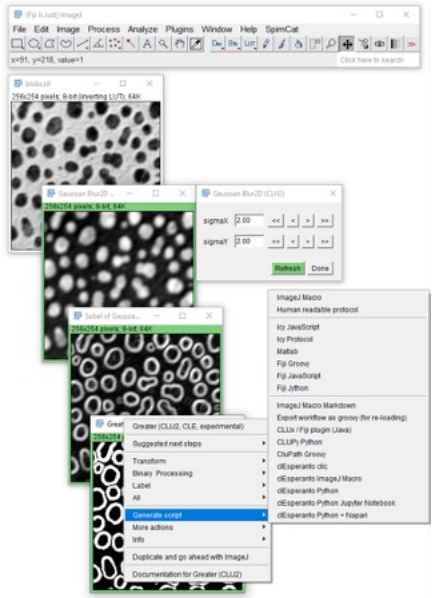
PoL Physics of Life TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

From Assistant to Notebooks

Till Korten



With material from: Robert Haase

June 2023

@TillKorten

Getting started

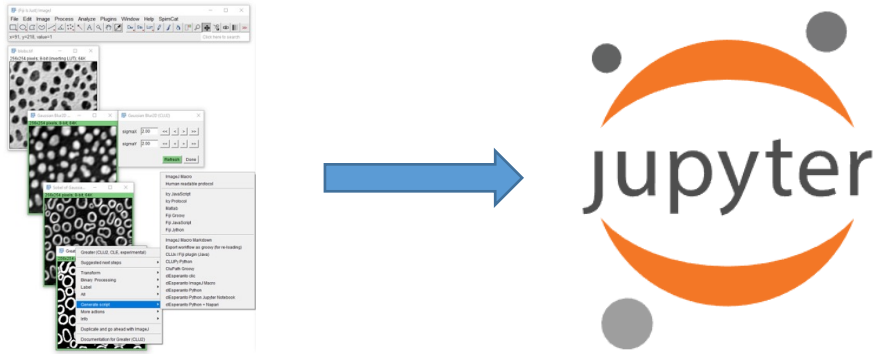
- Follow this blog post to install python, mamba and devbio-napari: https://t.ly/0BXB

Example/exercise materials and slides online

- https://t.ly/DVsyu

Outline

- Exporting workflows as notebooks



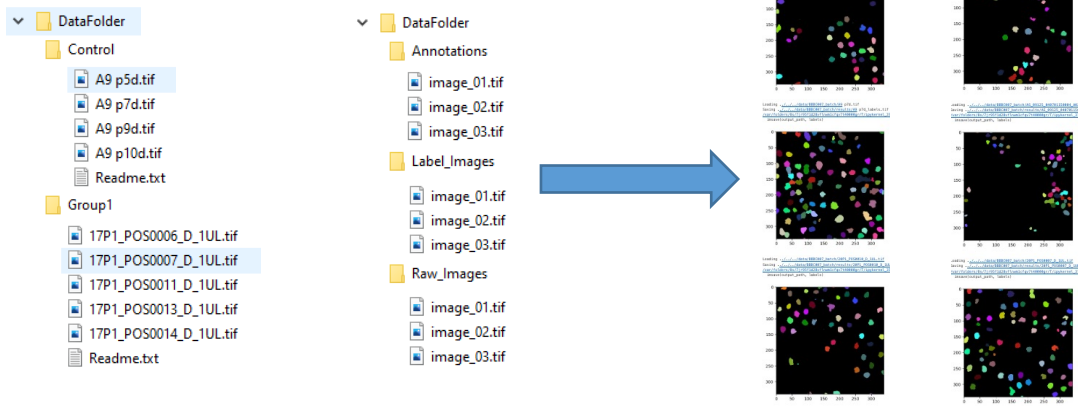
- Python programming basics

```
firstname = "Robert"
lastname = 'Haase'

print("Hello " + firstname + " " + lastname)
```

```
Hello Robert Haase
```

