



ScaDS.AI

DRESDEN LEIPZIG

CENTER FOR SCALABLE DATA ANALYTICS
AND ARTIFICIAL INTELLIGENCE

Bio-Image Data Science Training School 2024

Anja Neumann, Christian Martin, Dušan Praščević, Jan Ewald, Laura Žigutytė,
Marie-Sophie von Braun, Matthias Täschner, Robert Haase

GEFÖRDERT VOM



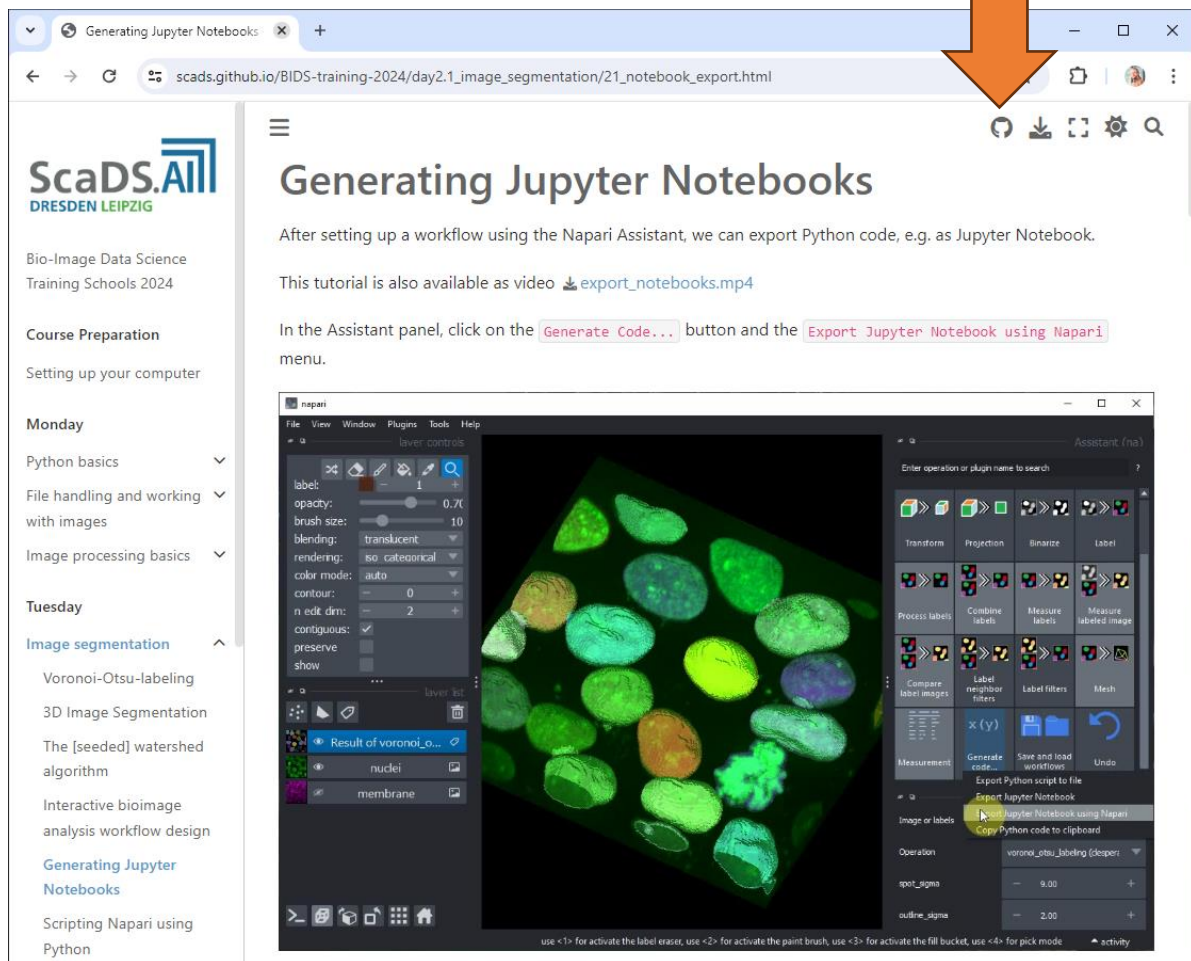
Bundesministerium
für Bildung
und Forschung



Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.

