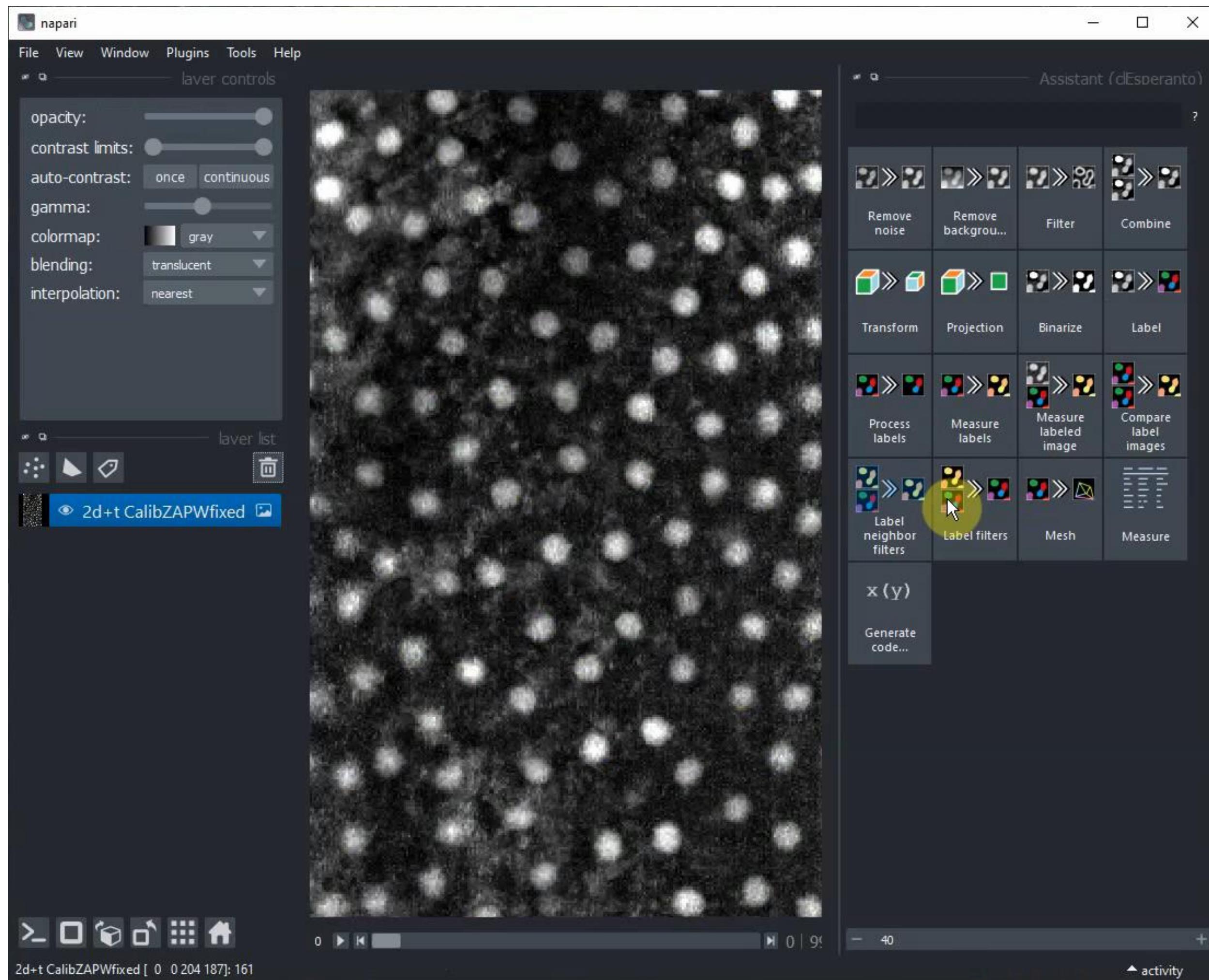
Napari, segment blobs and things with membranes!

- Filtering,
- thresholding,
- spot detection,
- seeded watershed segmentation,
- Voronoi-Otsu-labeling



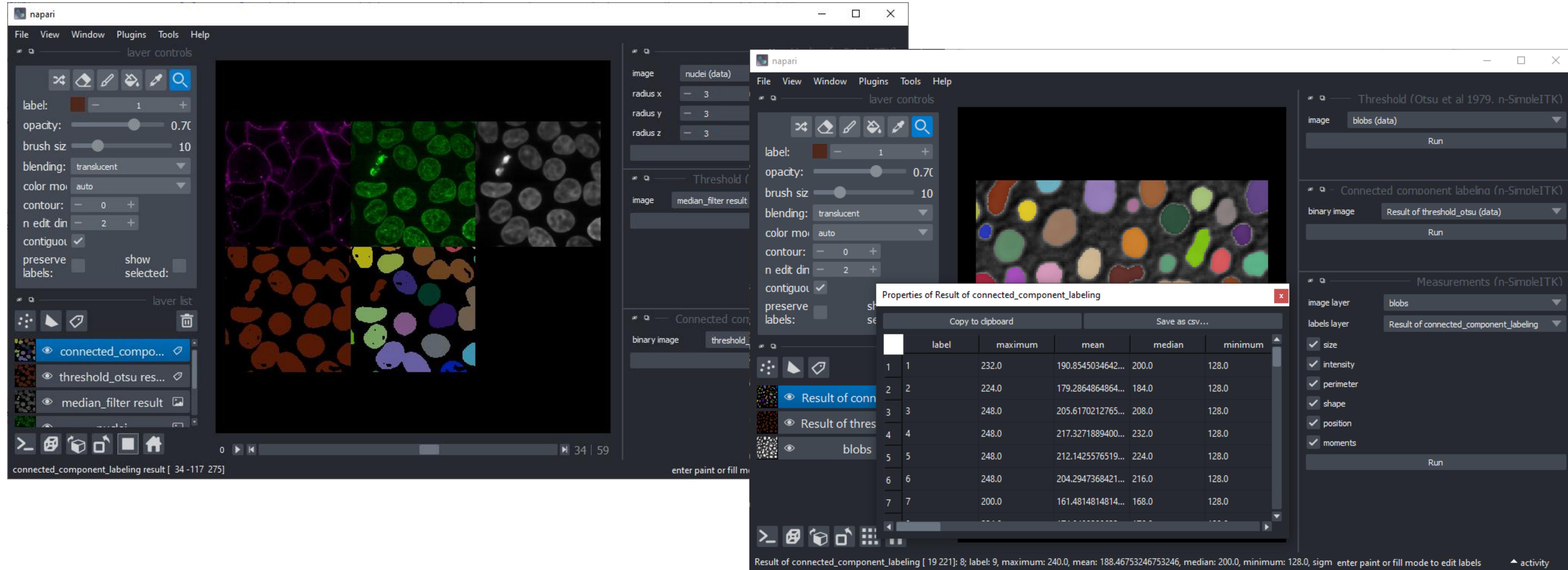
Napari + pyclesperanto + assistant

GPU-accelerated image processing with
a pocket-calculator like graphical user
interface