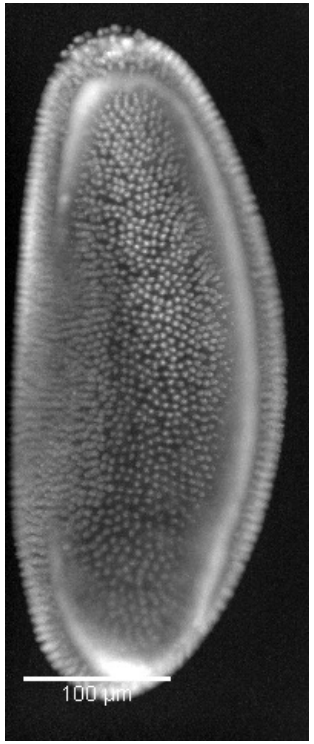
- Processing folders of images

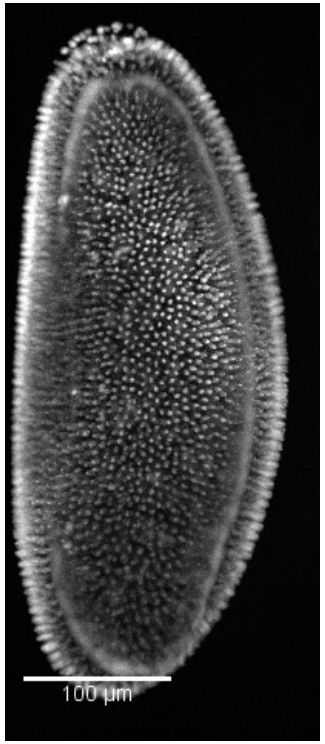


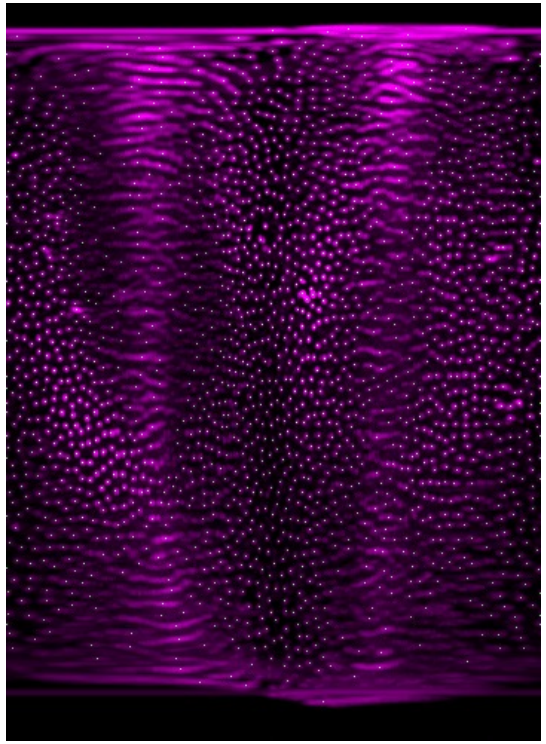
Recap – image analysis workflows

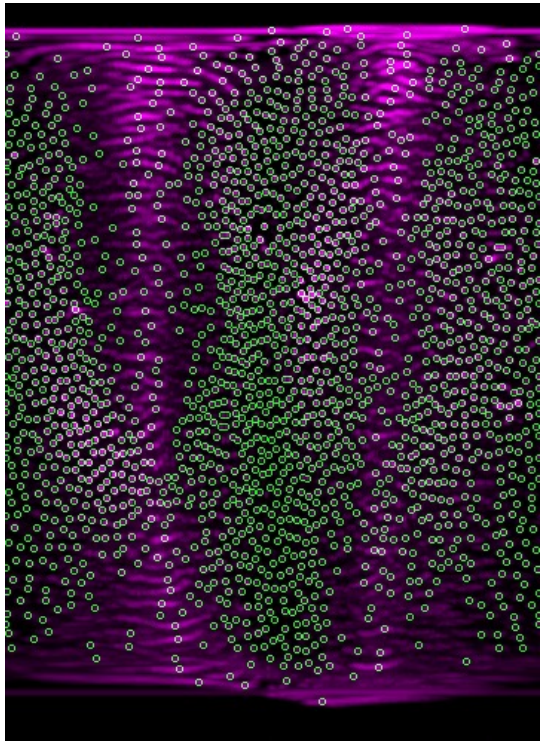
PoL Physics of Life TU Dresden

Load data → Preprocessing → Transformation → Segmentation → Save data









Haase et al Nat Methods (2020)

Why Notebooks?

- A great way to document your analysis workflow
  - Understandable
  - Reproducible
  - Publishable
  - Reusable - don't forget to add a license: https://choosealicense.com/
- Helps automate your workflow (some programming required)
  - Saves time
  - Improves reproducibility and repeatability
  - Reduces bias (but you still need to be careful when you set up the workflow)

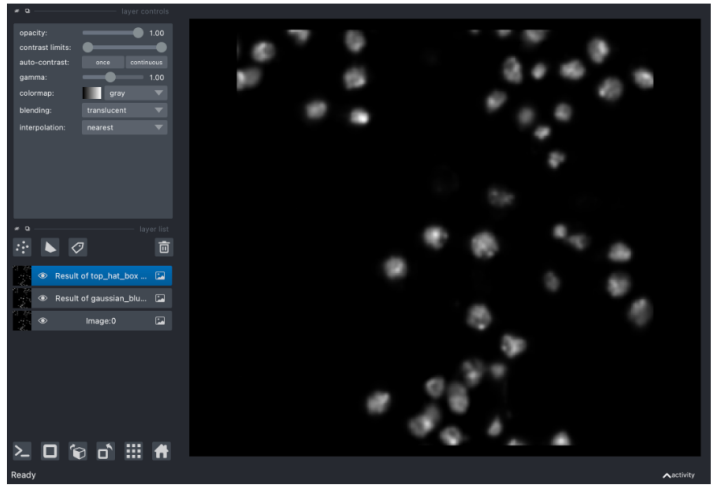
top hat box

Background was removed using a top-hat algorithm with a symmetric, box-shaped footprint of 10 by 10 pixels.

This algorithm treats objects that are significantly larger than the footprint as background and smaller objects as signal.

```python
image2_thb = cle.top_hat_box(image1_gb, None, 10.0, 10.0
viewer.add_image(image2_thb, name='Result of top_hat_box
napari.utils.nbscreenshot(viewer)
```

Python

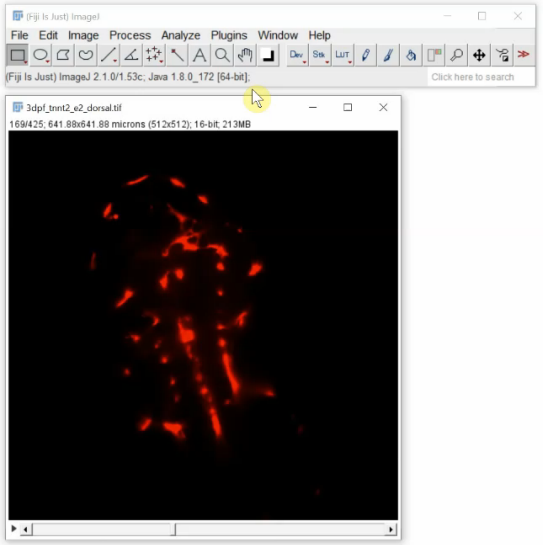


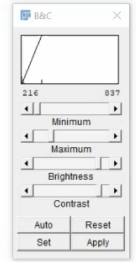
threshold otsu binarization

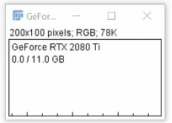
The image was binarized using a threshold determined by using Otsu's method [1]

