Advanced Methods and Human Cell Technologies (AMHCT)
RegBioMed Master's Course

Navigating the Reproducibility Storm with Bio-Image Analysis

Dr. Marcelo Leomil Zoccoler

Bio-Image Analysis Technology Development Group - Physics of Life (PoL) – TU Dresden

Re-using material from:

Robert Haase (Scads.AI Leipzig); Maleeha Hassan (Helmholtz AI Dresden); Johannes Soltwedel (PoL – TU Dresden);

Table of Contents

- Installation and introduction to **napari**
- Introduction to **OMERO** and **napari-omero**
- Segmentation with Machine Learning using **napari-apoc**
- Quick view of Segmentation with Machine Learning using **micro-sam**
- Feature Extraction with **napari-skimage-regionprops**
- Object Classification with **napari-apoc** and **napari-clusters-plotter**
- **Exercises:**
  - **Instance Segmentation with napari-apoc**
  - **Creating a workflow with napari-omero and napari-apoc**
  - **Reproducing a workflow**
- **Discussion**

https://biapol.github.io/AMHCT_Bio_Image_Analysis_2025/intro.html

napari Installation

- napari as a Python Library
- napari as a Bundle App
- Installing Plugins

Installing napari as a Python package

napari is a Python library

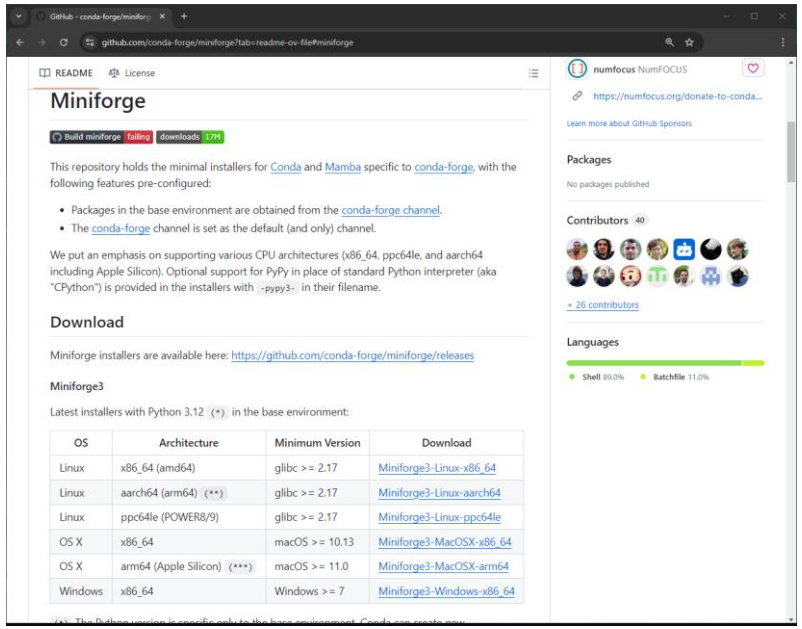
The recommended way to install napari is as a Python package

https://napari.org/stable/tutorials/fundamentals/installation.html#install-as-python-package-recommended

Your computer needs Python to run a Python package

— How do I install Python?

Install Miniforge!



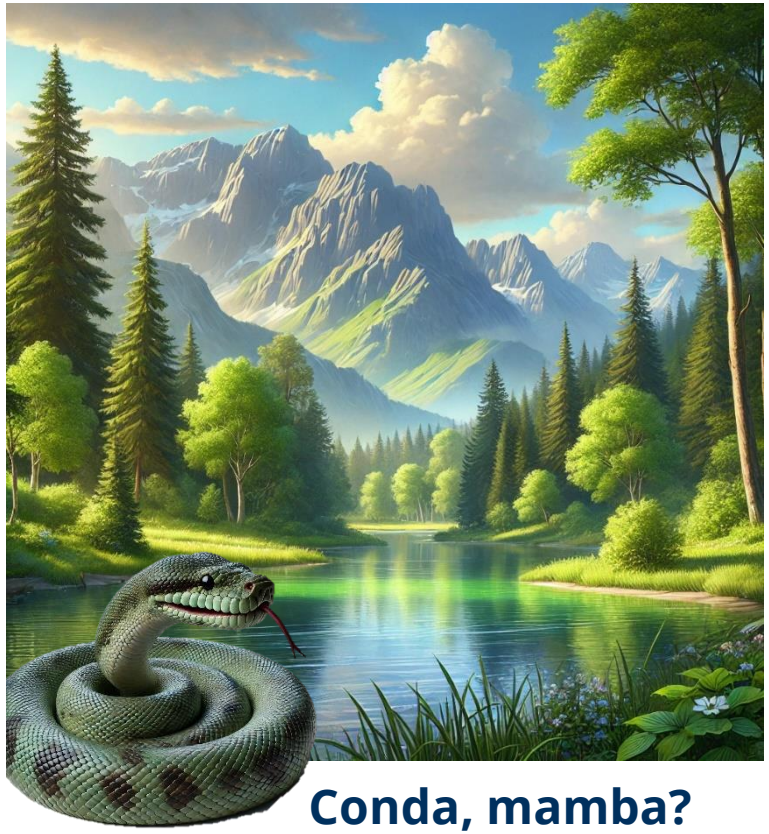
https://github.com/conda-forge/miniforge?tab=readme-ov-file#miniforge

- Miniforge is a minimal installer for `conda` and `mamba` (efficient conda in C++)
- `conda` is an environment and a package manager
- `conda` can create Virtual Enviroments and install Python (and other) packages, including napari

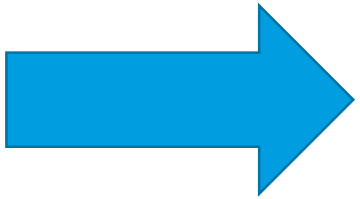
https://hackmd.io/@talley/SJB_lObBi#Terms

Installing napari as a Python package

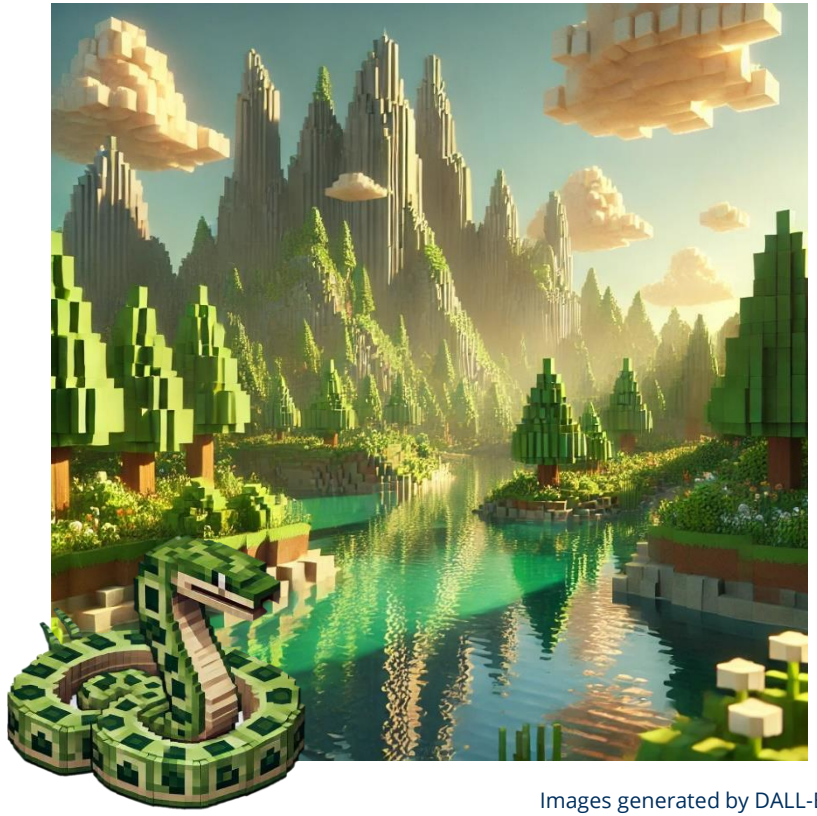
Wait, **environments ?**



Conda, mamba?



Yes, virtual environments!



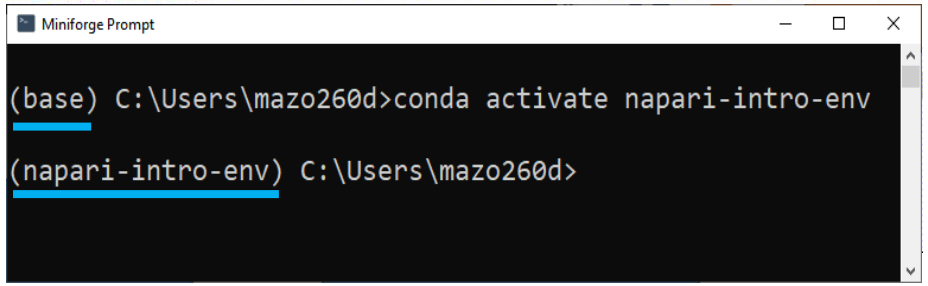
Images generated by DALL-E, developed by OpenAI.

Smiley icons from Flaticon.com

Installing napari as a Python package

Install it as a Python package in a Virtual Environment

— A virtual environment is an isolated collection of packages, settings, and an associated python interpreter



Commands:

```
mamba create -y -n napari-intro-env -c conda-forge python=3.10
mamba activate napari-intro-env
mamba install -c conda-forge napari pyqt
```

name — channel — package (version)

Some packages can only be installed with `pip`

- `conda` checks for version compatibility before installing
- `pip` tries to install, then lets you know if something went wrong
- `pip` cannot create environments

Installing napari

Napari can also be downloaded and installed like other software

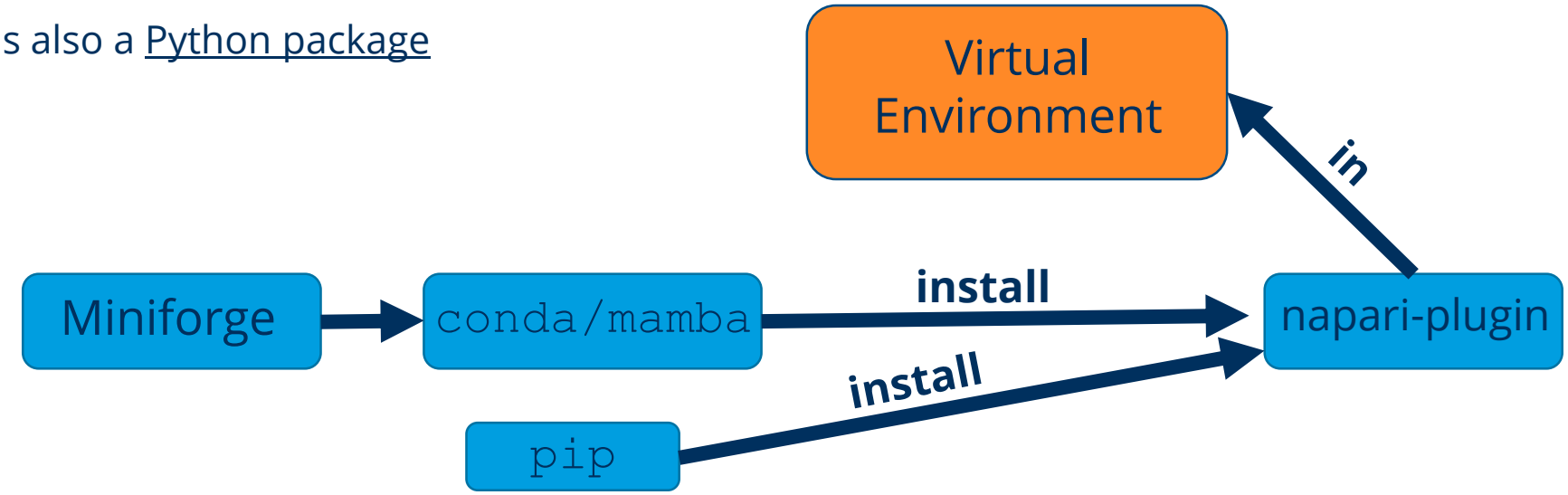
- https://napari.org/stable/tutorials/fundamentals/installation_bundle_conda.html#how-to-install-napari-as-a-bundled-app

This is more convenient, however it may be difficult to manage different plugin versions mid-long term

- In case of a dependency version mismatch, it may be necessary to uninstall and re-install the software and all the plugins again

Installing a plugin

A napari plugin is also a Python package



Commands:

mamba activate napari-intro-env

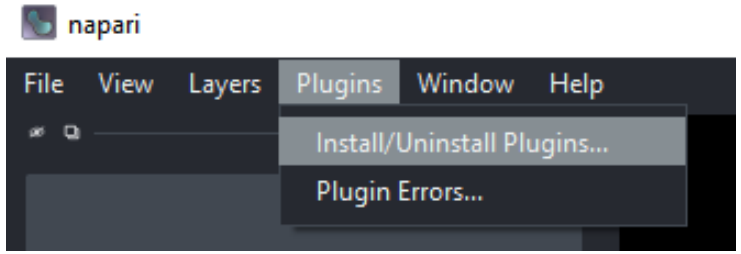
mamba install napari-plugin **OR** **pip** install napari-plugin

OR

conda install napari-plugin

Always check plugin installation instructions!

Via Plugins Menu:



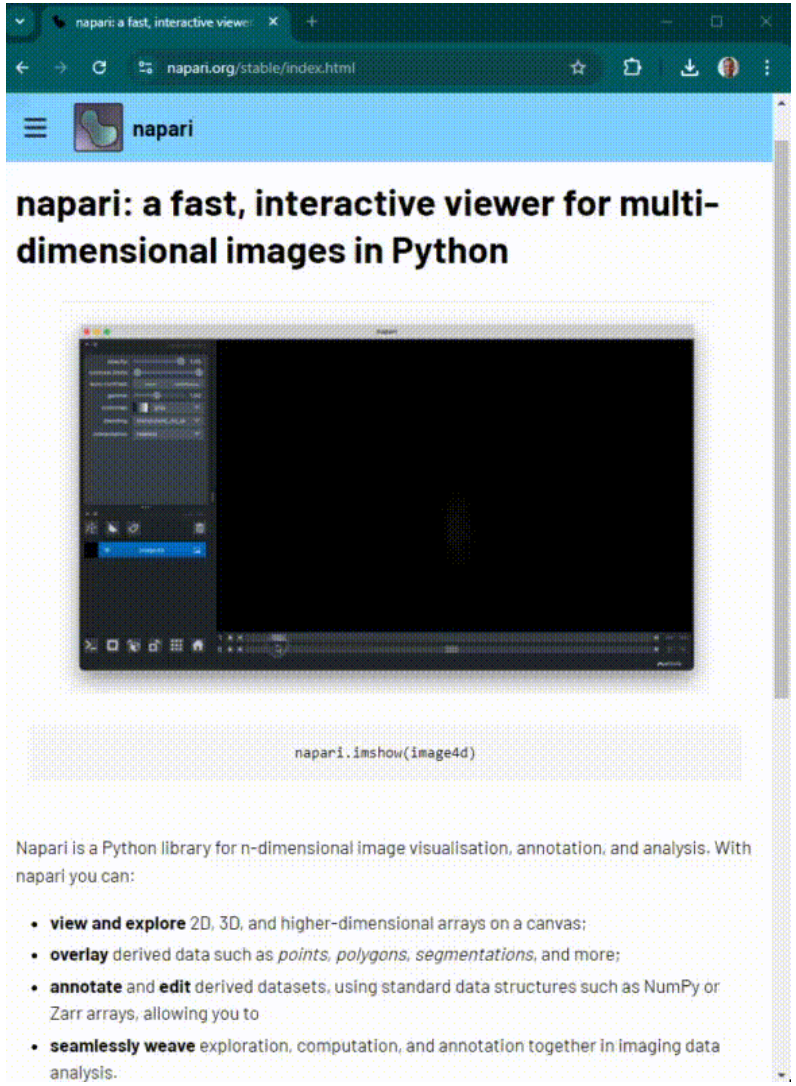
Introduction to napari

- napari Viewer
- Layer Types
- napari Plugins

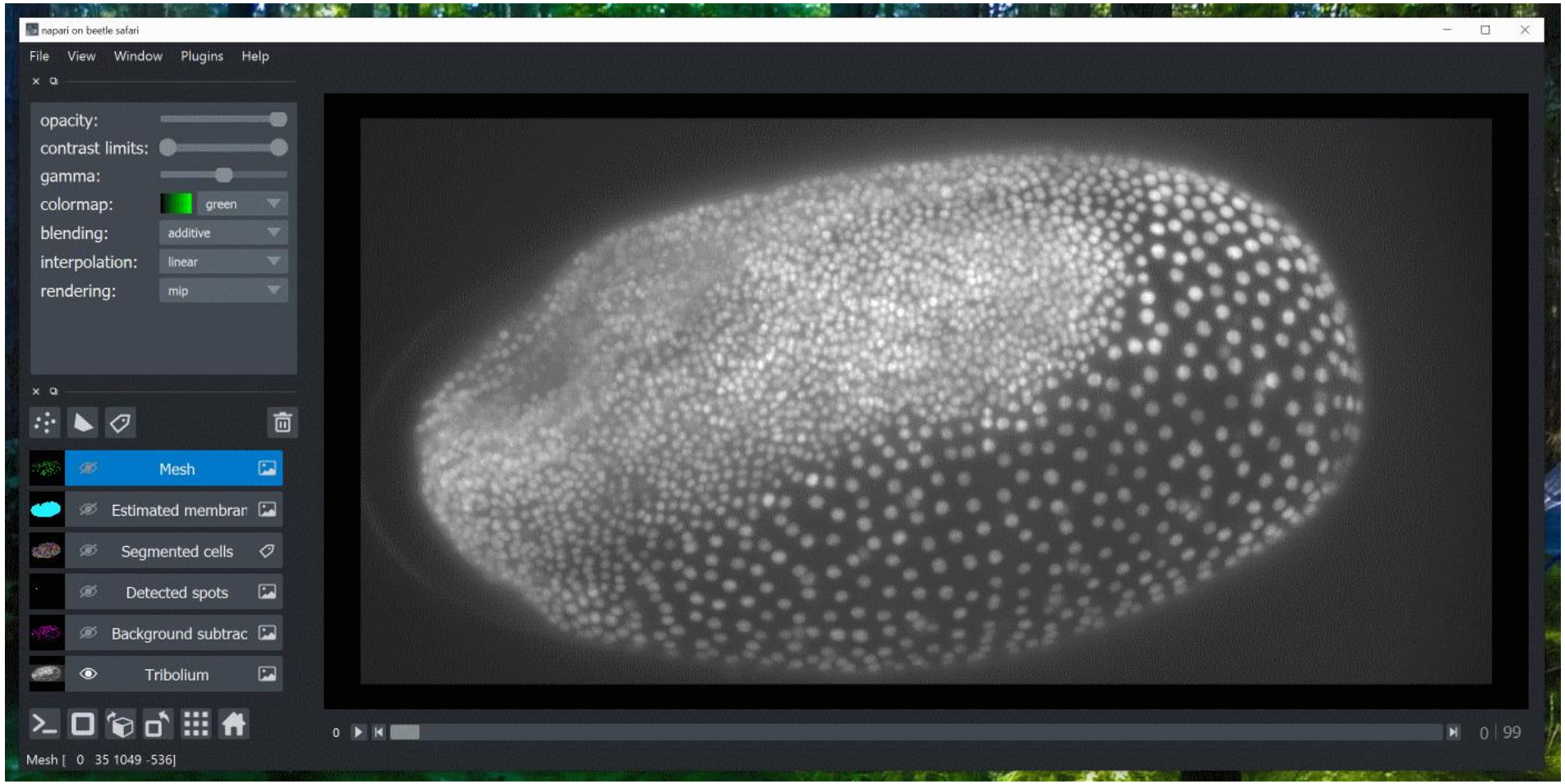
Napari: 3D viewer for Python

Multi-dimensional image viewer in python

https://napari.org/



Napari: 3D viewer for Python



https://napari.org/

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

Napari user interface

Menus

View configuration / tools

```
layer.opacity = 0.5
```

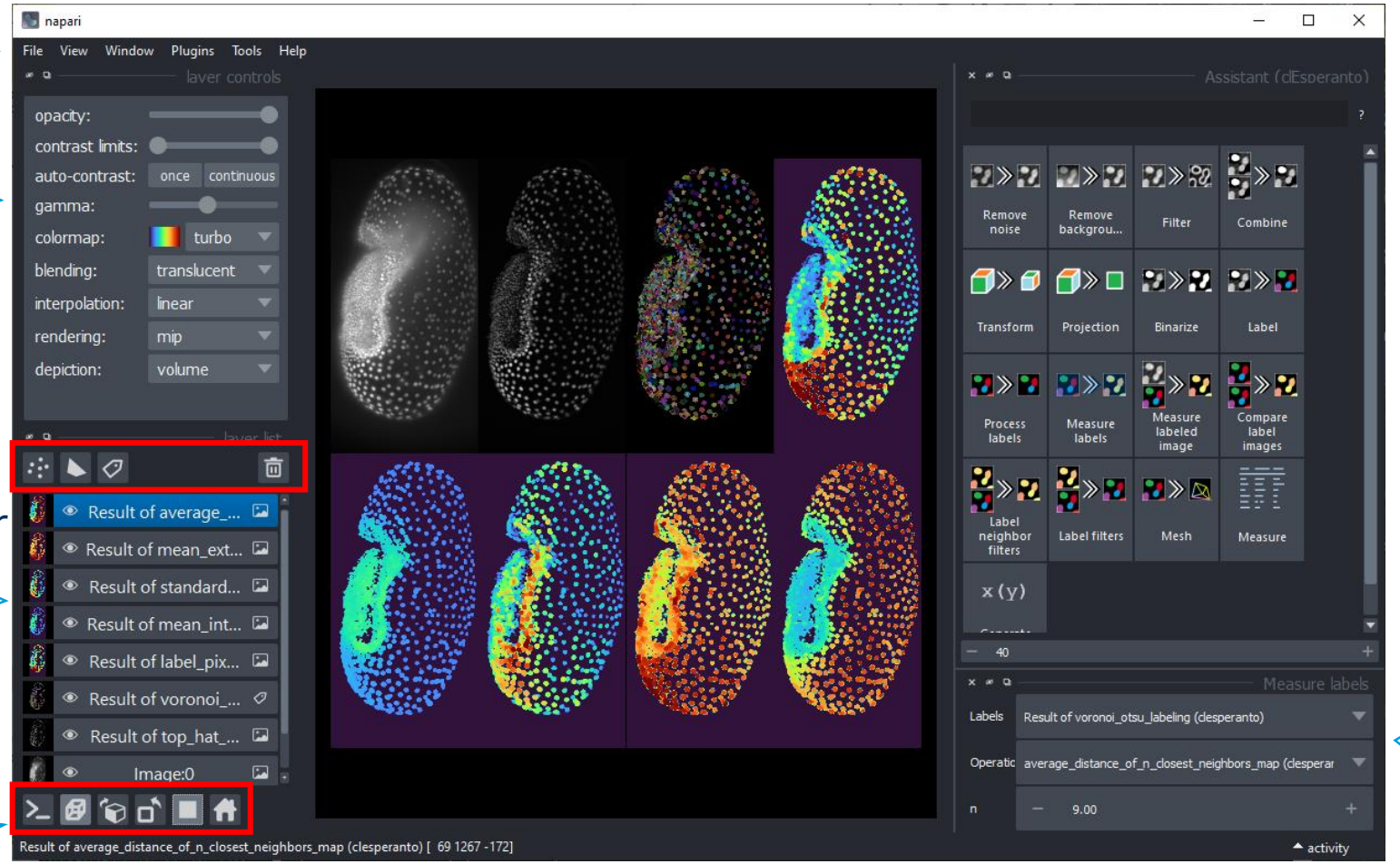
Add empty layers

Delete layer

Layers

```
layer.visible = False
```

Viewer controls



Dock widgets (custom plugins)

Function widgets (custom plugins)