Training Materials

We updated the materials during the training school, download them again



ScaDS.ALI
DRESDEN LEIPZIG

Bio-Image Data Science
Training Schools 2024

Course Preparation
Setting up your computer

Monday

- Python basics
- File handling and working with images
- Image processing basics

Tuesday

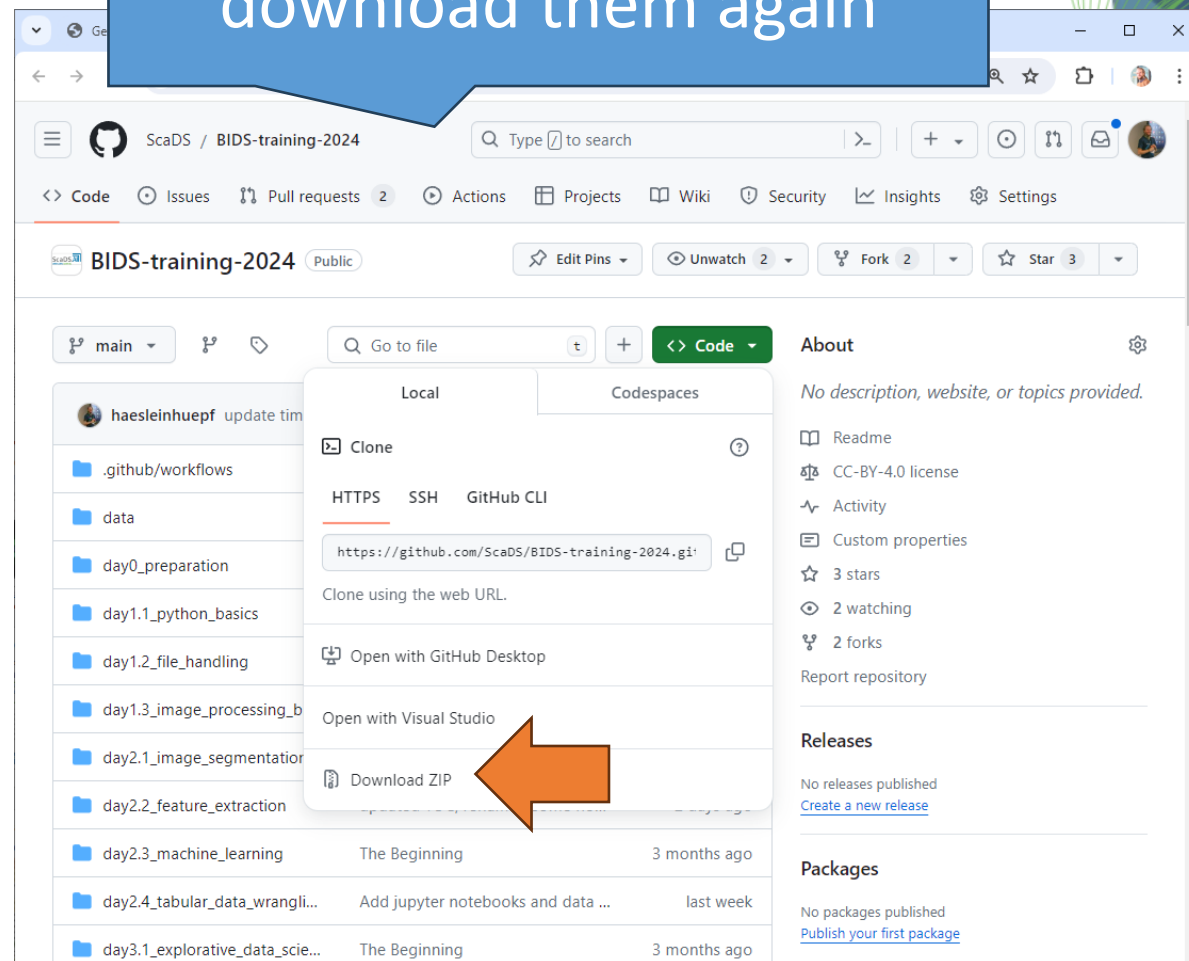
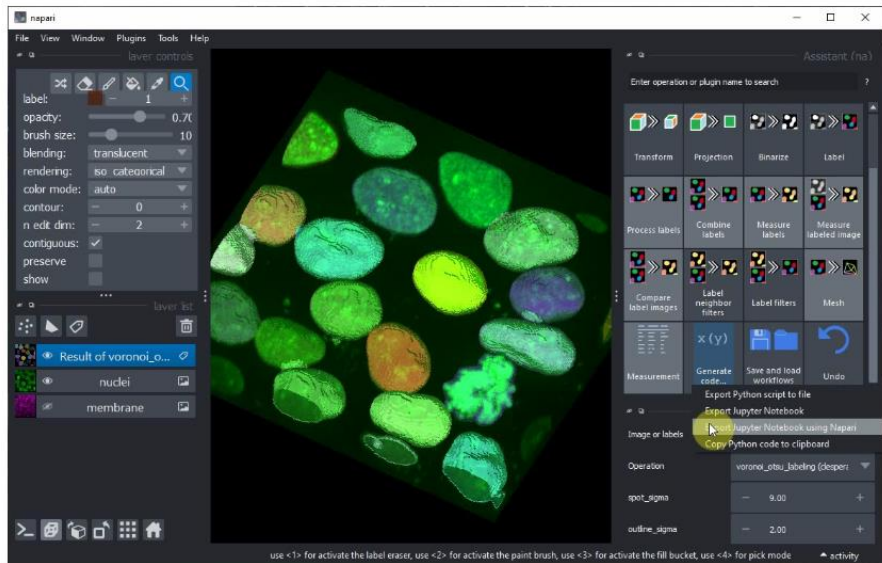
- Image segmentation
 - Voronoi-Otsu-labeling
 - 3D Image Segmentation
 - The [seeded] watershed algorithm
 - Interactive bioimage analysis workflow design
 - Generating Jupyter Notebooks
 - Scripting Napari using Python