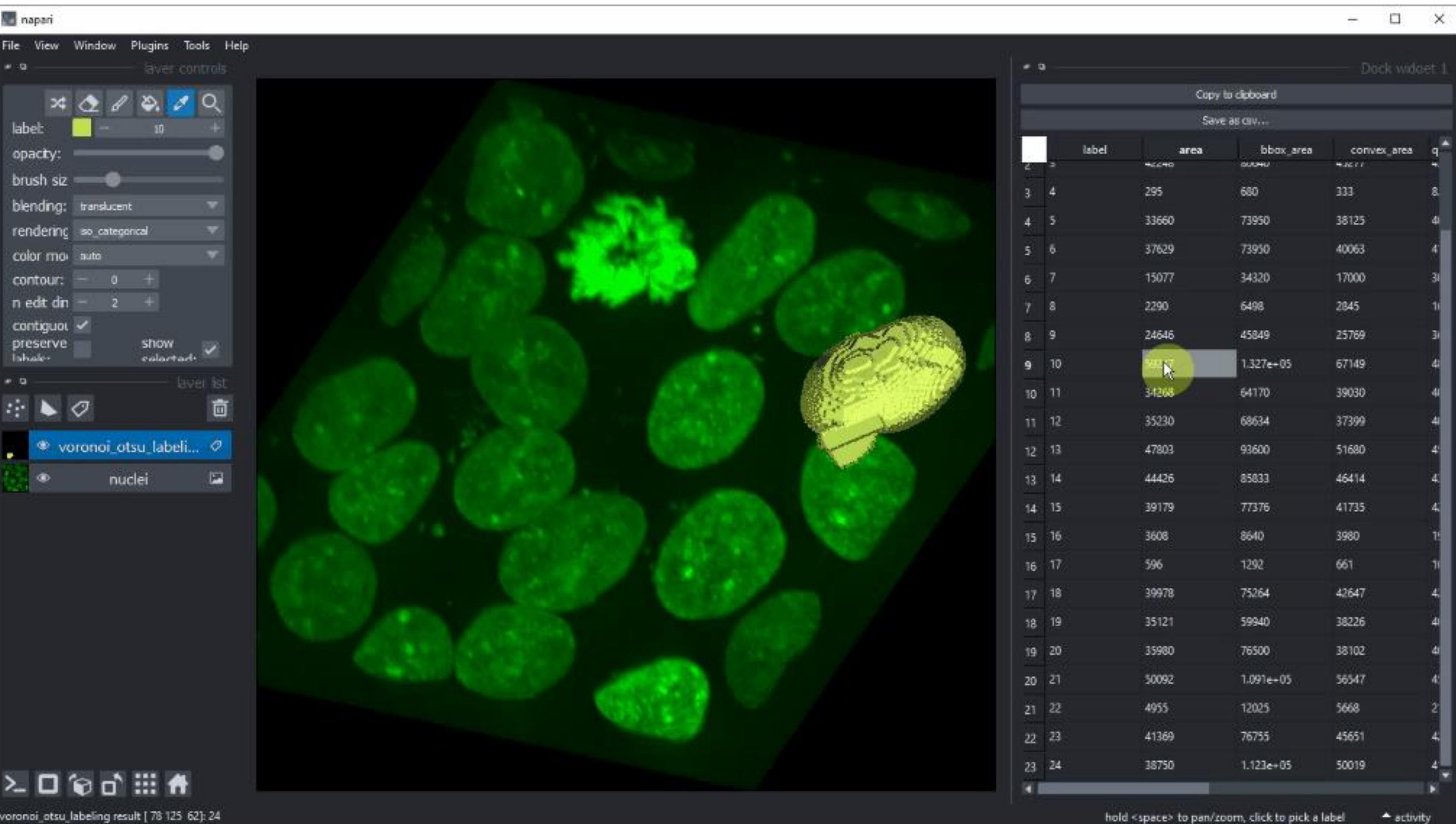
napari-simpleitk-image-processing

Recommended for 3D-measurements, based on the SimpleITK-project



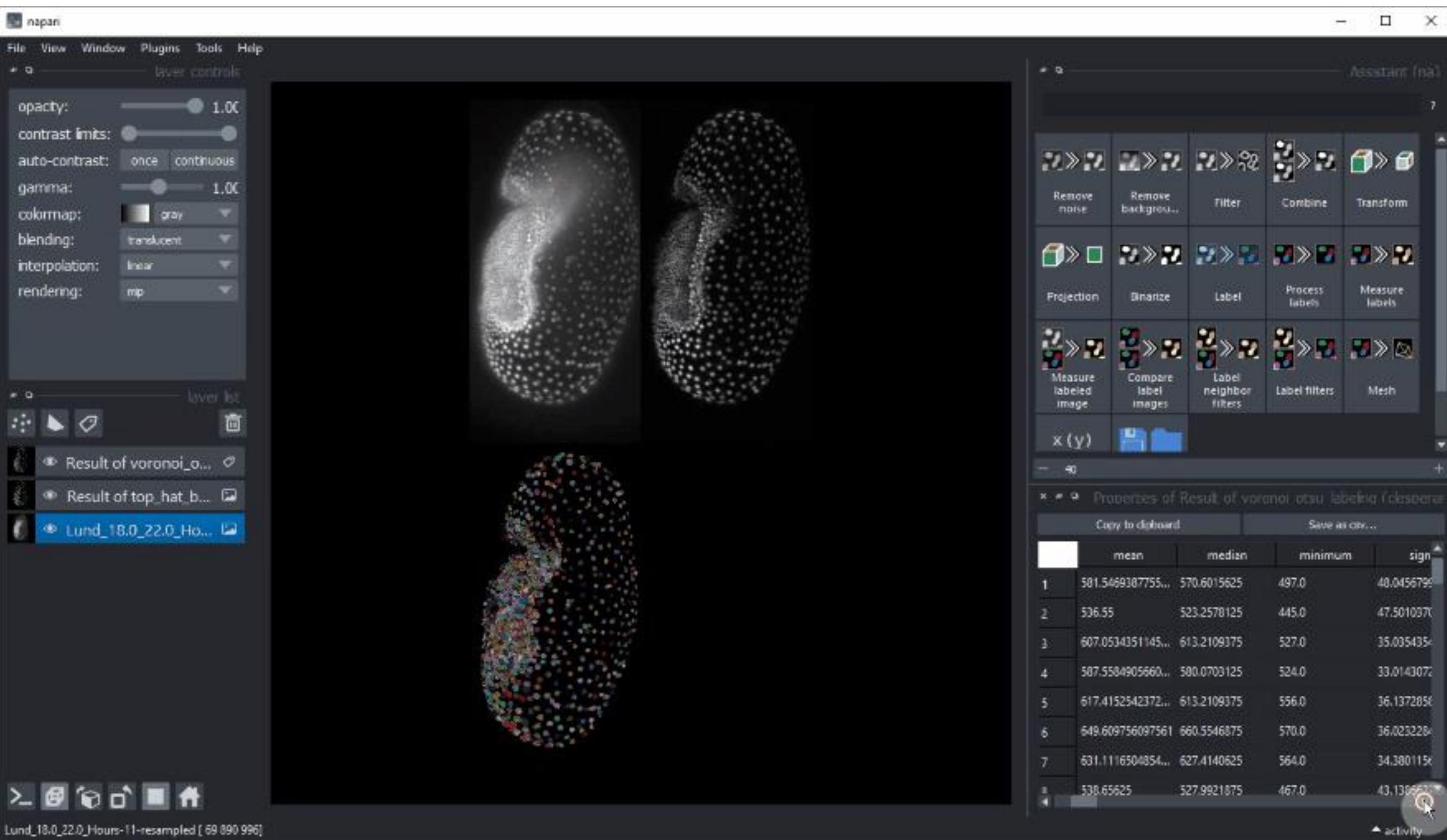
Data exploration

- Click on a cell to view the object the measurement belongs to