Fiji/clij: Set up a workflow

- Make x, y, z – projections match biological axes







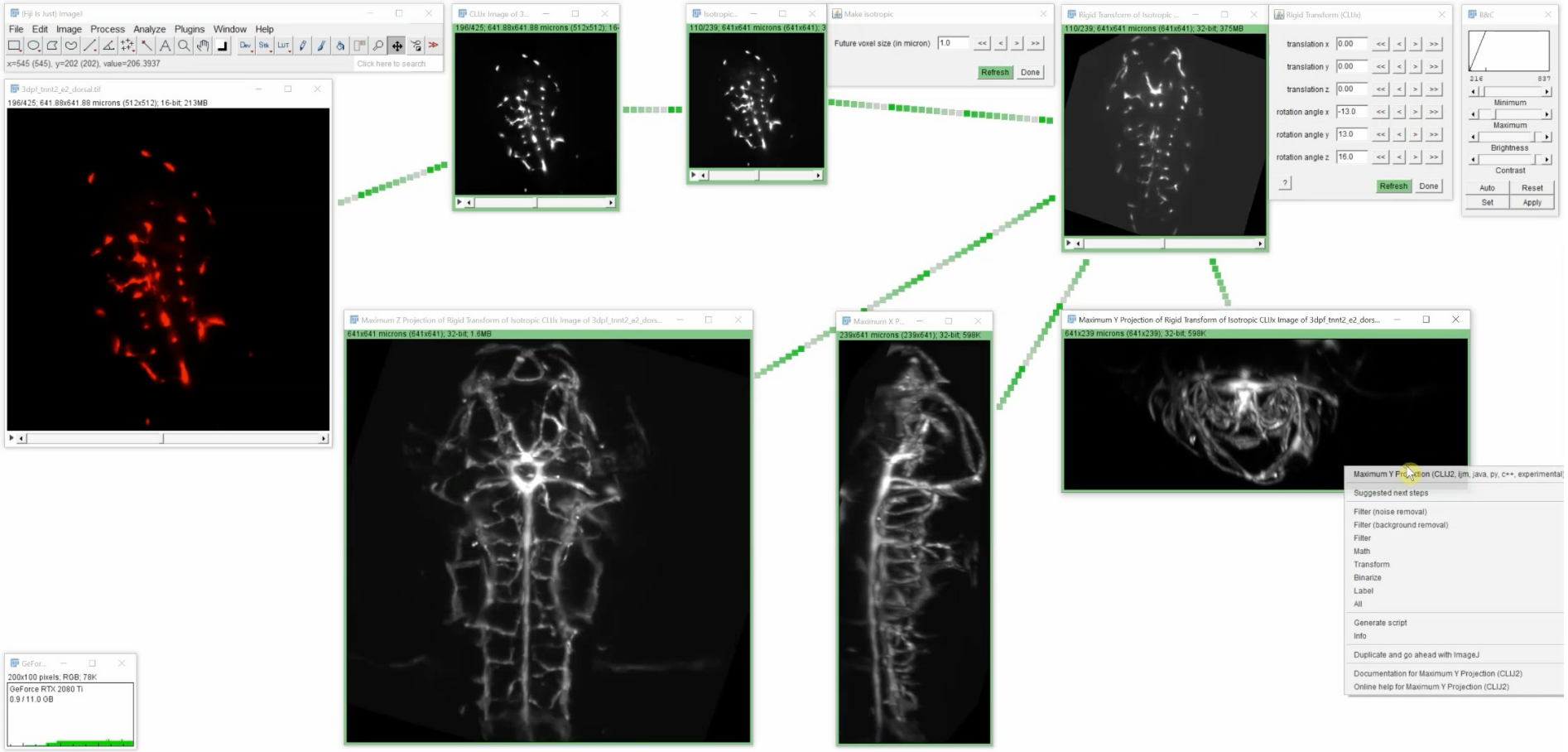
Special thanks to Elisabeth Kugler!

Elisabeth Kugle
@KuglerElisabeth

Image data source: Elisabeth Kugler; labs of Tim Chico and Paul Armitage, The University of Sheffield (UK)" https://zenodo.org/record/4204839#.X8DCRGj7Q2w

Fiji/clij: Generate code

- After setting up the workflow, generate code!