Generating Jupyter Notebooks

After setting up a workflow using the Napari Assistant, we can export Python code, e.g. as Jupyter Notebook.

This tutorial is also available as video [export_notebooks.mp4](#)

In the Assistant panel, click on the **Generate Code...** button and the **Export Jupyter Notebook using Napari** menu.



ScaDS / BIDS-training-2024

Type to search

<> Code Issues Pull requests 2 Actions Projects Wiki Security Insights Settings

BIDS-training-2024 Public

Edit Pins Unwatch 2 Fork 2 Star 3

main

Go to file

Clone

HTTPS SSH GitHub CLI

<https://github.com/ScaDS/BIDS-training-2024.git>

Clone using the web URL.

Open with GitHub Desktop

Open with Visual Studio

Download ZIP

haesleinhuepf update time

- .github/workflows
- data
- day0_preparation
- day1.1_python_basics
- day1.2_file_handling
- day1.3_image_processing_b
- day2.1_image_segmentation
- day2.2_feature_extraction
- day2.3_machine_learning
- day2.4_tabular_data_wrangli...
- day3.1_explorative_data_scie...

The Beginning 3 months ago

Add jupyter notebooks and data ... last week

The Beginning 3 months ago

About

No description, website, or topics provided.

Readme

CC-BY-4.0 license

Activity

Custom properties

3 stars

2 watching

2 forks

Report repository

Releases

No releases published

[Create a new release](#)

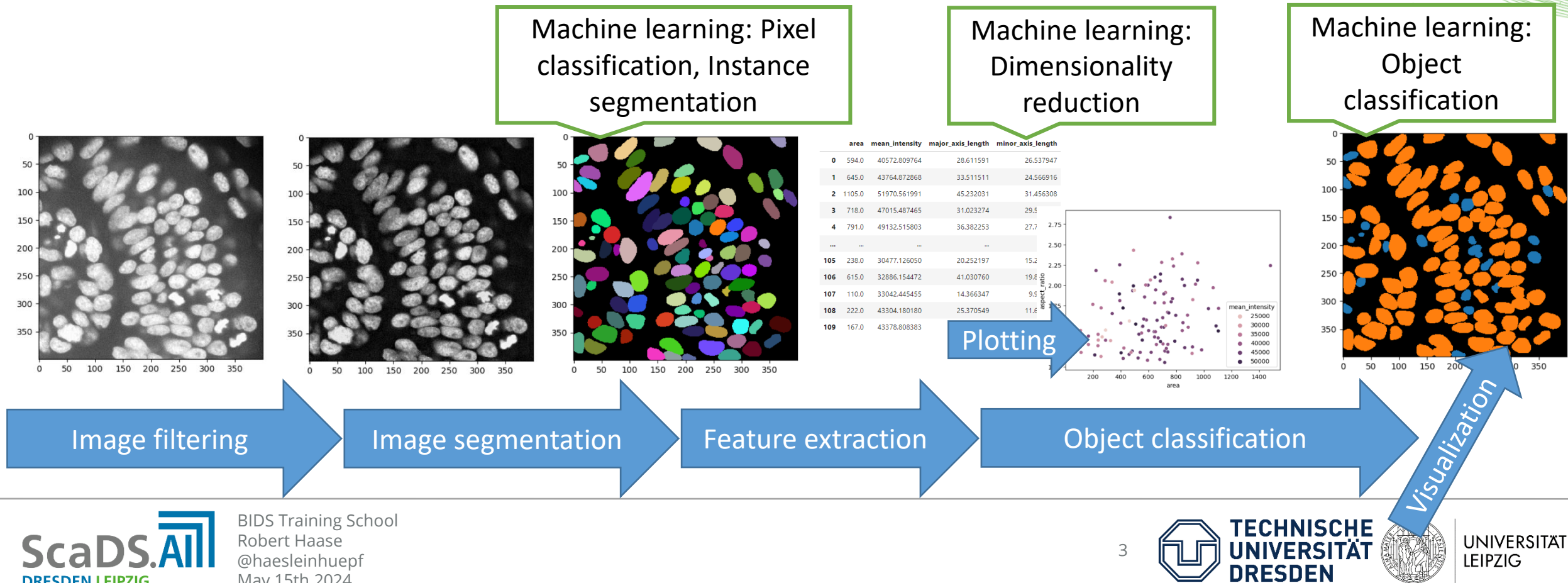
Packages

No packages published

[Publish your first package](#)

Overview: Bio-image Data Science

- [Image] Data Analysis workflows
- Goal: **Quantify observations, substantiate conclusions with numbers**



Python error messages

```
[1]: from skimage.io import imread
```

```
[2]: imread("unknown_file.tif")
```

```
-----  
FileNotFoundError                                Traceback (most recent call last)  
~\AppData\Local\Temp\ipykernel_17160\2534596731.py in ?()  
----> 1 imread("unknown_file.tif")  
  
~\mambaforge\envs\bio399\lib\site-packages\skimage\io\_io.py in ?(fname, as_g  
ray, plugin, **plugin_args)
```

FileNotFoundError: [Errno 2] No such file or directory: 'C:\\Users\\haase\\un
known_file.tif'

Where did
it happen?