Data exploration

- Double-click on a column of measurements to view a parametric image



Data exploration

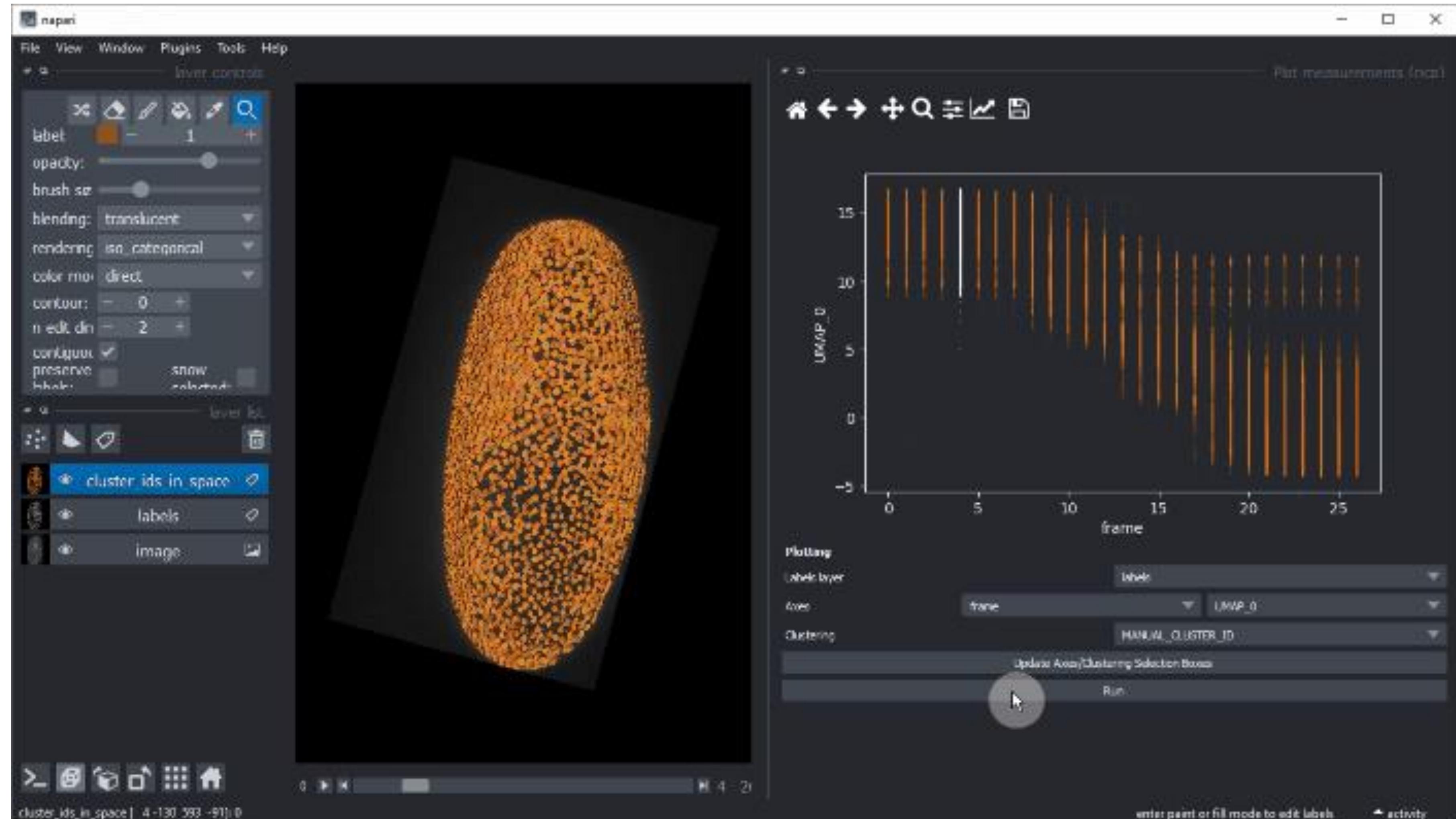
- Manual clustering to gain deeper insights in relationships between measured parameters



