PoL
Physics of Life
TU Dresden

BIOTEC
Biotechnology Center
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Bio-image analysis

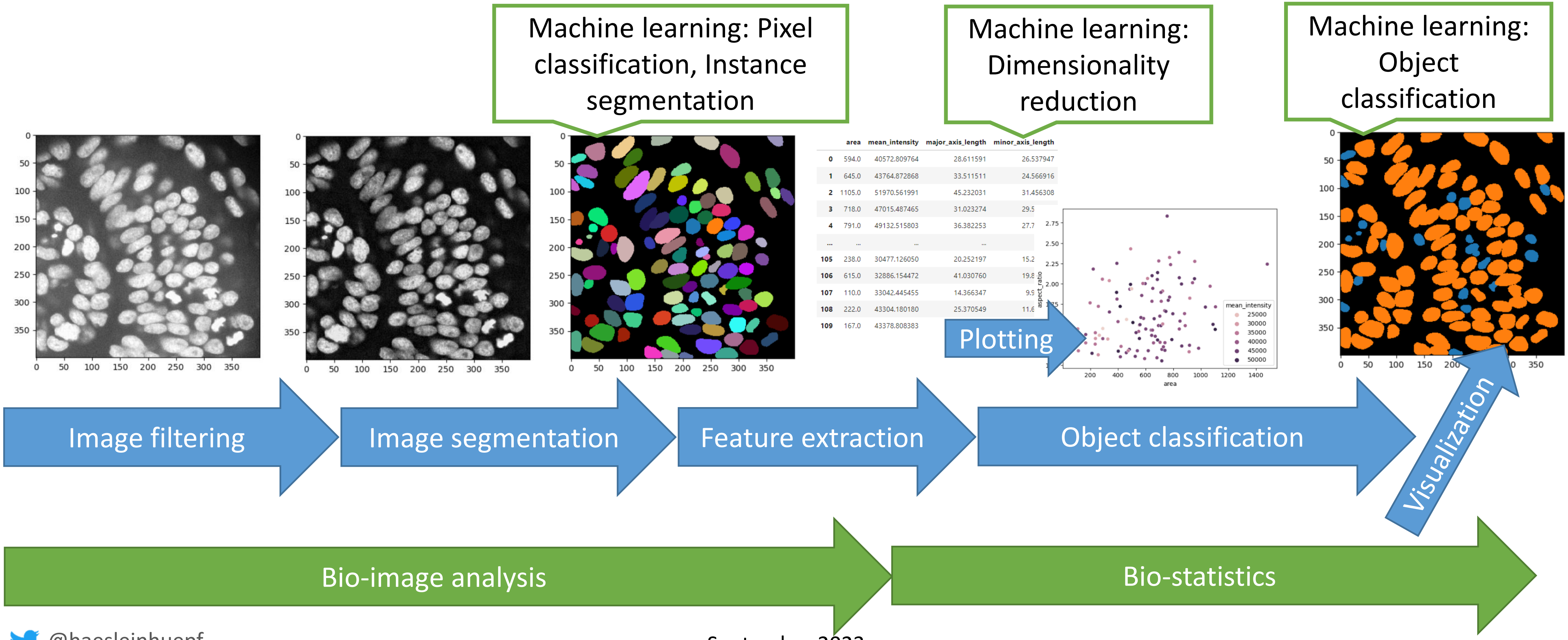
Robert Haase, Till Korten

@haesleinhuepf

September 2023

Lecture overview: Bio-image Analysis

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



Machine learning: Pixel classification, Instance segmentation

Machine learning: Dimensionality reduction

Machine learning: Object classification

	area	mean_intensity	major_axis_length	minor_axis_length
0	594.0	40572.809764	28.611591	26.537947
1	645.0	43764.872868	33.511511	24.566916
2	1105.0	51970.561991	45.232031	31.456308
3	718.0	47015.487465	31.023274	29.5
4	791.0	49132.515803	36.382253	27.7
...	...	...	...	
105	238.0	30477.126050	20.252197	15.2
106	615.0	32886.154472	41.030760	19.8
107	110.0	33042.445455	14.366347	9.9
108	222.0	43304.180180	25.370549	11.6
109	167.0	43378.808383		

Plotting

Image filtering

Image segmentation

Feature extraction

Object classification

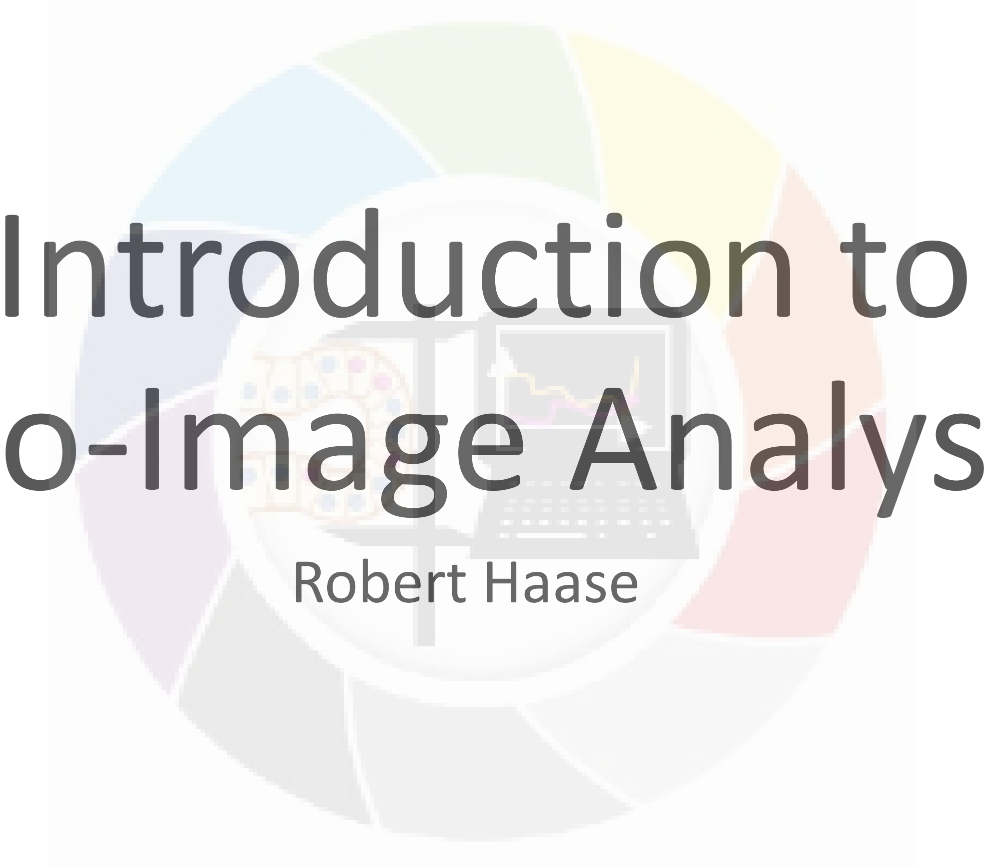
Visualization

Bio-image analysis

Bio-statistics

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN



Introduction to Bio-Image Analysis

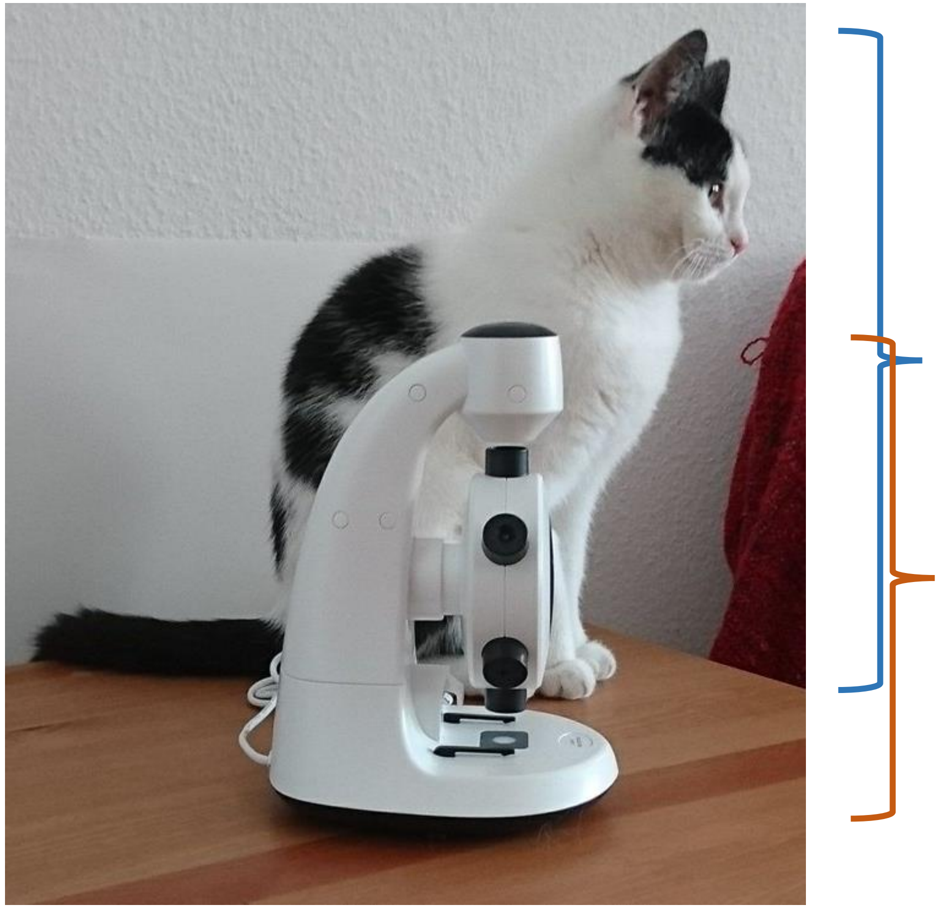
Robert Haase

@haesleinhuepf

September 2023

Quantitative bio-image analysis

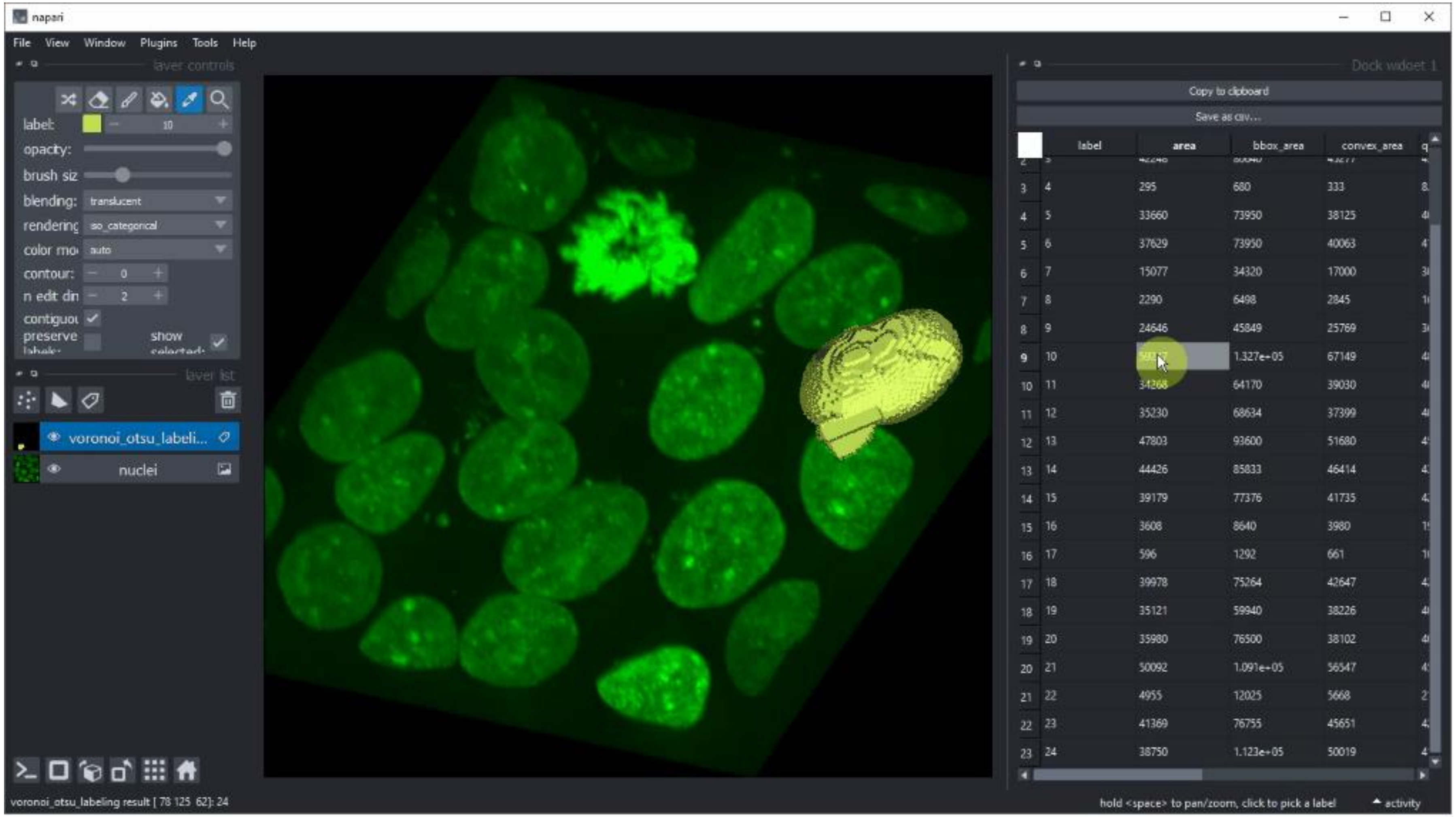
- Deriving quantitative information from images of biological samples taken with microscopes



cat height = 1.5 x microscope height

Quantitative bio-image analysis

- Deriving quantitative information from images of biological samples taken with microscopes + visualization



label: 10
opacity:
brush siz
blending: translucent
renderinc iso_categorical
color mo auto
contour: 0
n edit din 2
contiguou ✓
preserve labels: ☐
show selected: ✓