Open Python Console

2D/3D view

Change axes order

Transpose visible axes

Grid mode

Reset view

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

@mazoc.bsky.social

TECHNISCHE UNIVERSITÄT DRESDEN

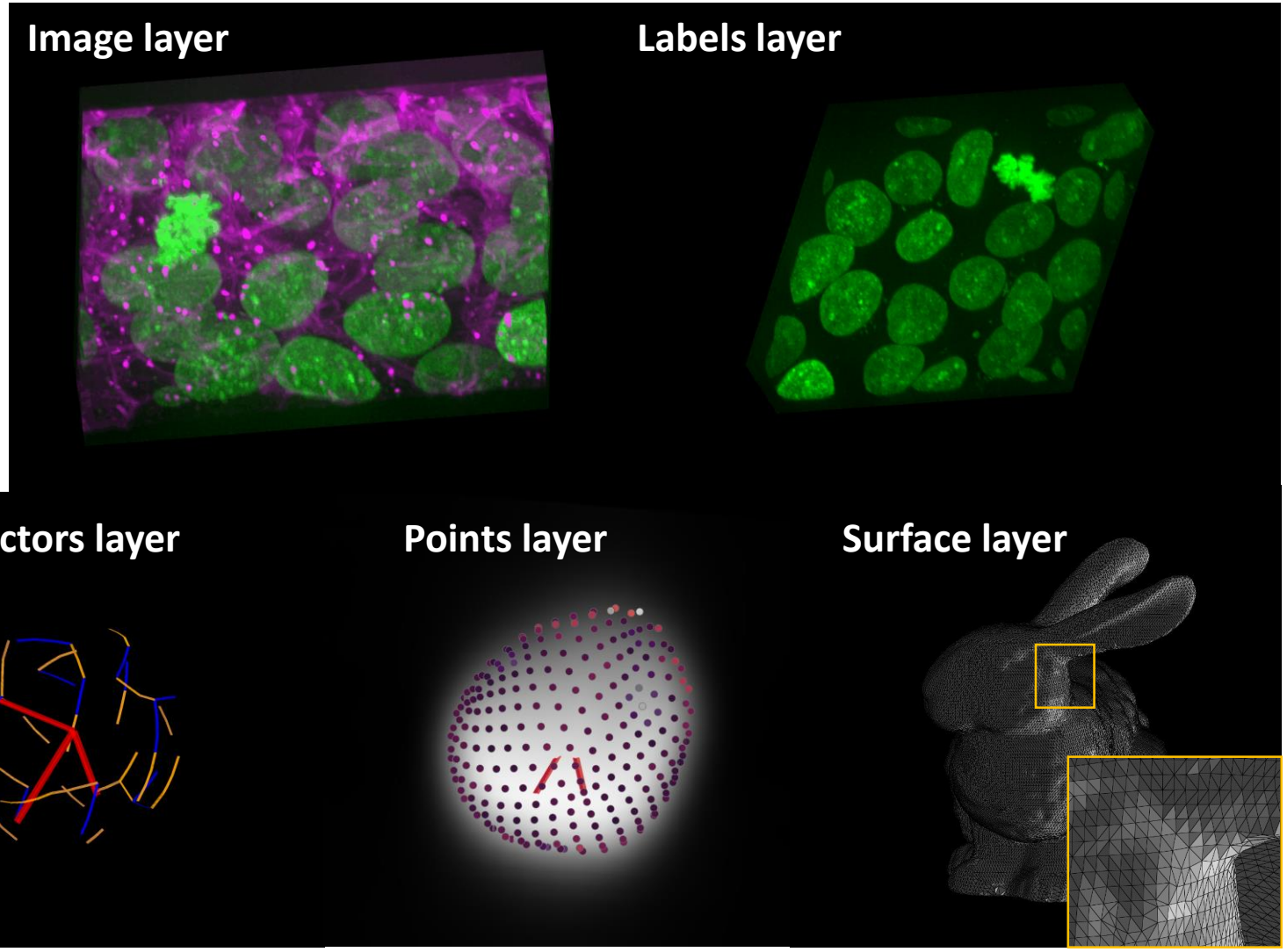
BioDIP Biopolis Dresden Imaging Platform

PoL Physics of Life TU Dresden

DRESDEN concept

Layer types

- **Image layer:** Can be n-dimensional grayscale data (e.g., [CTZYX])
- **Labels layer:** Similar to image layer, but contains only integer numbers (e.g., 0,1,2,3,…)
- **Points layer:** List of coordinates in space
- **Vectors layer**: Direction from point A to point B
- **Surface layer:** Mesh (Vertices, Faces, Values)
- **Tracks layer:** Follow objects through time



https://github.com/quantumjot/arboretum licensed under MIT license

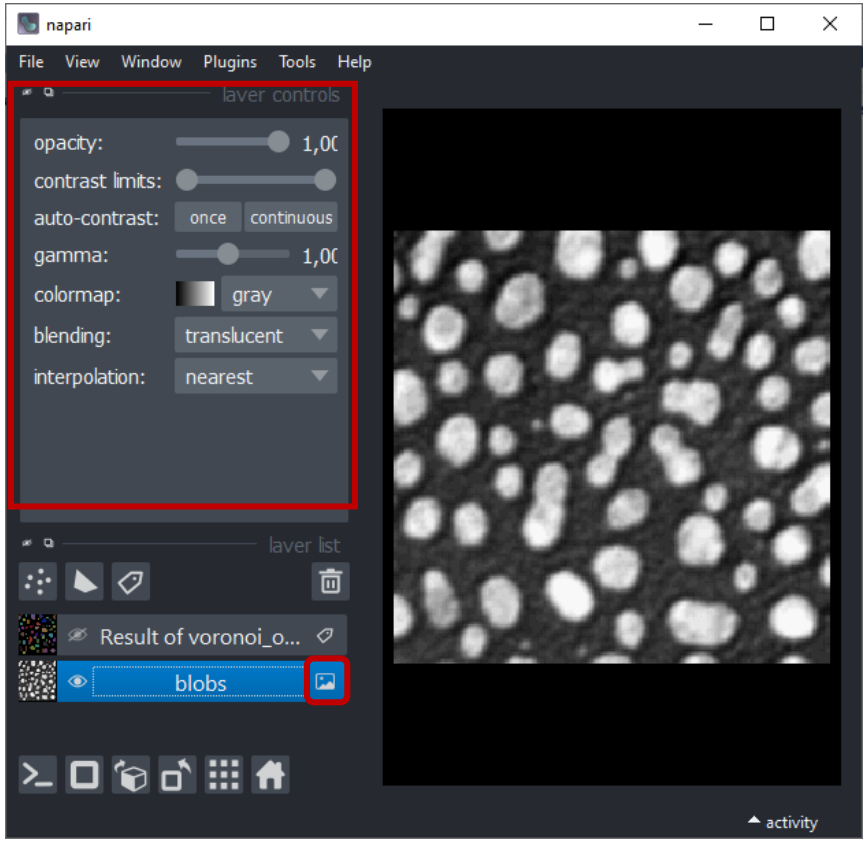
https://github.com/campaslab/napari-stress

https://gitlab.kitware.com/vtk/vtk

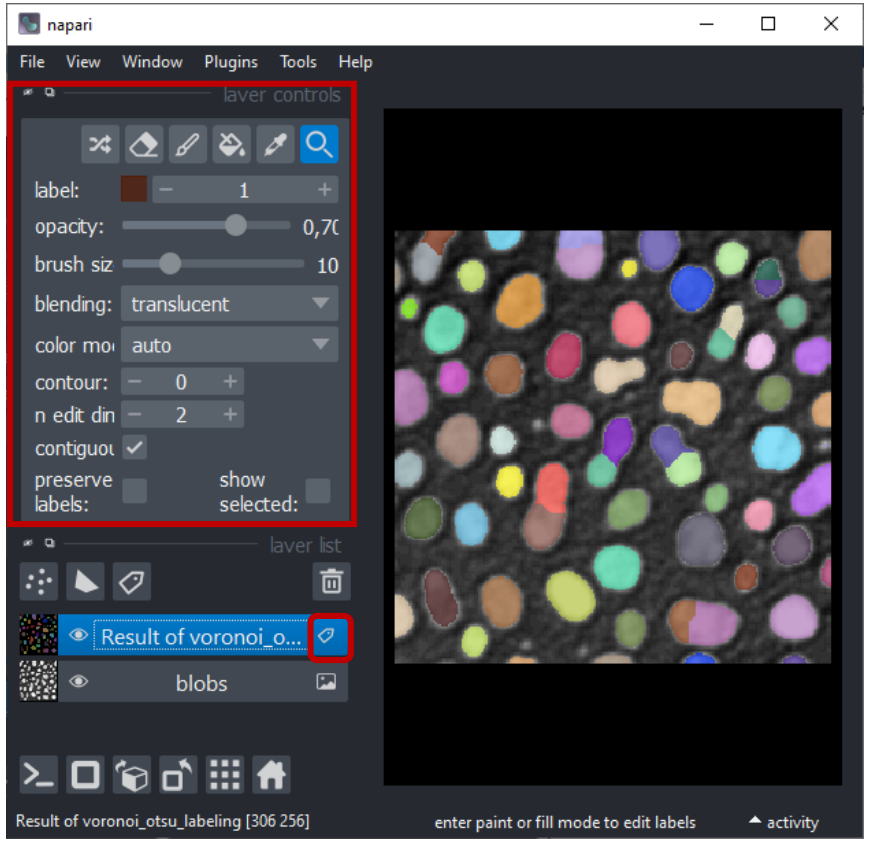
Layer types

Different layers have different tools and options

Image Layer



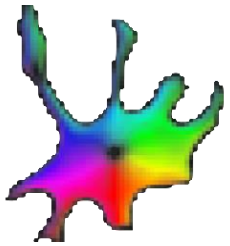
Labels Layer



Python image & data analysis tools are powerful!

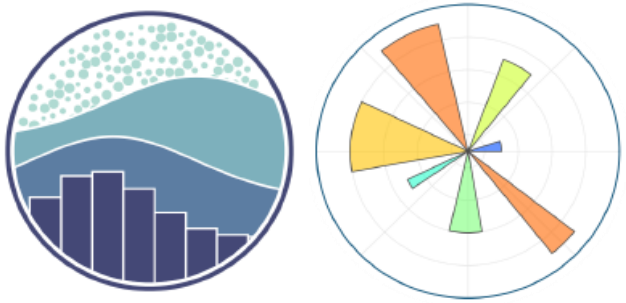
Python scientific image computing toolbox
https://github.com/scikit-image

Python machine-learning toolbox
https://github.com/scikit-learn



Cell segmentation
https://github.com/MouseLand/cellpose

Object detection with Stardist
https://github.com/stardist



Data visualization & exploration
https://github.com/matplotlib
https://github.com/seaborn

cle._

GPU-accelerated image processing
https://github.com/clEsperanto

Napari

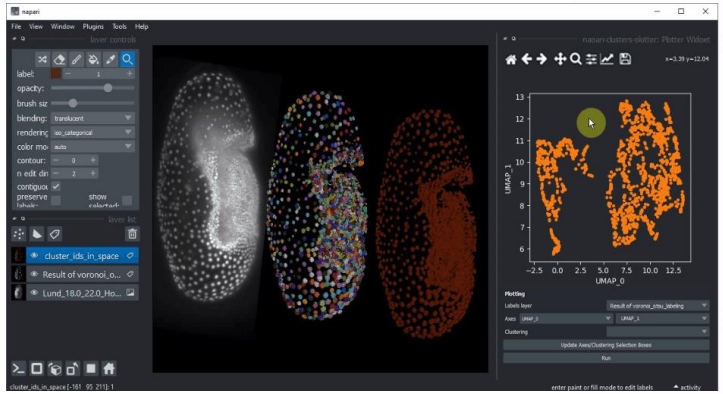
Existing functionality can be turned into plugins!

→ Interactivity

→ Automatic GUI generation

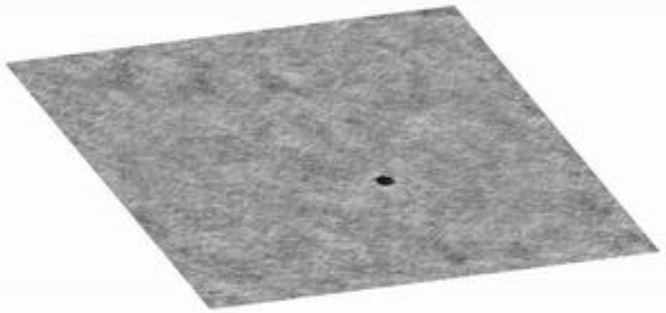
napari plugins

clusters-plotter



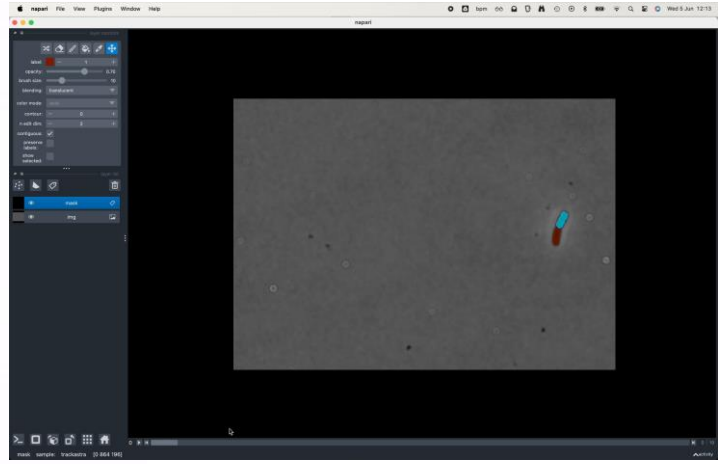
https://github.com/BiAPoL/napari-clusters-plotter

animation



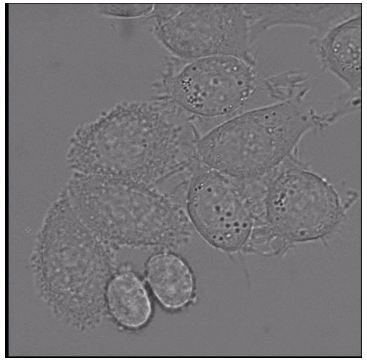
https://github.com/napari/napari-animation

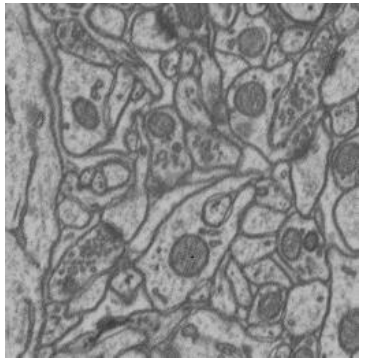
trackastra



https://github.com/weigertlab/trackastra

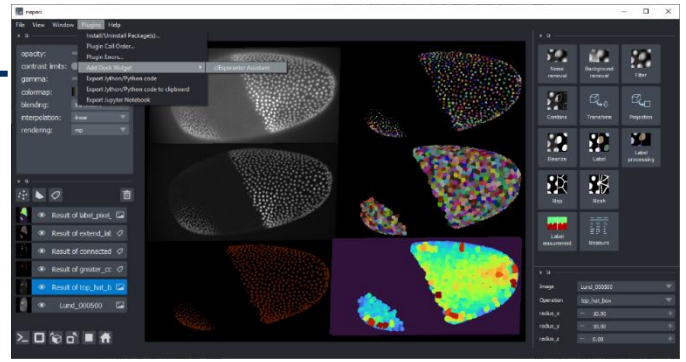
micro-sam





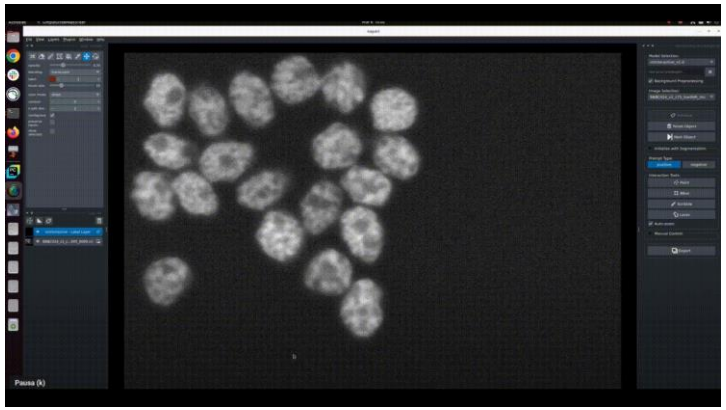
https://github.com/computational-cell-analytics/micro-sam

devbio-napari



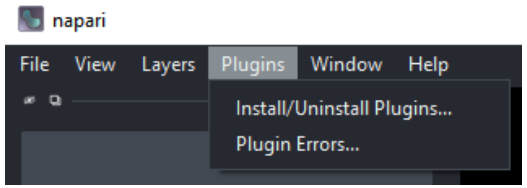
https://github.com/haesleinhuepf/devbio-napari

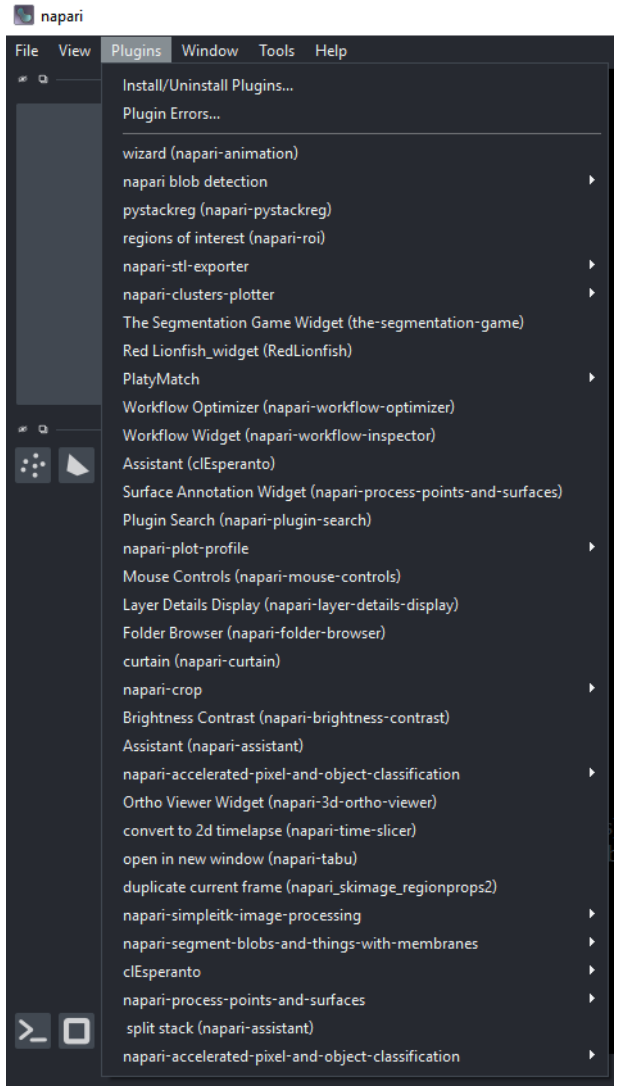
nninteractive

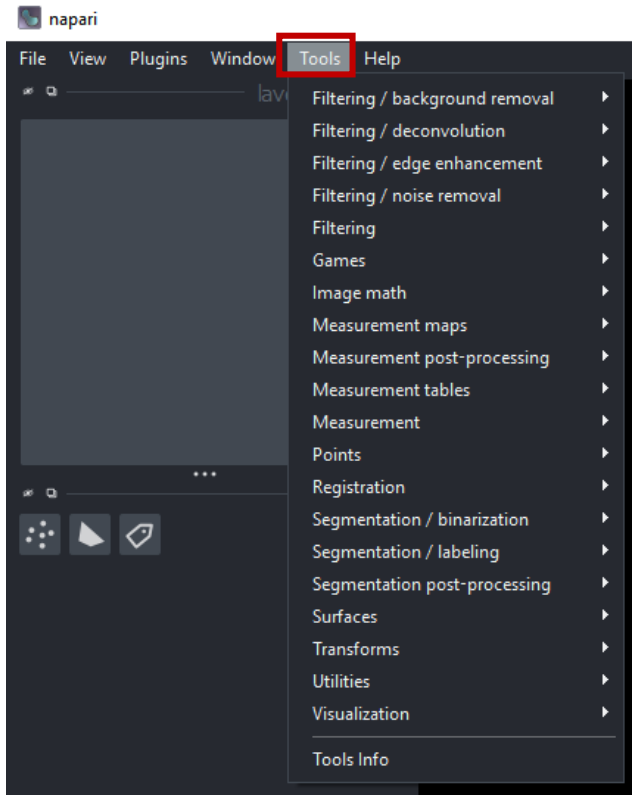


https://github.com/MIC-DKFZ/napari-nninteractive

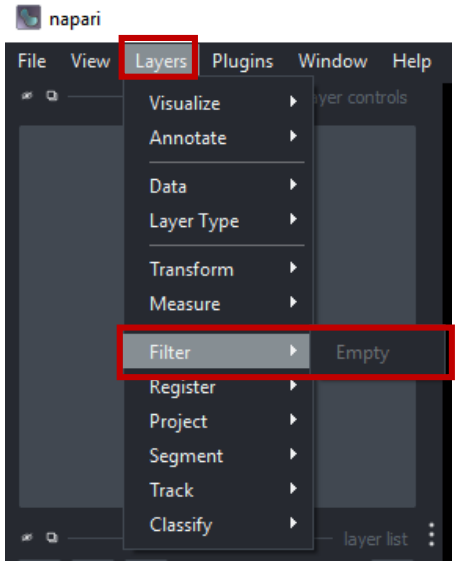
Plugins and Layers Menus







devbio-napari plugin bundle



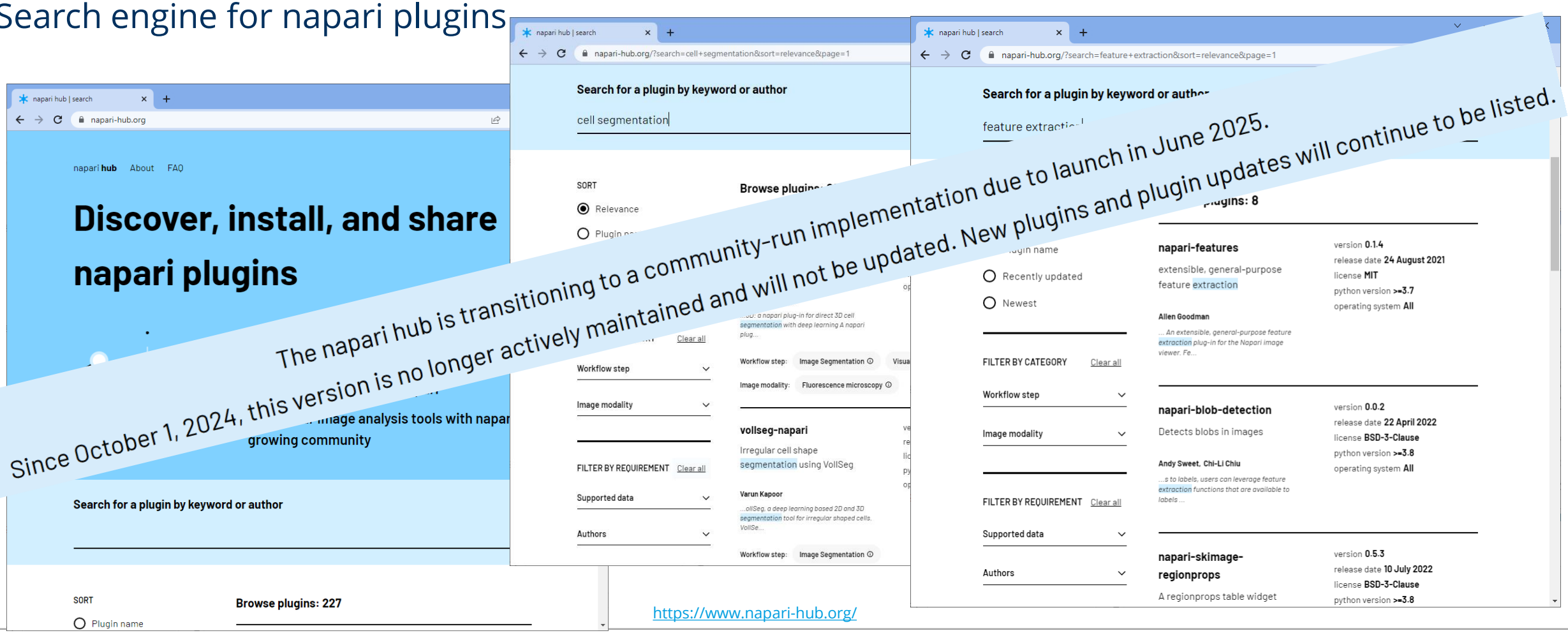
napari >= 0.5.0

Layers sub-menu items depend on plugin developers populating them

The Napari Hub

The plugin you are looking for may be near you!