Laura Žigutytė
@zigutyte

Ryan Savill
@RyanSavill4

Marcelo Zoccoler
@zoccolermarcelo



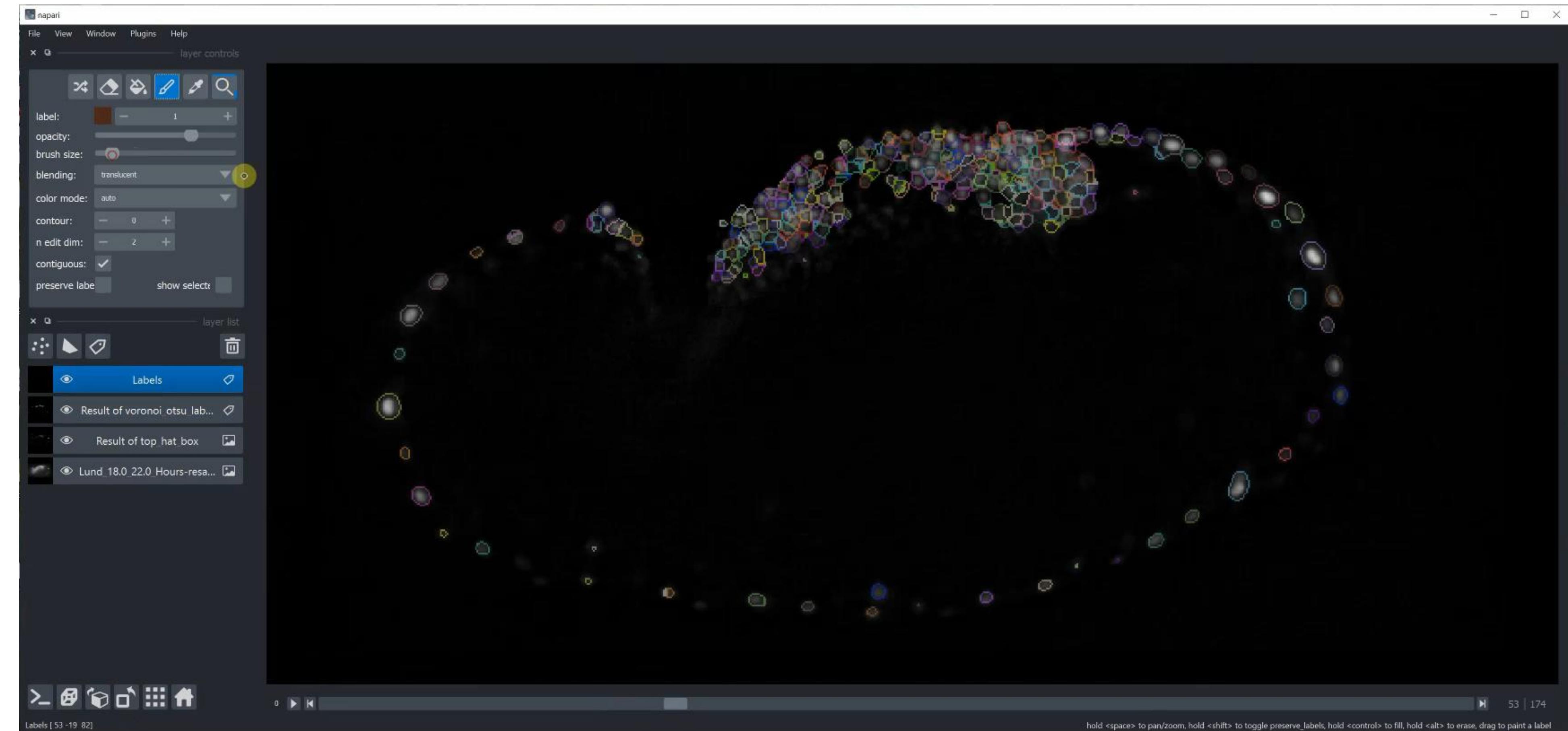
<https://github.com/BiAPoL/napari-clusters-plotter>