voronoi_otsu_labeli...
nuclei

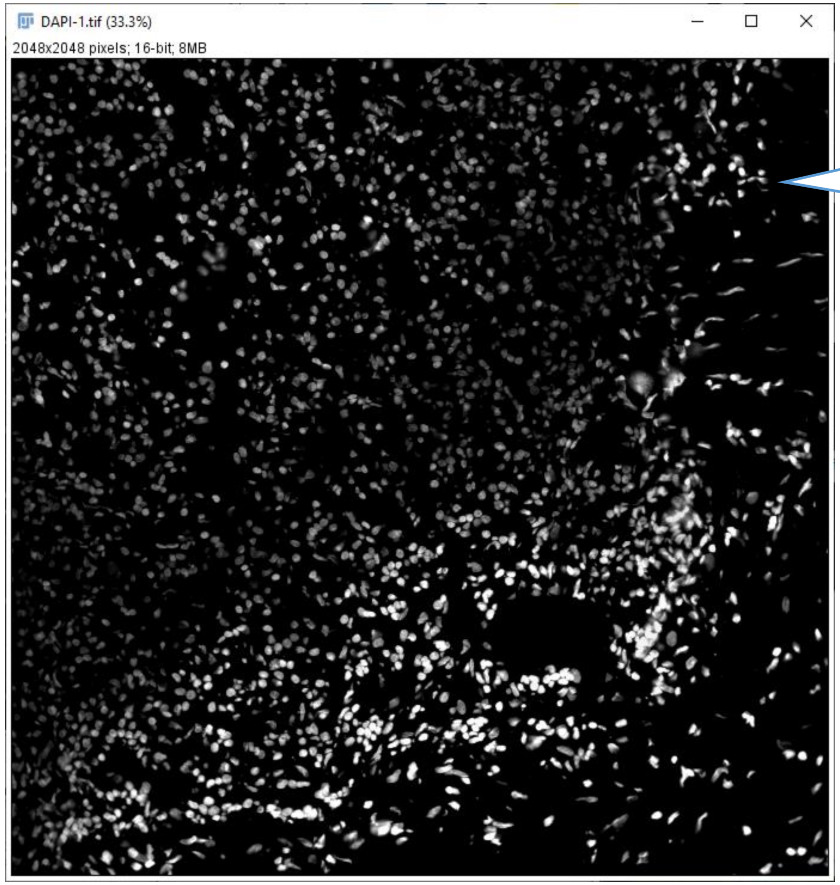
	label	area	bbox_area	convex_area	q
2	3	[illegible]	[illegible]	[illegible]	4
3	4	295	680	333	8.
4	5	33660	73950	38125	4
5	6	37629	73950	40063	4
6	7	15077	34320	17000	3
7	8	2290	6498	2845	1
8	9	24646	45849	25769	3
9	10	[illegible]	1.327e+05	67149	4
10	11	34268	64170	39030	4
11	12	35230	68634	37399	4
12	13	47803	93600	51680	4
13	14	44426	85833	46414	4
14	15	39179	77376	41735	4
15	16	3608	8640	3980	1
16	17	596	1292	661	1
17	18	39978	75264	42647	4
18	19	35121	59940	38226	4
19	20	35980	76500	38102	4
20	21	50092	1.091e+05	56547	4
21	22	4955	12025	5668	2
22	23	41369	76755	45651	4
23	24	38750	1.123e+05	50019	4

voronoi_otsu_labeling result [78 125 62]: 24

hold <space> to pan/zoom, click to pick a label

Objective bio-image analysis

- Measurements should be objective, not influenced by human interpretation

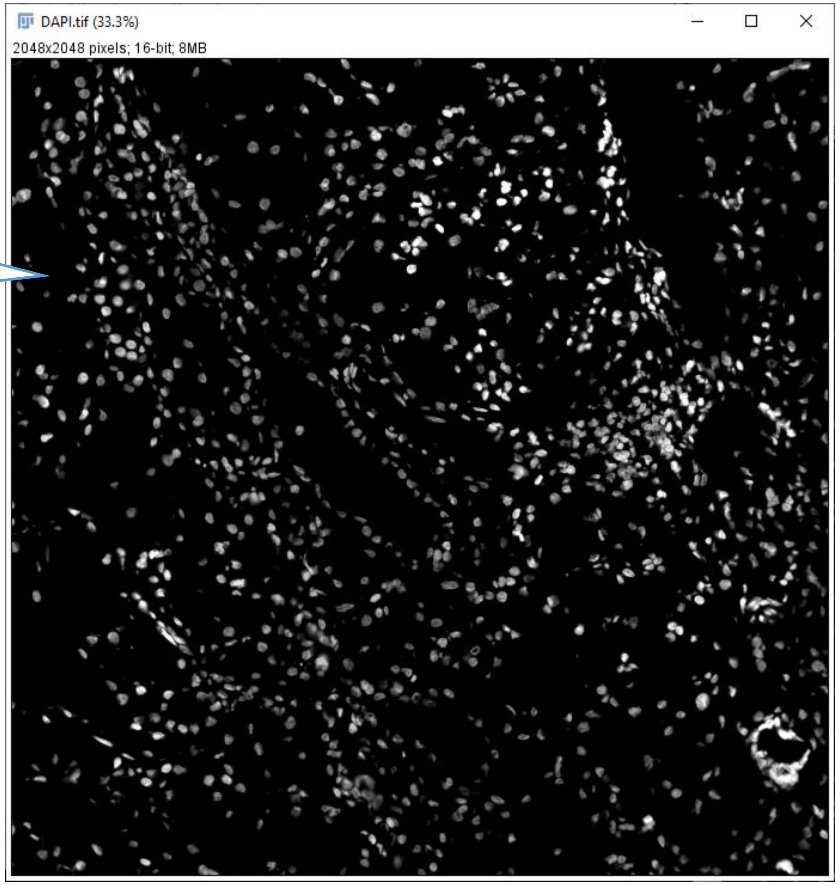


DAPI-1.tif (33.3%)

2048x2048 pixels; 16-bit; 8MB

Nuclei in this image are … _

… more dense than in this image.



DAPI.tif (33.3%)

2048x2048 pixels; 16-bit; 8MB

Use automation for less subjective analysis.

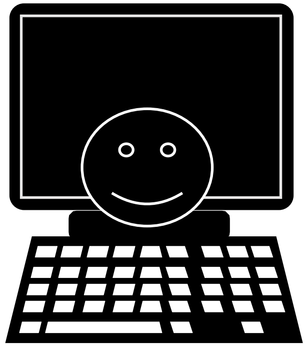
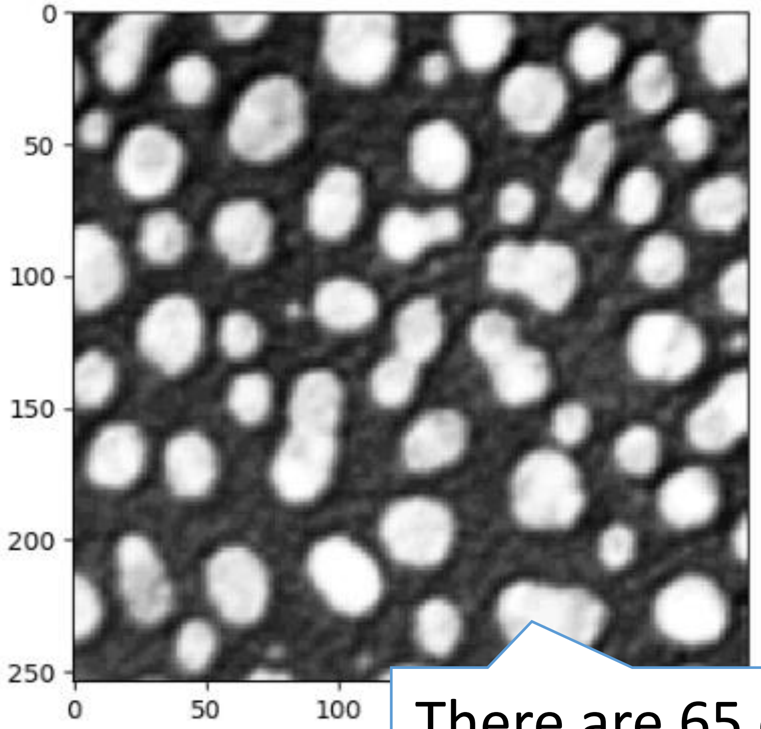


Image data source: Pascual-Reguant, Anna. (2021). Immunofluorescence staining of a human kidney (#2, peri-tumor area) obtained by MELC [Data set]. Zenodo. http://doi.org/10.5281/zenodo.4434462 licensed CC-BY 4.0

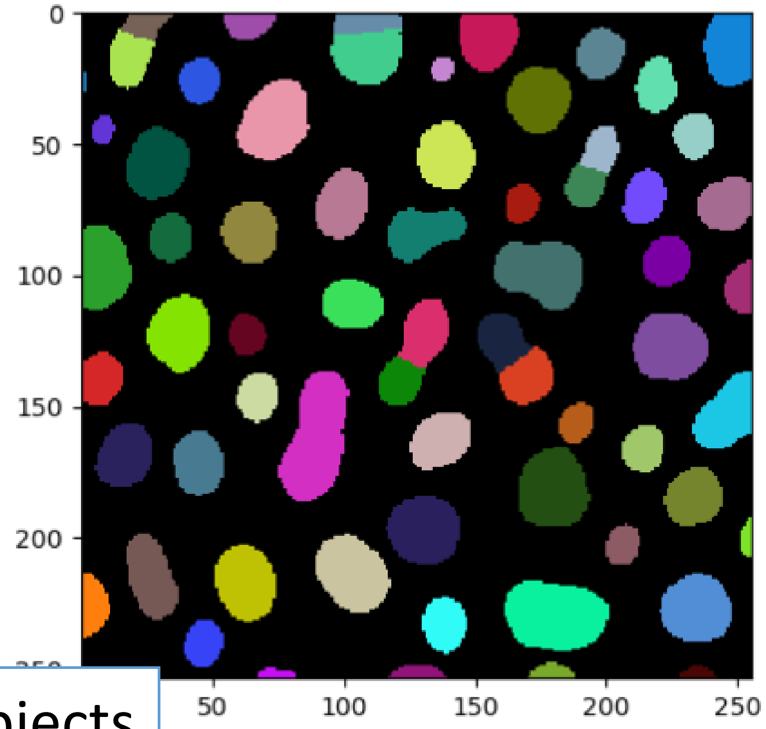
Reliable bio-image analysis

- Algorithms must be reliable (trustworthy).
- Visualization helps gaining trust in automated methods.

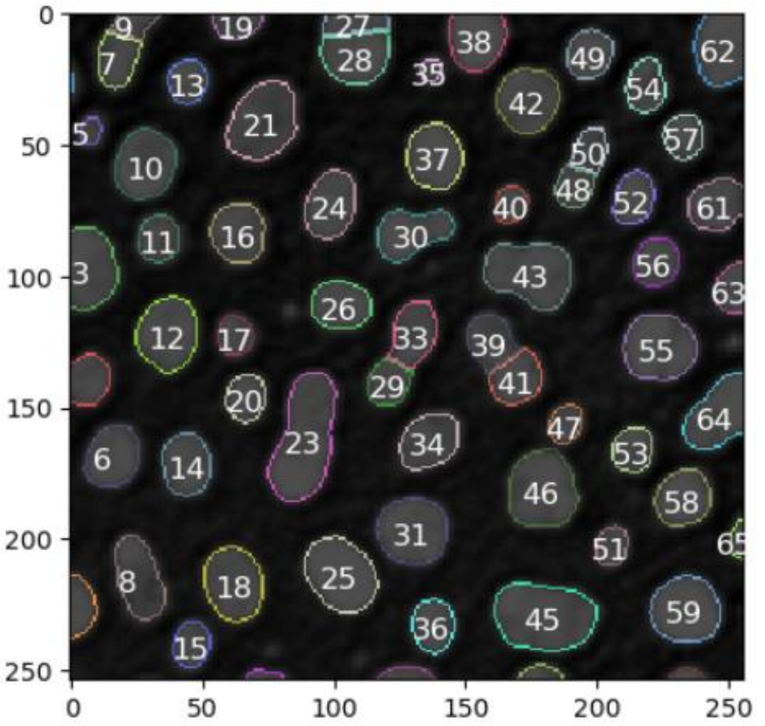
Original image



Label image



Overlay

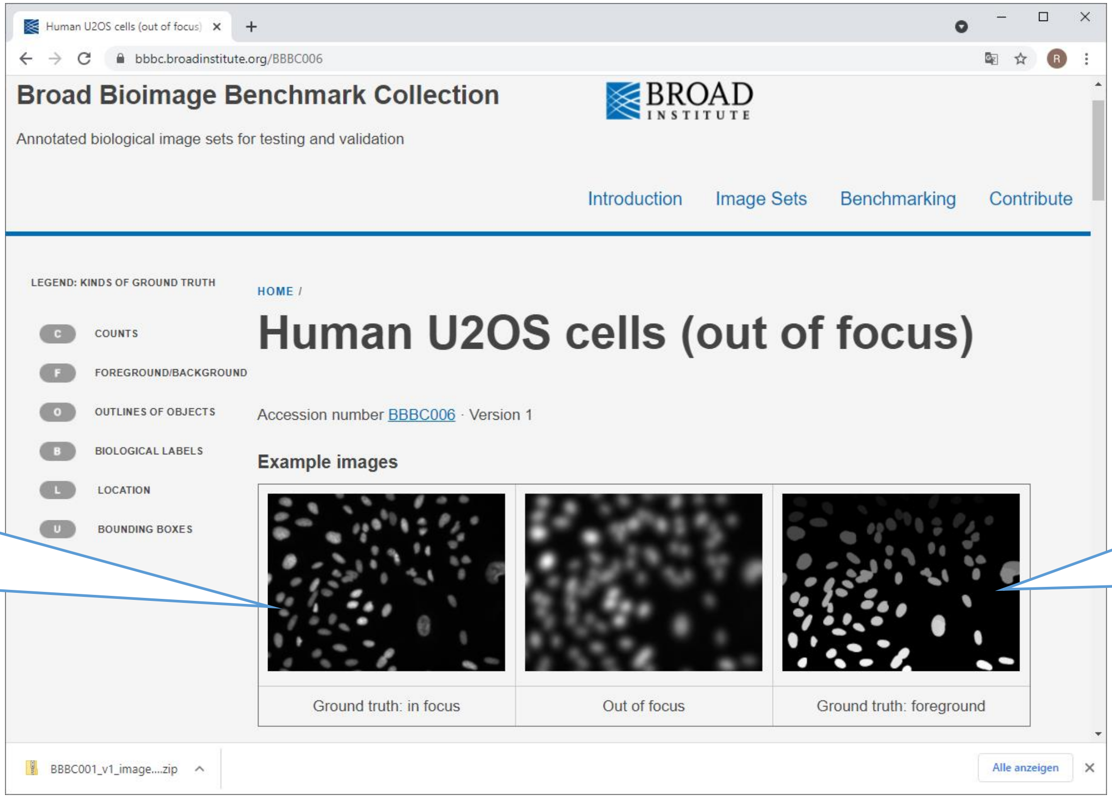


There are 65 objects in this image.