Search engine for napari plugins



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

Inspecting plugin usage and maintenance - GitHub

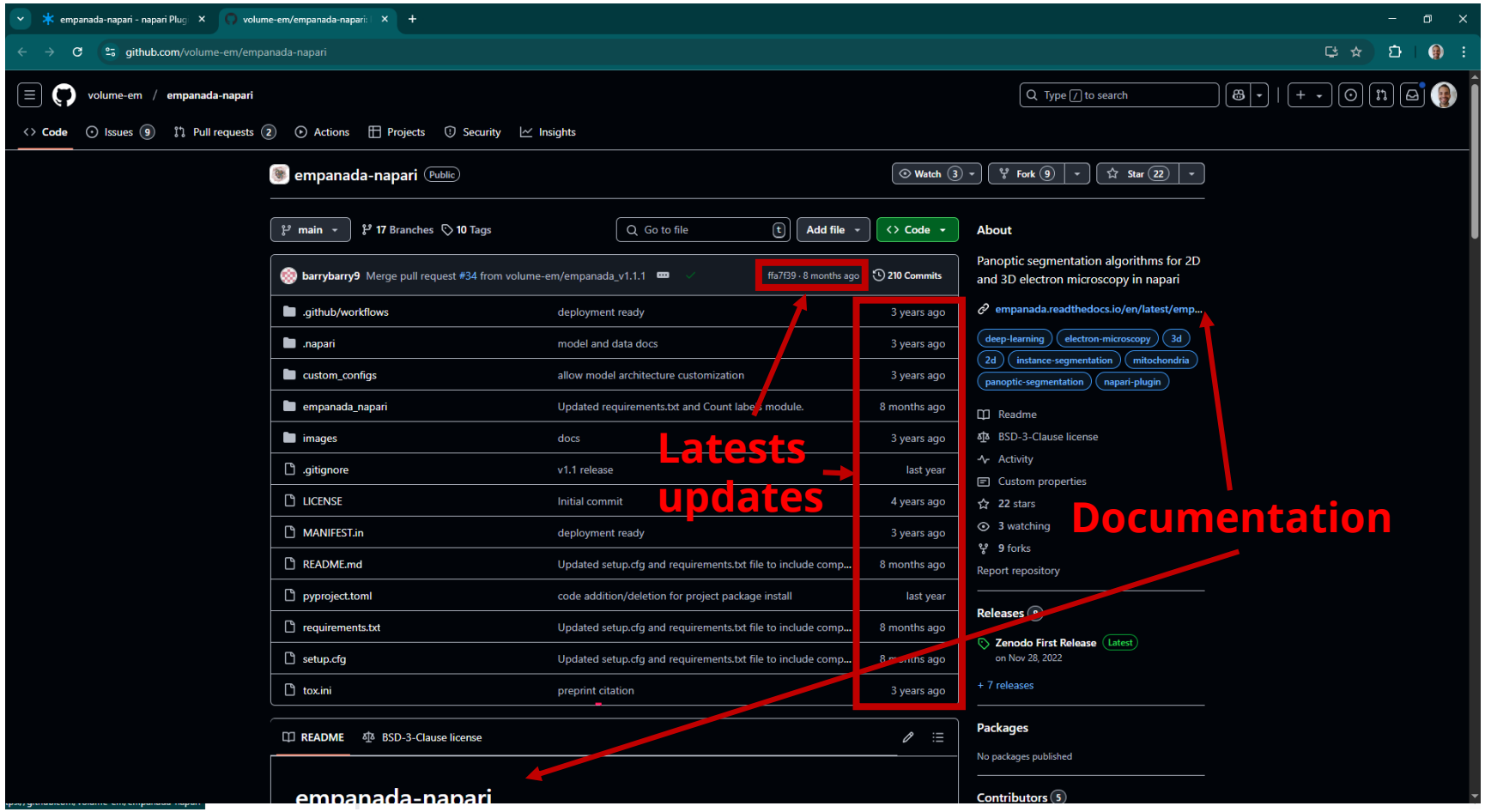


Image Data Management and Access: Napari – OMERO integration

- Demonstration: napari-OMERO plugin

Working with images in python

Open images

```
from skimage.io import imread

image = imread("blobs.tif")
```

```
image
```

```
array([[ 40,  32,  24, ..., 216, 200, 200],
       [ 56,  40,  24, ..., 232, 216, 216],
       [ 64,  48,  24, ..., 240, 232, 232],
       ...,
       [ 72,  80,  80, ...,  48,  48,  48],
       [ 80,  80,  80, ...,  48,  48,  48],
       [ 96,  88,  80, ...,  48,  48,  48]], dtype=uint8)
```

Images are *just* multi-dimensional arrays or “arrays of arrays”.

https://biapol.github.io/AMHCT_Bio_Image_Analysis_2025/01_images_are_numpy_arrays.html

Opening images should, but it is not always simple

- **Paths** need attention
  - Backslashes often cause problems on Windows

    OSError: [Errno 22] Invalid argument:

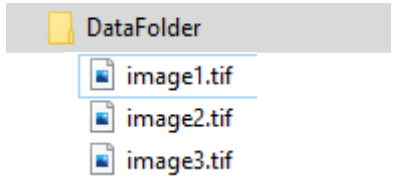
    Add a lowercase 'r' before path to fix that: `image = imread(r'C:\data\blobs.tif')`
- **Data may be too big** for your computer
  - They would need to be opened *lazily*
- **Proprietary file formats** ('.czi', '.lif', '.nd2', '.ims', etc) may need additional packages
  - In Fiji, Bioformats usually takes care of opening many different proprietary file formats
  - In Python/napari, you may need to install:
    - https://github.com/bioio-devs/bioio
    - https://github.com/AllenCellModeling/napari-aicsimageio
- **Metadata** may be lost
  - For example, some libraries return an image with dimensions like this

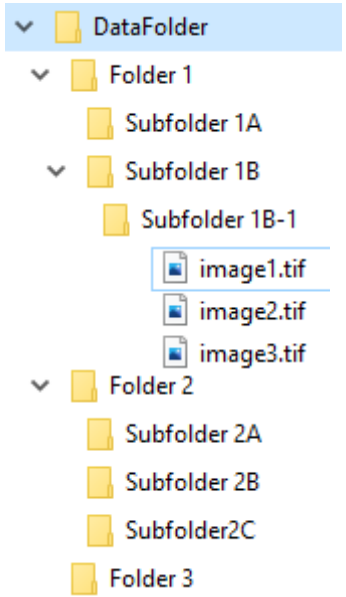
    What dimension is this? `(1, 1, 1, 80, 520, 692, 1)`

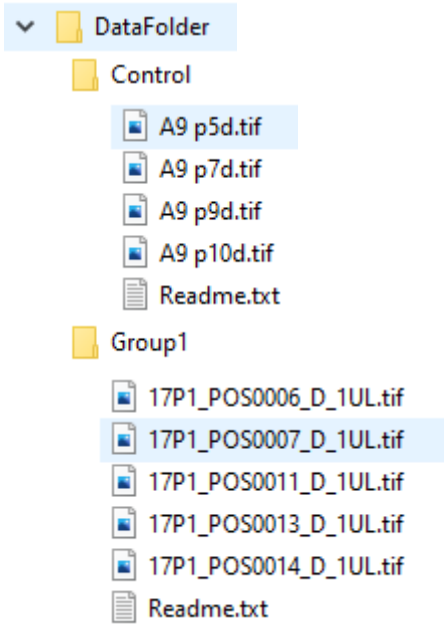
    ?

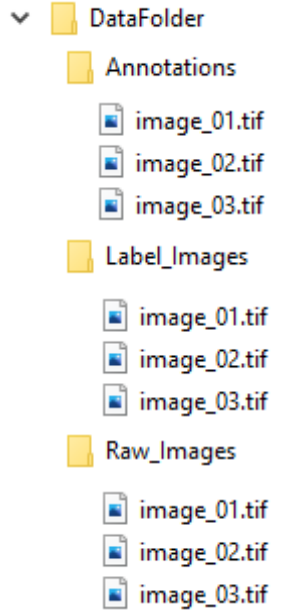
    Tile Z Time I don't know

Data Structure









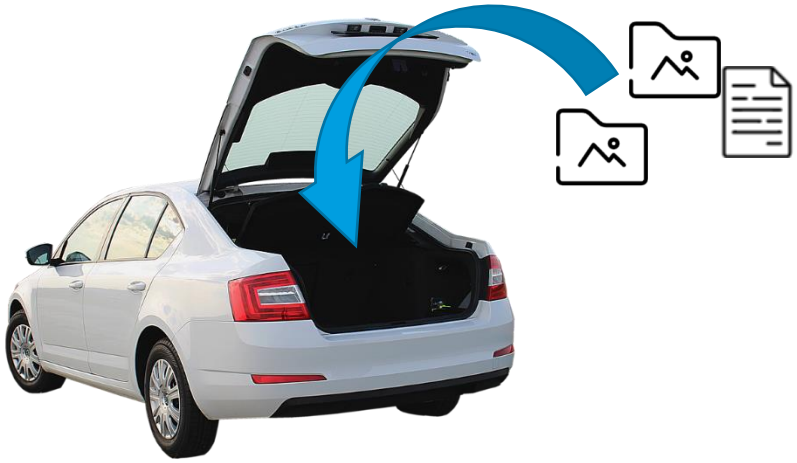


Image by eikira from Pixabay (https://pixabay.com/photos/trunk-automobile-vehicle-luggage-1478832/)

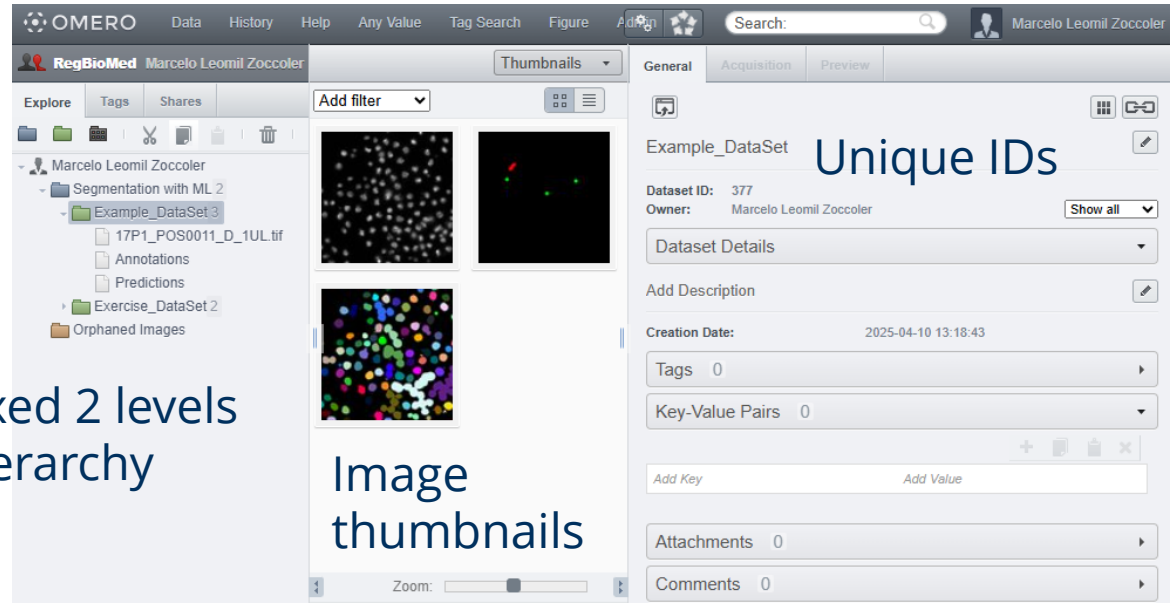
OMERO

OMERO is designed with scientists in mind

OMERO handles all your images using a secure central repository, from the microscope to publication.

Using the power of Bio-Formats, OMERO supports over 140 image file formats, including all major microscope formats.

Filter by tags and more...



Unique IDs

Fixed 2 levels hierarchy

Image thumbnails

Metadata

IMPORT DATA

Upload your data via the desktop app or command line, use our custom dropbox tool, or have your facility manager import it for you

ORGANIZE

View and organize your data from anywhere you have internet access

VIEW

Work with your images using the OMERO.insight desktop app, with OMERO.web in your browser, or from 3rd party software

ANALYZE

Use ImageJ, MATLAB, R, scripts and other tools to run analysis on data in OMERO and save results on the server

SHARE

Share data with collaborators or your PI using the Groups/Users permission system.

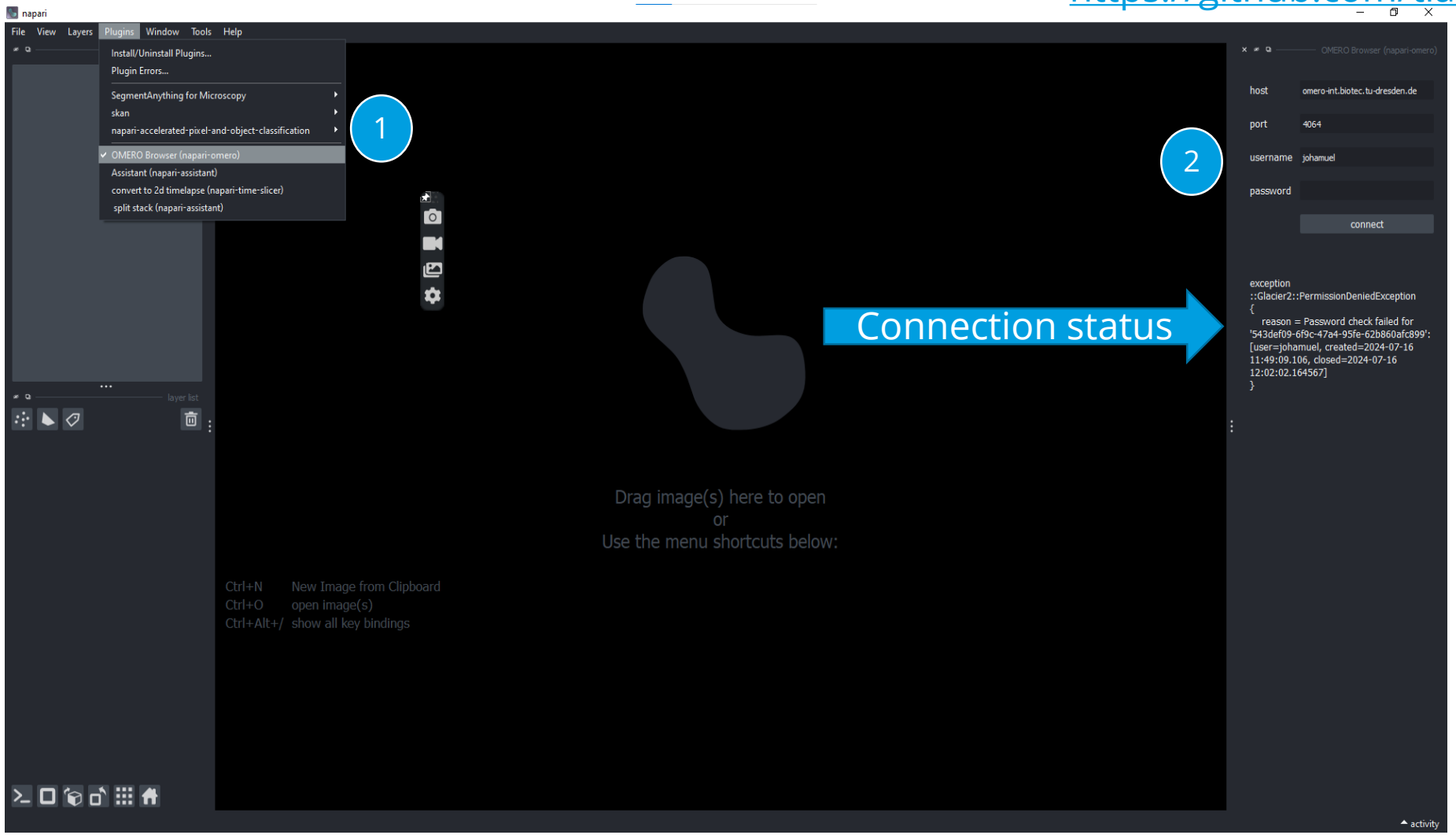
PUBLISH

Publish direct from the server via a customized website or embedded viewer

https://www.openmicroscopy.org/omero/scientists/

Napari-OMERO usage

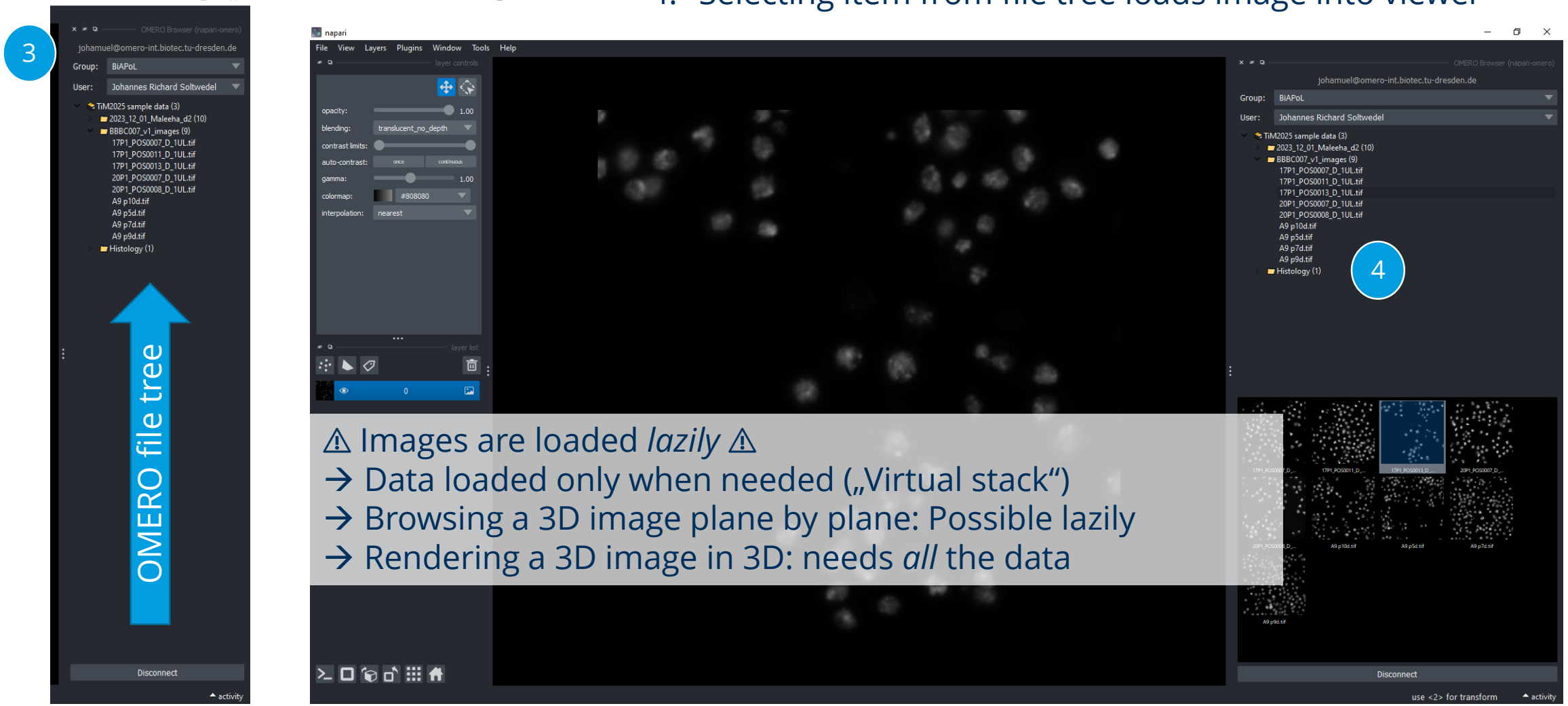
https://github.com/tlambert03/napari-omero



1. Select Plugin from plugin menu
2. Log into BiAPoL OMERO with ZIH credentials

Napari-OMERO usage

3. Select group and user
4. Selecting item from file tree loads image into viewer

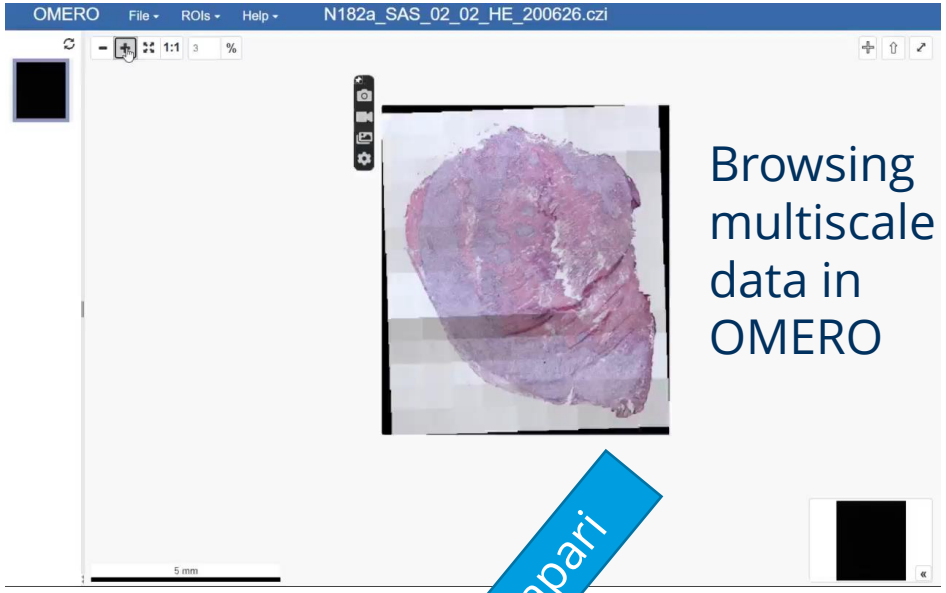


⚠ Images are loaded *lazily* ⚠
→ Data loaded only when needed („Virtual stack“)
→ Browsing a 3D image plane by plane: Possible lazily
→ Rendering a 3D image in 3D: needs *all* the data

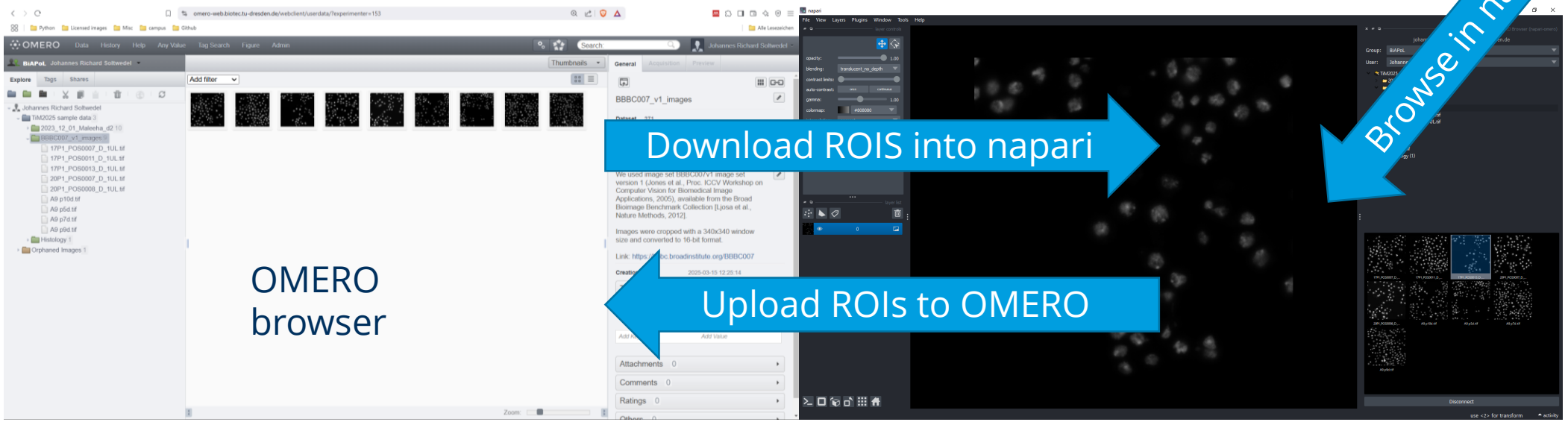
Some tips and tricks for OMERO from Phython: https://biapol.github.io/omero-tools

Napari-OMERO usage

- ROI Upload
- Multiscale support: Browse larger-than-RAM samples (*3D limitation remains*)
- **Coming soon:**
- ROI download