What
happened?

```
[ ]:
```



Ask ChatGPT



Du

What does the code `image[:,::5]` do?



Du

Explain me like I'm 12 years old and keep it short

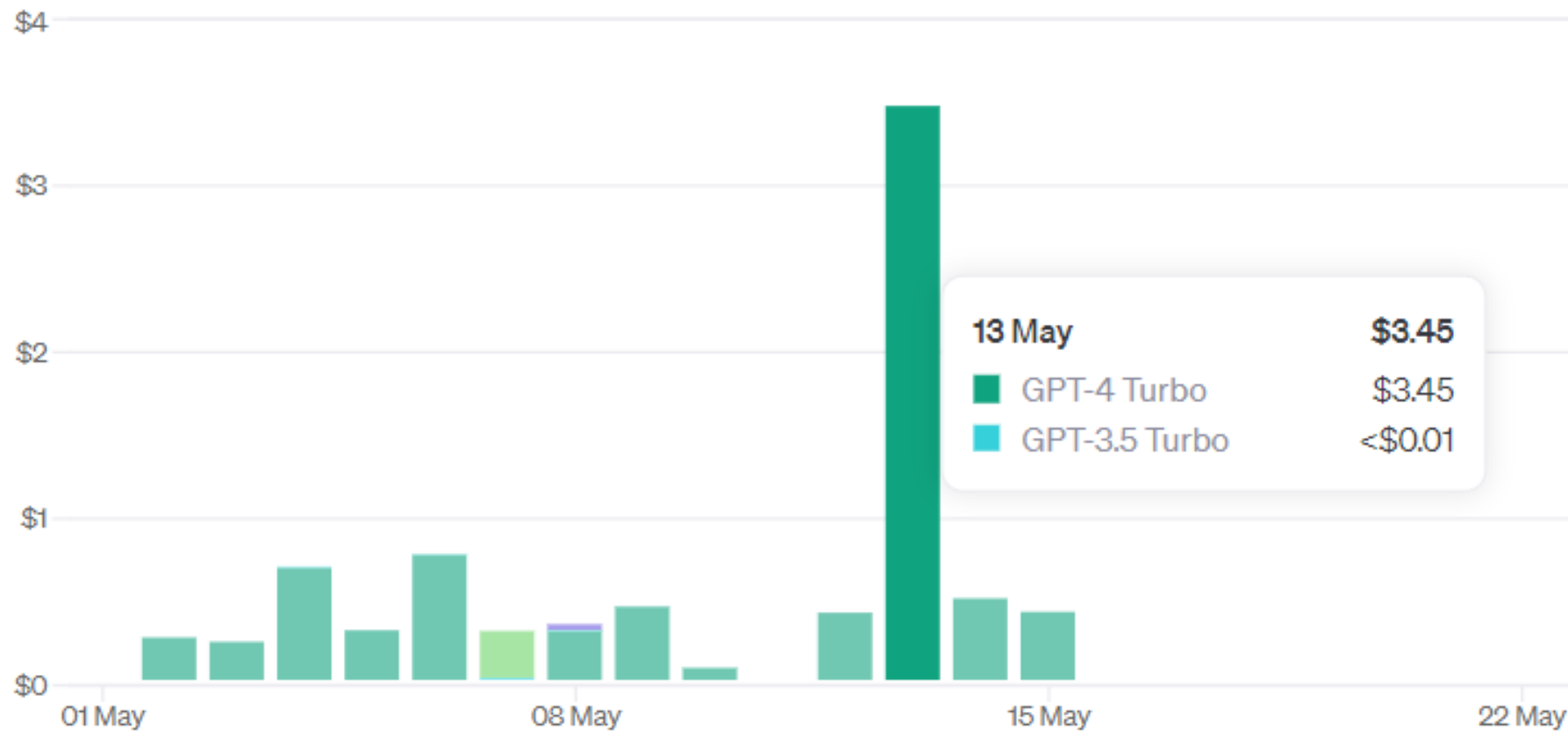


ChatGPT

Imagine you have a big grid of colored squares, which is a picture. If you take every fifth square from each row and each column, and put those squares together to make a smaller grid, that's what the code ``image[:,::5, ::5]`` does. It makes the big picture smaller by keeping only some of the squares and skipping a lot in between. This makes the picture smaller and simpler!