Source: M. Zoccoler & R. Haase licensed CC-BY https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/60_data_visualization/overlay_text_on_image.html

Reliable bio-image analysis

- Algorithms must be reliable (validated methods).
- Publicly available benchmark data sets allow to compare algorithms on common data.


Human U2OS cells (out of focus)
bbbc.broadinstitute.org/BBBC006

Broad Bioimage Benchmark Collection

Annotated biological image sets for testing and validation

Introduction
Image Sets
Benchmarking
Contribute

LEGEND: KINDS OF GROUND TRUTH

C	COUNTS
F	FOREGROUND/BACKGROUND
O	OUTLINES OF OBJECTS
B	BIOLOGICAL LABELS
L	LOCATION
U	BOUNDING BOXES

HOME /

Human U2OS cells (out of focus)

Accession number BBBC006 · Version 1

Example images

Ground truth: in focus | Out of focus | Ground truth: foreground

Original image data

“Ground truth” label images

https://bbbc.broadinstitute.org/BBBC006

Reproducible data analysis

- Allowing others to do your experiment again.

- “The image data was analyzed with Python.”

Can you reproduce what they did?

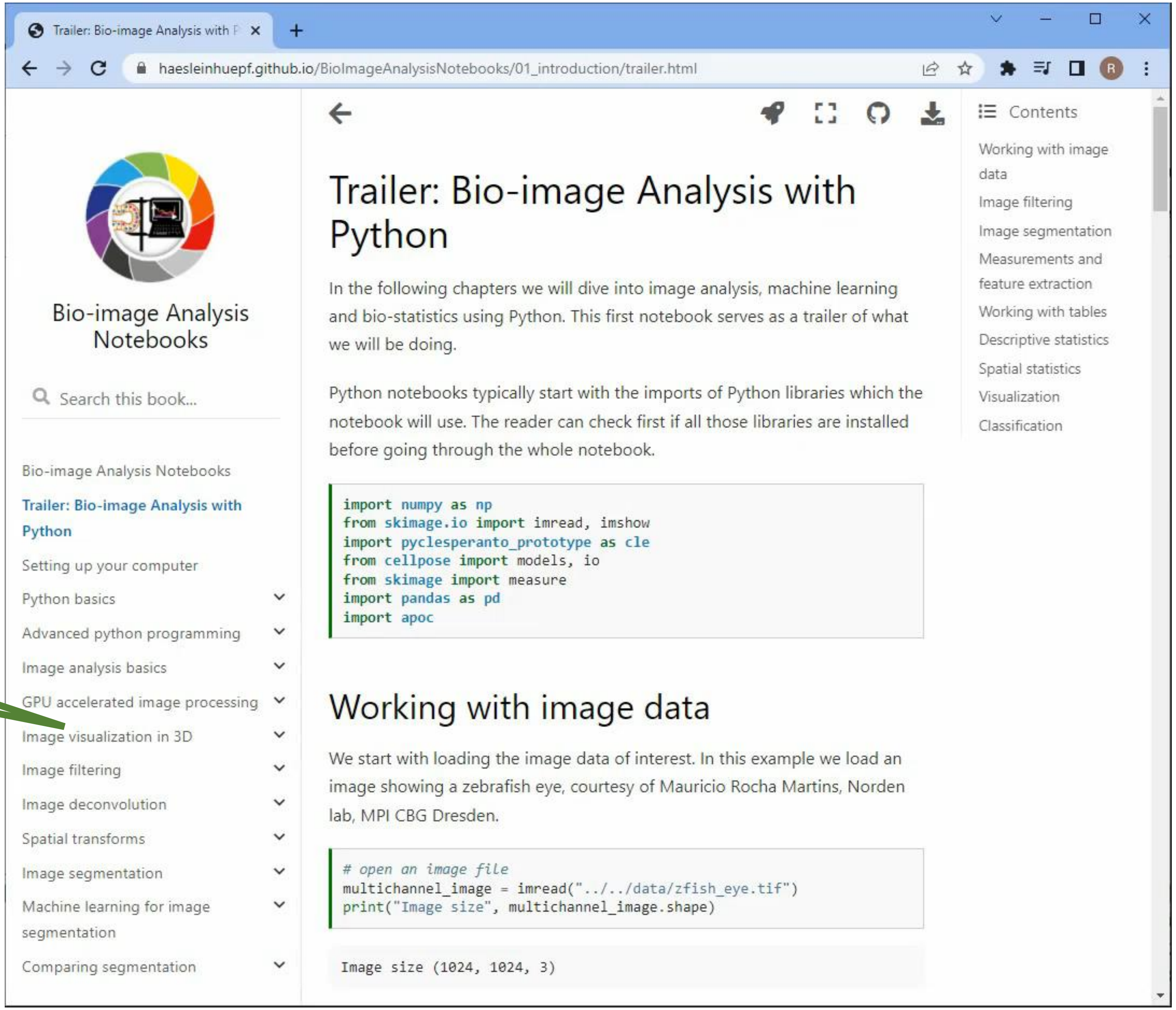
Reproducible bio-image analysis

- Allowing others to do your experiment again.

- “The image data was analyzed with Python.”

Can you reproduce what they did?

Can you reproduce what they did?


Trailer: Bio-image Analysis with P
haesleinhuepf.github.io/BioImageAnalysisNotebooks/01_introduction/trailer.html
Bio-image Analysis Notebooks
Search this book...
Bio-image Analysis Notebooks
Trailer: Bio-image Analysis with Python
Setting up your computer
Python basics
Advanced python programming
Image analysis basics
GPU accelerated image processing
Image visualization in 3D
Image filtering
Image deconvolution
Spatial transforms
Image segmentation
Machine learning for image segmentation
Comparing segmentation
Contents
Working with image data
Image filtering
Image segmentation
Measurements and feature extraction
Working with tables
Descriptive statistics
Spatial statistics
Visualization
Classification

Trailer: Bio-image Analysis with Python

In the following chapters we will dive into image analysis, machine learning and bio-statistics using Python. This first notebook serves as a trailer of what we will be doing.

Python notebooks typically start with the imports of Python libraries which the notebook will use. The reader can check first if all those libraries are installed before going through the whole notebook.

```
import numpy as np
from skimage.io import imread, imshow
import pyclesperanto_prototype as cle
from cellpose import models, io
from skimage import measure
import pandas as pd
import apoc
```

Working with image data

We start with loading the image data of interest. In this example we load an image showing a zebrafish eye, courtesy of Mauricio Rocha Martins, Norden lab, MPI CBG Dresden.

```
# open an image file
multichannel_image = imread("../../data/zfish_eye.tif")
print("Image size", multichannel_image.shape)
```

```
Image size (1024, 1024, 3)
```

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/01_introduction/trailer.html

Replicable bio-image analysis

- Others run the same analysis on their data and have consistent results / same conclusions.
- Can only be achieved if data analysis protocol was documented reproducibly.

- See also: *Replication crisis*
  - In Psychology (surveys)
  - In Medicine (clinical trials)
  - In Computer Science (executable code)
  - …

Open access, freely available online

Essay

Why Most Published Research Findings Are False

John P. A. Ioannidis

Summary

There is increasing concern that most current published research findings are false. The probability that a research claim is true may depend on study power and bias, the number of other studies on the same question, and, importantly, the ratio of true to no relationships among the relationships probed in each scientific field. In this framework, a research finding is less likely to be true when the studies conducted in a field are smaller; when effect sizes are smaller; when there is a greater number and lesser preselection of tested relationships; where there is greater flexibility in designs, definitions, outcomes, and analytical modes; when there is greater financial and other interest and prejudice; and when more teams are involved in a scientific field in chase of statistical significance. Simulations show that for most study designs and settings, it is more likely for

factors that influence this problem and some corollaries thereof.

Modeling the Framework for False Positive Findings

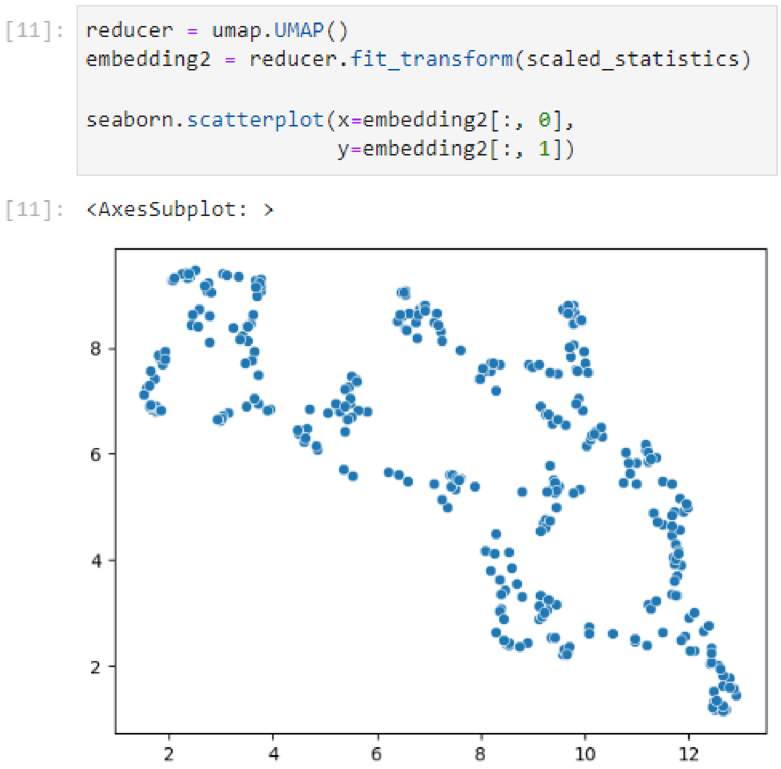
Several methodologists have pointed out [9–11] that the high rate of nonreplication (lack of confirmation) of research discoveries is a consequence of the convenient, yet ill-founded strategy of claiming conclusive research findings solely on the basis of a single study assessed by formal statistical significance, typically for a p-value less than 0.05. Research is not most appropriately represented and summarized by p-values, but, unfortunately, there is a widespread notion that medical research articles is characteristic of the field and can vary a lot depending on whether the field targets highly likely relationships or searches for only one or a few true relationships among thousands and millions of hypotheses that may be postulated. Let us also consider, for computational simplicity, circumscribed fields where either there is only one true relationship (among many that can be hypothesized) or the power is similar to find any of the several existing true relationships. The pre-study probability of a relationship being true is $R/(R + 1)$. The probability of a study finding a true relationship reflects the power $1 - \beta$ (one minus the Type II error rate). The probability of claiming a relationship when none truly exists reflects the Type I error rate, α. Assuming that c relationships are being probed in the field, the expected values of the 2×2 table are given in Table 1. After a research finding has been claimed based on

It can be proven that most claimed research findings are false.

https://en.wikipedia.org/wiki/Replication_crisis

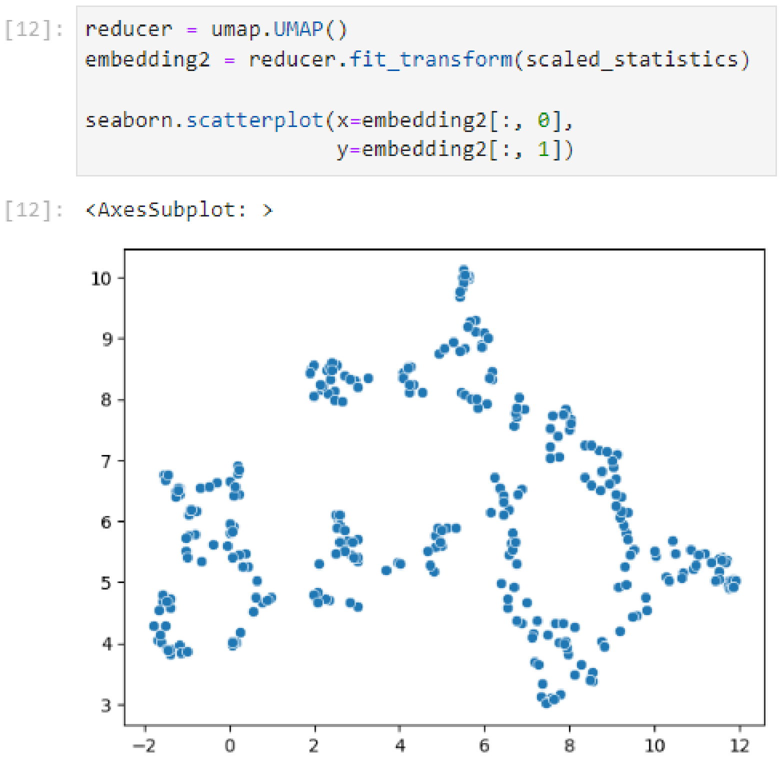
Repeatable data analysis

- In wet-lab experiments, samples may get destroyed while executing the experiment.
- Repeatability is a property of the experiment / algorithm. You cannot improve repeatability by better documentation.



```
[11]: reducer = umap.UMAP()
      embedding2 = reducer.fit_transform(scaled_statistics)

      seaborn.scatterplot(x=embedding2[:, 0],
                          y=embedding2[:, 1])
```

```
[11]: <AxesSubplot: >
```