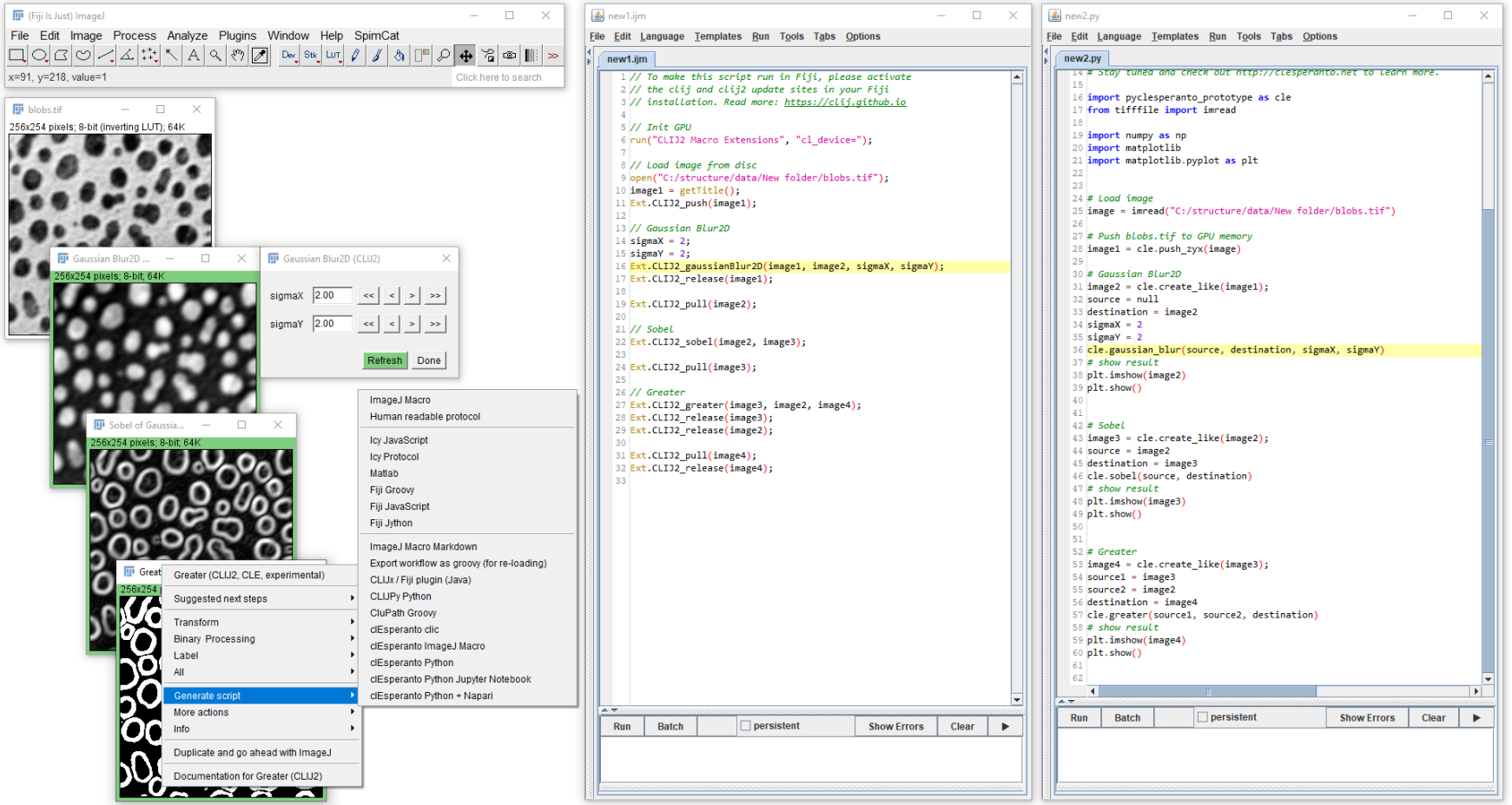


Special thanks to Elisabeth Kugler!

Elisabeth Kugle
@KuglerElisabeth

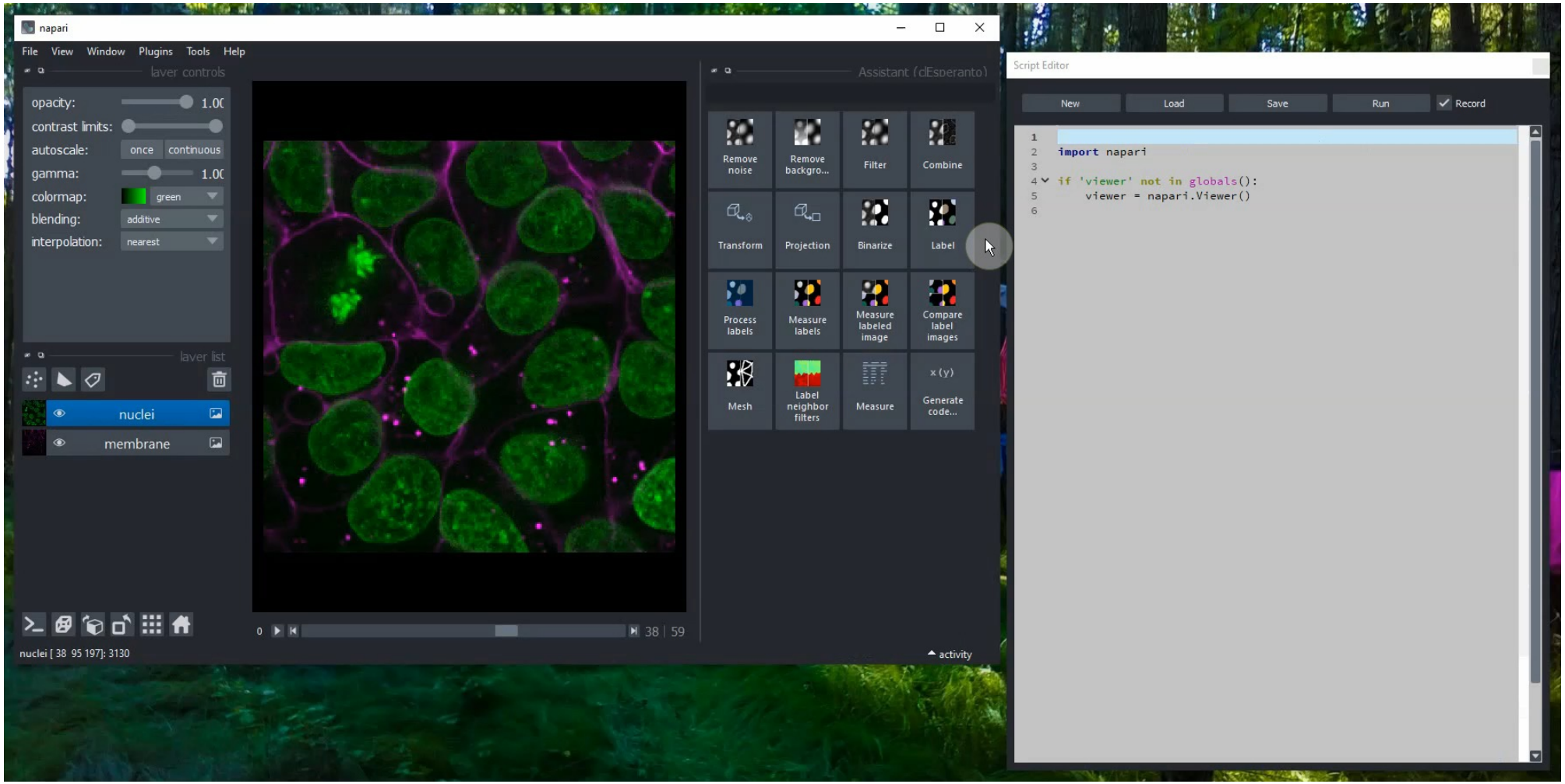
Image data source: Elisabeth Kugler; labs of Tim Chico and Paul Armitage, The University of Sheffield (UK)" https://zenodo.org/record/4204839#.X8DCRGj7Q2w