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

Supervised machine learning for tissue classification

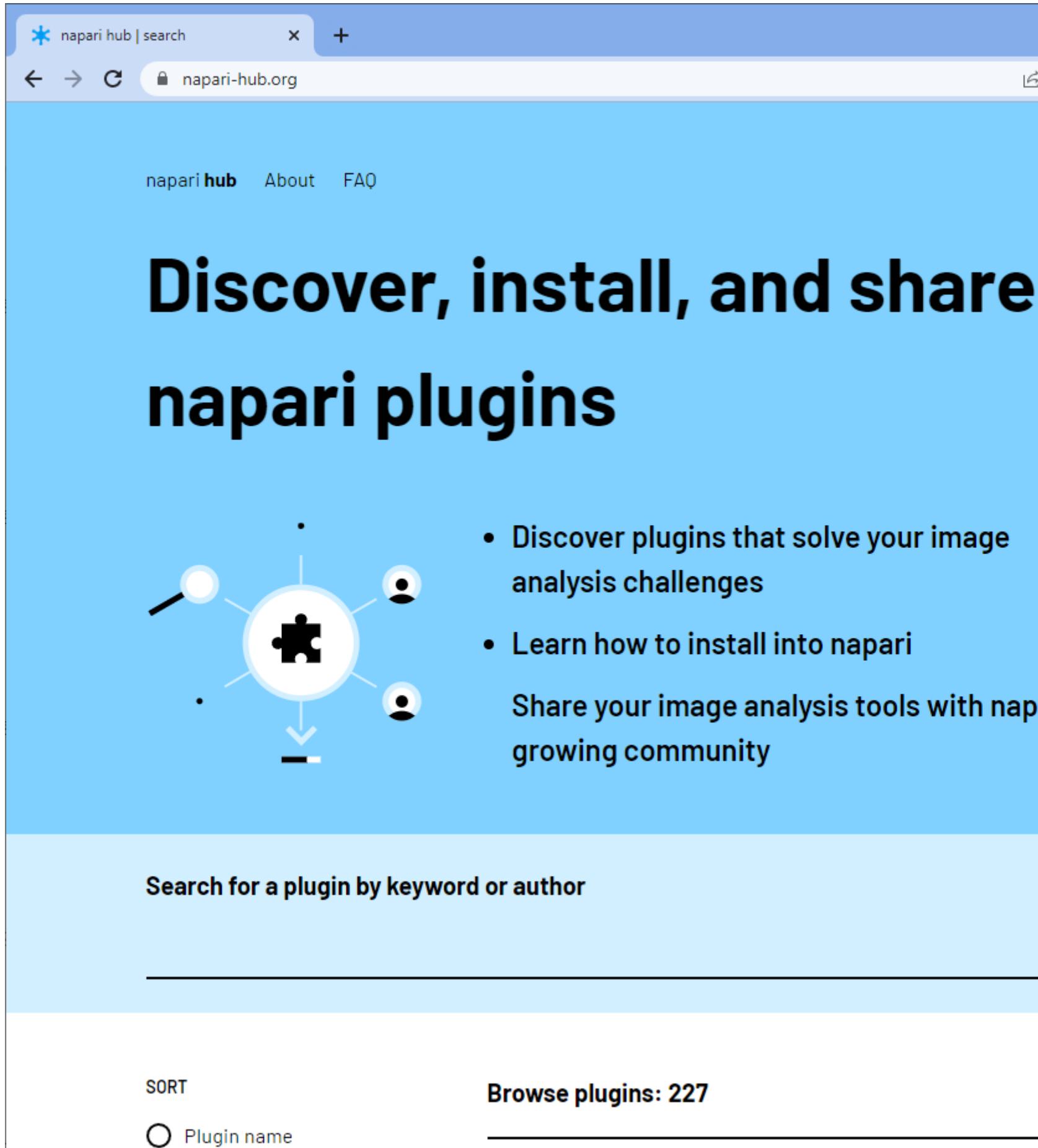
GPU-accelerated
Random Forest
Classifiers based on

- scikit-learn and
- clesperanto

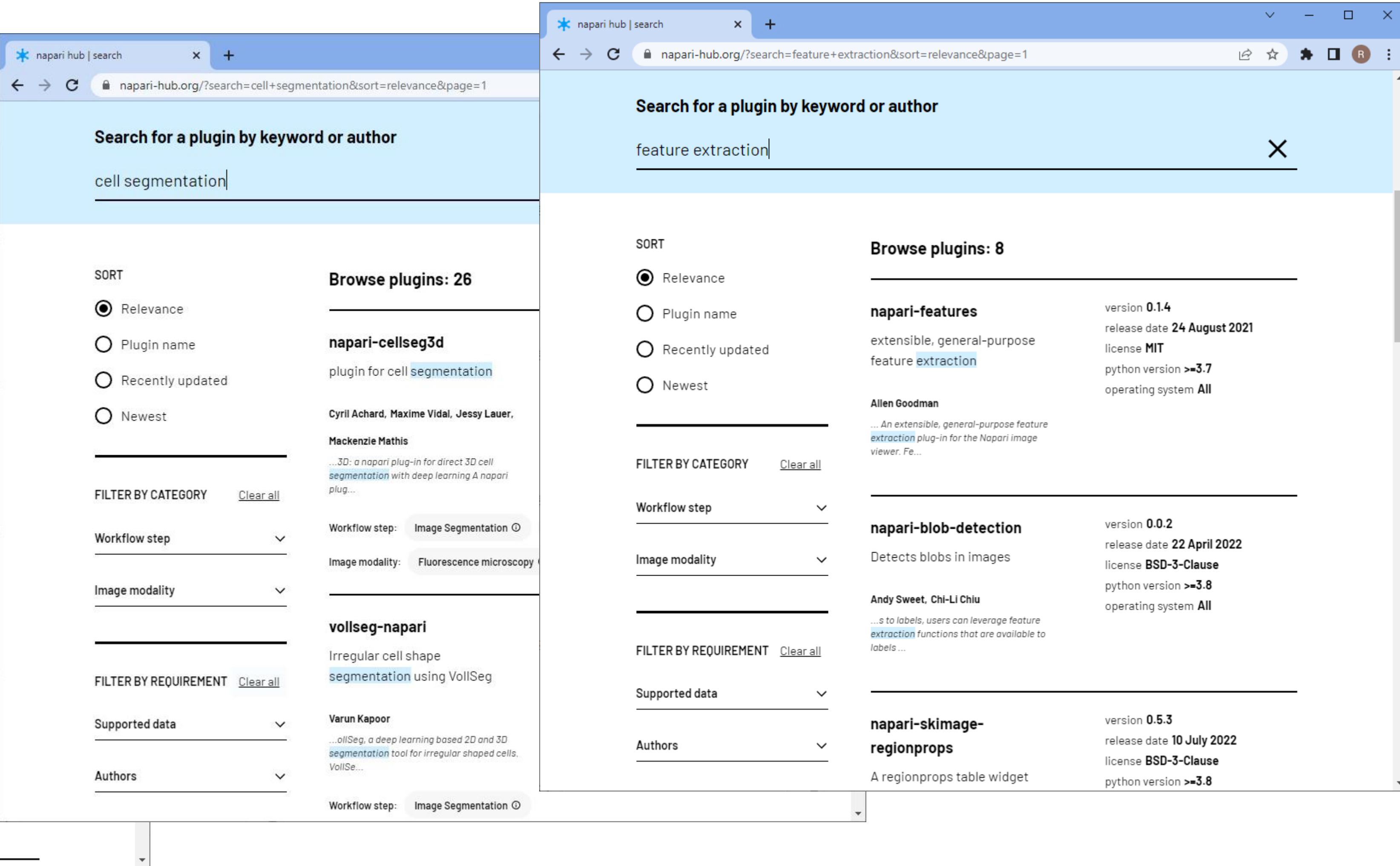


The Napari Hub

Search engine for napari plugins



The screenshot shows the main landing page of the Napari Hub. At the top, there's a navigation bar with links for "napari hub", "About", and "FAQ". Below this, a large heading reads "Discover, install, and share napari plugins". To the left of the text is a graphic featuring a central puzzle piece surrounded by four user icons, symbolizing community and collaboration. On the right side of the heading, there are three bullet points: "Discover plugins that solve your image analysis challenges", "Learn how to install into napari", and "Share your image analysis tools with napari's growing community". Below the heading is a search bar labeled "Search for a plugin by keyword or author". Underneath the search bar are two filter options: "SORT" (set to "Plugin name") and "Browse plugins: 227".