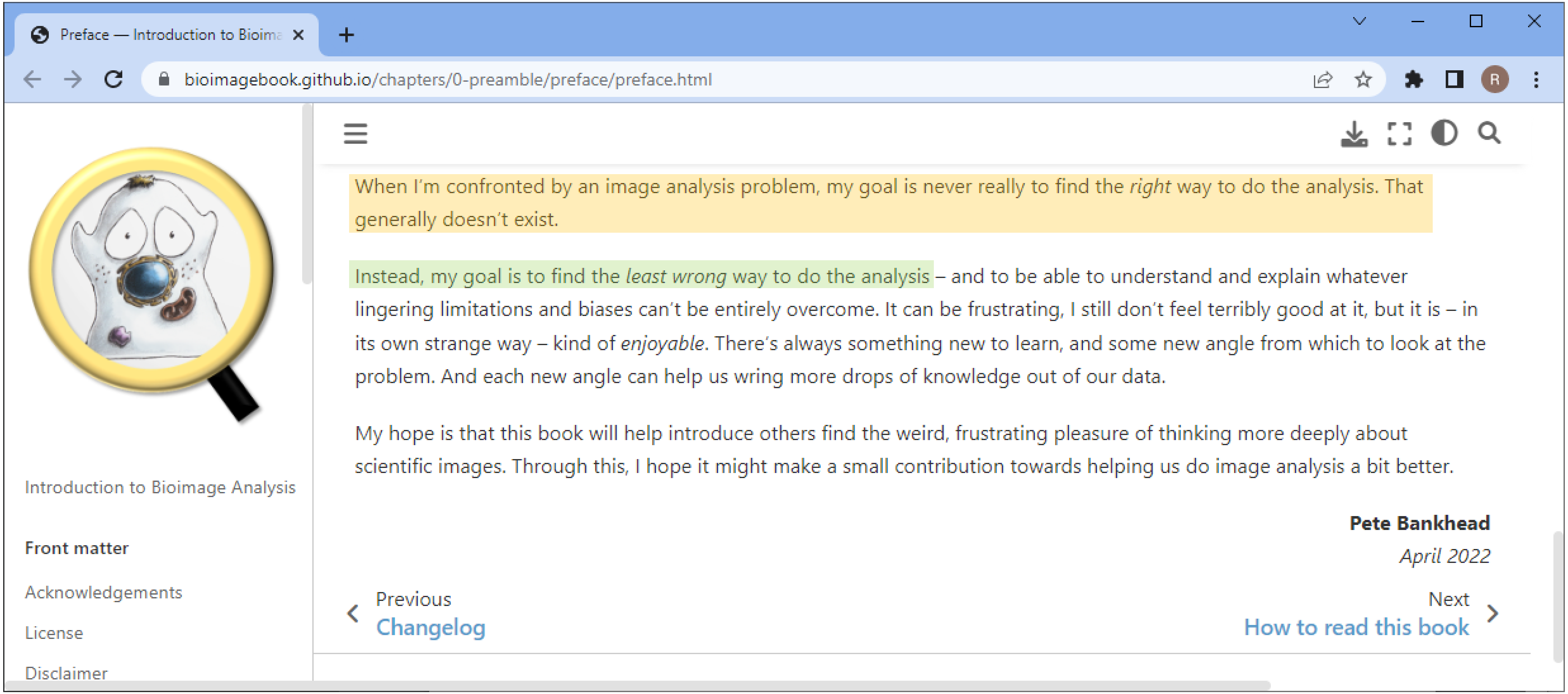


```
[12]: reducer = umap.UMAP()
      embedding2 = reducer.fit_transform(scaled_statistics)

      seaborn.scatterplot(x=embedding2[:, 0],
                          y=embedding2[:, 1])
```

```
[12]: <AxesSubplot: >
```

https://haesleinhuepf.github.io/BioImageAnalysisNotebooks/47_clustering/umap.html?highlight=umap#a-note-on-repeatability

Bio-image Analysis: good scientific practice



Preface — Introduction to Bioima
bioimagebook.github.io/chapters/0-preamble/preface/preface.html
Introduction to Bioimage Analysis
Front matter
Acknowledgements
License
Disclaimer

When I'm confronted by an image analysis problem, my goal is never really to find the *right* way to do the analysis. That generally doesn't exist.

Instead, my goal is to find the *least wrong* way to do the analysis – and to be able to understand and explain whatever lingering limitations and biases can't be entirely overcome. It can be frustrating, I still don't feel terribly good at it, but it is – in its own strange way – kind of *enjoyable*. There's always something new to learn, and some new angle from which to look at the problem. And each new angle can help us wring more drops of knowledge out of our data.

My hope is that this book will help introduce others find the weird, frustrating pleasure of thinking more deeply about scientific images. Through this, I hope it might make a small contribution towards helping us do image analysis a bit better.

Pete Bankhead
April 2022

Previous
Changelog

Next
How to read this book

@haesleinhuepf

September 2023 https://bioimagebook.github.io/chapters/0-preamble/preface/preface.html

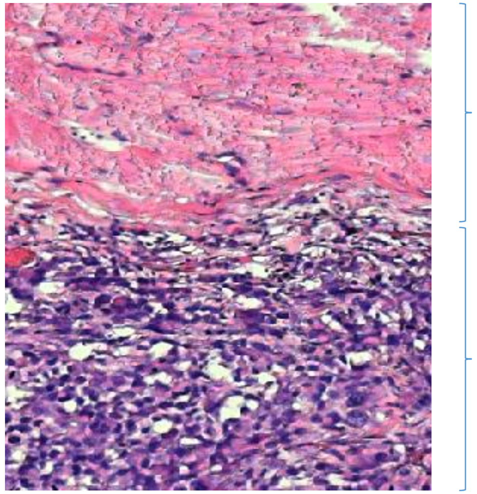
Introduction to bio-image analysis

- Bio-image analysis is supposed to be

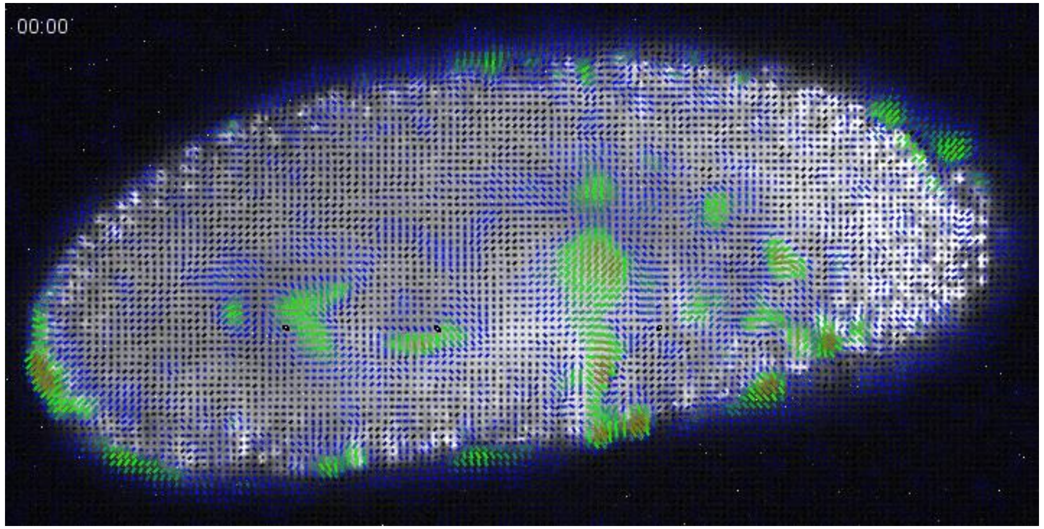
  - **Quantitative**
    - We derive numbers from images which describe physical properties of the observed sample.
  - **Objective**
    - The derived measurement does not depend on who did the measurement. The measurement is free of interpretation.
  - **Reliable (trustworthy / validated)**
    - We are confident that the measurement is describing what it is supposed to describe.
  - **Reproducible**
    - Enabling others to re-do the experiment. For this, documentation is crucial!
  - **Replicability**
    - Others *do* execute the same analysis, potentially on other data, and see consistent results.
  - **Repeatable**
    - We can do the same experiment twice under the *same conditions* and get the same measurements.

Common topics

- Typical questions bio-image analysts deal with
  - Is signal intensity different under varying conditions?
  - How many cells are in my image?
  - How high is cell density?
    - ➢ Bio-statistics / medicine / disease staging
  - How are different tissues characterized?
    - ➢ Machine learning



- Typical questions bio-image analysts struggle with
  - What force drives the observed processes?
  - What is the lineage tree of one particular cell?
  - Are observation A and observation B related?
  - Are structures observed in different color channels colocalized?



Hypothesis-driven quantitative biology

- Hypothesis: Cell shape can be influenced by modifying X.
- Null-Hypothesis: Circularity of modified cells is similar to cells in the control group.

- Sample preparation
- Imaging
- Cell segmentation
- Circularity measurement
- Statistics

Shall we use a different microscope?

Should we use a different segmentation algorithm?

Is circularity the right parameter to measure?

Hypothesis *generating* quantitative biology

- ~~Hypothesis: Cell shape can be influenced by modifying X.~~
- Question: Which image-derived parameter is influenced when modifying X?
  - Sample preparation
  - Imaging
  - Cell segmentation algorithm A, algorithm B, algorithm C
  - Measurement of circularity, solidity, elongation, extend, texture, intensity, topology …
  - Statistics

Which segmentation algorithms allow measurements that show a relationship with X?

Why?

Which parameter shows any relationship with X?

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Python Programming

Robert Haase

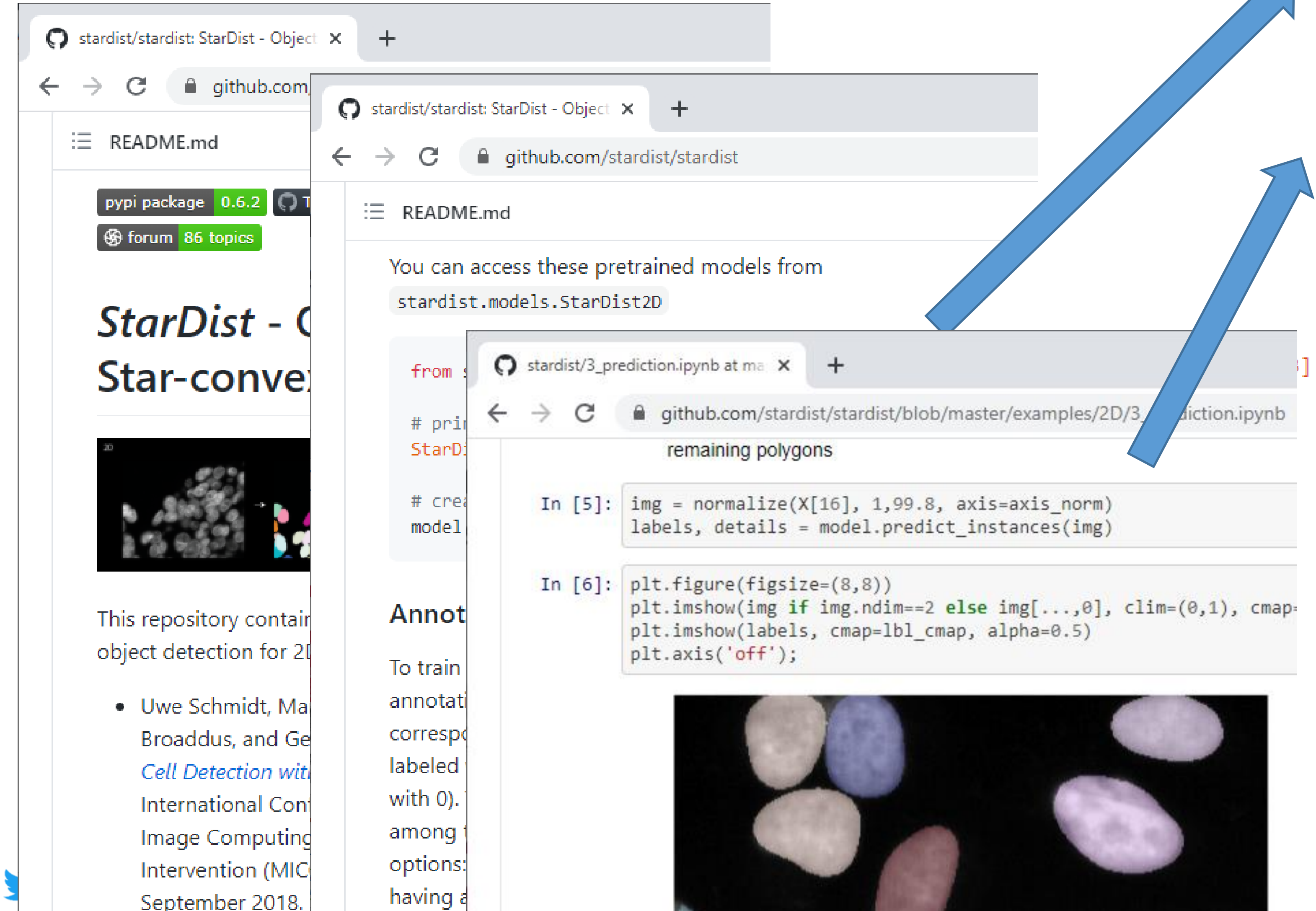
April 2023

@haesleinhuepf

Data science with python

- Why Python?