Browsing multiscale data in OMERO



OMERO browser

Download ROIS into napari

Upload ROIs to OMERO

Browse in napari

TECHNISCHE UNIVERSITÄT DRESDEN

BioDIP Biopolis Dresden Imaging Platform

PoL Physics of Life TU Dresden

Segmentation and Supervised Machine Learning

Random Forest Classifiers

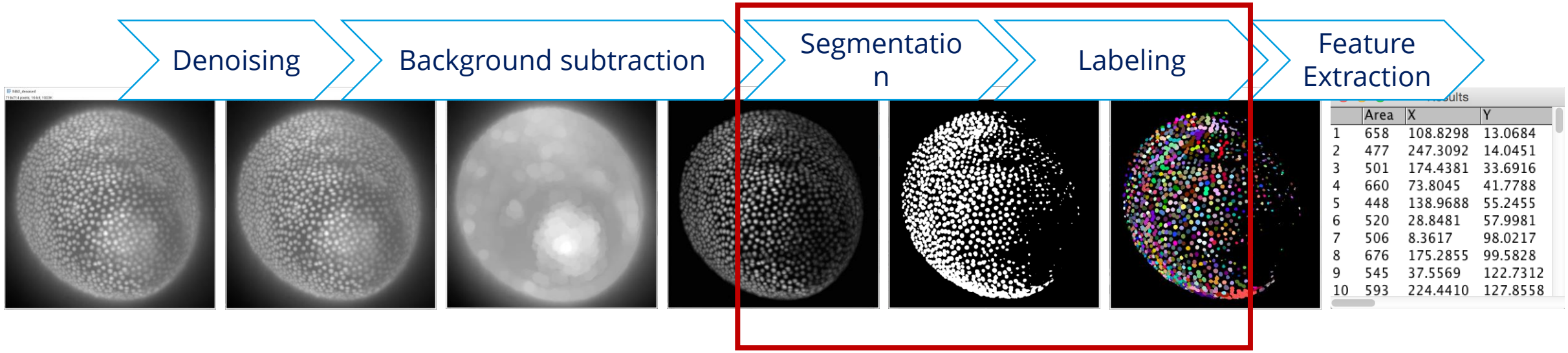
- Pixel Classifier

DRESDEN concept

Image Analysis Classic Workflows

Image analysis workflow is a series of processes/functions (not always linear) applied to images to achieve a certain goal (usually some measurements)

Here is a classic example:



Application: Segmentation

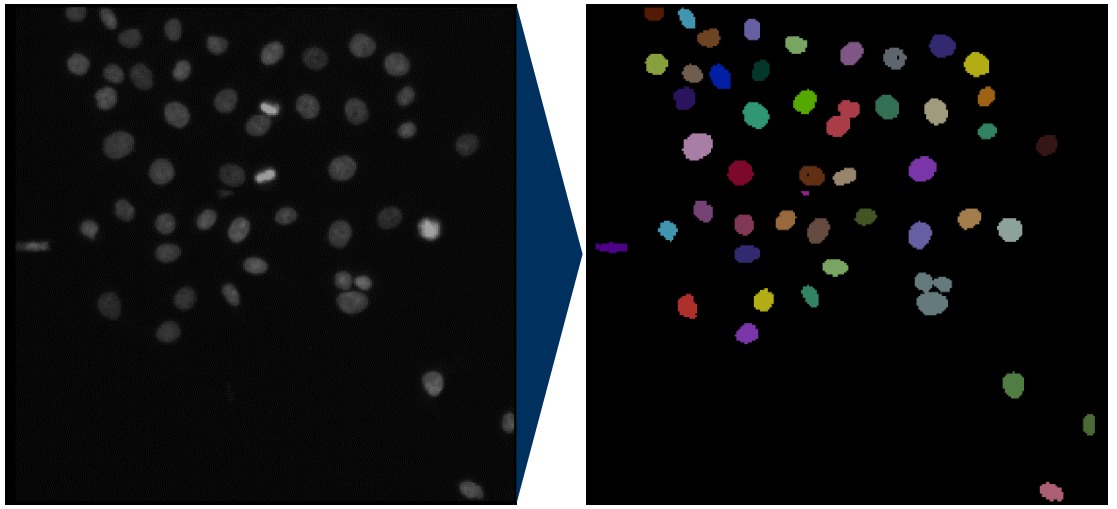
https://scikit-image.org/docs/stable/api/skimage.data.html

Aim:

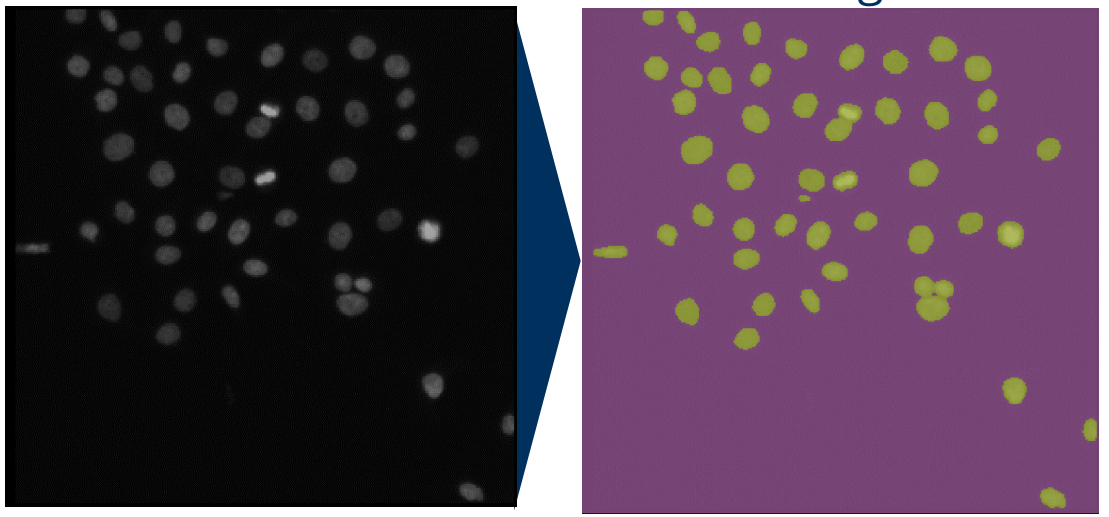
Separate background from foreground

Vocabulary:

- **Segmentation:**
  → Assigning a meaningful *label* to each pixel
  → Segmentation is a *classification* problem
- **Semantic segmentation:**
  Differentiate pixels into multiple *classes* (e.g., membrane, nucleus, cytosol, etc.)
- **Instance segmentation:**
  Differentiate multiple occurrences of the same class into separate instances of this class (e.g., separate *label* for each cell in image)



Instance segmentation



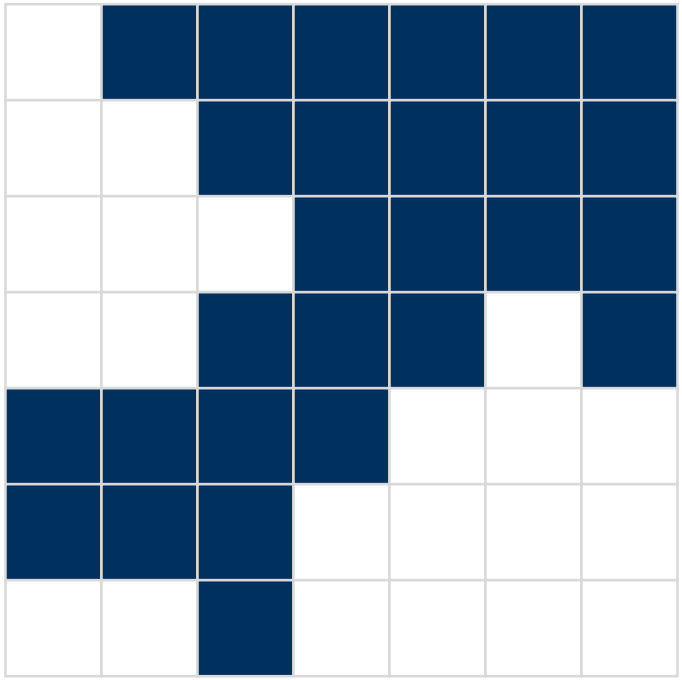
Semantic segmentation

Instance segmentation

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

Image segmentation using thresholding

Finding the right workflow towards a good segmentation takes time

A priori, we usually don't know which information in the image is useful for a good segmentation

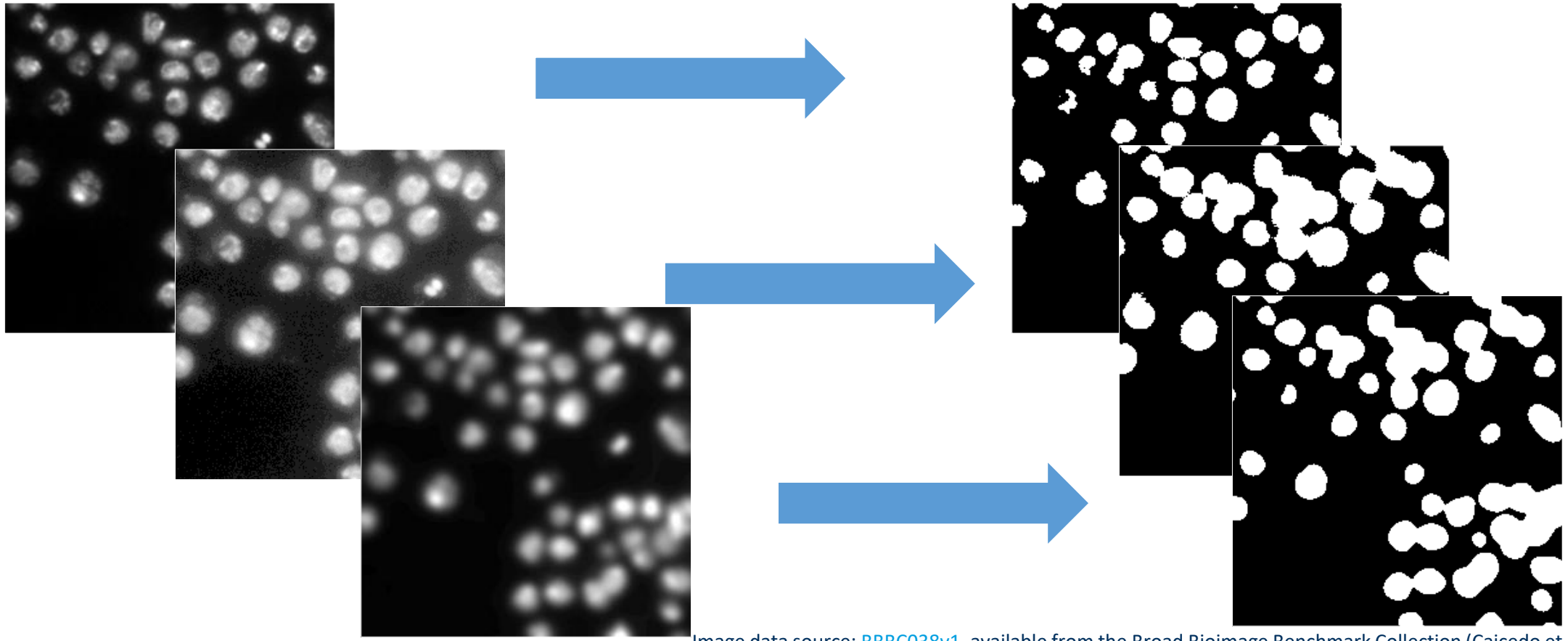
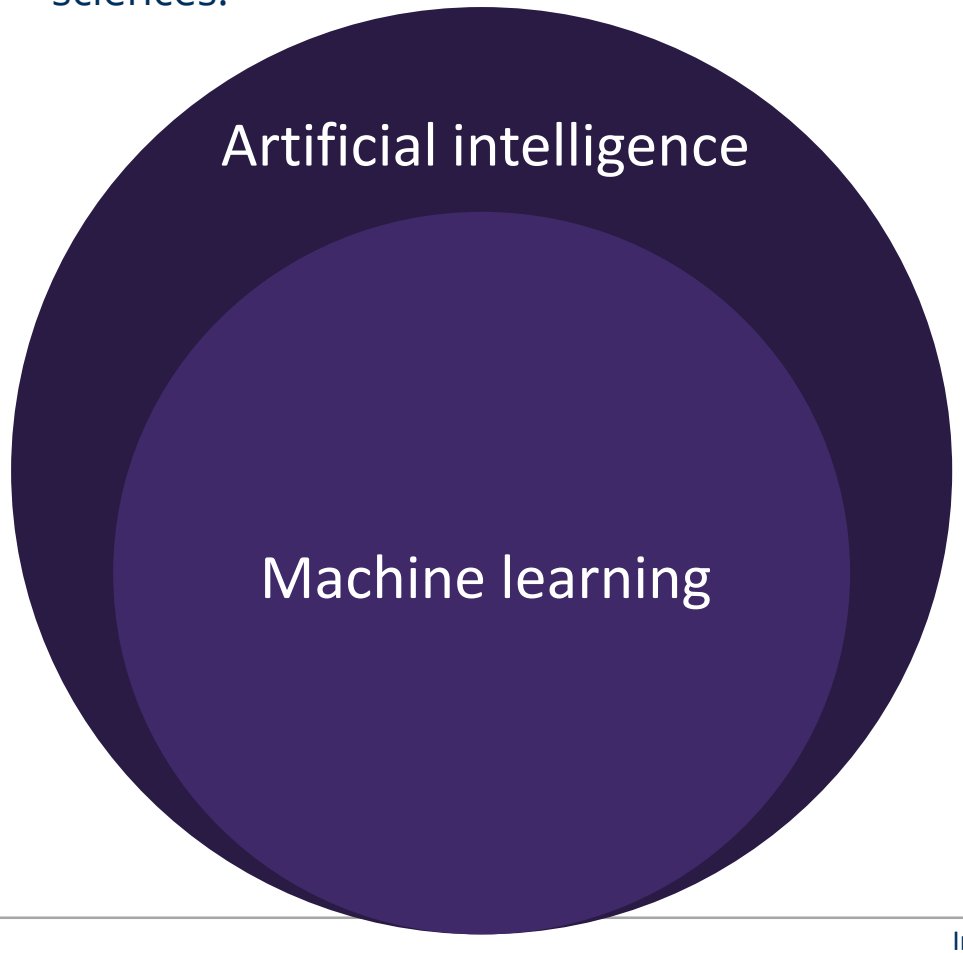
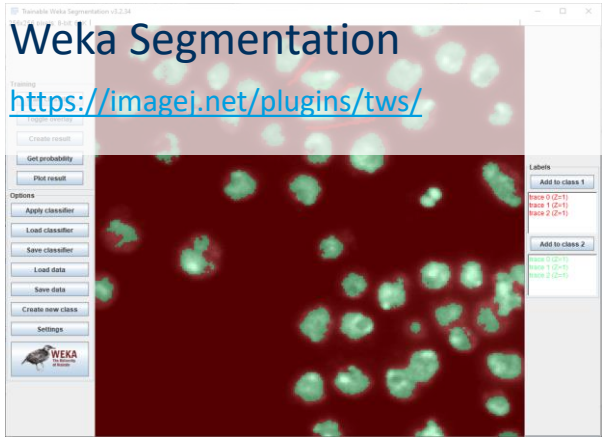


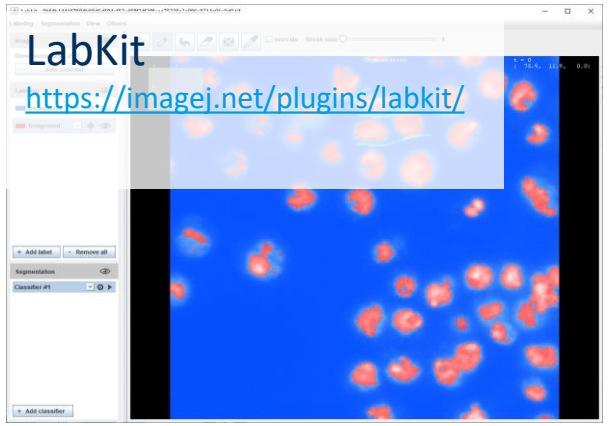
Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

Machine learning

- A research field in computer science
- Finds more and more applications, also in life sciences.







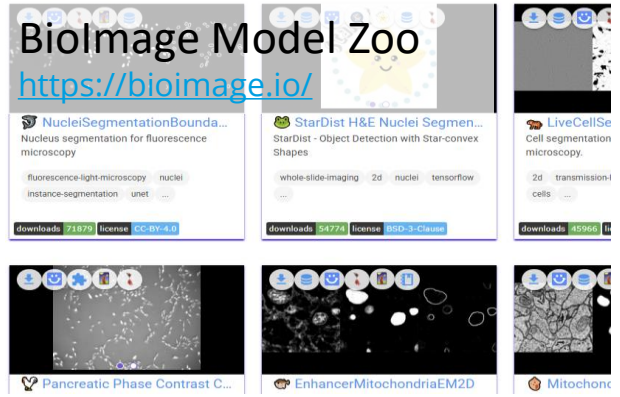
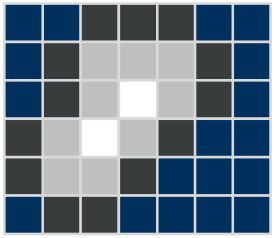


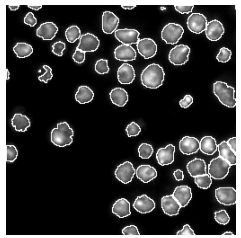
Image data source: BBBC038v1, available from the Broad Bioimage Benchmark Collection (Caicedo et al., Nature Methods, 2019)

Machine learning

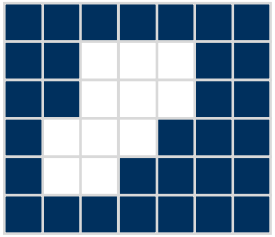
Automatic construction of predictive models from given data

Pixels, Objects, Images, Audio, Text, Measurements, …





Dense Segmentation / Binarization



Object classification

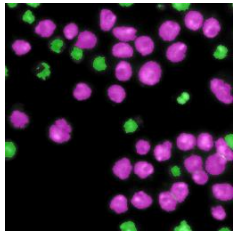
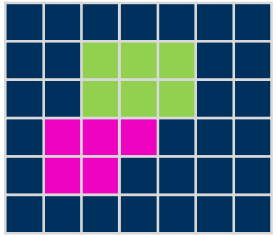


Image classification

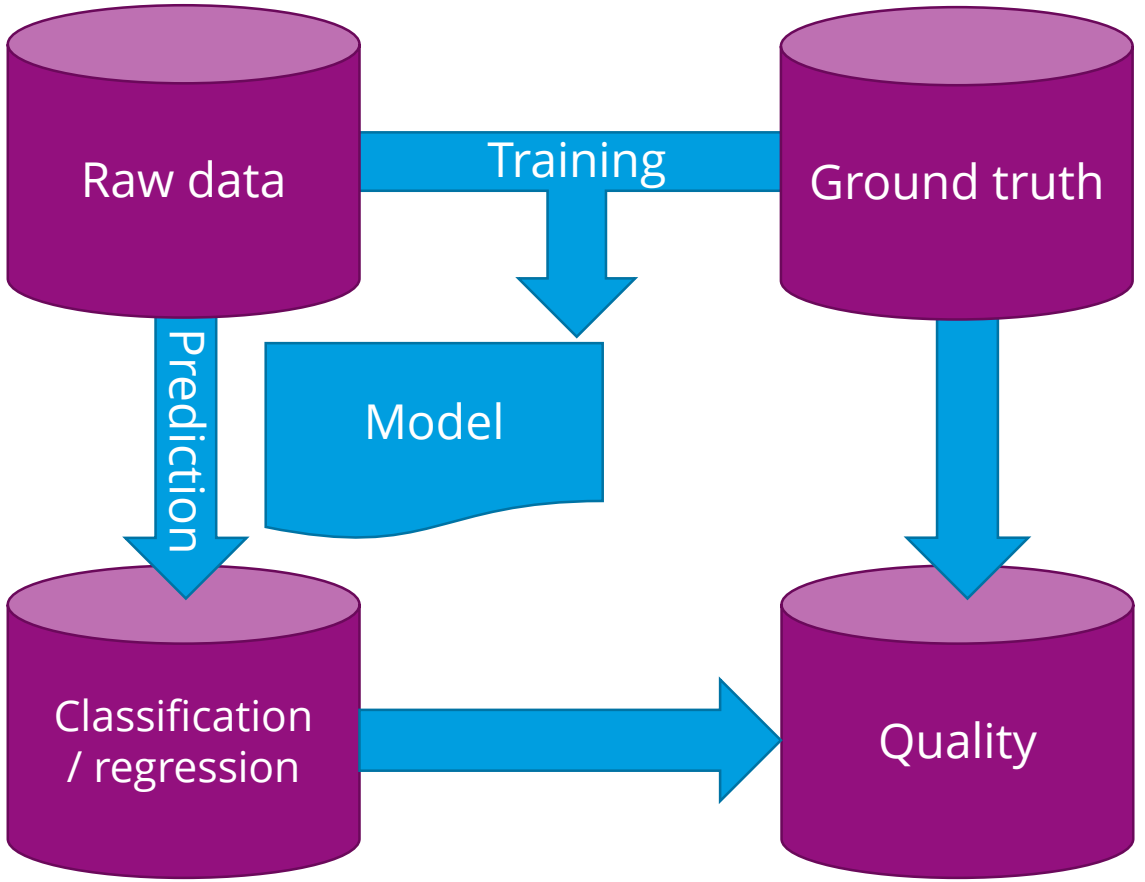
"Cat"

Instance segmentation



Cont. quantity

P_{Cat}= 0.5
$P_{Microscope}$= 0.4
Height = 80 cm



Annotated raw data, usually generated by humans

Precision, Recall

Segmentation: Latest developments

1970s-2010: Filtering, thresholding

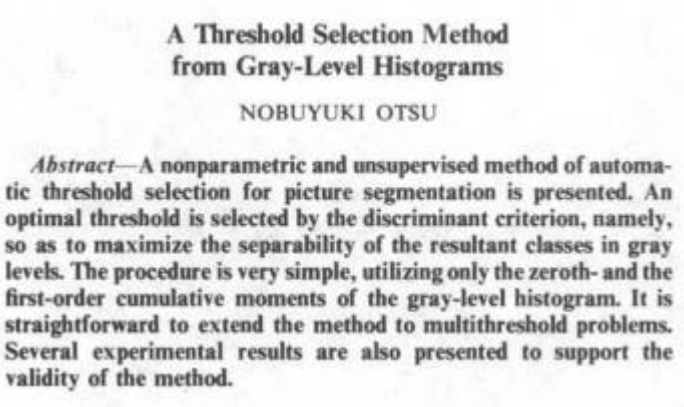
2010s: Random forests et al.

Deep learning

2015: UNet
2018: Stardist

Foundational models

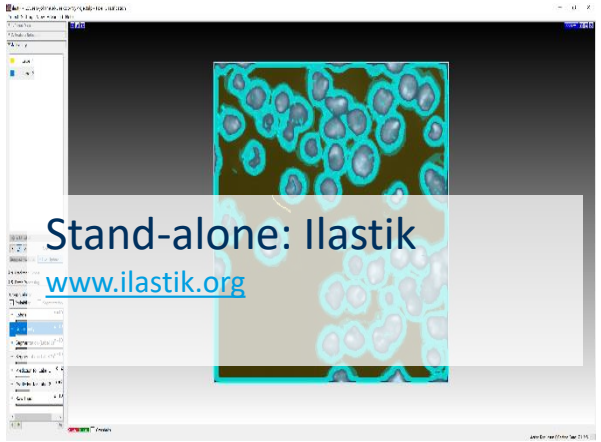
2023: Segment anything (SAM)
2024: SAM 2



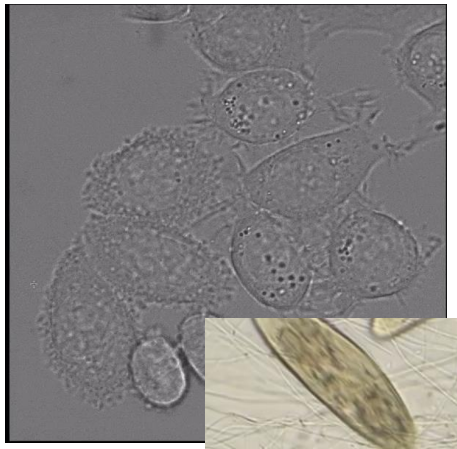
A Threshold Selection Method from Gray-Level Histograms

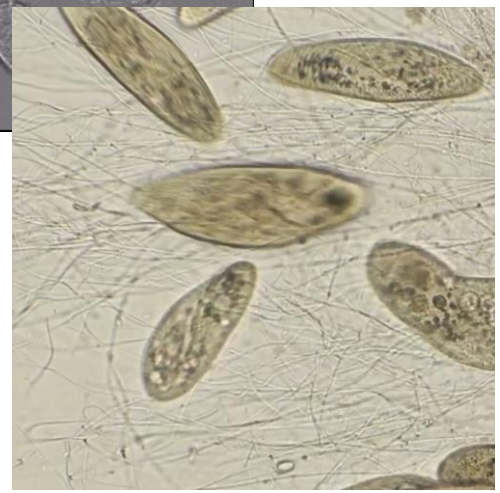
NOBUYUKI OTSU

Abstract—A nonparametric and unsupervised method of automatic threshold selection for picture segmentation is presented. An optimal threshold is selected by the discriminant criterion, namely, so as to maximize the separability of the resultant classes in gray levels. The procedure is very simple, utilizing only the zeroth- and the first-order cumulative moments of the gray-level histogram. It is straightforward to extend the method to multithreshold problems. Several experimental results are also presented to support the validity of the method.