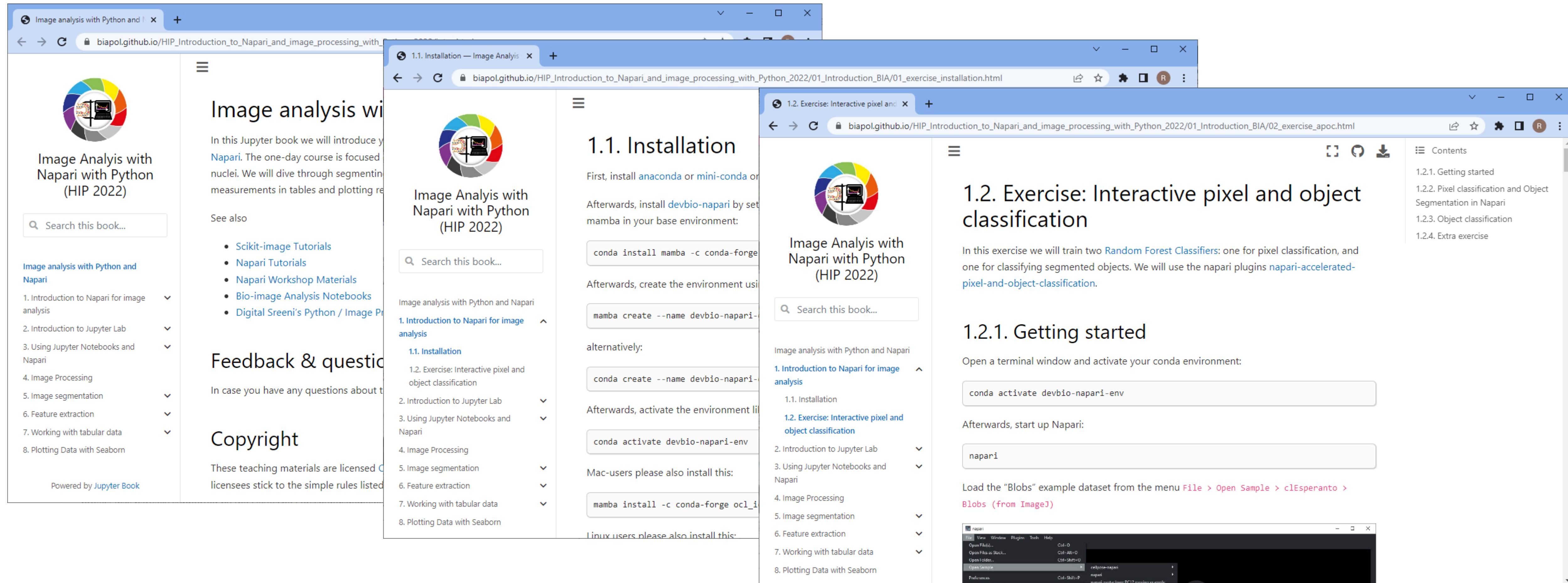


The screenshot shows the search results for the query "feature extraction". The search bar at the top contains the text "feature extraction". On the left, there are filtering options: "SORT" (set to "Relevance"), "FILTER BY CATEGORY" (set to "Workflow step: Image Segmentation"), "FILTER BY REQUIREMENT" (set to "Supported data"), and "Authors". The main results area shows a list of plugins:

- napari-cellseg3d**: A plugin for cell segmentation by Cyril Achard, Maxime Vidal, Jessy Lauer, Mackenzie Mathis. It's described as "...3D: a napari plug-in for direct 3D cell segmentation with deep learning. A napari plug-in for cell segmentation using a 3D U-Net trained on a dataset of fluorescence microscopy images." It has a "Workflow step: Image Segmentation" and "Image modality: Fluorescence microscopy".
- vollseg-napari**: A plugin for irregular cell shape segmentation using VollSeg by Varun Kapoor. It's described as "...lSeg, a deep learning based 2D and 3D segmentation tool for irregular shaped cells. VollSeg is a Python library that provides a simple API for performing 2D and 3D segmentation of irregular shaped cells using a U-Net architecture trained on a dataset of fluorescence microscopy images." It has a "Workflow step: Image Segmentation" and "Image modality: Fluorescence microscopy".
- napari-blob-detection**: A plugin for detecting blobs in images by Andy Sweet, Chi-Li Chiu. It's described as "...s to labels, users can leverage feature extraction functions that are available to labels...".
- napari-skimage-regionprops**: A regionprops table widget by a user named "Technische Universität Dresden". It's described as "A regionprops table widget".