Choosing a programming language



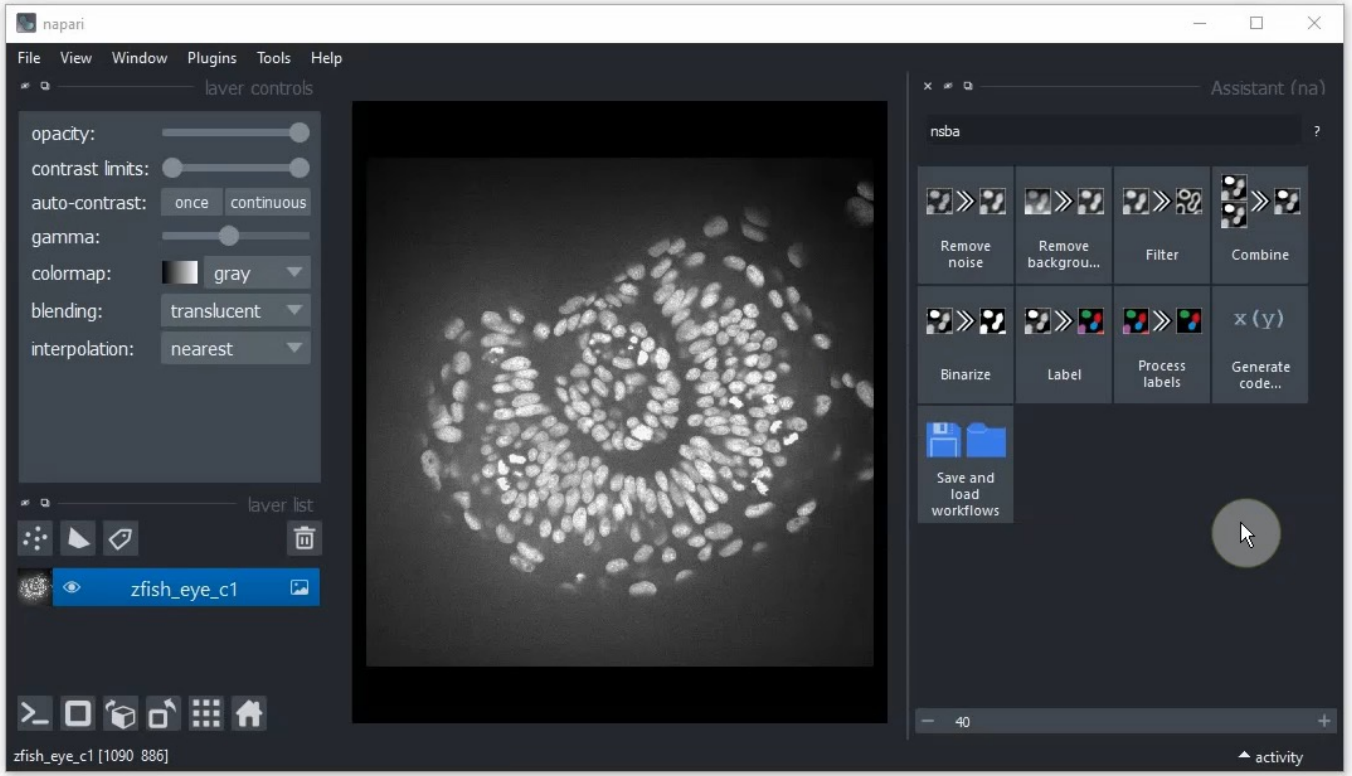
- Do you have access to existing scripts from others (and does the license allow you to use them)?
- Do you need other tools/packages (e.g. a deep learning python package)?
- Personal preference