Because copy&paste works so great.



stardist/stardist: StarDist - Object

github.com/

README.md

pypi package 0.6.2

forum 86 topics

StarDist - O
Star-convex

This repository contai
object detection for 2D

- Uwe Schmidt, Ma
Broaddus, and Ge
Cell Detection wit
International Con
Image Computing
Intervention (MIC
September 2018.

stardist/stardist: StarDist - Object

github.com/stardist/stardist

README.md

You can access these pretrained models from `stardist.models.StarDist2D`

```
from s

# pri
StarD

# cre
model
```

Annot

To train
annotat
correspo
labeled
with 0).
among
options:
having a

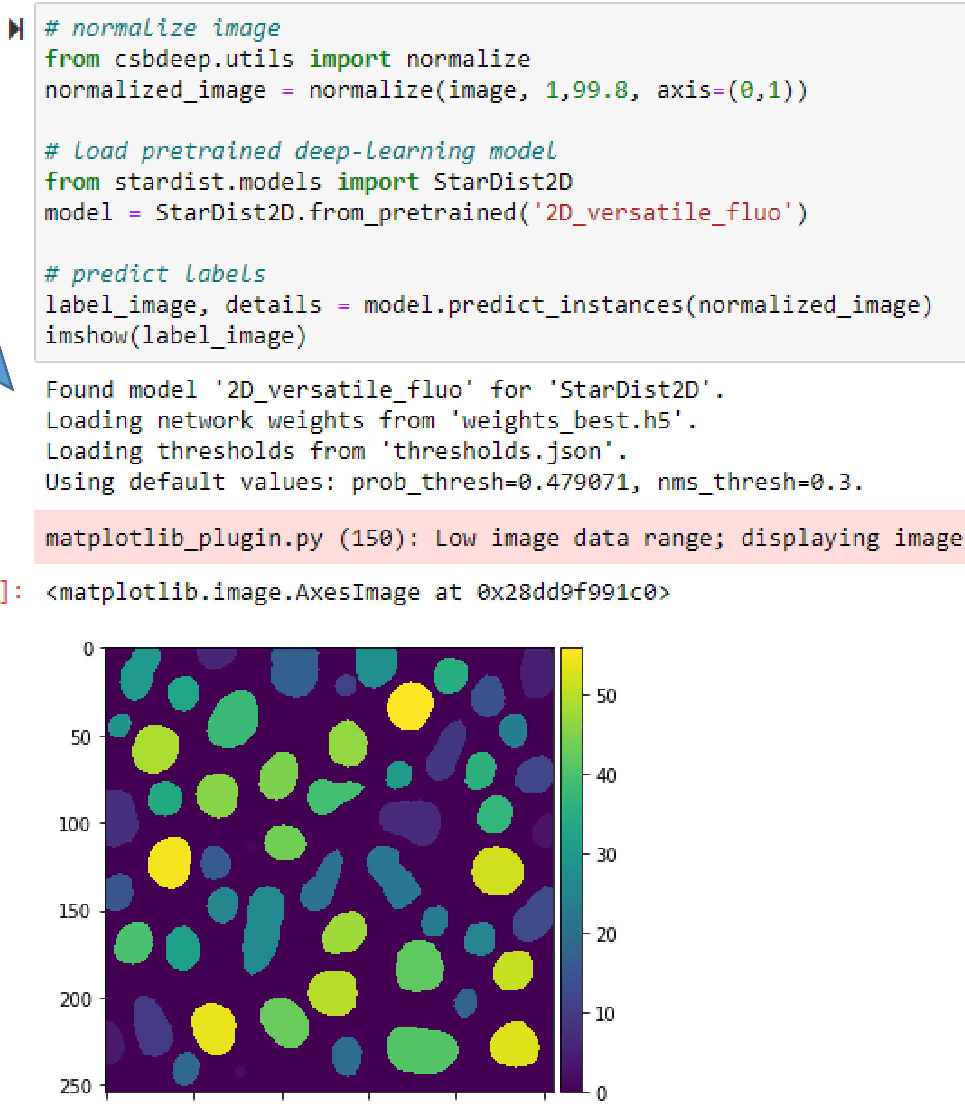
stardist/3_prediction.ipynb at ma

github.com/stardist/stardist/blob/master/examples/2D/3_ diction.ipynb

remaining polygons

```
In [5]: img = normalize(X[16], 1,99.8, axis=axis_norm)
        labels, details = model.predict_instances(img)

In [6]: plt.figure(figsize=(8,8))
        plt.imshow(img if img.ndim==2 else img[...,0], clim=(0,1), cmap=
        plt.imshow(labels, cmap=lbl_cmap, alpha=0.5)
        plt.axis('off');
```

In [3]:



```
# normalize image
from csbdeep.utils import normalize
normalized_image = normalize(image, 1,99.8, axis=(0,1))

# load pretrained deep-learning model
from stardist.models import StarDist2D
model = StarDist2D.from_pretrained('2D_versatile_fluo')

# predict labels
label_image, details = model.predict_instances(normalized_image)
imshow(label_image)
```

```
Found model '2D_versatile_fluo' for 'StarDist2D'.
Loading network weights from 'weights_best.h5'.
Loading thresholds from 'thresholds.json'.
Using default values: prob_thresh=0.479071, nms_thresh=0.3.
```

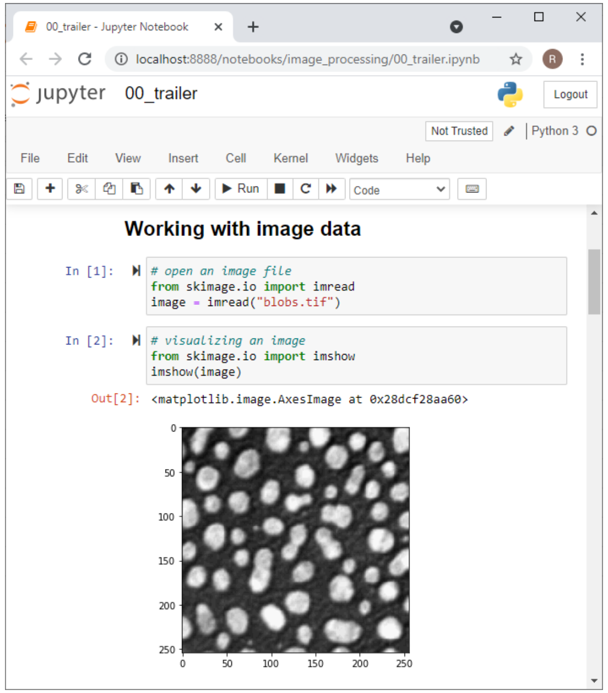
```
matplotlib_plugin.py (150): Low image data range; displaying image
```

```
]: <matplotlib.image.AxesImage at 0x28dd9f991c0>
```

https://github.com/stardist/stardist

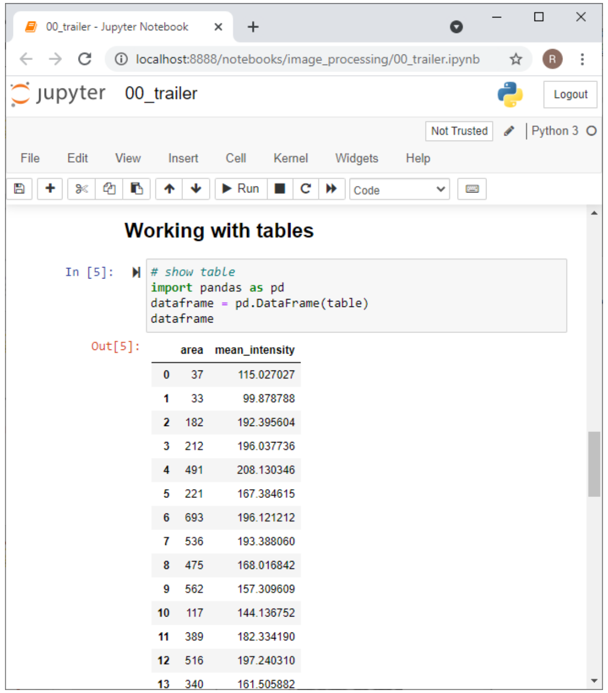
Python

- Major goals of [image] data analysis via scripting:
  - reproducible workflows for processing images (raw data) into quantitative information and visualizing biological properties.
  - automation
  - Sharing code, knowledge
  - Prevent reinventing the wheel



Working with image data

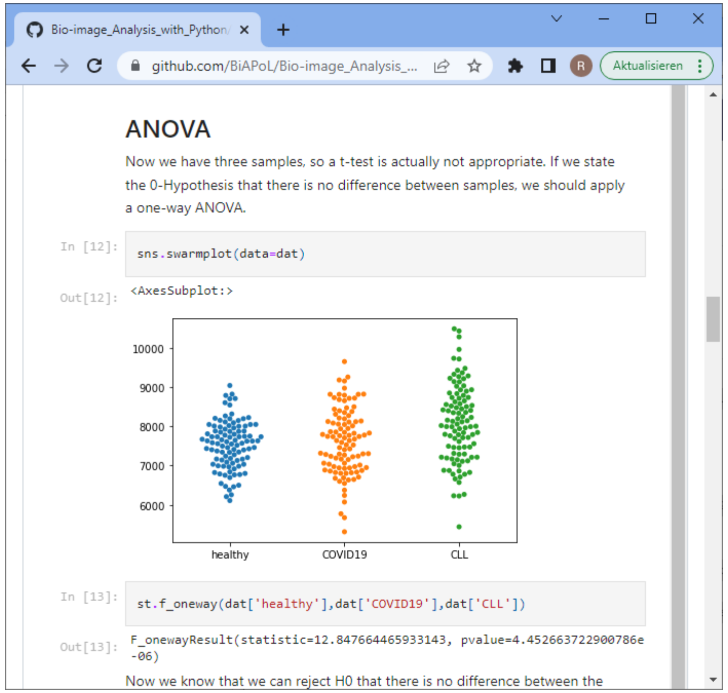
```
In [1]:  # open an image file
         from skimage.io import imread
         image = imread("blobs.tif")

In [2]:  # visualizing an image
         from skimage.io import imshow
         imshow(image)

Out[2]:  <matplotlib.image.AxesImage at 0x28dcf28aa60>
```



Working with tables

```
In [5]:  # show table
         import pandas as pd
         dataframe = pd.DataFrame(table)
         dataframe

Out[5]:
```

	area	mean_intensity
0	37	115.027027
1	33	99.878788
2	182	192.395604
3	212	196.037736
4	491	208.130346
5	221	167.384615
6	693	196.121212
7	536	193.388060
8	475	168.016842
9	562	157.309609
10	117	144.136752
11	389	182.334190
12	516	197.240310
13	340	161.505882


Bio-image_Analysis_with_Python/
github.com/BiAPoL/Bio-image_Analysis_...
Aktualisieren
10000
9000
8000
7000
6000
healthy
COVID19
CLL

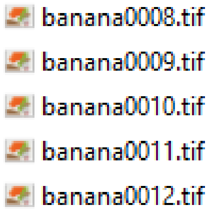
ANOVA

Now we have three samples, so a t-test is actually not appropriate. If we state the 0-Hypothesis that there is no difference between samples, we should apply a one-way ANOVA.

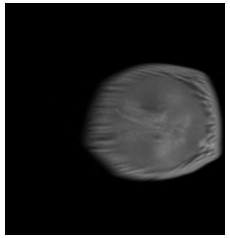
```
In [12]:  sns.swarmplot(data=dat)

Out[12]:  <AxesSubplot:>

In [13]:  st.f_oneway(dat['healthy'],dat['COVID19'],dat['CLL'])

Out[13]:  F_onewayResult(statistic=12.847664465933143, pvalue=4.452663722900786e
          -06)
```

Now we know that we can reject H0 that there is no difference between the

How to eat a banana the computational way using Python

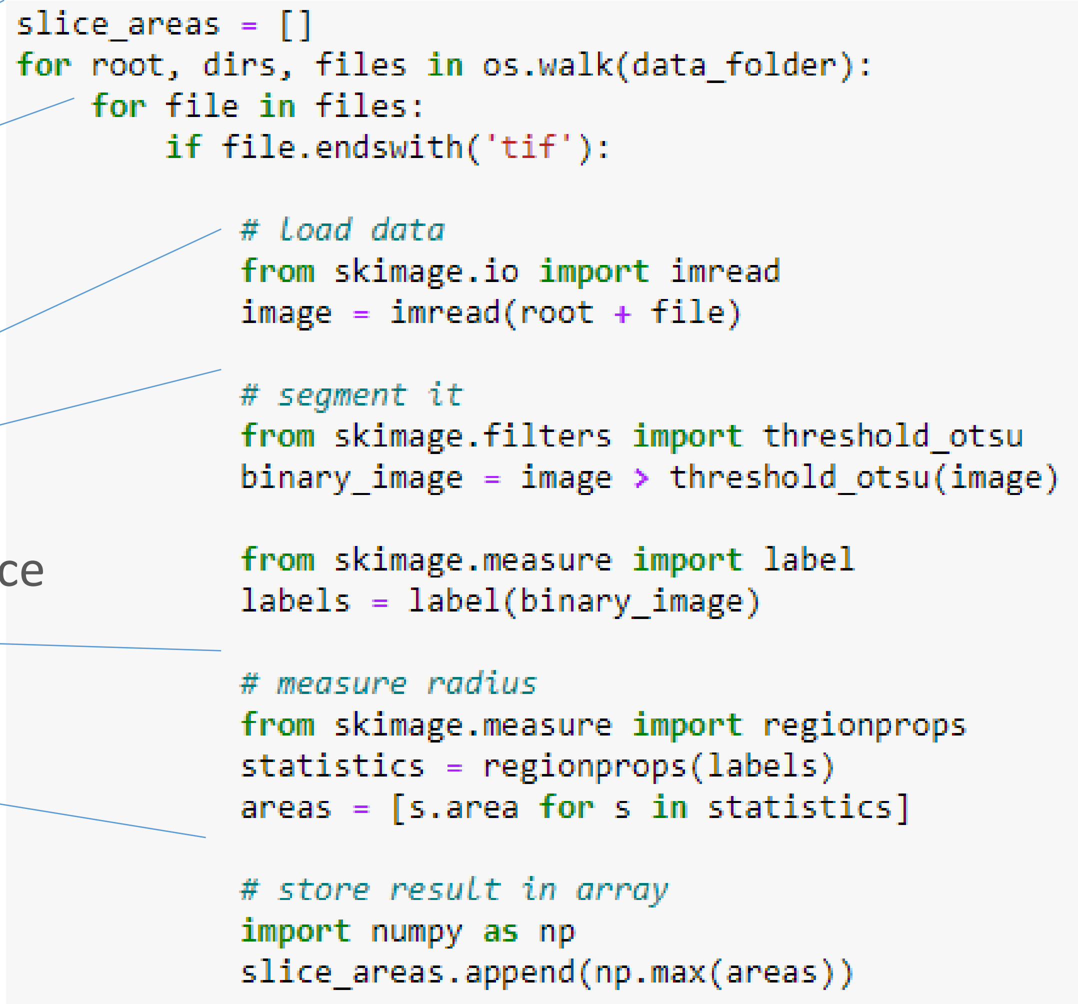


banana0008.tif
banana0009.tif
banana0010.tif
banana0011.tif
banana0012.tif



- Remove shell
- Repeat until nothing left:
    - Take a bite
    - Chew
    - Swallow
- Digest

- Access folder
- Repeat for all images:
    - Open an image file
    - Segment the banana slice
    - Analyse it
- Save measurements



```
slice_areas = []
for root, dirs, files in os.walk(data_folder):
    for file in files:
        if file.endswith('tif'):

            # load data
            from skimage.io import imread
            image = imread(root + file)

            # segment it
            from skimage.filters import threshold_otsu
            binary_image = image > threshold_otsu(image)

            from skimage.measure import label
            labels = label(binary_image)

            # measure radius
            from skimage.measure import regionprops
            statistics = regionprops(labels)
            areas = [s.area for s in statistics]

            # store result in array
            import numpy as np
            slice_areas.append(np.max(areas))
```