Exercises

Explore the Jupyter Book for our course today

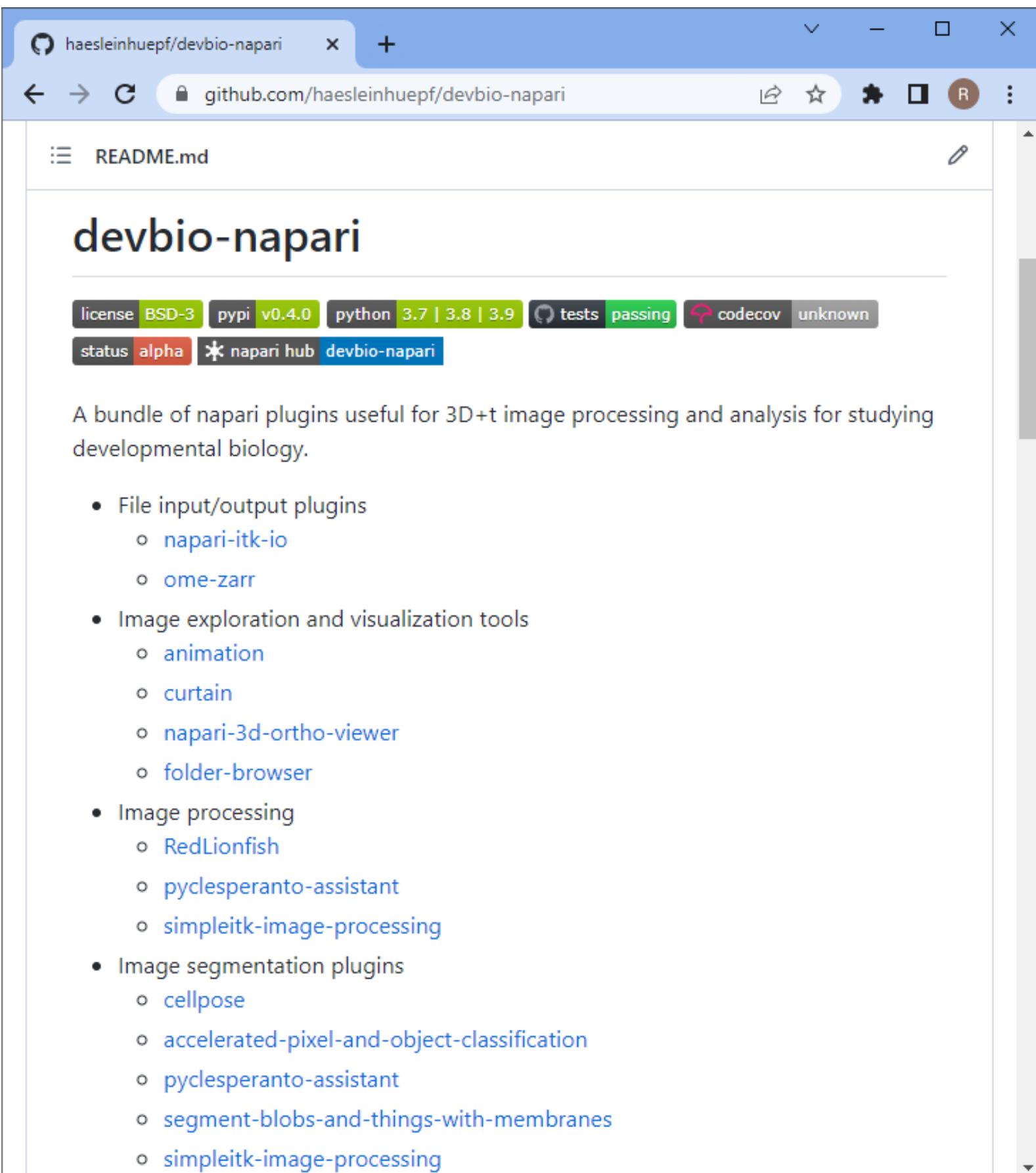


The screenshot shows three browser tabs open, each displaying a page from a Jupyter Book:

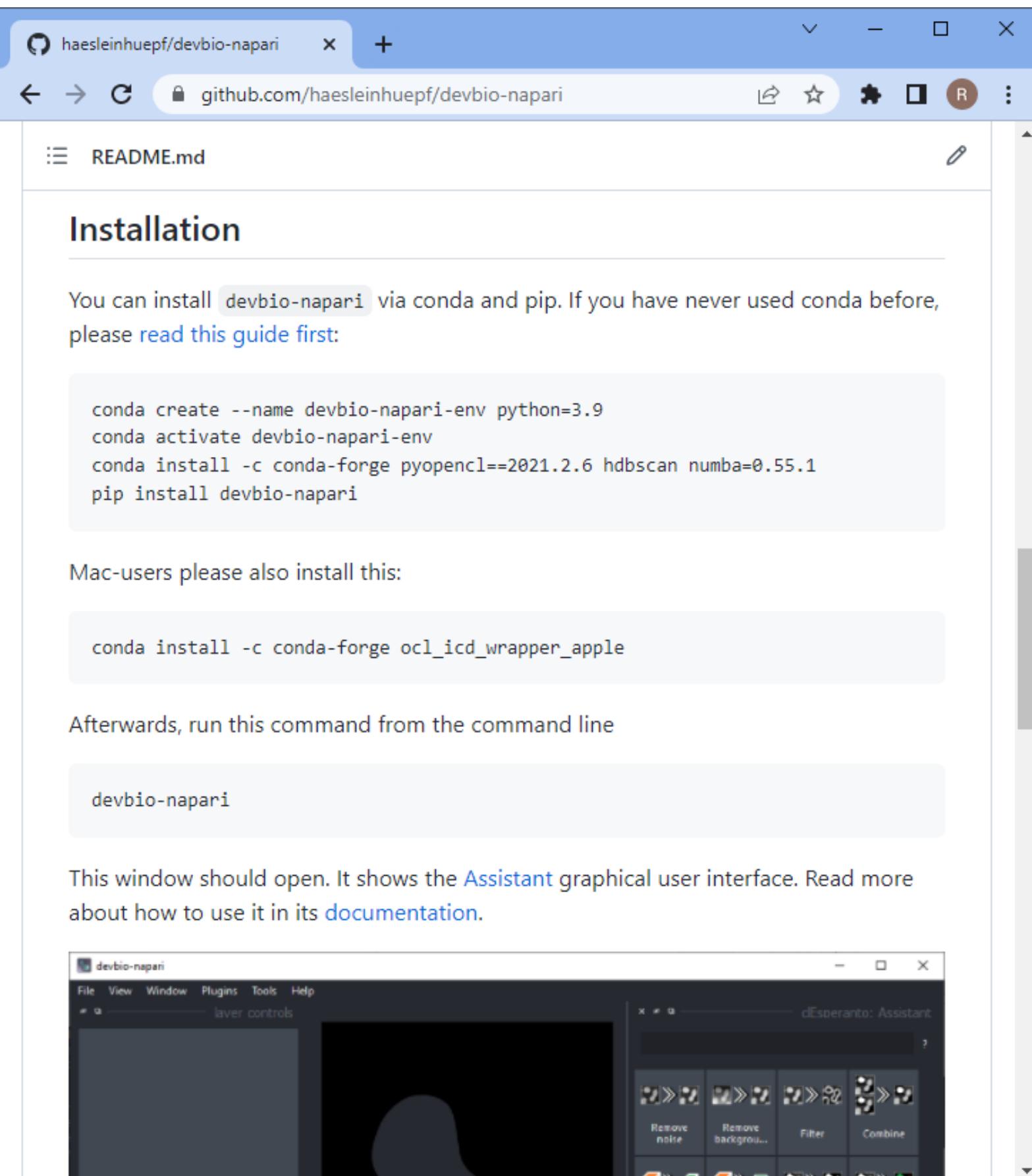
- Tab 1: Image analysis with Python and Napari (HIP 2022)**
 - Image analysis with Napari with Python (HIP 2022)**: Main page with introduction, installation instructions, and exercises.
 - Feedback & questions**: Placeholder for feedback.
 - Copyright**: License information.
- Tab 2: 1.1. Installation — Image Analysis**
 - 1.1. Installation**: Detailed steps for installing Anaconda or miniconda, followed by devbio-napari using mamba.
 - Code snippets**:
 - conda install mamba -c conda-forge
 - mamba create --name devbio-napari
 - Alternatives**:
 - conda create --name devbio-napari
 - conda activate devbio-napari-env
 - Mac users**: mamba install -c conda-forge ocl_i
 - Linux users**: mamba install -c conda-forge ocl_i
- Tab 3: 1.2. Exercise: Interactive pixel and object classification**
 - 1.2. Exercise: Interactive pixel and object classification**: Overview of the exercise.
 - Getting started**: Instructions to open a terminal, activate the environment, and start Napari.
 - Code snippets**:
 - conda activate devbio-napari-env
 - napari
 - Load dataset**: File > Open Sample > c1Esperanto > Blobs (from ImageJ)