Create a workflow with Napari-Assistant



https://www.napari-hub.org/plugins/napari-script-editor

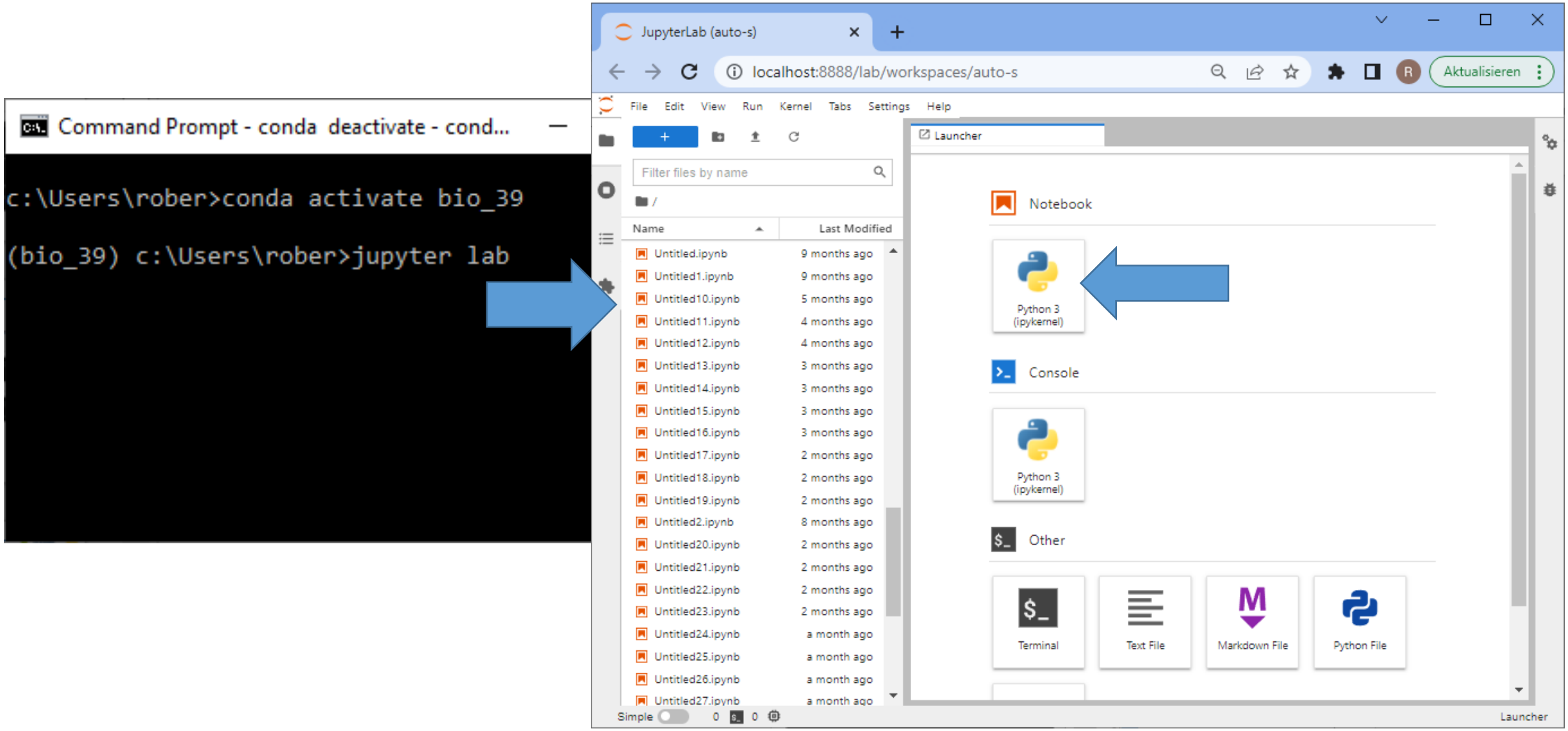
Export code to Jupyter Notebooks



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

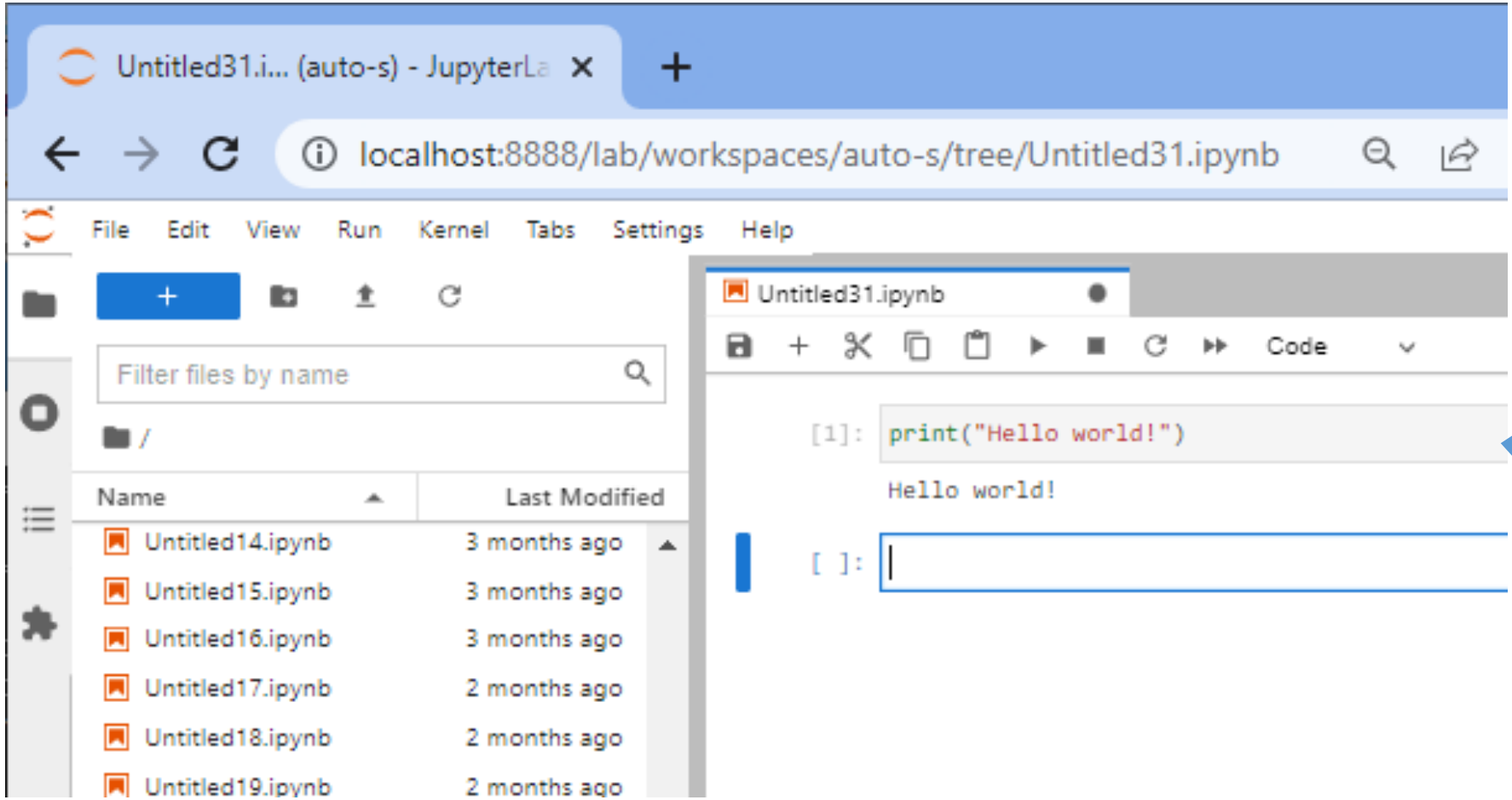
Image data source: Mauricio Rocha Martins, Norden lab, MPI CBG (now at IGC Oeiras)