Image data source: Nasreddin Abolmaali, TU Dresden

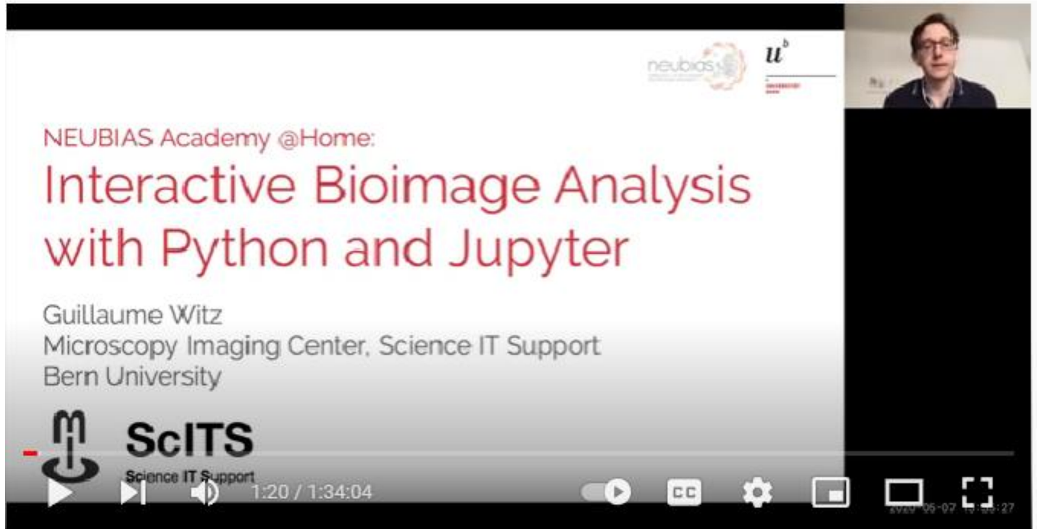
# Comments

Comments should contain additional information such as

- User documentation
  - What does the program do?
  - How can this program be used?
- Your name / institute in case a reader has a question
- Comment why things are done.
- Do not comment what is written in the code already!

```
#
# This program sums up two numbers.
#
# Usage:
# * Run it in Python 3.8
#
# Author: Robert Haase, PoL TUD
#         Robert.haase@tu-dresden.de
# April 2021

# initialise program
a = 1
b = 2.5

# run complicated algorithm
final_result = a + b

#~~ print the final result~~
print( final_result )
```

More resources


NEUBIAS Academy @Home:

Interactive Bioimage Analysis with Python and Jupyter

Guillaume Witz
Microscopy Imaging Center, Science IT Support
Bern University

Guillaume Witz, NEUBIAS Academy 2020

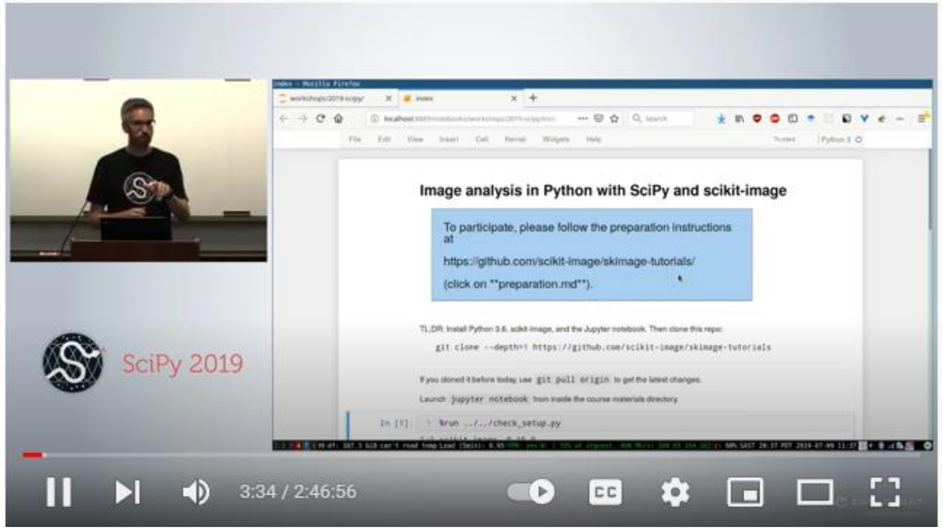


Image analysis in Python with SciPy and scikit-image

To participate, please follow the preparation instructions at

https://github.com/scikit-image/skimage-tutorials/

(click on "preparation.md").

TL;DR: Install Python 3.6, scikit-image, and the Jupyter notebook. Then clone this repo:

```
git clone --depth=1 https://github.com/scikit-image/skimage-tutorials
```

If you cloned it before today, use `git pull origin` to get the latest changes.

Launch `jupyter notebook` from inside the course materials directory.

```
In [1]:  %run ../../check_setup.py
```

Stéfan van der Walt, Juan Nunez-Iglesias, SciPy 2019

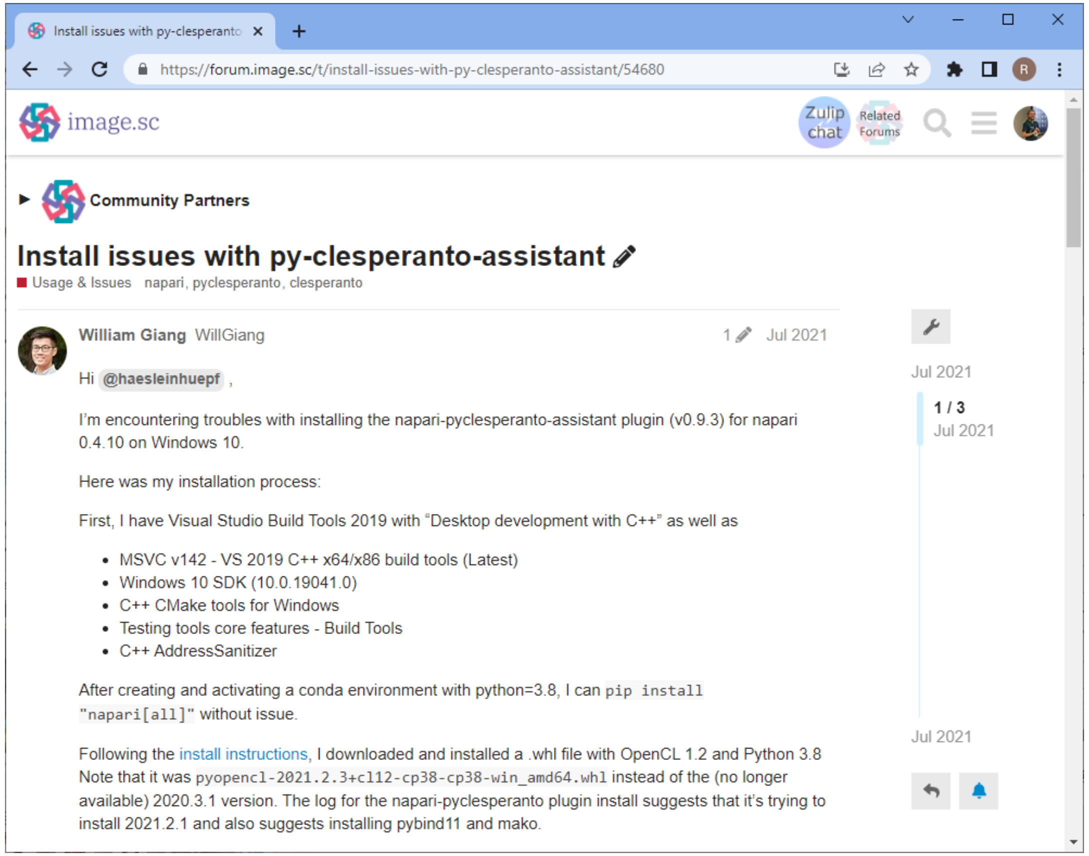
Sreenivas Bhattiprolu, Python for Microscopists @Youtube 2019-…

Watch more:

- https://www.youtube.com/watch?v=2KF8vBrp3Zw
- https://www.youtube.com/watch?v=d1CIV9irQAY
- https://www.youtube.com/watch?v=X_pCiVQ4c4E

The Image Science Community

- Ask your question online and an expert will likely reply the same day ☺


Install issues with py-clesperanto
https://forum.image.sc/t/install-issues-with-py-clesperanto-assistant/54680
image.sc
Zulip chat
Related Forums
Community Partners
1 / 3
Jul 2021
Jul 2021
Jul 2021

Install issues with py-clesperanto-assistant

Usage & Issues napari, pyclesperanto, clesperanto

William Giang WillGiang 1 Jul 2021

Hi @haesleinhuepf ,

I'm encountering troubles with installing the napari-pyclesperanto-assistant plugin (v0.9.3) for napari 0.4.10 on Windows 10.

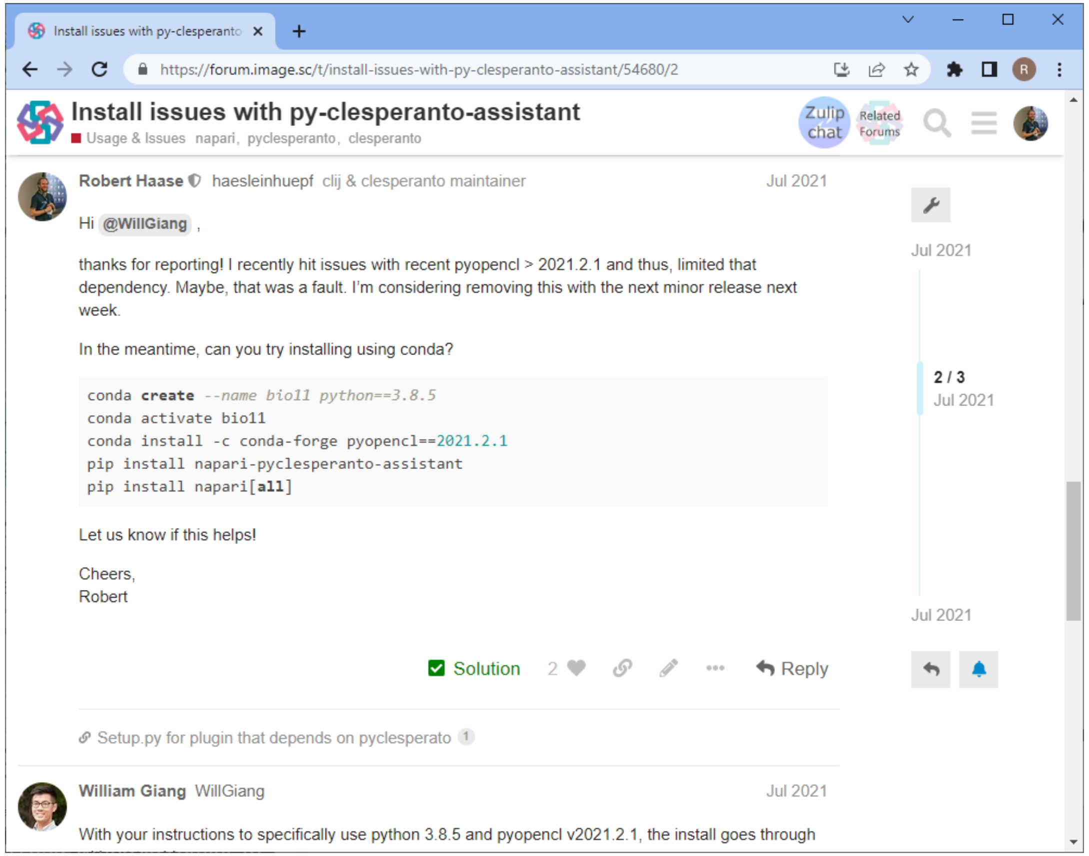
Here was my installation process:

First, I have Visual Studio Build Tools 2019 with "Desktop development with C++" as well as

- 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
- C++ AddressSanitizer

After creating and activating a conda environment with python=3.8, I can `pip install "napari[all]"` without issue.

Following the install instructions, I downloaded and installed a .whl file with OpenCL 1.2 and Python 3.8 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.


Install issues with py-clesperanto
https://forum.image.sc/t/install-issues-with-py-clesperanto-assistant/54680/2
Zulip chat
Related Forums
Jul 2021
2 / 3
Jul 2021
Jul 2021
Solution
2
Reply

Install issues with py-clesperanto-assistant

Usage & Issues napari, pyclesperanto, clesperanto

Robert Haase haesleinhuepf clij & clesperanto maintainer Jul 2021

Hi @WillGiang ,

thanks for reporting! I recently hit issues with recent pyopencl > 2021.2.1 and thus, limited that dependency. Maybe, that was a fault. I'm considering removing this with the next minor release next week.

In the meantime, can you try installing using conda?

```
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]
```

Let us know if this helps!

Cheers,
Robert

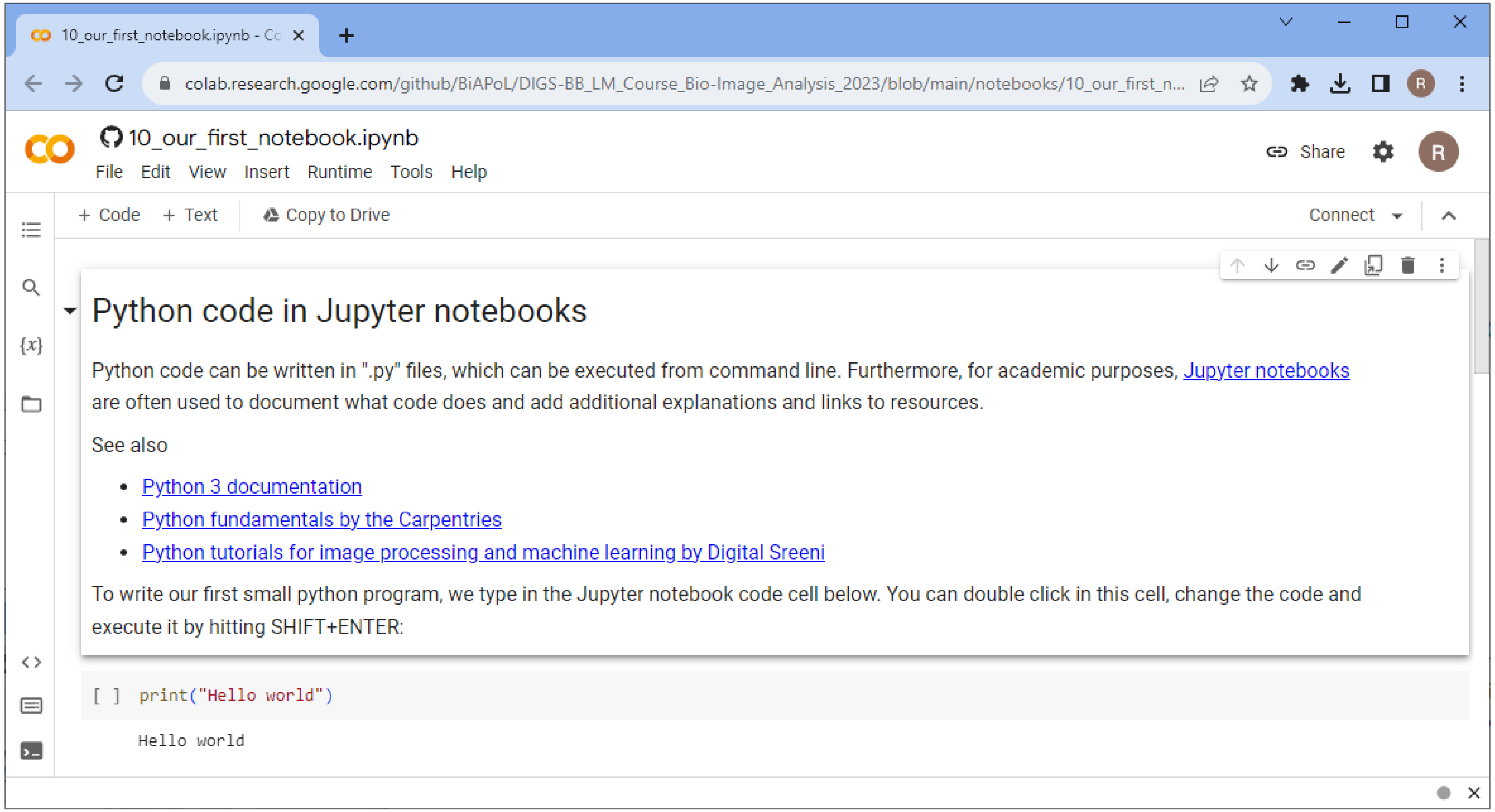
Setup.py for plugin that depends on pyclesperato 1