BiA-Bob costs

Monthly Spend \$8.11



Exploring Python modules and functions

```
[1]: import napari_segment_blobs_and_things_with_membranes as nsbatwm
```

```
[ ]: nsbatwm.g
```

```
[ ]: f gauss_otsu_labeling function
f gaussian function
f gaussian_blur function
f gaussian_laplace function
f grayscale_closing function
f grayscale_dilation function
f grayscale_erosion function
f grayscale_opening function
f regionprops function
f register_function function
```

TAB key

Exploring Python modules and functions

```
[1]: import napari_segment_blobs_and_things_with_membranes as nsbatwm
```

```
[ ]: nsbatwm.gauss_otsu_labeling(
```

```
[ ]:
```

SHIFT +
TAB key

Signature:

```
nsbatwm.gauss_otsu_labeling(  
    image: 'napari.types.ImageData',  
    outline_sigma: float = 2,  
    *,  
    viewer: 'napari.viewer.Viewer' = None,  
) -> 'napari.types.LabelsData'
```

Docstring:

Gauss-Otsu-Labeling can be used to segment objects such as nuclei with bright intensity on low intensity background images.

The outline_sigma parameter allows tuning how precise segmented objects are outlined. Under the hood, this filter applies a Gaussian blur, Otsu-thresholding and connected component labeling.