Open the notebook in Jupyter lab



Jupyter lab

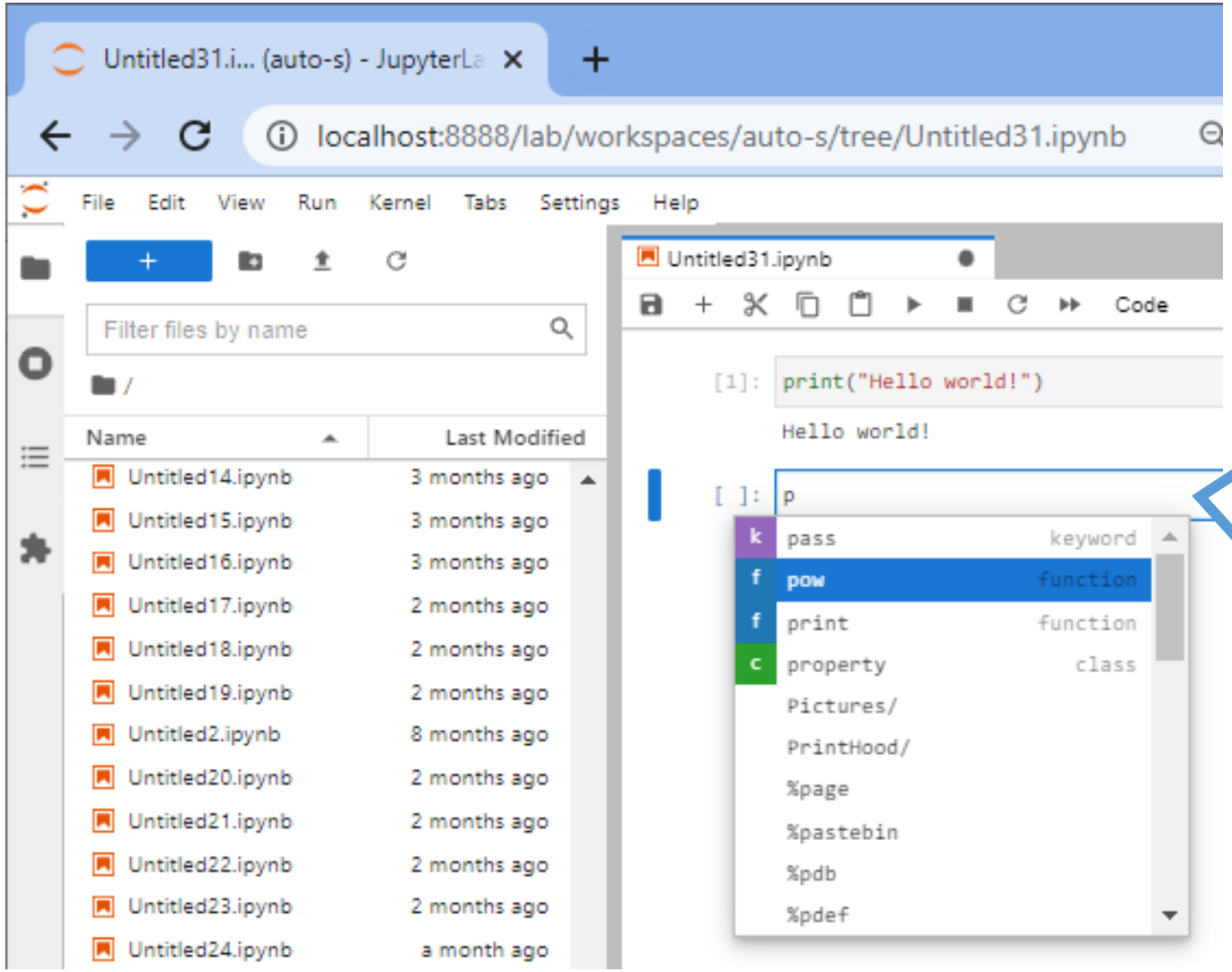
- Execute code cell-by-cell and see results instantaneously



SHIFT + ENTER to execute a code cell

Jupyter lab

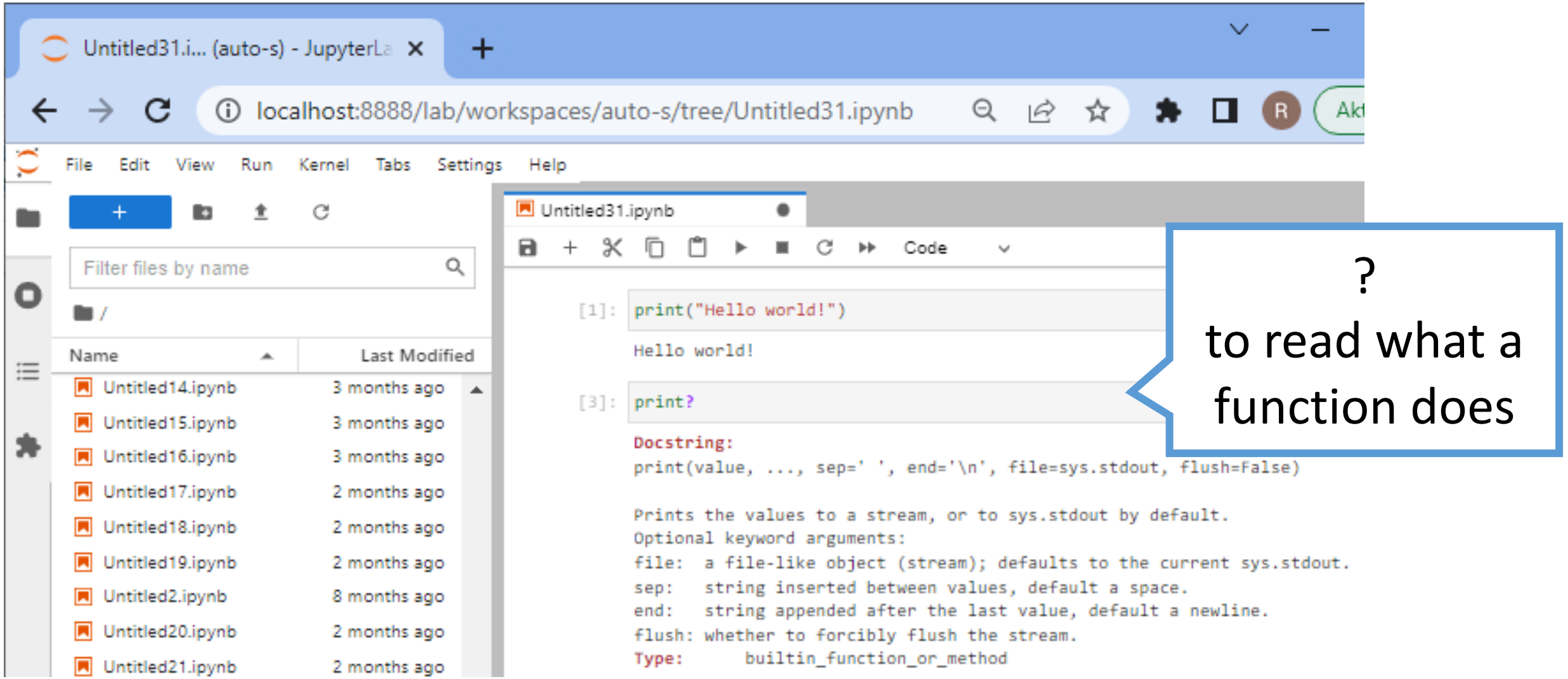
- Context-specific help, auto-completion