Stand-alone: Ilastik
www.ilastik.org





Computational demand

TECHNISCHE UNIVERSITÄT DRESDEN

BioDIP Biopolis Dresden Imaging Platform

PoL Physics of Life TU Dresden

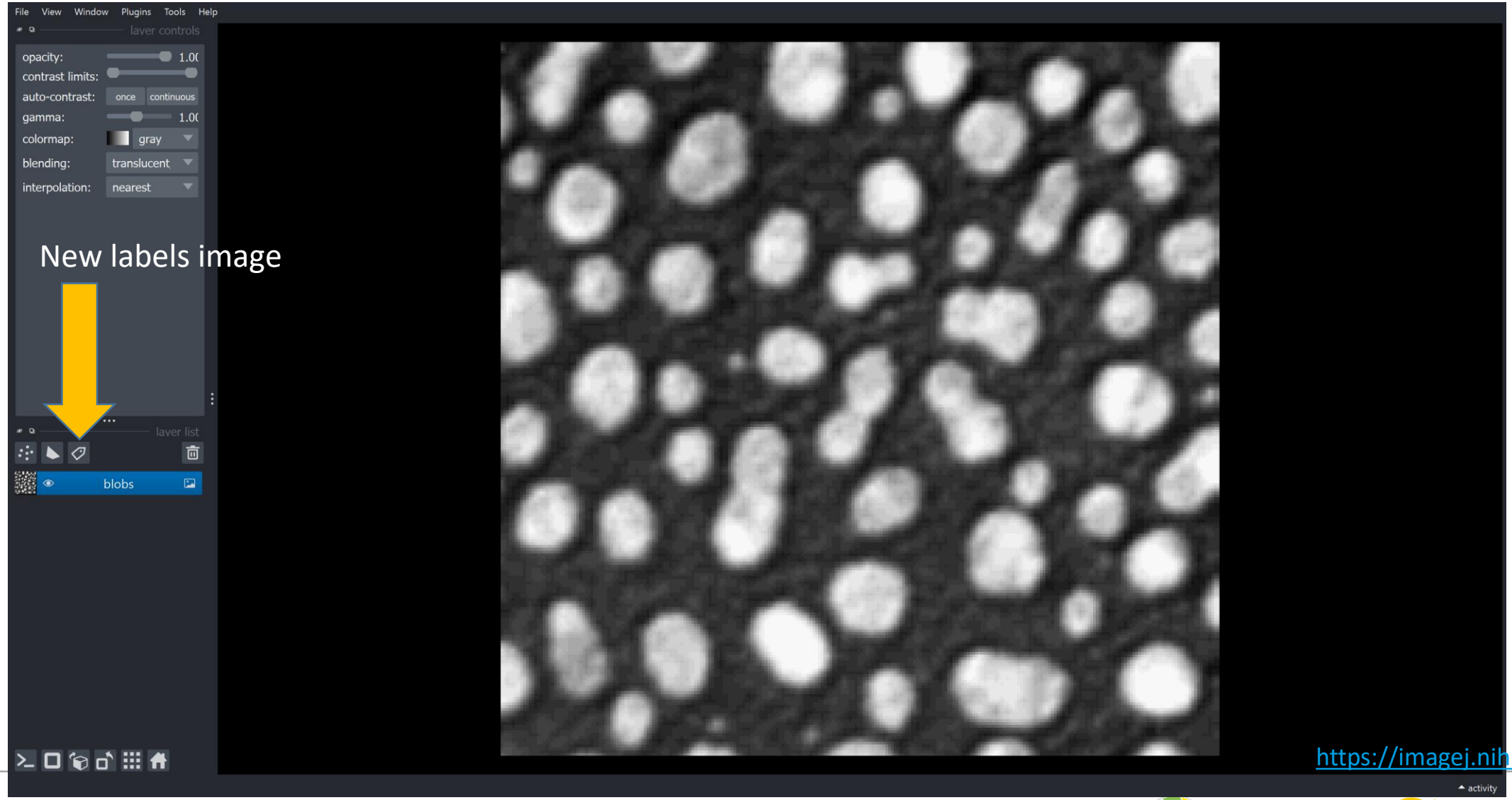
Segmentation and Supervised Machine Learning in napari

Random Forest Classifiers

- Pixel Classifier
- **napari-apoc plugin**

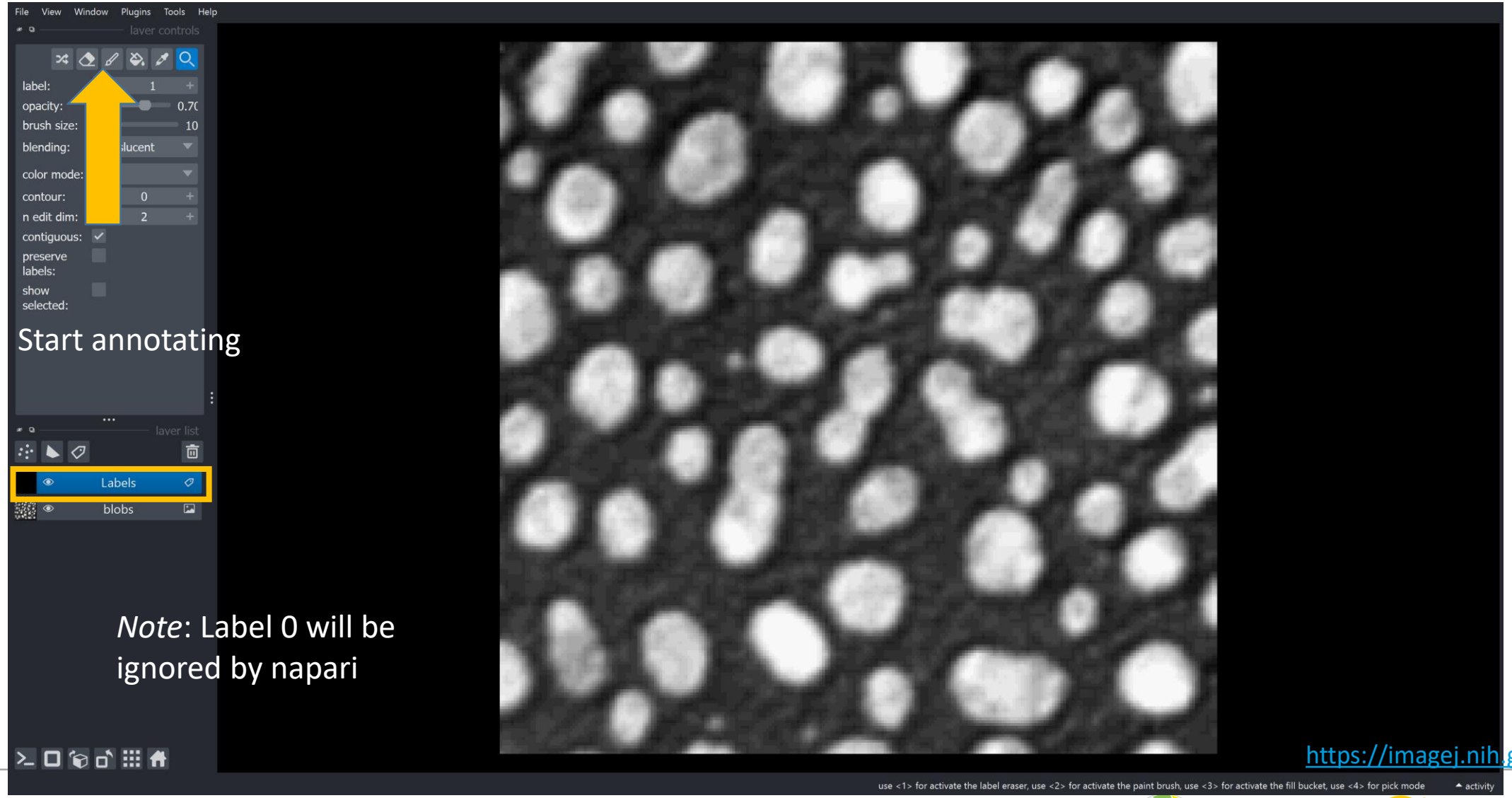
DRESDEN concept

In napari: annotation



https://imagej.nih.gov/ij/images/

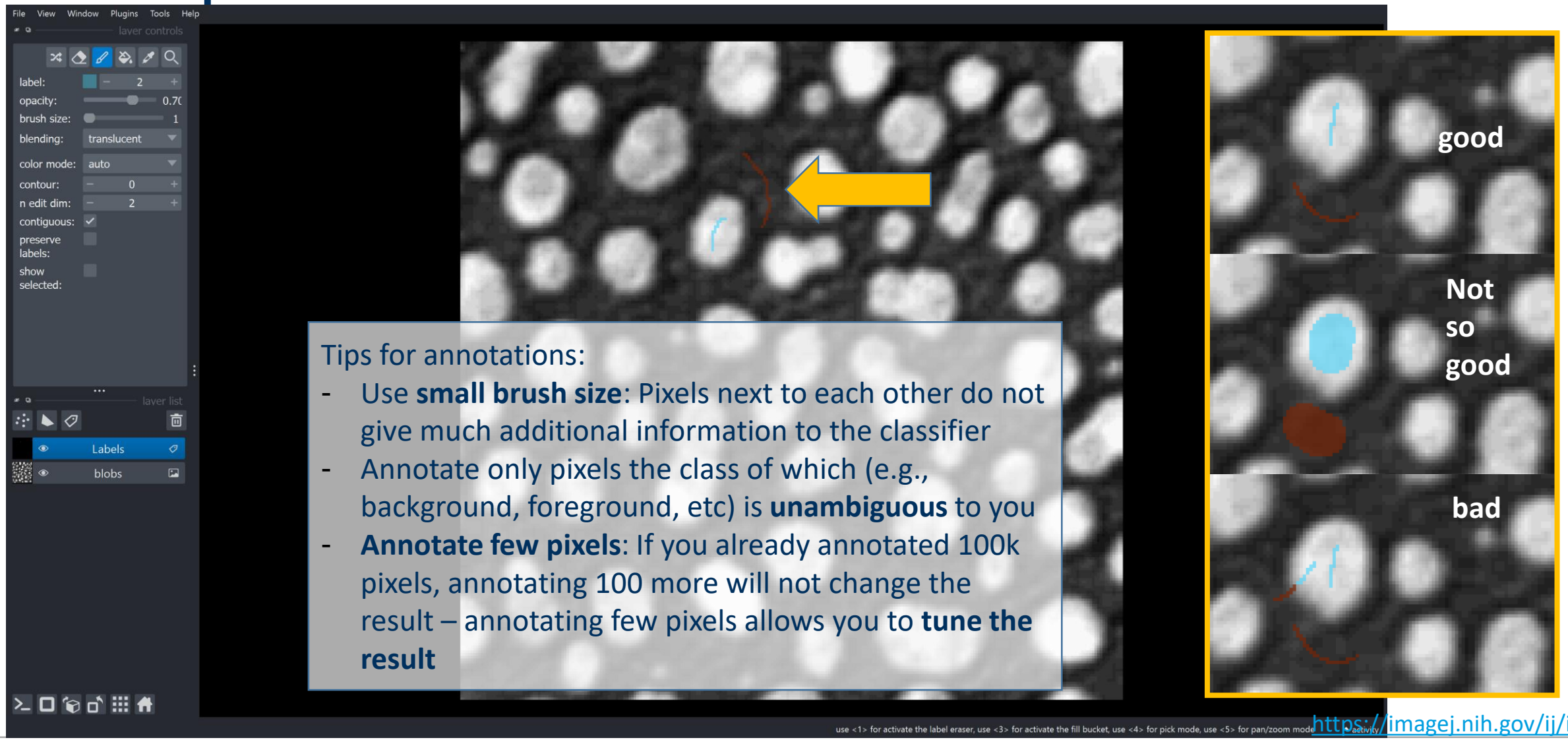
In napari: annotation



Note: Label 0 will be ignored by napari

https://imagej.nih.gov/ij/images/

In napari: annotation



Tips for annotations:

- Use **small brush size**: Pixels next to each other do not give much additional information to the classifier
- Annotate only pixels the class of which (e.g., background, foreground, etc) is **unambiguous** to you
- **Annotate few pixels**: If you already annotated 100k pixels, annotating 100 more will not change the result – annotating few pixels allows you to **tune the result**

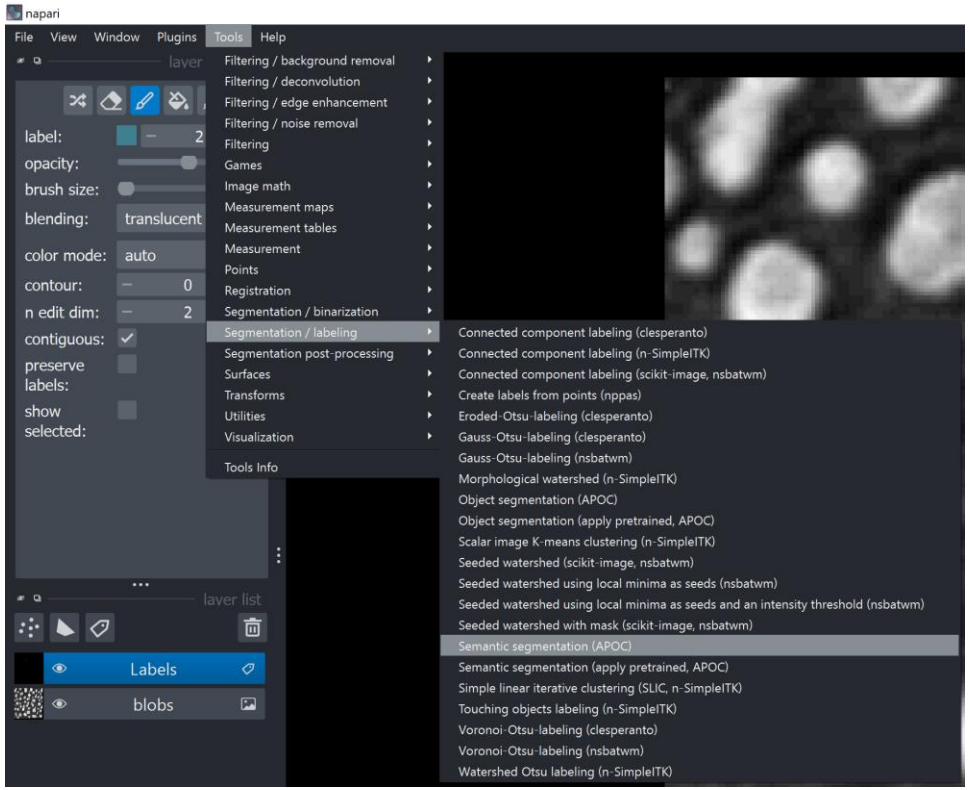
https://imagej.nih.gov/ij/images/

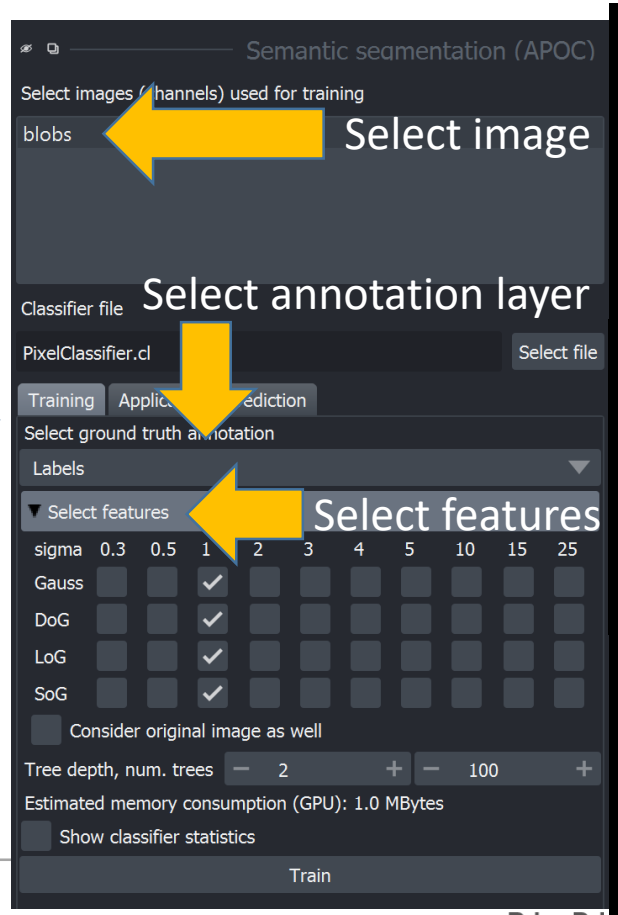
In napari: Semantic segmentation (training)

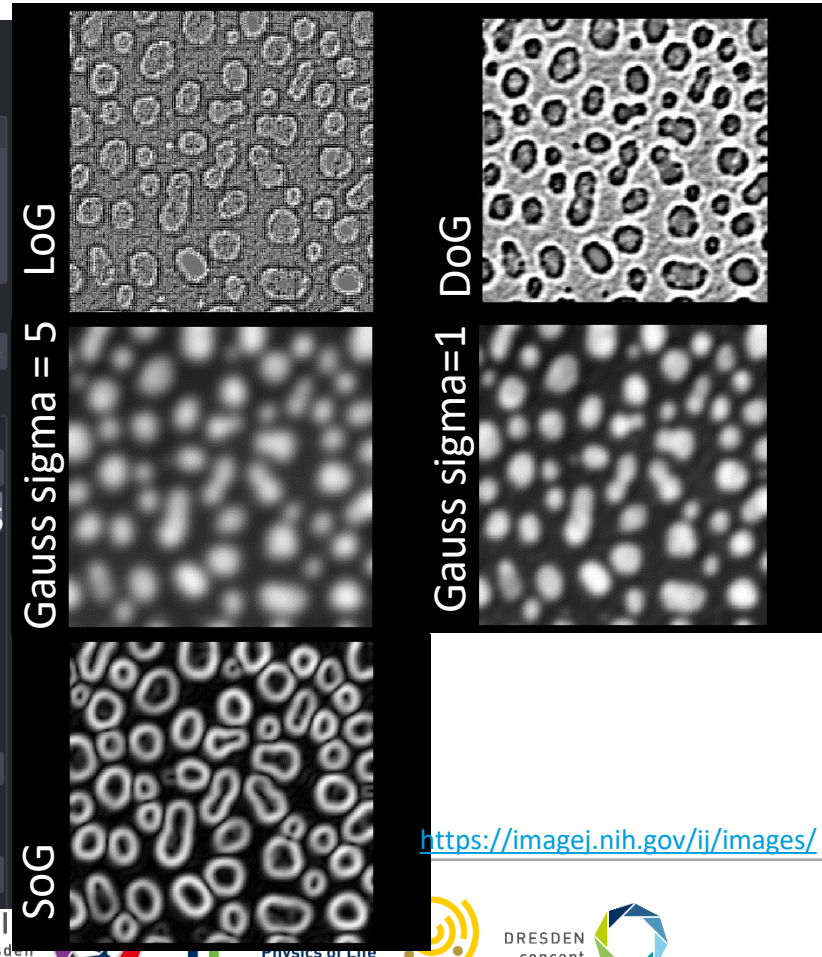
Two options:

Semantic segmentation: Predict class of every pixel according to annotation

Object segmentation: Assumes that class “1” refers to background – applies connected component analysis to other class

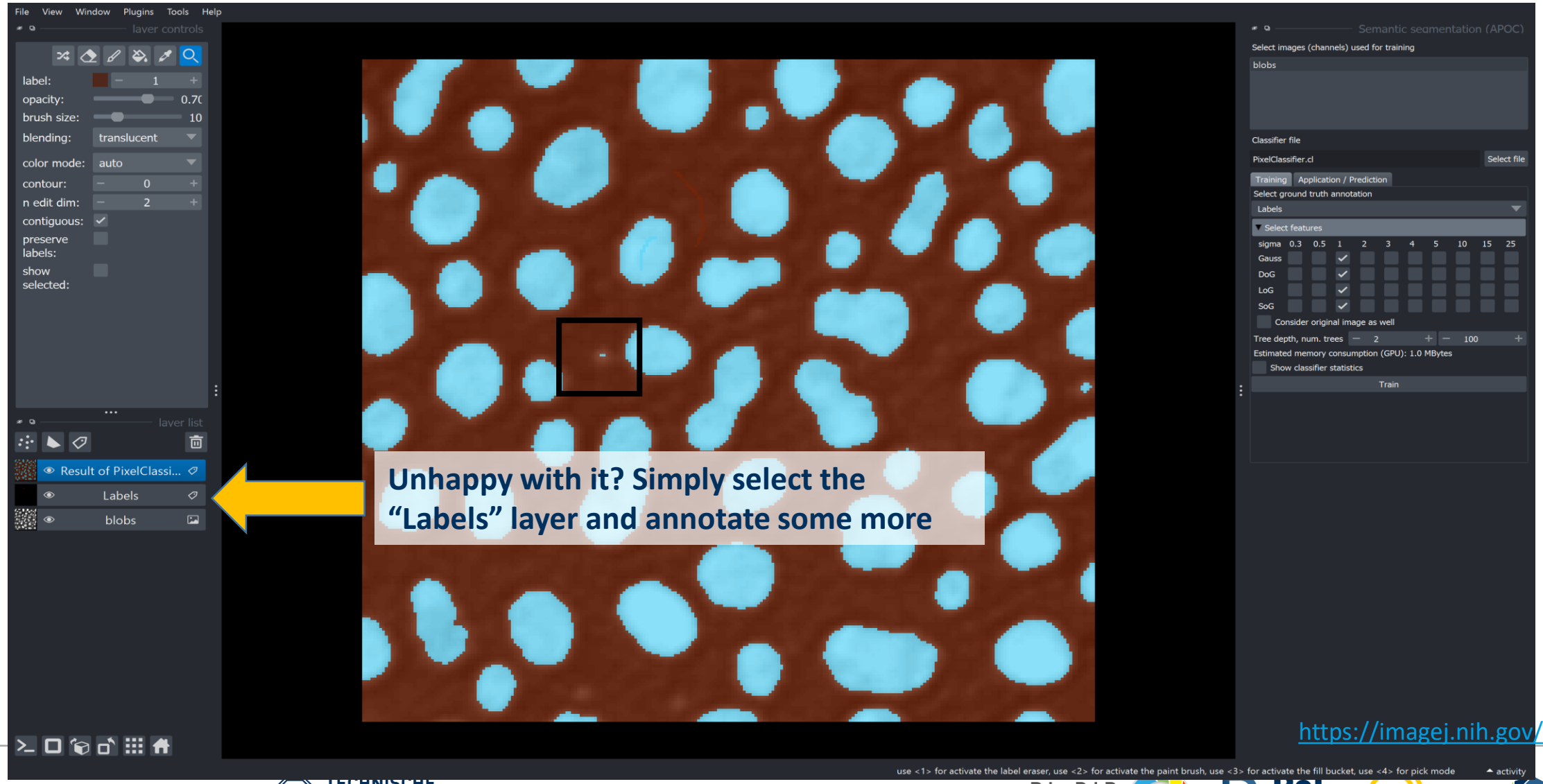






https://imagej.nih.gov/ij/images/

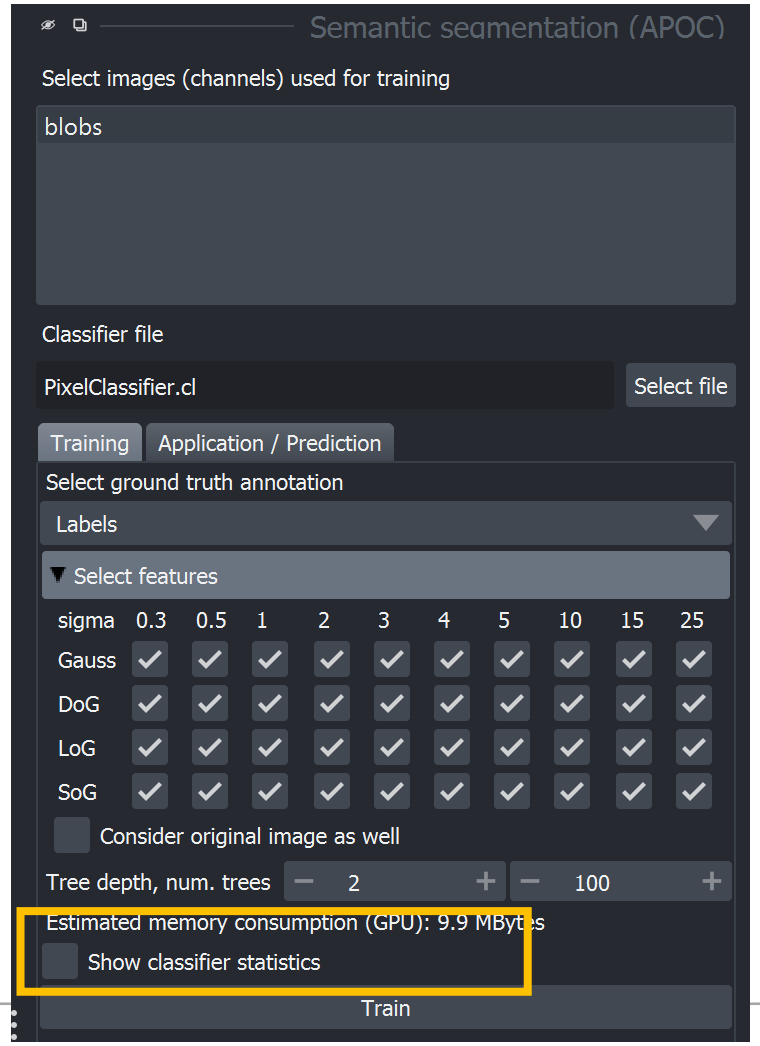
In napari: Semantic segmentation (training)