See also

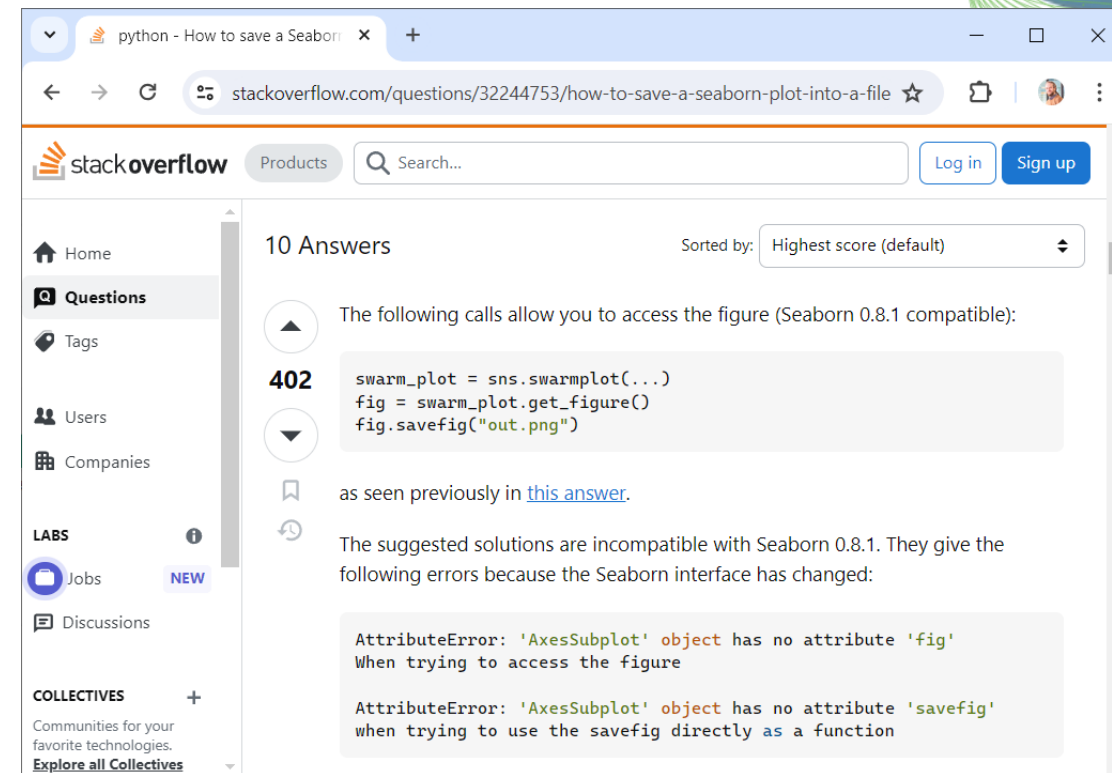
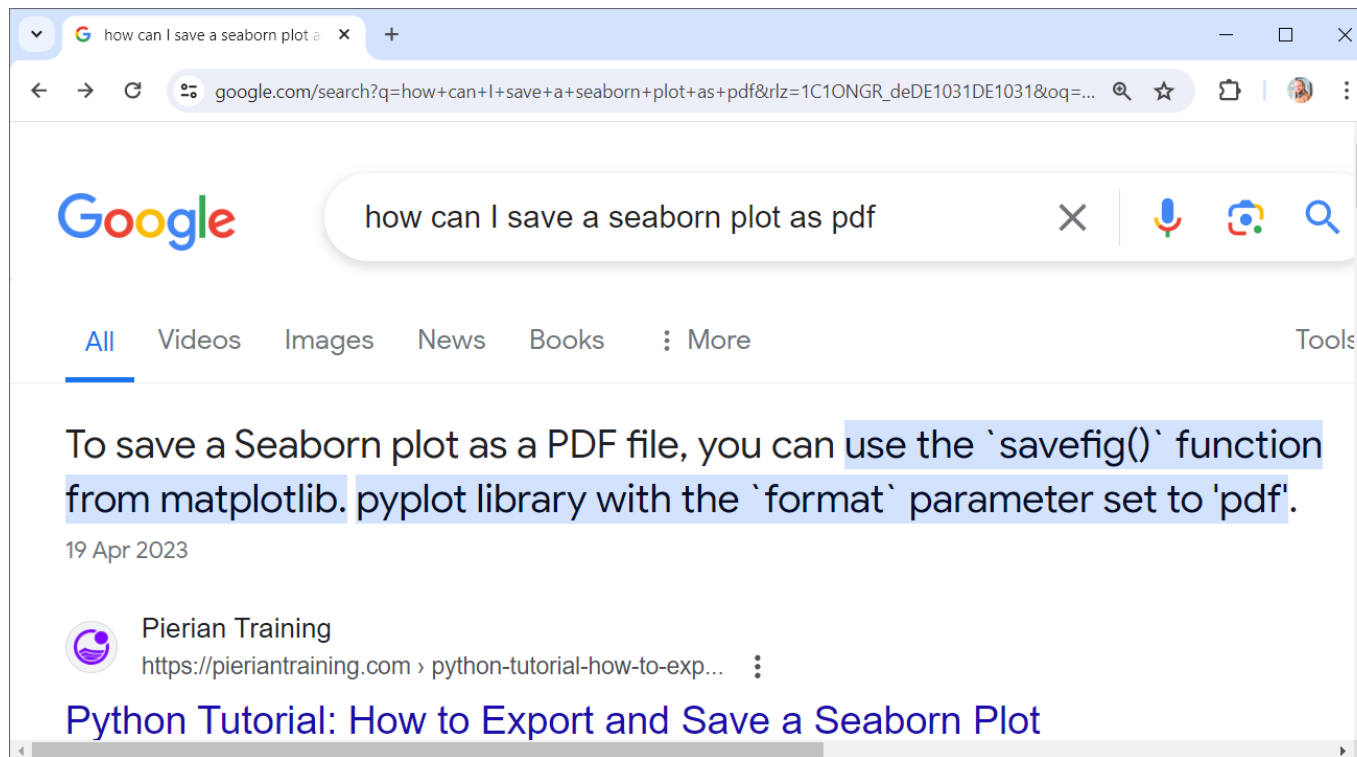
.. [0]

https://github.com/clEsperanto/pyclesperanto_prototype/blob/master/demo/segmentation/gauss_otsu_labeling.py