Exercise: Install devbio-napari

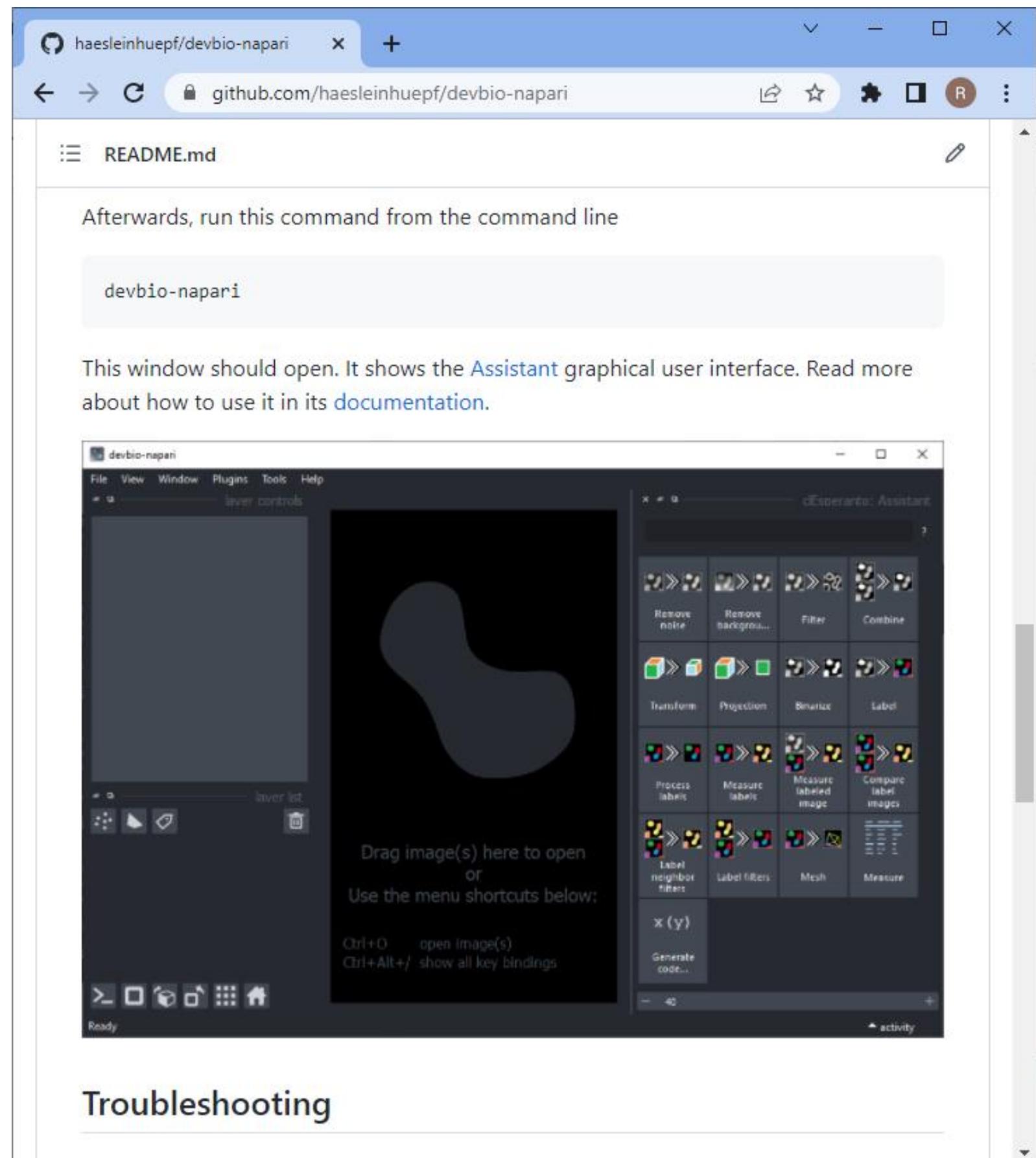
A collection of napari plugins for developmental biologists



The screenshot shows the GitHub README.md page for the devbio-napari repository. It features a header with the repository name, a table of contents, and a main content area. The main content area includes sections for installation instructions, command-line commands, and a screenshot of the graphical user interface.



This screenshot shows the 'Installation' section of the GitHub README.md page. It provides instructions for installing via conda and pip, along with specific command-line commands for different operating systems. It also includes a screenshot of the graphical user interface.



This screenshot shows the 'Troubleshooting' section of the GitHub README.md page. It contains a screenshot of the graphical user interface and a note about running the command 'devbio-napari' from the command line.

Exercise: Pixel and object classification

Segment and classify the blobs in this image.