William Giang WillGiang Jul 2021

With your instructions to specifically use python 3.8.5 and pyopencl v2021.2.1, the install goes through

Google Colab

- Why Google Colab?
  - Great for teaching
  - Great for [remote] collaboration
  - No installation necessary
- Why not Google Colab?
  - Processed data must be available via the internet
- Alternatives:
  - Local mambaforge installation
  - Jupyter-lab at the compute center (ZIH)


10_our_first_notebook.ipynb - Co
colab.research.google.com/github/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023/blob/main/notebooks/10_our_first_n...
10_our_first_notebook.ipynb
File Edit View Insert Runtime Tools Help
Share
R
+ Code + Text
Copy to Drive
Connect

Python code in Jupyter notebooks

Python code can be written in ".py" files, which can be executed from command line. Furthermore, for academic purposes, Jupyter notebooks are often used to document what code does and add additional explanations and links to resources.

See also

- Python 3 documentation
- Python fundamentals by the Carpentries
- Python tutorials for image processing and machine learning by Digital Sreeni

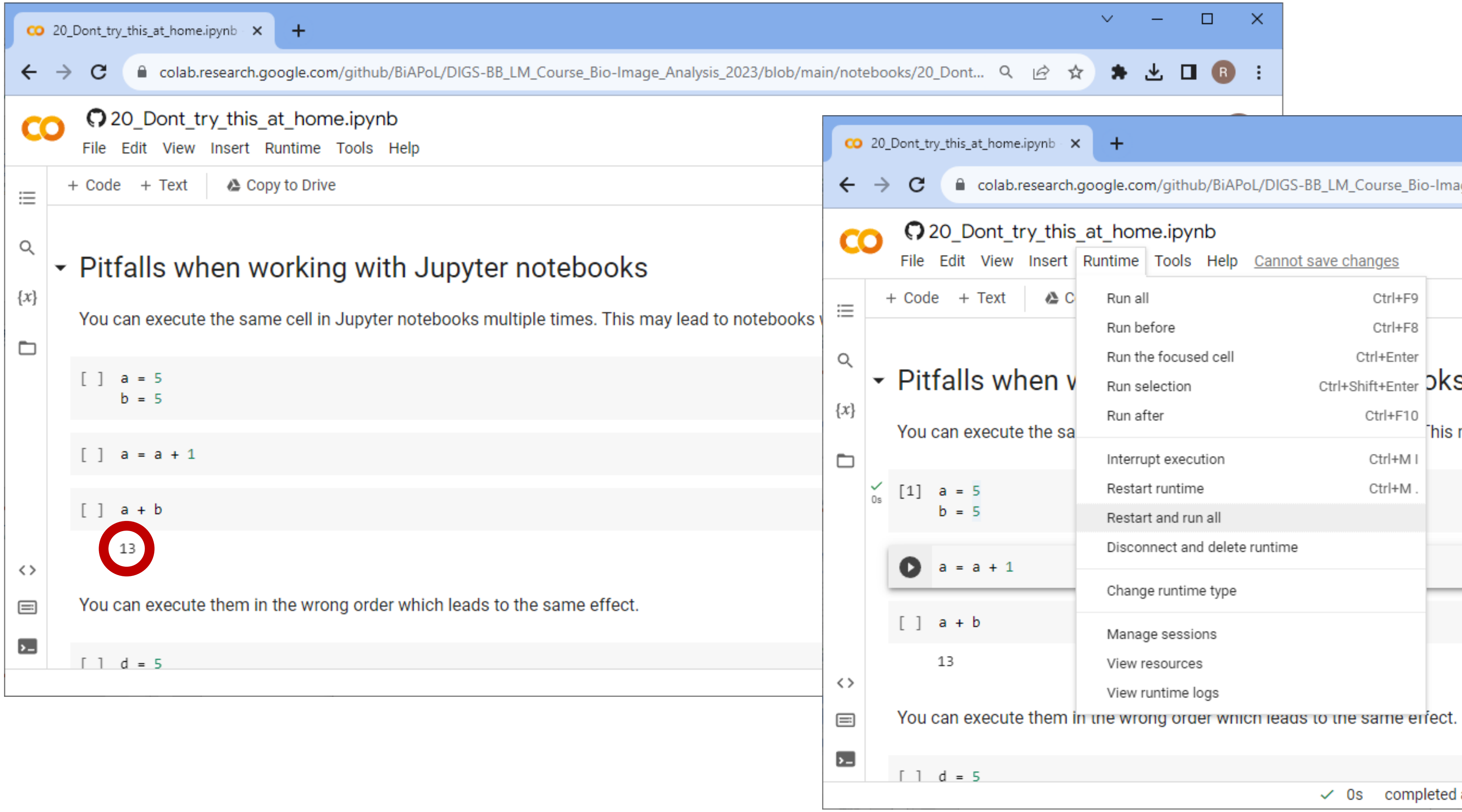
To write our first small python program, we type in the Jupyter notebook code cell below. You can double click in this cell, change the code and execute it by hitting SHIFT+ENTER:

```
[ ] print("Hello world")
```

```
Hello world
```

Pitfalls when working with notebooks

- Make sure to run cells in order
- Before finishing a session, click on *Runtime > Restart and run all*



20_Dont_try_this_at_home.ipynb
colab.research.google.com/github/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023/blob/main/notebooks/20_Dont...
File Edit View Insert Runtime Tools Help
+ Code + Text Copy to Drive
Cannot save changes
Run all Ctrl+F9
Run before Ctrl+F8
Run the focused cell Ctrl+Enter
Run selection Ctrl+Shift+Enter
Run after Ctrl+F10
Interrupt execution Ctrl+M I
Restart runtime Ctrl+M .
Restart and run all
Disconnect and delete runtime
Change runtime type
Manage sessions
View resources
View runtime logs
0s completed

Pitfalls when working with Jupyter notebooks

You can execute the same cell in Jupyter notebooks multiple times. This may lead to notebooks

```
[ ] a = 5
    b = 5
```

```
[ ] a = a + 1
```

```
[ ] a + b
    13
```

You can execute them in the wrong order which leads to the same effect.

```
[ ] d = 5
```

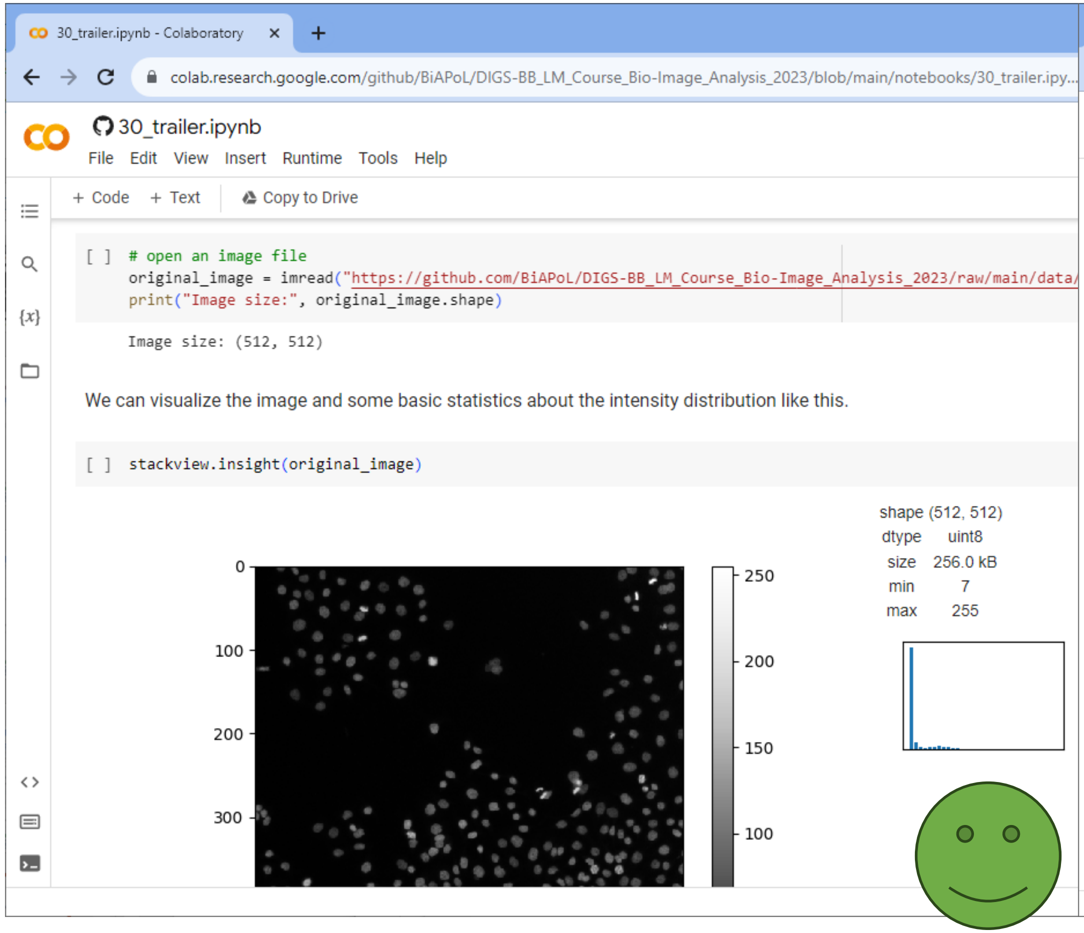
```
[1] a = 5
    b = 5
```

```
a = a + 1
```

```
[ ] a + b
    13
```

Reproducible science

- Python Notebooks allow executing a minimal set of code and showing intermediate results.


30_trailer.ipynb - Colaboratory
colab.research.google.com/github/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023/blob/main/notebooks/30_trailer.ipy...
30_trailer.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
+ Code
+ Text
Copy to Drive

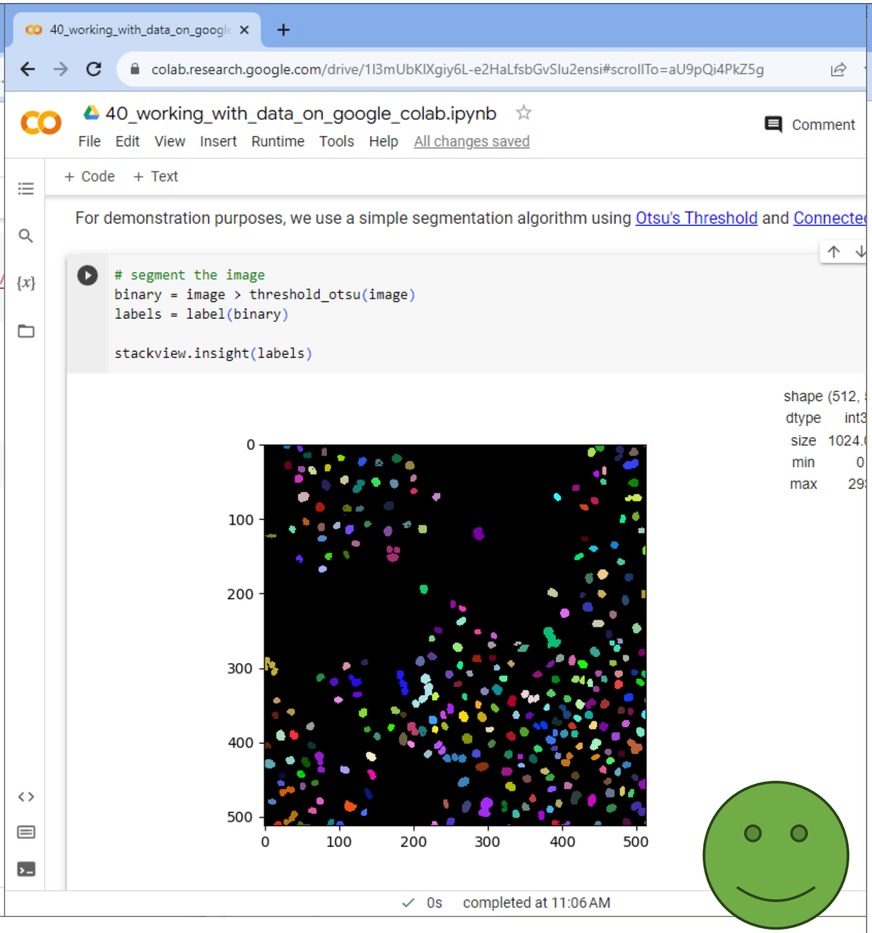
```
[ ] # open an image file
    original_image = imread("https://github.com/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023/raw/main/data/
    print("Image size:", original_image.shape)

    Image size: (512, 512)
```

We can visualize the image and some basic statistics about the intensity distribution like this.

```
[ ] stackview.insight(original_image)
```

```
shape (512, 512)
dtype    uint8
 size  256.0 kB
  min        7
  max      255
```