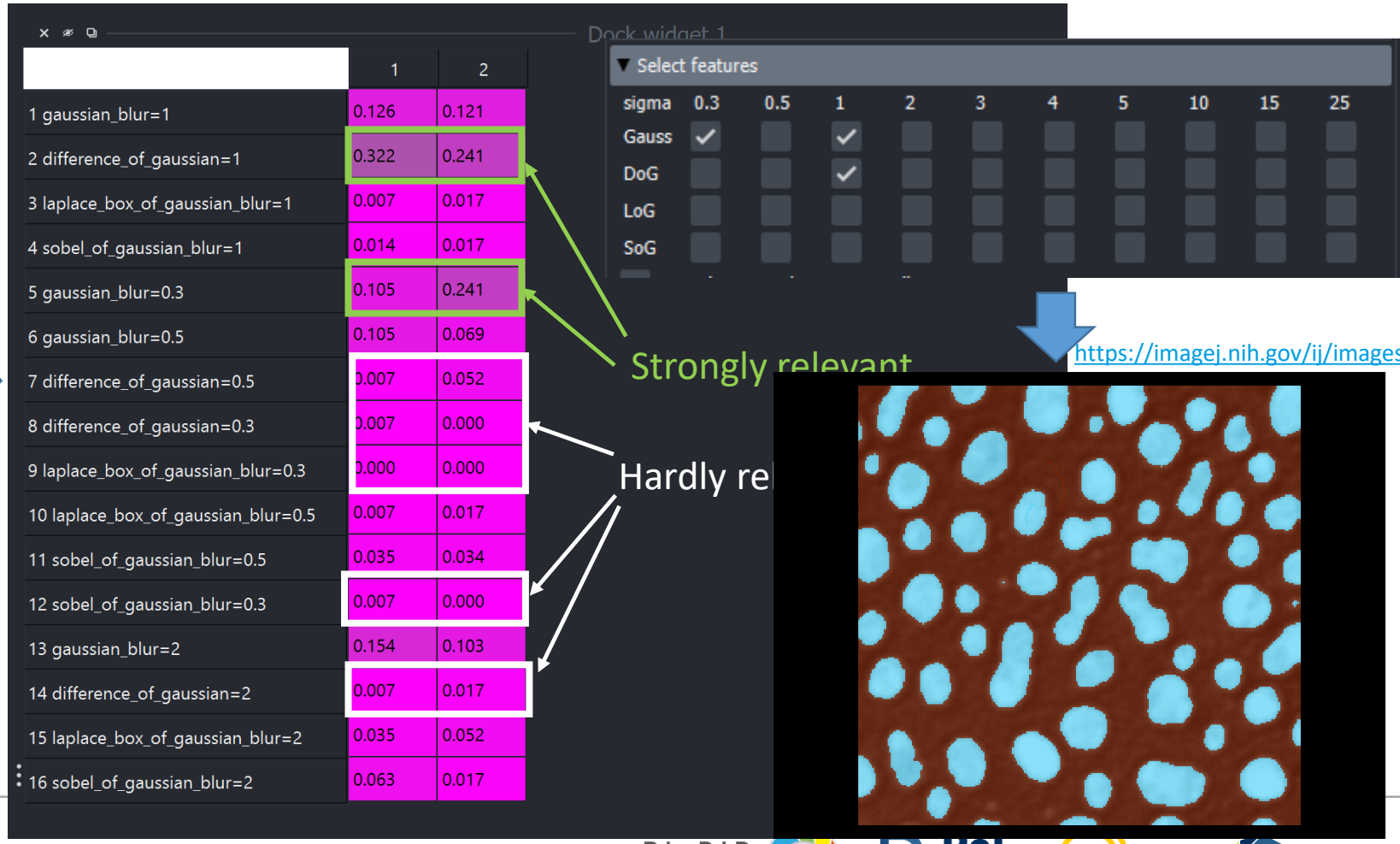
https://imagej.nih.gov/ij/images/

Semantic segmentation: Choosing the right features

Why not just do this?



- Not all features are equally relevant!
- Calculating the features takes time and computation resources!



https://imagej.nih.gov/ij/images/

TECHNISCHE UNIVERSITÄT DRESDEN

BioDIP Biopolis Dresden Imaging Platform

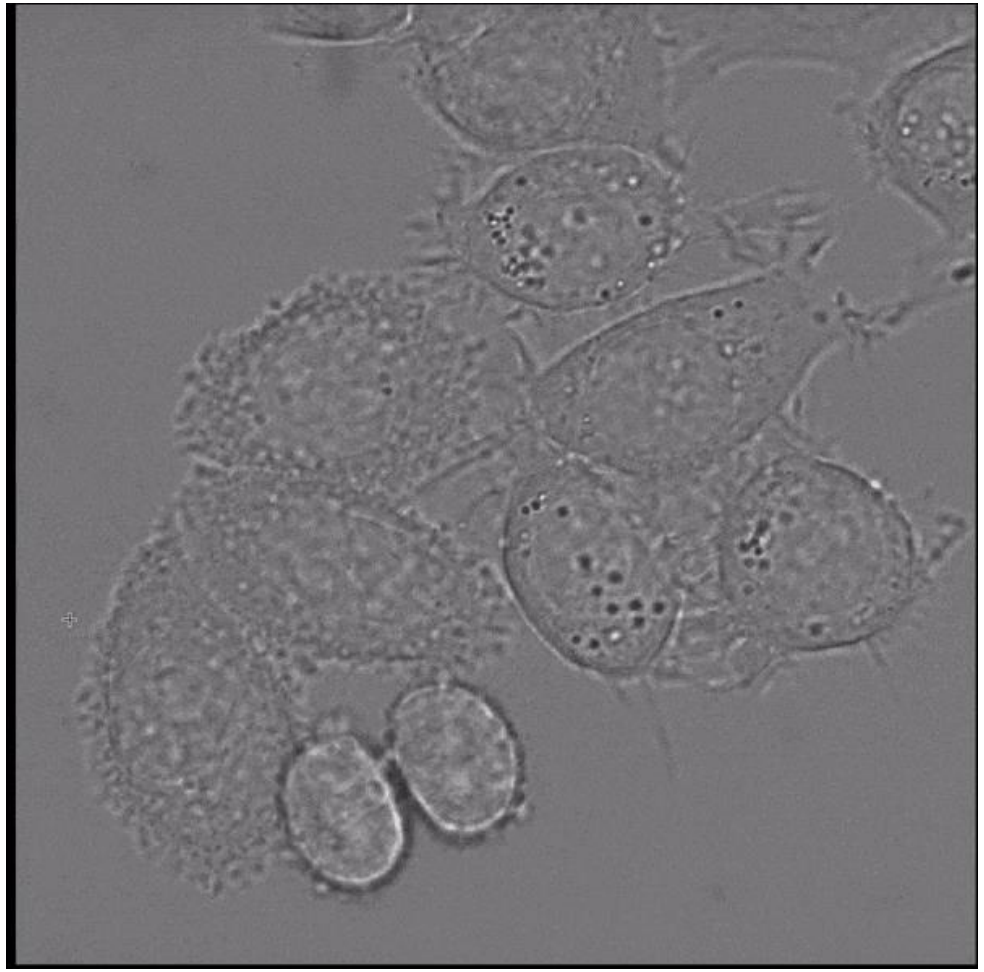
PoL Physics of Life TU Dresden

Segmentation and Supervised Machine Learning in napari

- Demonstration: Micro-sam plugin

DRESDEN concept

Micro-sam



https://github.com/computational-cell-analytics/micro-sam

Feature Extraction

- napari-skimage-regionprops

Feature extraction

- A feature is a countable or measurable property of an image or object.
- Goal of feature extraction is finding a minimal set of features to describe an object well enough to differentiate it from other objects.

- **Intensity based**
  - Mean intensity
  - Standard deviation
  - Total intensity
  - Textures
- **Shape based /spatial**
  - Area / Volume
  - Roundness
  - Solidity
  - Circularity / Sphericity
  - Elongation
  - Centroid
  - Bounding box
- **Spatio-temporal**
  - Displacement,
  - Speed,
  - Acceleration
- **Others**
  - Overlap
  - Colocalization
  - Neighborhood
- **Mixed features**
  - Center of mass
  - Local minima / maxima

Further reading:
https://focalplane.biologists.com/2023/05/03/feature-extraction-in-napari/

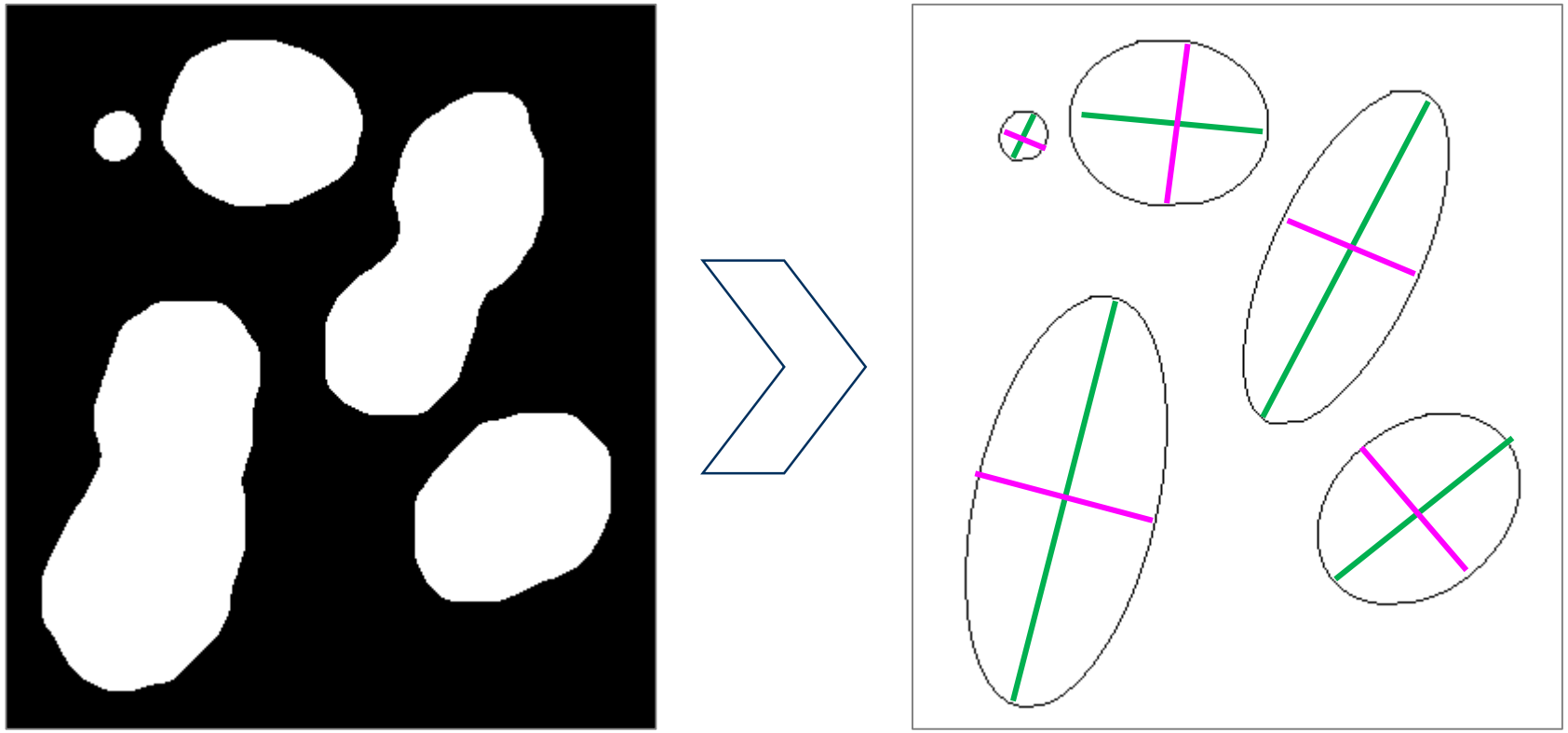
Fit ellipse

For every object, find the optimal ellipse simplifying the object.

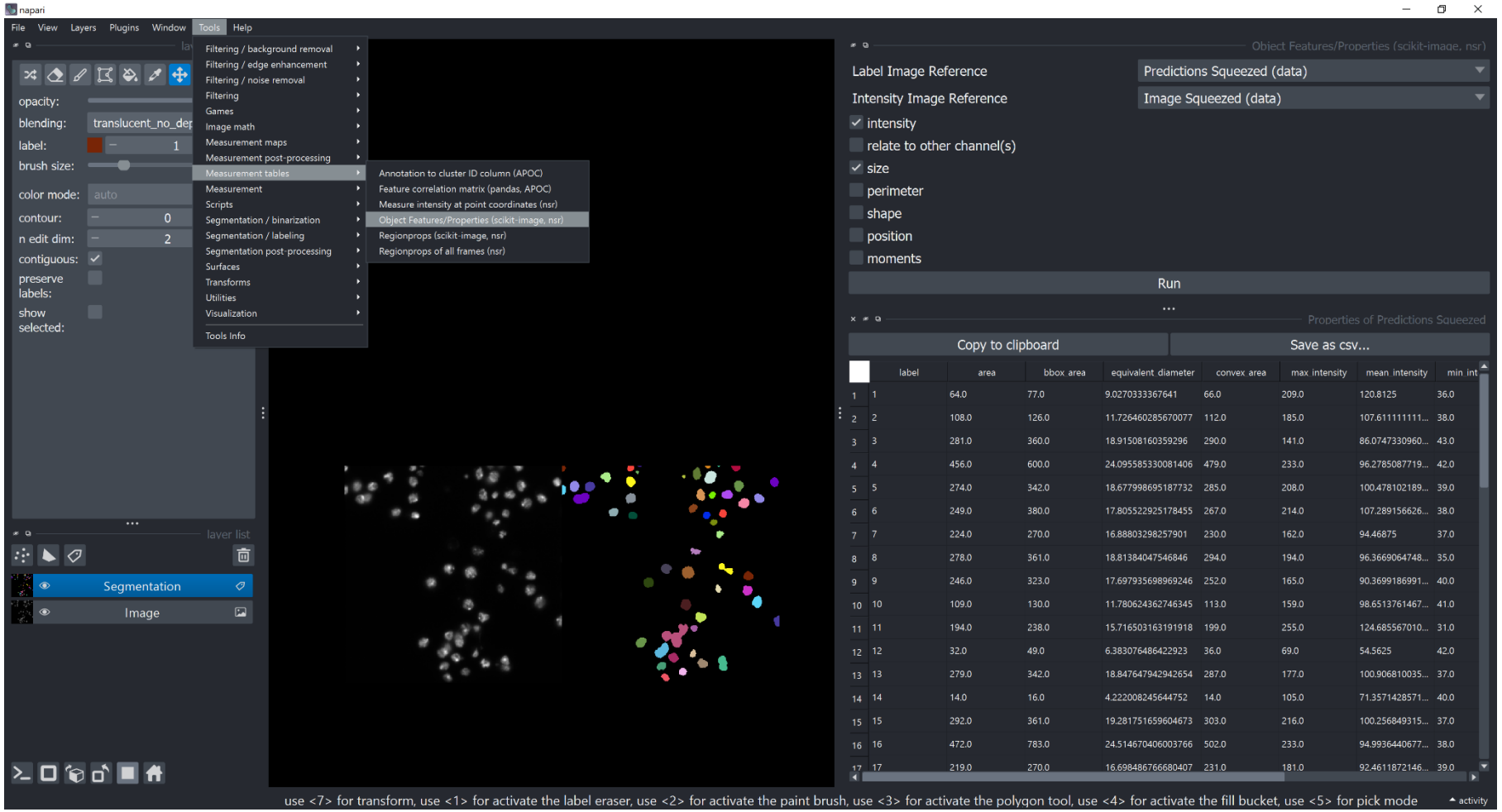
Major axis … long diameter

Minor axis … short diameter

Major and minor axis are perpendicular to each other



napari-skimage-regionprops



https://github.com/haesleinhuepf/napari-skimage-regionprops

TECHNISCHE UNIVERSITÄT DRESDEN

BioDIP Biopolis Dresden Imaging Platform

PoL Physics of Life TU Dresden

Object Classification and Supervised Machine Learning in napari

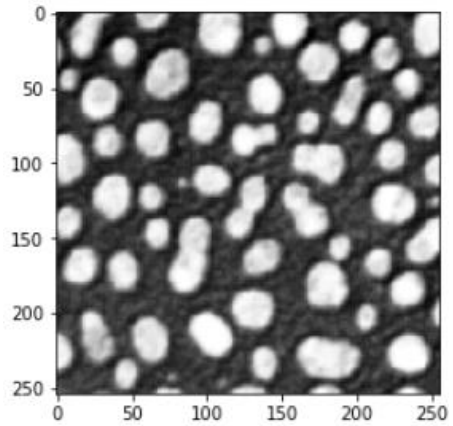
Random Forest Classifiers

- Object Classifier
- **napari-apoc plugin**

DRESDEN concept

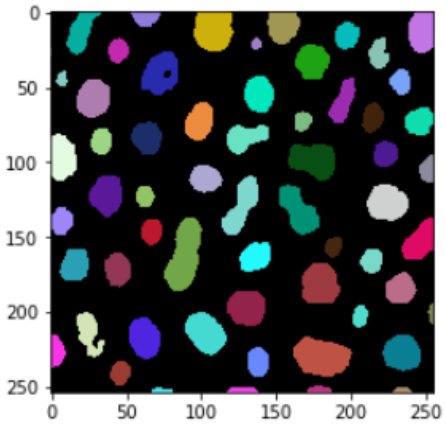
Object classification

- Object classification is a task that can be applied to an existing instance segmentation in order to identify a particular group of objects