TAB
to open auto-completion

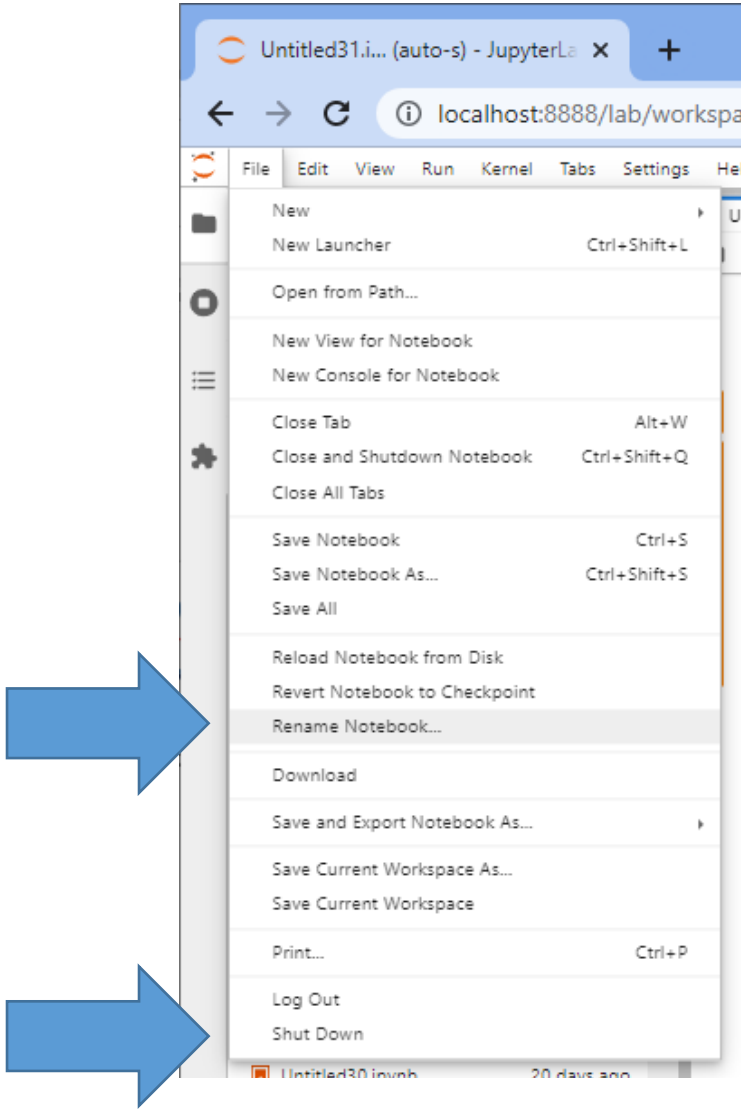
Jupyter lab

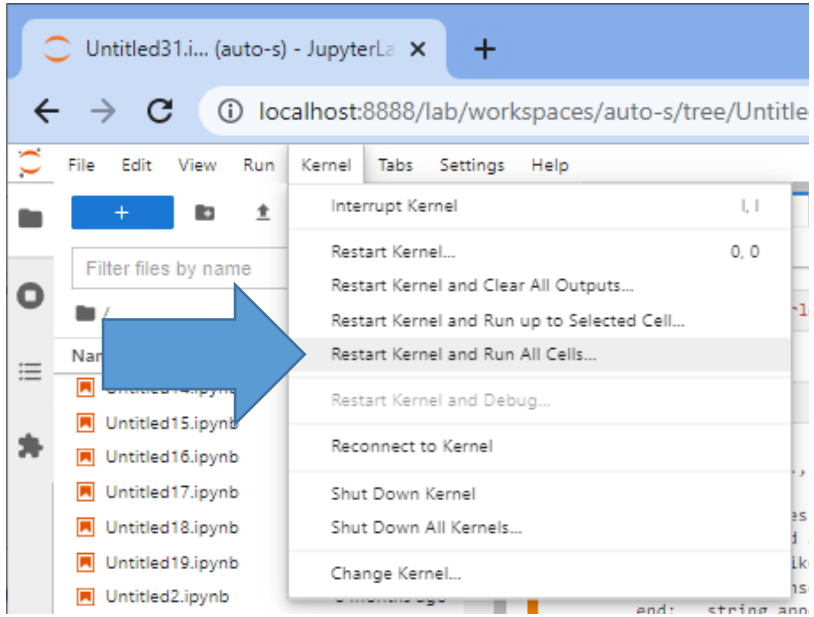
- Help / “docstrings”



Jupyter lab

- Saving / renaming / closing





Enforcing a “clean” execution state is important for ensuring reproducibility and repeatability

Live demo: create workflow notebook using napari

PoL Physics of Life TU Dresden

- Example/exercise notebooks online: https://t.ly/DVsyu