File: c:\users\haase\mambaforge\envs\bio399\lib\site-

The Python community

... is googling ... to end up on stackoverflow.com



The Image Science Community

- Developers of common image-analysis libraries answer your questions for free 😊

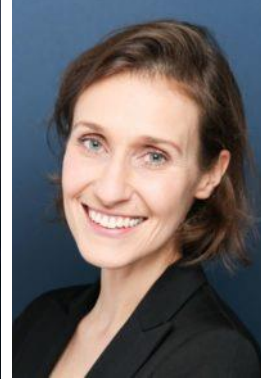
The screenshot shows a forum post on the image.sc website. The post is titled "Install issues with py-clesperanto-assistant" and is categorized under "Usage & Issues". It was posted by William Giang (@haesleinhuepf) on July 2021. The post describes a problem with installing the napari-pyclesperanto-assistant plugin (v0.9.3) for napari 0.4.10 on Windows 10. The user lists their system requirements: MSVC v142 - VS 2019 C++ x64/x86 build tools (Latest), Windows 10 SDK (10.0.19041.0), C++ CMake tools for Windows, Testing tools core features - Build Tools, and C++ AddressSanitizer. They mention that after creating and activating a conda environment with python=3.8, they can pip install "napari[all]" without issue. They also mention following the install instructions, downloading and installing a .whl file with OpenCL 1.2 and Python 3.8, and note that it was pyopencl-2021.2.3+cl12-cp38-cp38-win_amd64.whl instead of the (no longer available) 2020.3.1 version. The log for the napari-pyclesperanto plugin install suggests that it's trying to install 2021.2.1 and also suggests installing pybind11 and mako.

The screenshot shows a reply to the forum post. The reply is from Robert Haase (@haesleinhuepf), who is the maintainer of clij and clesperanto. He responds to the user's post, thanking them for reporting the issue. He mentions that he recently hit issues with recent pyopencl > 2021.2.1 and thus, limited that dependency. Maybe, that was a fault. He's considering removing this with the next minor release next week. He suggests trying installing using conda and provides the following commands:

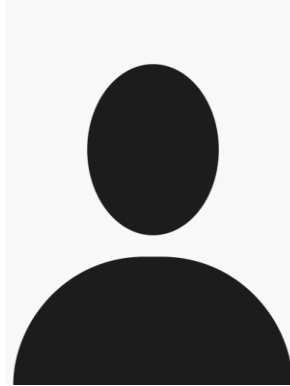
```
conda create --name bio11 python==3.8.5
conda activate bio11
conda install -c conda-forge pyopencl==2021.2.1
pip install napari-pyclesperanto-assistant
pip install napari[all]
```

He asks if this helps and signs off with "Cheers, Robert". The reply is marked as a "Solution" and has 2 likes. It was posted on July 2021.

Trainers & Supporters



Anja Neumann, Christian Martin, Dušan Prašević, Jan Ewald, Laura Žigutytė, Marie-Sophie von Braun, Matthias Täschner, Robert Haase



Mara Lampert, Susann Tilger, Nicole Spieß, Marvin Rosé

Acknowledgements

Big thanks to our funders

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung



Diese Maßnahme wird gefördert durch die Bundesregierung aufgrund eines Beschlusses des Deutschen Bundestages. Diese Maßnahme wird mitfinanziert durch Steuermittel auf der Grundlage des von den Abgeordneten des Sächsischen Landtags beschlossenen Haushaltes.