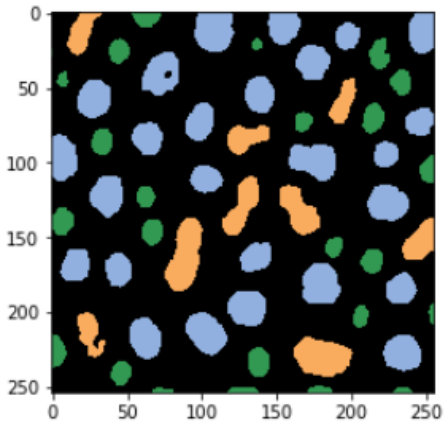
Raw image

Object segmentation

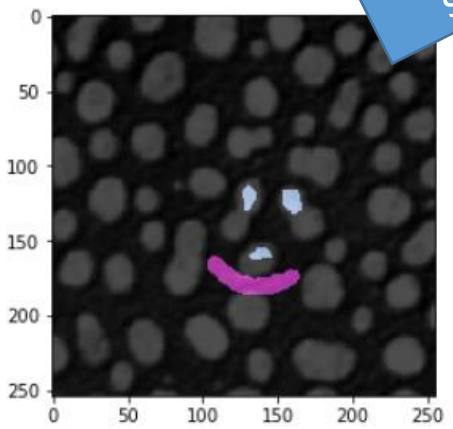


Object label image

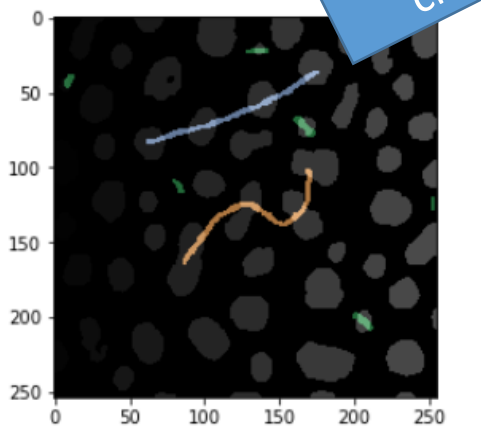
Object classification



Class label image



Pixel annotation



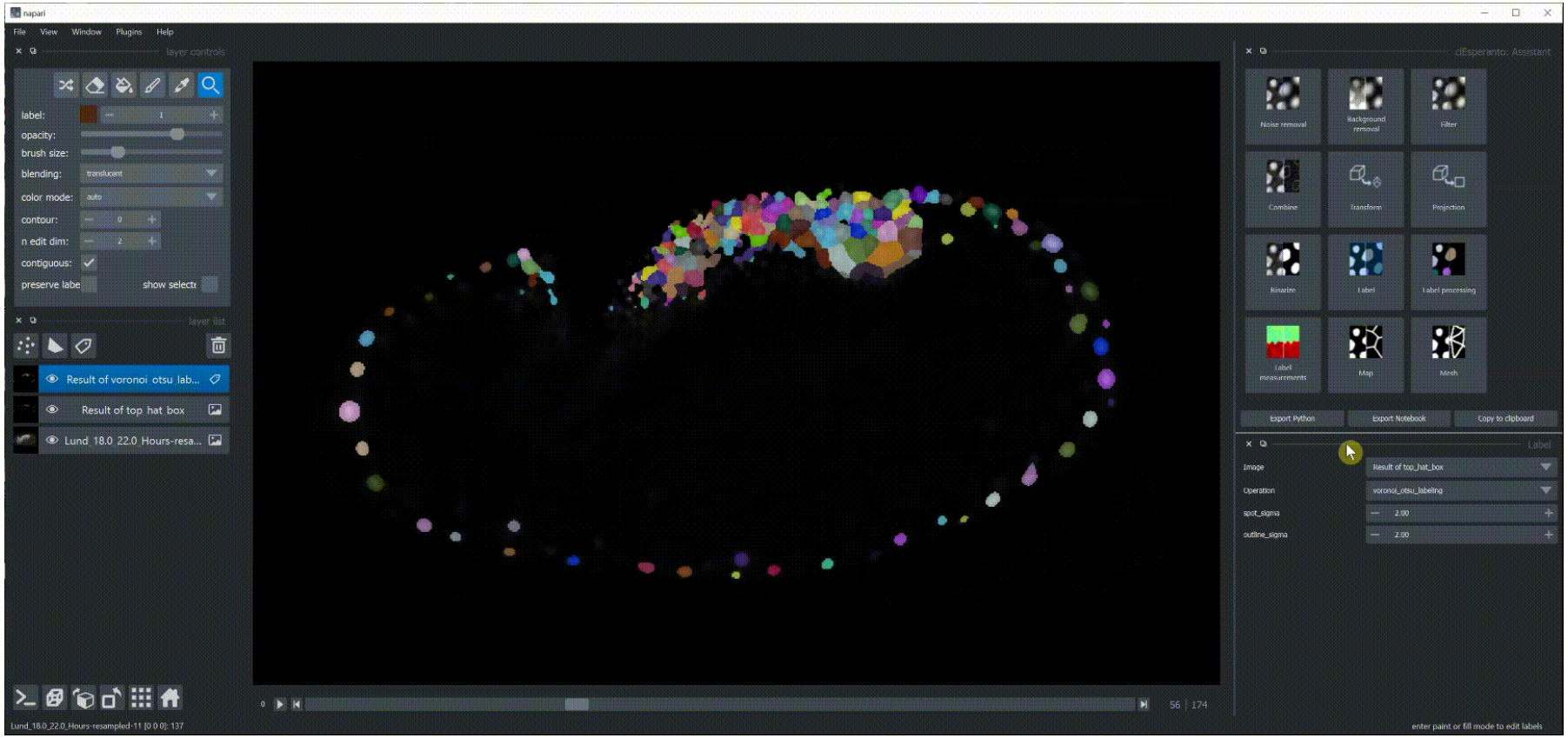
Object annotation

https://imagej.nih.gov/ij/images/

Object classification

Random Forest Classifiers based on

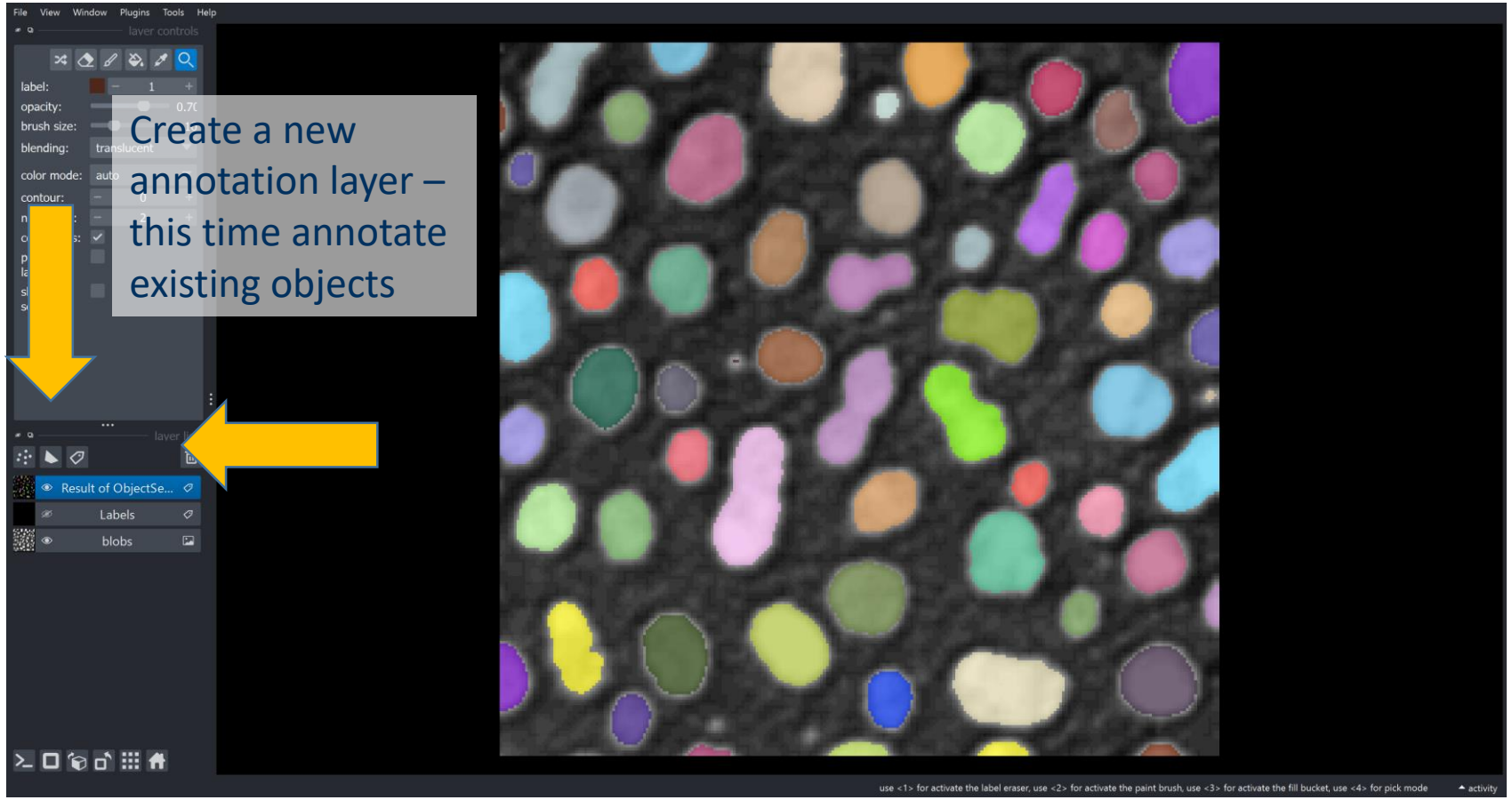
- scikit-learn and
- clesperanto



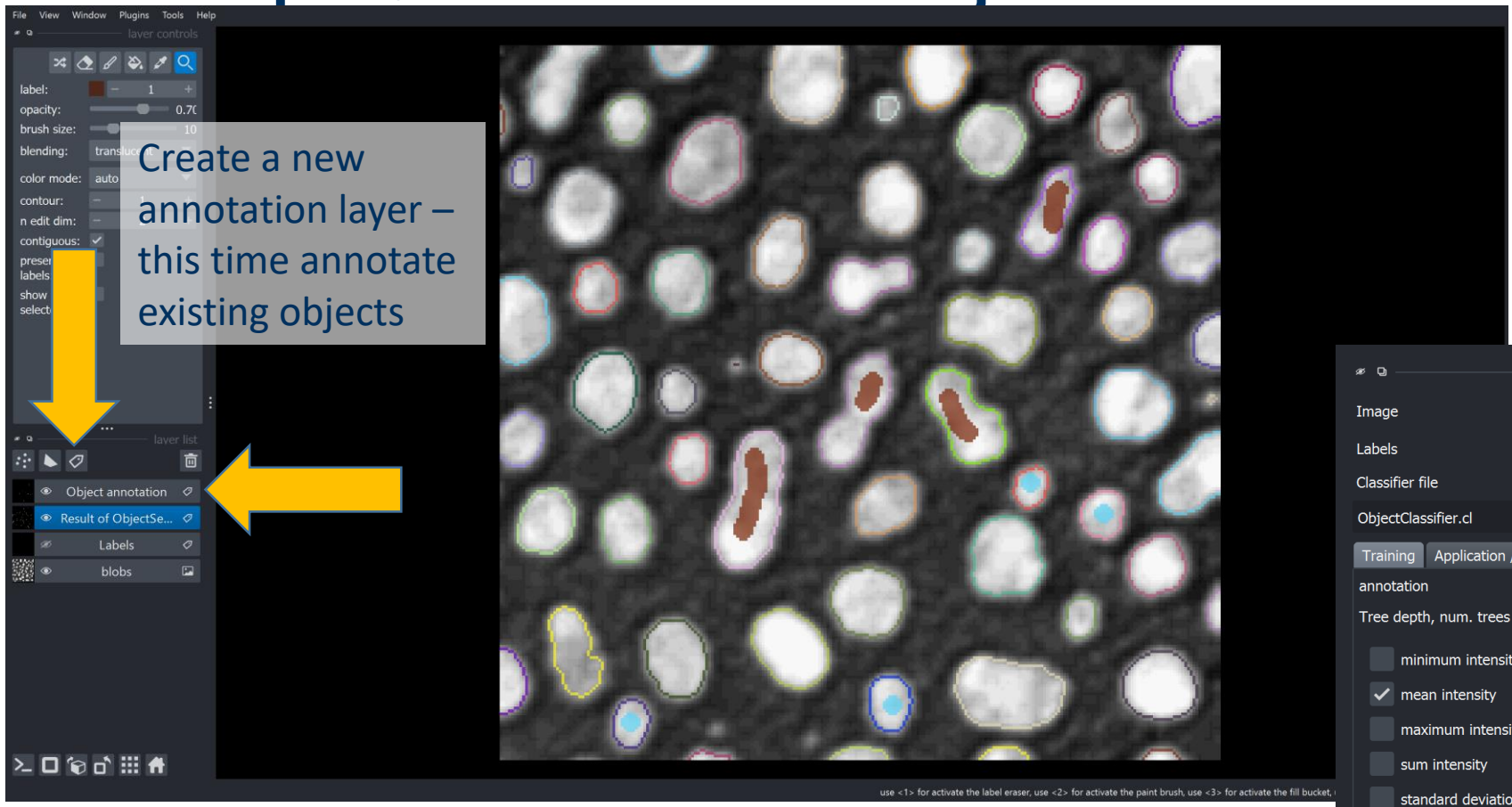
https://github.com/haesleinhuepf/napari-accelerated-pixel-and-object-classification

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

In napari: annotation

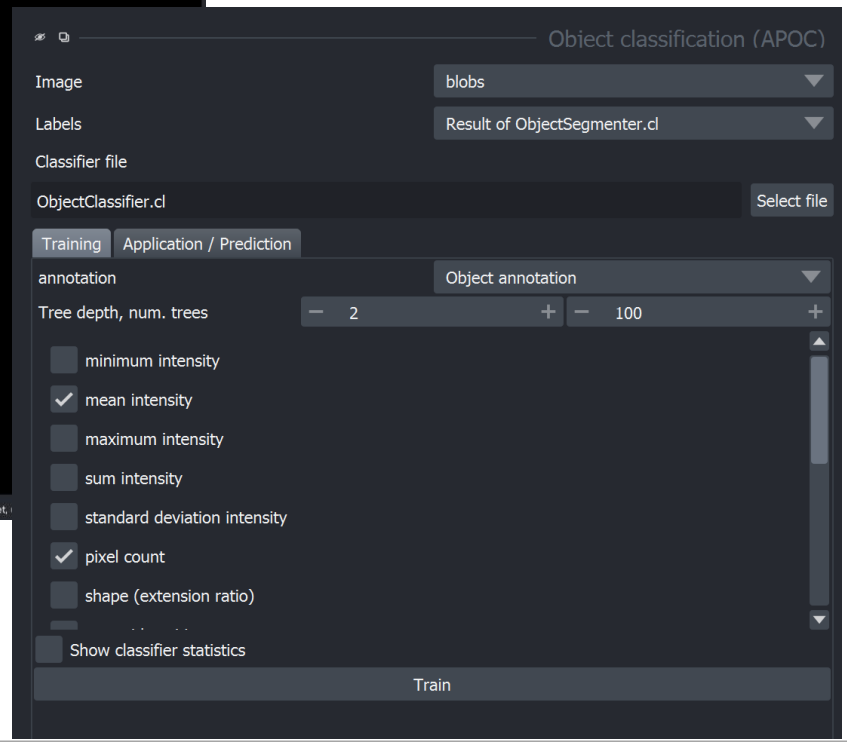


In napari: annotation and object classification



Create a new annotation layer – this time annotate existing objects

Tools ->
Segmentation post-processing ->
Object Classification (APOC)



Telling different classes of objects apart requires:

- An **annotation** for some example objects
- **Features** for each objects upon which to make a prediction

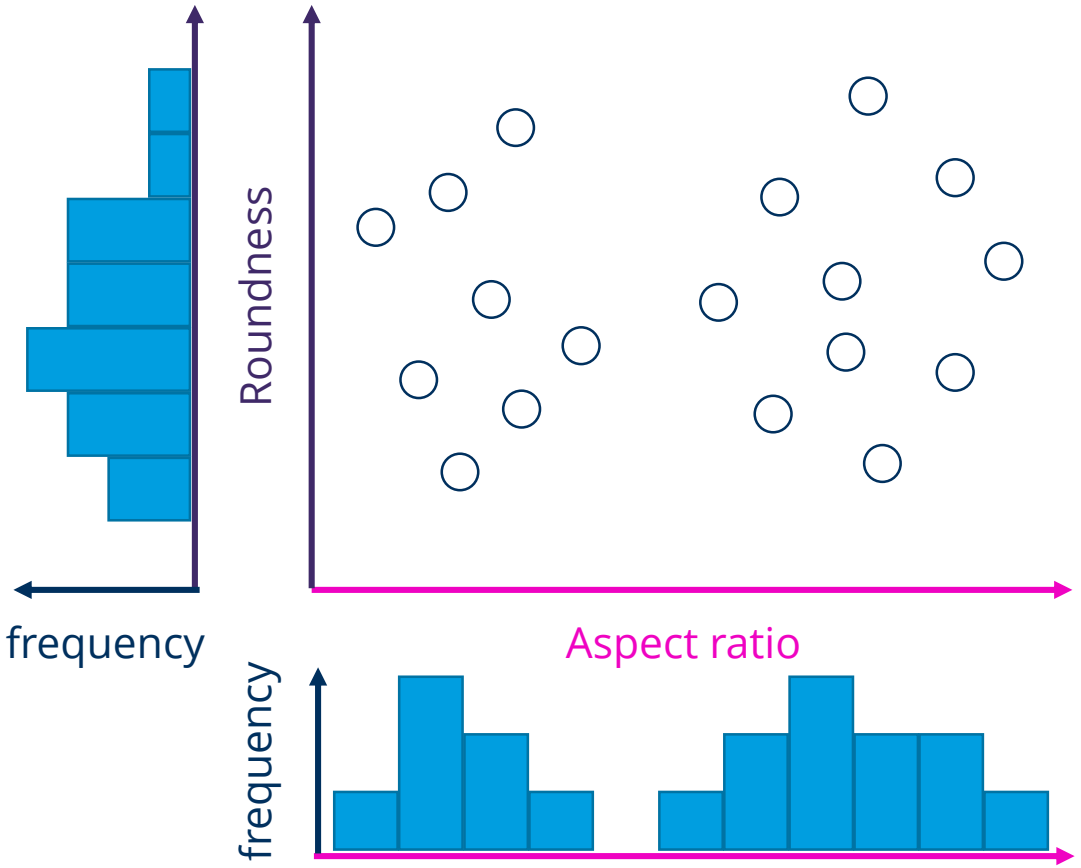
https://imagej.nih.gov/ij/images/

Object Classification and Unsupervised Machine Learning

- Dimensionality Reduction
- Clustering
- **napari-clusters-plotter plugin**

Unsupervised machine learning

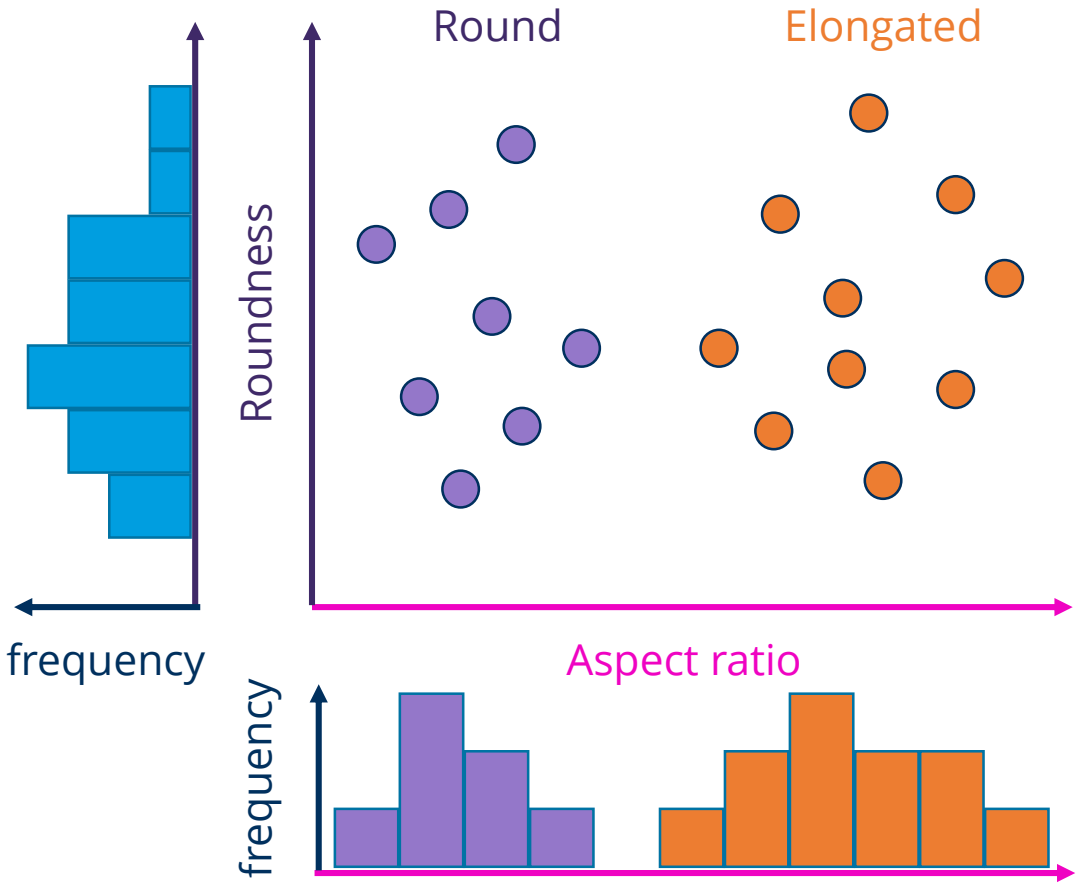
If you don’t provide ground truth, the algorithm is *unsupervised*.



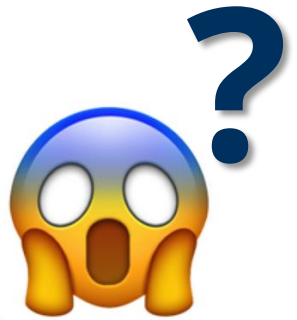
Unsupervised machine learning

If you don’t provide ground truth, the algorithm is unsupervised.

Nevertheless, algorithms can tell us something about the data



- Mean intensity
- Standard deviation
- Total intensity
- Textures
- Area / Volume
- Roundness
- Solidity
- Circularity / Sphericity
- Elongation
- Centroid
- Bounding box
- …



Dimensionality reduction

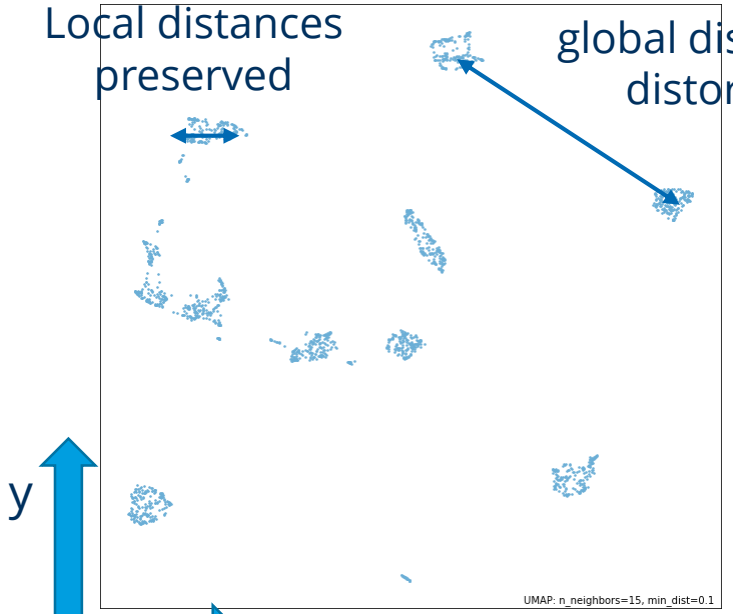
Challenge: Find a representation (embedding) of your data that represents the data in fewer dimensions

Preserve local distances at the expense of global distortions

x y z ...

	label	area	bbox_area	convex_area	quivalent_diamete	max_intensity	mean_intensity	min_intensity	solidity	extent	eret_diameter_ma	local_centroid-0	l
1	1	3379	13949	5120	18.61786412639...	613.0	345.6717963894...	259.0	0.6599609375	0....	37.3496987939662	15.77952056821...	18
2	2	2319	7448	3491	16.42230229224...	421.0	297.8434670116...	240.0	0....	0....	38.65229618017...	4....	17
3	3	2304	14415	4281	16.38681751812...	456.0	300.8298611111...	245.0	0....	0....	34.19064199455...	17.73828125	13
4	4	3278	13804	5139	18.43048549951...	467.0	316.1446003660...	249.0	0....	0....	34.84250278036...	15.52287980475...	10
5	5	1501	3315	1681	14.20563625190...	458.0	302.147235176549	236.0	0....	0....	17.97220075561...	6....	6.
6	6	2341	6061	2714	16.47407088948...	594.0	355.4446817599...	261.0	0....	0....	30.67572330035...	16.54250320375...	6.
7	7	1725	3584	1940	14.87979081163...	568.0	343.7866666666...	257.0	0....	0....	17.72004514666...	7.804637681159 42	7.
8	8	1502	3840	1753	14.20879025650...	431.0	290.0659121171...	235.0	0....	0....	18.57417562100...	8....	6.
9	9	1602	4080	1894	14.51737058294...	475.0	297.8008739076...	241.0	0....	0....	18.70828693386...	8....	8.
10	10	1395	3600	1624	13.86304166283...	424.0	304.8494623655...	247.0	0....	0.3875	17.60681686165...	7....	7.
11	11	609	1100	697	10.51654029260...	323.0	274.2528735632...	241.0	0....	0....	13.45362404707...	3....	4.
12	12	1686	3757	1894	14.76679738567...	460.0	303.8303677342...	240.0	0....	0....	17.97220075561...	9....	7.
13	13	2157	5184	2531	16.03062694504...	576.0	339.990264255911	270.0	0....	0....	19.54482028569...	8....	8.
14	14	863	2340	1032	11.81237949737...	327.0	272.4449594438...	237.0	0....	0....	16.0312195418814	6....	5.

Many dimensions



Few dimensions

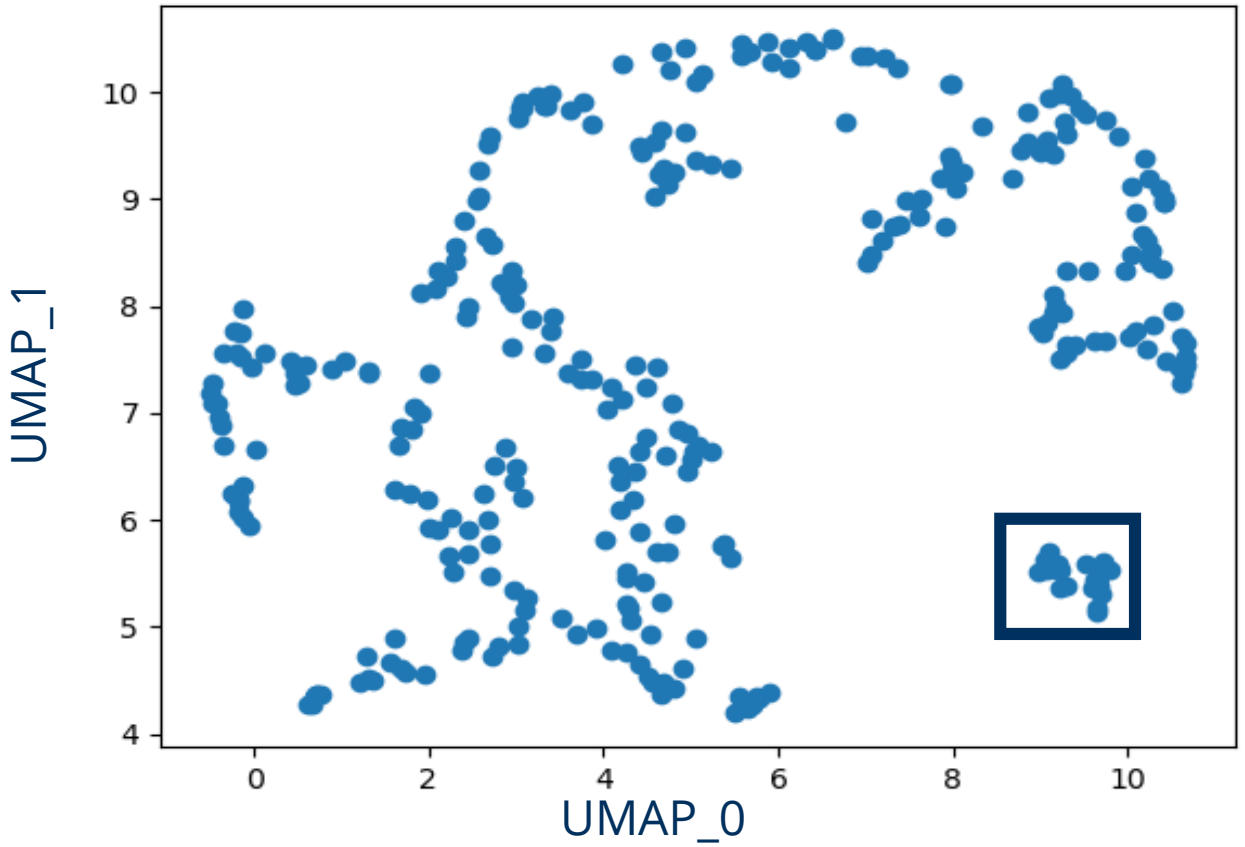
UNIFORM MANIFOLD UMAP APPROXIMATION & PROJECTION

https://umap-learn.readthedocs.io/en/latest/index.html

Clustering

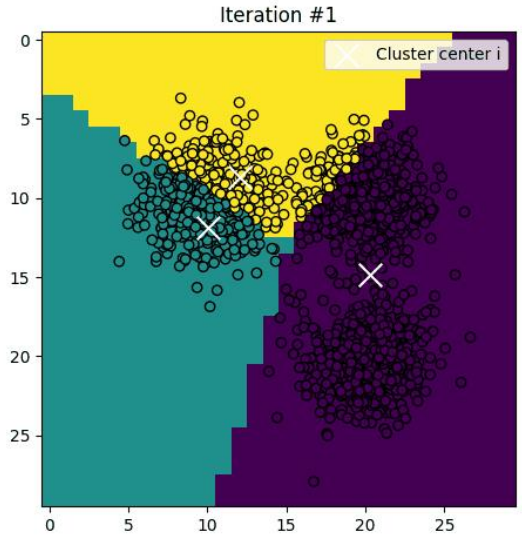
Starting point: Feature space or dimensionality reduction reveals “groups” in our data

Can we automatically identify these groups?



→ **Clustering** allows to stratify data into groups *without previous annotations*

K-means clustering



Dimensionality reduction

Uniform manifold approximation and projection (UMAP)

t-distributed stochastic neighbor embedding (t-SNE)

Principal component analysis (PCA)

Laura Žigutytė
@zigutyte

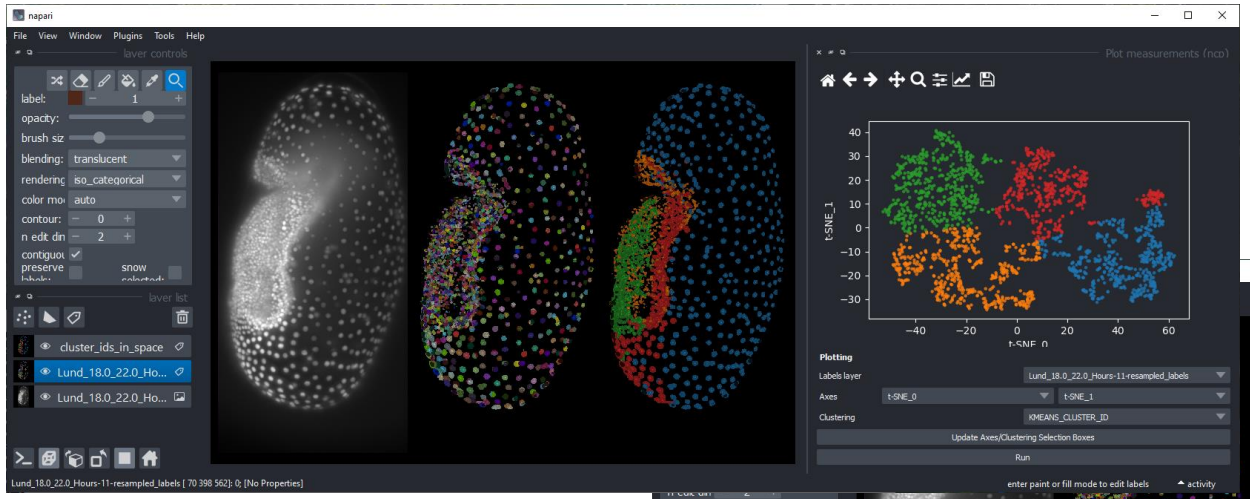
Ryan Savill
@RyanSavill4

Marcelo L. Zoccoler
Robert Haase,
Thorsten Wagner,
Johannes Soltwedel

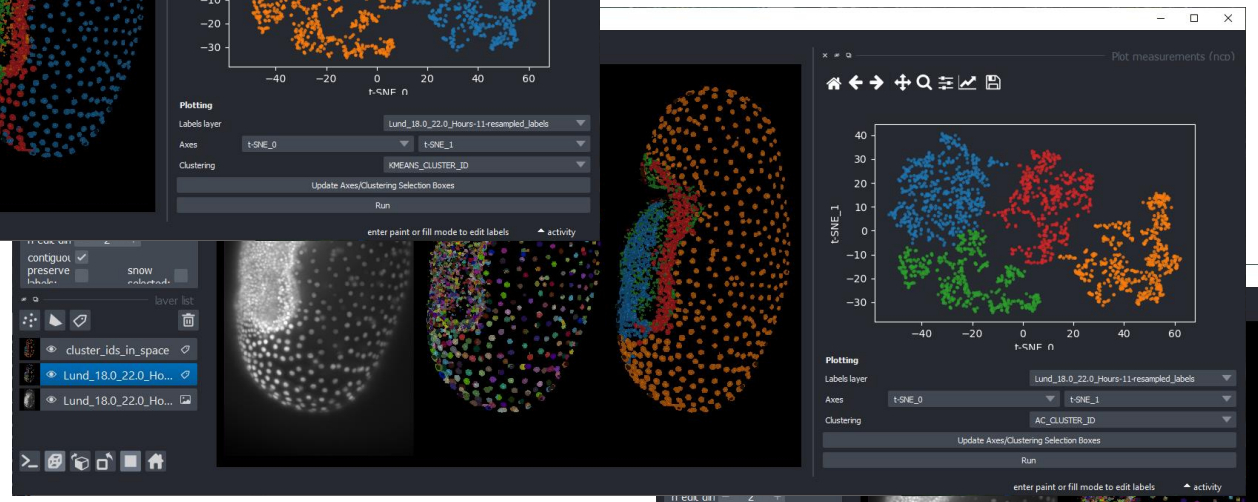
https://github.com/BiAPoL/napari-clusters-plotter

Image Data Source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD Dresden

Clustering

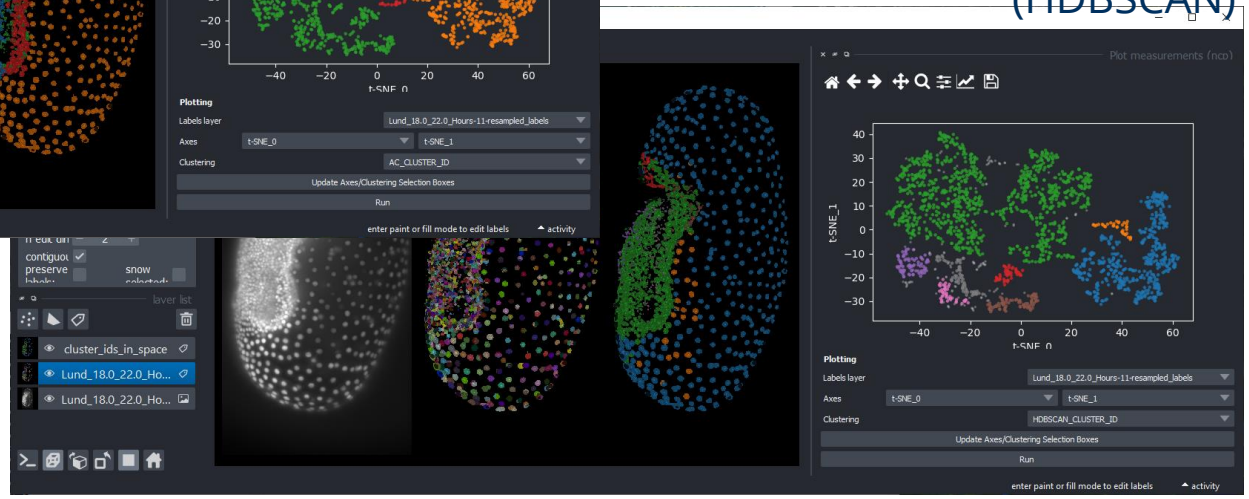


K-means clustering



Agglomerative clustering

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)



Laura Žigutytė
@zigutyte

Ryan Savill
@RyanSavill4

Marcelo L. Zoccoler
Robert Haase,
Thorsten Wagner,
Johannes Soltwedel

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

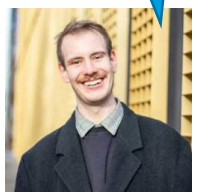
Image Data Source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD Dresden

Data Exploration

... using interactive clustering

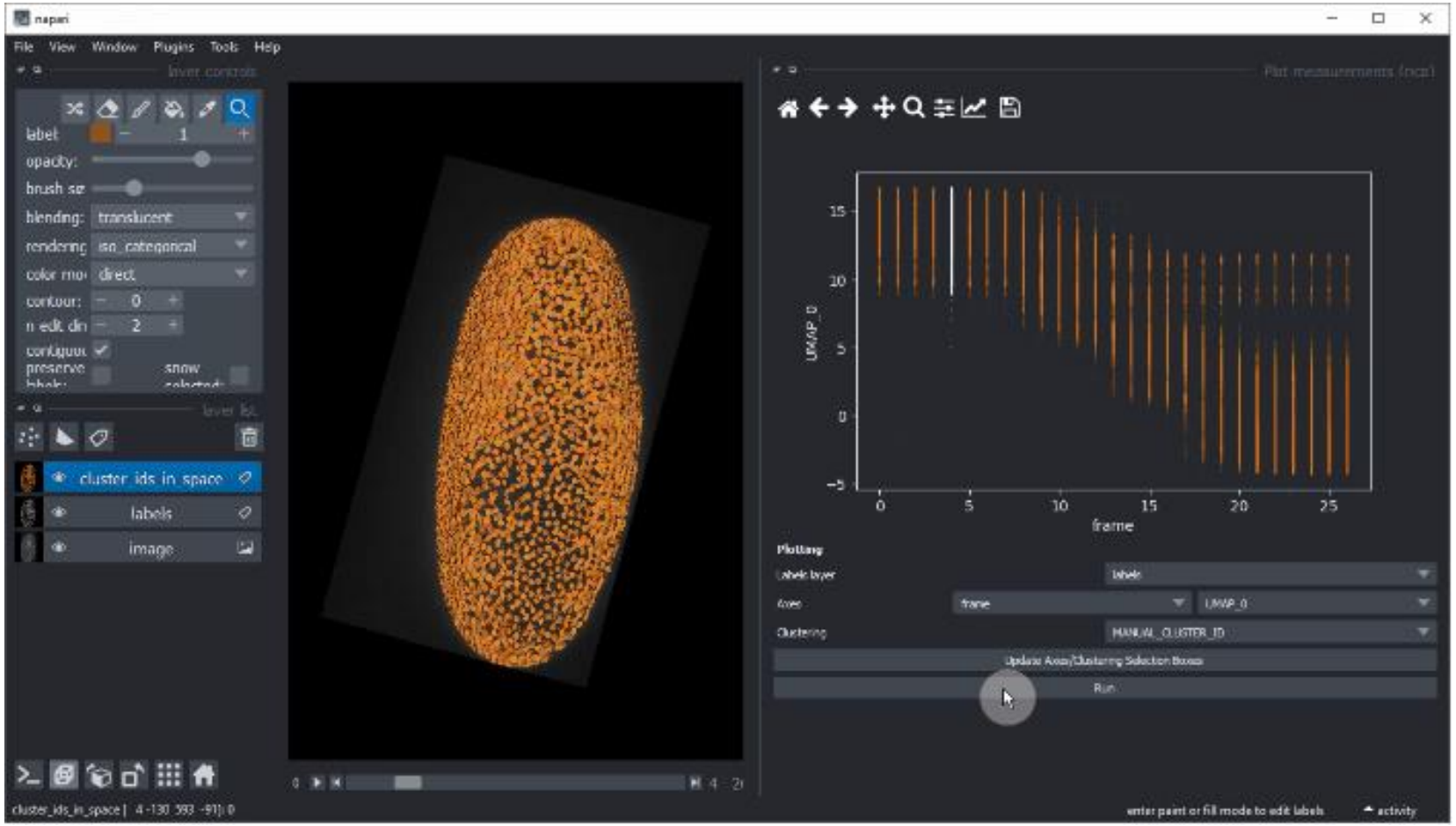
Multi-layer analysis coming soon in 0.9.0!

Laura Žigutytė
@zigutyte



Ryan Savill
@RyanSavill4

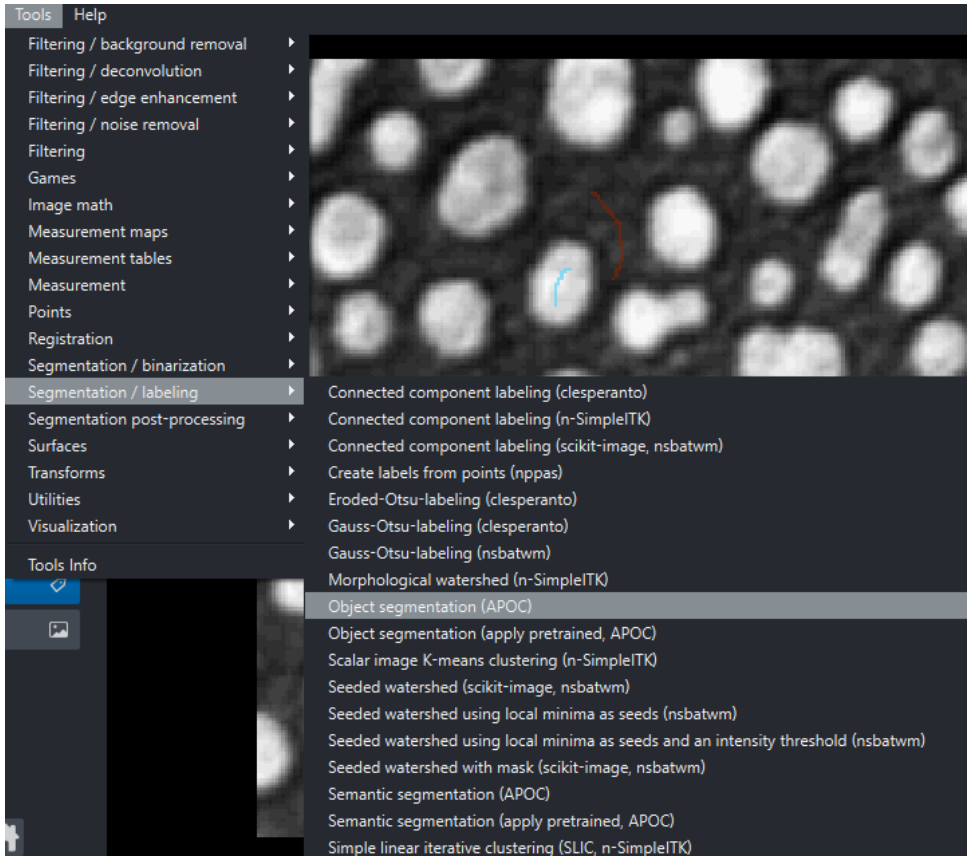
Marcelo L. Zoccoler
Robert Haase,
Thorsten Wagner,
Johannes Soltwedel



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

Image Data Source: Daniela Vorkel, Myers lab, MPI-CBG/CSBD Dresden

Exercise: Object segmentation



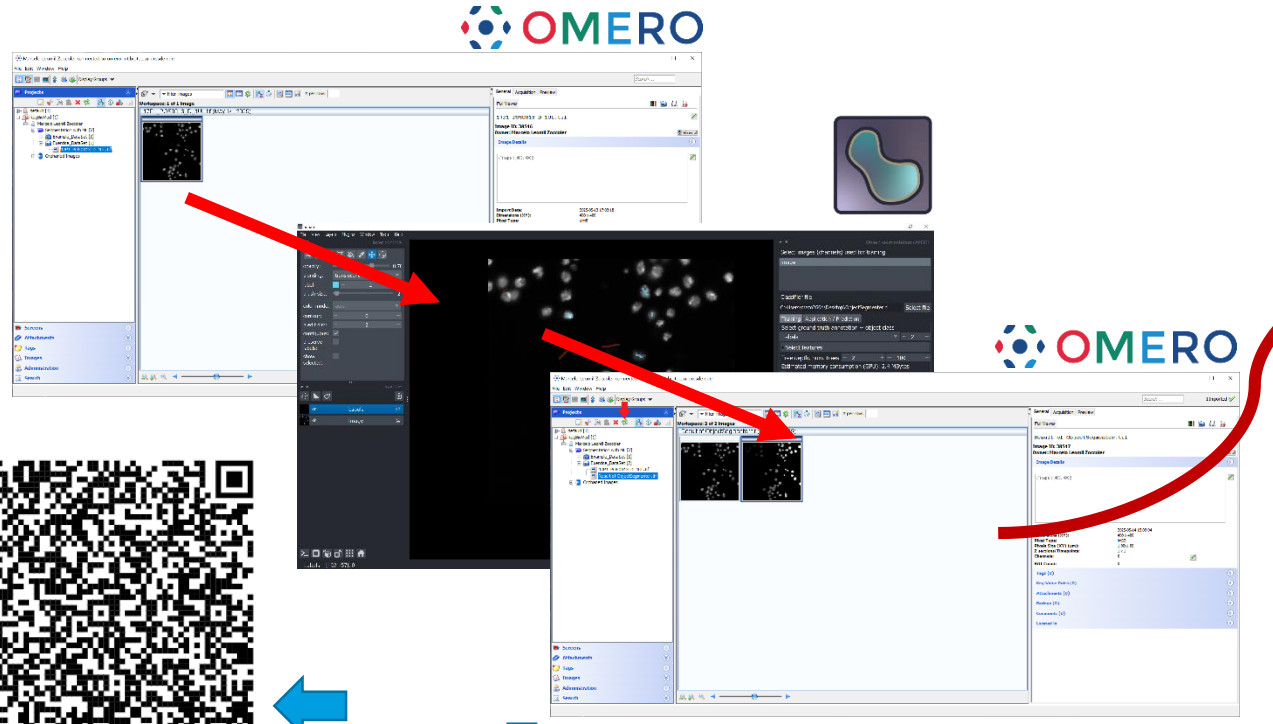
1. Activate the environment and open napari
   ```
   mamba activate amhct
   napari
   ```
2. Open the blobs image
3. Perform object segmentation on blobs sample dataset

https://imagej.nih.gov/ij/images/

https://biapol.github.io/AMHCT_Bio_Image_Analysis_2025/interactive_pixel_classification/readme.html

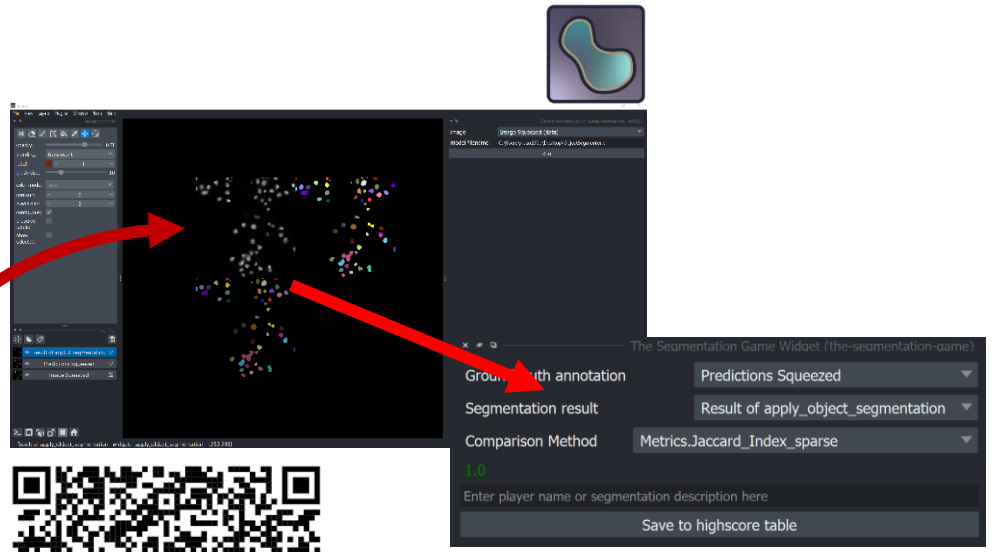
Exercise: Recording and Reproducing a Bio-Image Analysis Workflow

Creating a workflow:



https://biapol.github.io/AMHCT_Bio_Image_Analysis_2025/napari_apoc_omero_workflow/workflow_creation.html

Reproducing your colleague's workflow:



https://biapol.github.io/AMHCT_Bio_Image_Analysis_2025/napari_apoc_omero_workflow/workflow_reproduction.html

Discussion

- How many succeeded in **creating** the workflow?
- How many succeeded in **reproducing** the workflow?
- Which were key items to ensure reproducibility?
- Which part was the hardest?
- What could hamper reproducibility?

Further resources:

Napari-hub:
www.napari-hub.org

BiaPoL image analysis course materials:

https://github.com/BiAPoL/Bio-image_Analysis_with_Python

Bio-image analysis materials:
https://bioimagebook.github.io/README.html

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/intro.html

Bio-Image Workflow Tools:

https://galaxyproject.org/

https://github.com/nextflow-io/nextflow

Image Science Community Forum:
https://forum.image.sc/

Acknowledgements

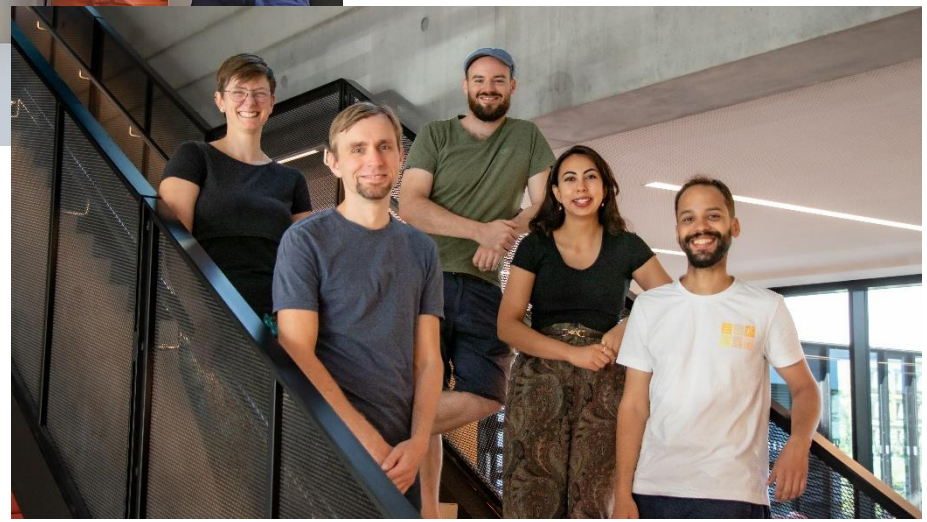
PoL Physics of Life TU Dresden

BiAPoL team

- Marcelo Zoccoler
- Johannes Soltwedel
- Stefan Hahmann

Former lab members:

- Robert Haase
- Allyson Ryan
- Till Korten
- Mara Lampert
- Svetlana Iarovenko
- Ryan George Savill
- Laura Zigutyte
- Somashekhar Kulkarni
- Maleeha Hassan
- Tina Smejka



Networks

BioDIP Biopolis Dresden Imaging Platform

CENTER FOR SYSTEMS BIOLOGY DRESDEN

image.sc

NFDI4 BIOIMAGE

Funding

DFG public

@mazoc.bsky.social
@jo-soltwedel.bsky.social

https://physics-of-life.tu-dresden.de/bia

TECHNISCHE UNIVERSITÄT DRESDEN