0
100
200
300
250
200
150
100


40_working_with_data_on_googl
colab.research.google.com/drive/1I3mUbKIXgiy6L-e2HaLfsbGvSlu2ensi#scrollTo=aU9pQi4PkZ5g
40_working_with_data_on_google_colab.ipynb
File
Edit
View
Insert
Runtime
Tools
Help
All changes saved
Comment
+ Code
+ Text

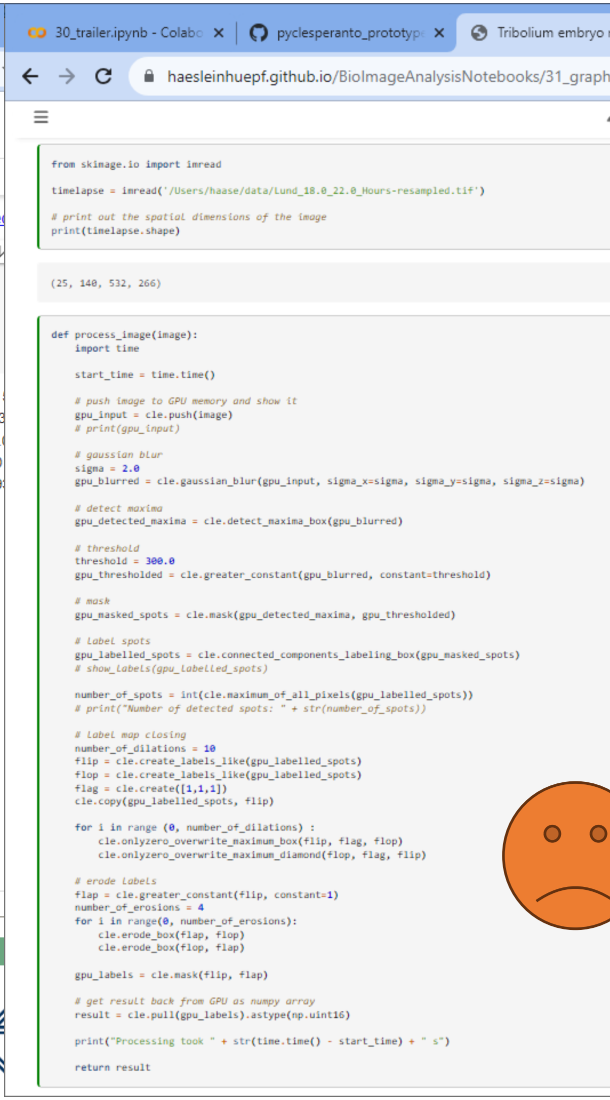
For demonstration purposes, we use a simple segmentation algorithm using Otsu's Threshold and Connecte

```
# segment the image
binary = image > threshold_otsu(image)
labels = label(binary)

stackview.insight(labels)
```

```
shape (512,
dtype    int3
 size  1024.(
  min       0
  max      29
```

0
100
200
300
400
500
0
100
200
300
400
500

✓ 0s completed at 11:06 AM


30_trailer.ipynb - Colabo
pyclesperanto_prototyp
Tribolium embryo
haesleinhuepf.github.io/BioImageAnalysisNotebooks/31_graph

```
from skimage.io import imread

timelapse = imread('/Users/haase/data/Lund_18.0_22.0_Hours-resampled.tif')

# print out the spatial dimensions of the image
print(timelapse.shape)
```

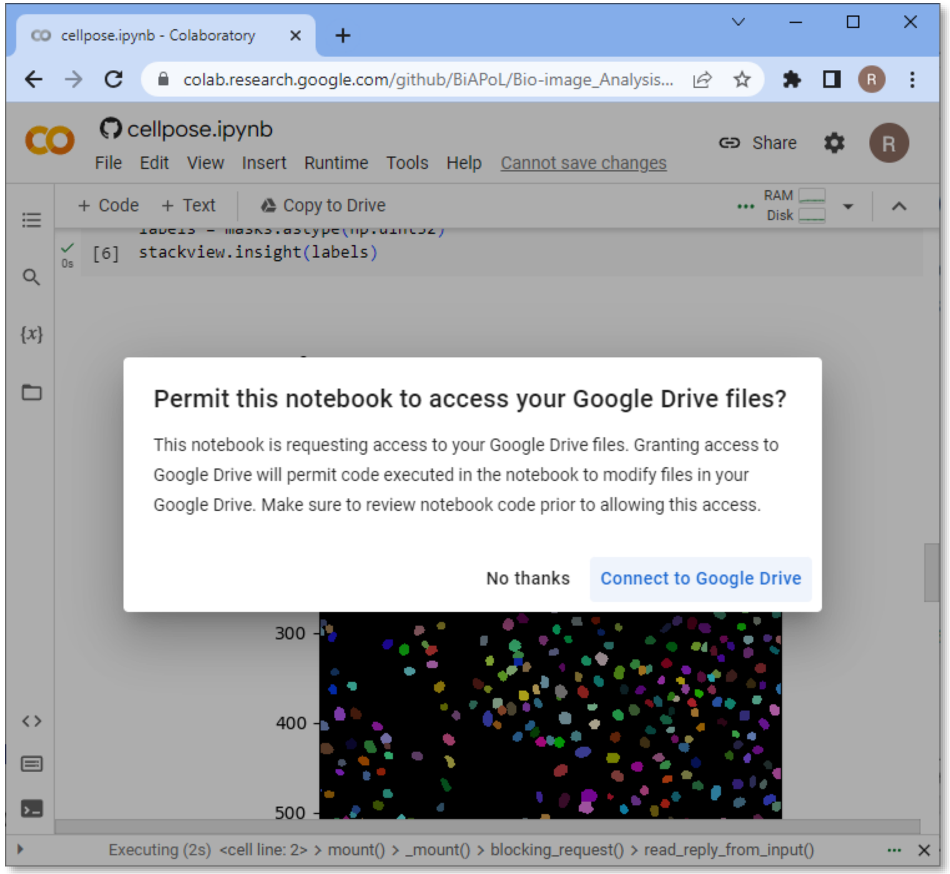
```
(25, 140, 532, 266)
```

```
def process_image(image):
    import time

    start_time = time.time()

    # push image to GPU memory and show it
    gpu_input = cle.push(image)
    # print(gpu_input)

    # gaussian blur
    sigma = 2.0
    gpu_blurred = cle.gaussian_blur(gpu_input, sigma_x=sigma, sigma_y=sigma, sigma_z=sigma)

    # detect maxima
    gpu_detected_maxima = cle.detect_maxima_box(gpu_blurred)

    # threshold
    threshold = 300.0
    gpu_thresholded = cle.greater_constant(gpu_blurred, constant=threshold)

    # mask
    gpu_masked_spots = cle.mask(gpu_detected_maxima, gpu_thresholded)

    # label spots
    gpu_labelled_spots = cle.connected_components_labeling_box(gpu_masked_spots)
    # show_labels(gpu_labelled_spots)

    number_of_spots = int(cle.maximum_of_all_pixels(gpu_labelled_spots))
    # print("Number of detected spots: " + str(number_of_spots))

    # label map closing
    number_of_dilations = 10
    flip = cle.create_labels_like(gpu_labelled_spots)
    flop = cle.create_labels_like(gpu_labelled_spots)
    flag = cle.create([1,1,1])
    cle.copy(gpu_labelled_spots, flip)

    for i in range (0, number_of_dilations) :
        cle.onlyzero_overwrite_maximum_box(flip, flag, flop)
        cle.onlyzero_overwrite_maximum_diamond(flop, flag, flip)

    # erode labels
    flap = cle.greater_constant(flip, constant=1)
    number_of_erosions = 4
    for i in range(0, number_of_erosions):
        cle.erode_box(flap, flop)
        cle.erode_box(flop, flap)

    gpu_labels = cle.mask(flip, flap)

    # get result back from GPU as numpy array
    result = cle.pull(gpu_labels).astype(np.uint16)

    print("Processing took " + str(time.time() - start_time) + " s")

    return result
```

Connecting Google Drive

Note: Do not store hot research data on Google Drive. Use institutional infrastructure instead.

- Work with data on your Google Drive


cellpose.ipynb - Colaboratory
colab.research.google.com/github/BiAPoL/Bio-image_Analysis...
cellpose.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
Share
+ Code + Text Copy to Drive
RAM
Disk
300
400
500

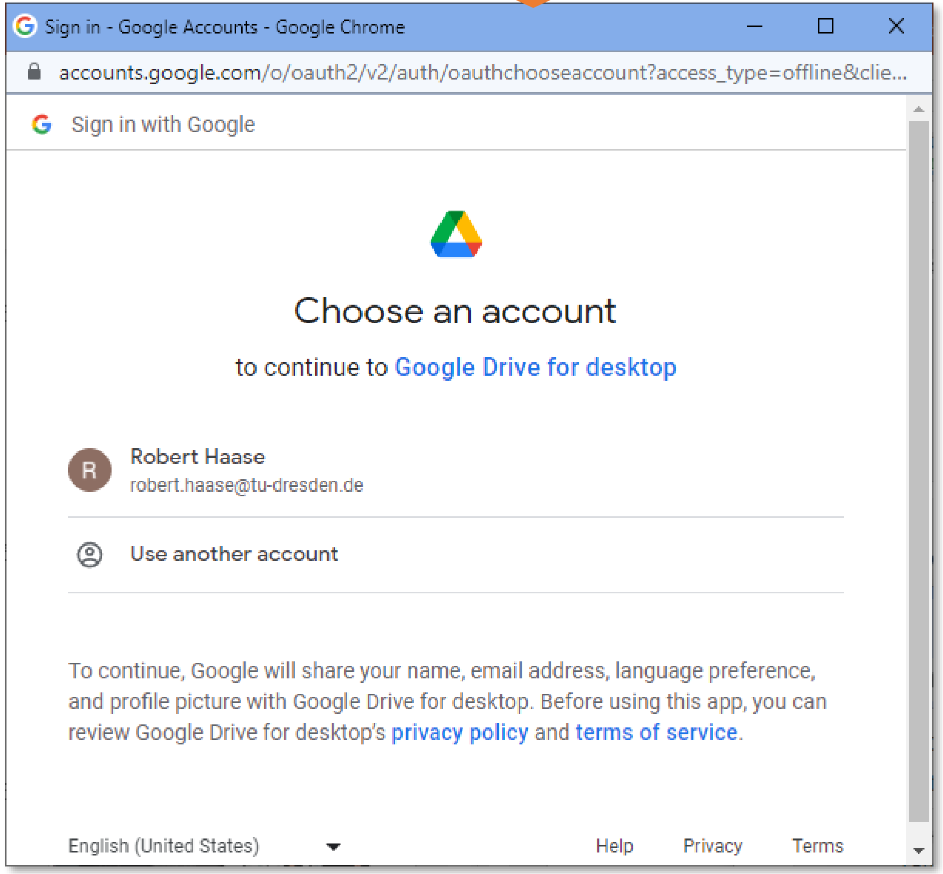
```
[6] stackview.insight(labels)
```

Permit this notebook to access your Google Drive files?

This notebook is requesting access to your Google Drive files. Granting access to Google Drive will permit code executed in the notebook to modify files in your Google Drive. Make sure to review notebook code prior to allowing this access.

No thanks | Connect to Google Drive

Executing (2s) <cell line: 2> > mount() > _mount() > blocking_request() > read_reply_from_input()


Sign in - Google Accounts - Google Chrome
accounts.google.com/o/oauth2/v2/auth/oauthchooseaccount?access_type=offline&clie...
Sign in with Google

Choose an account

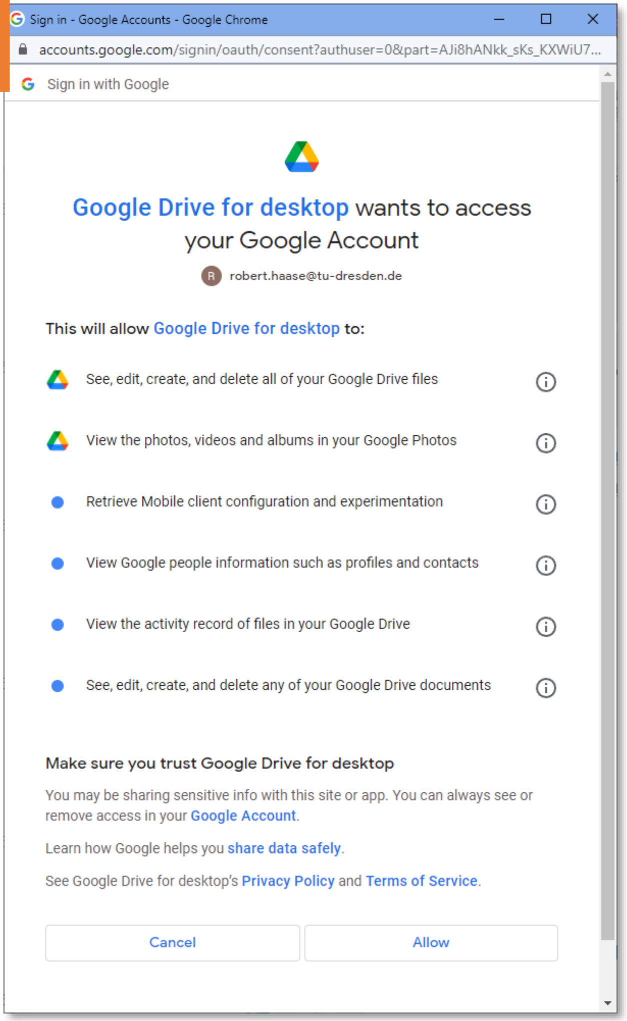
to continue to Google Drive for desktop

Robert Haase
robert.haase@tu-dresden.de

Use another account

To continue, Google will share your name, email address, language preference, and profile picture with Google Drive for desktop. Before using this app, you can review Google Drive for desktop's privacy policy and terms of service.

English (United States)
Help
Privacy
Terms


Sign in - Google Accounts - Google Chrome
accounts.google.com/signin/oauth/consent?authuser=0&part=AJi8hANkk_sKs_KXWiU7...
Sign in with Google

Google Drive for desktop wants to access your Google Account

robert.haase@tu-dresden.de

This will allow Google Drive for desktop to:

- See, edit, create, and delete all of your Google Drive files
- View the photos, videos and albums in your Google Photos
- Retrieve Mobile client configuration and experimentation
- View Google people information such as profiles and contacts
- View the activity record of files in your Google Drive
- See, edit, create, and delete any of your Google Drive documents

Make sure you trust Google Drive for desktop

You may be sharing sensitive info with this site or app. You can always see or remove access in your Google Account.

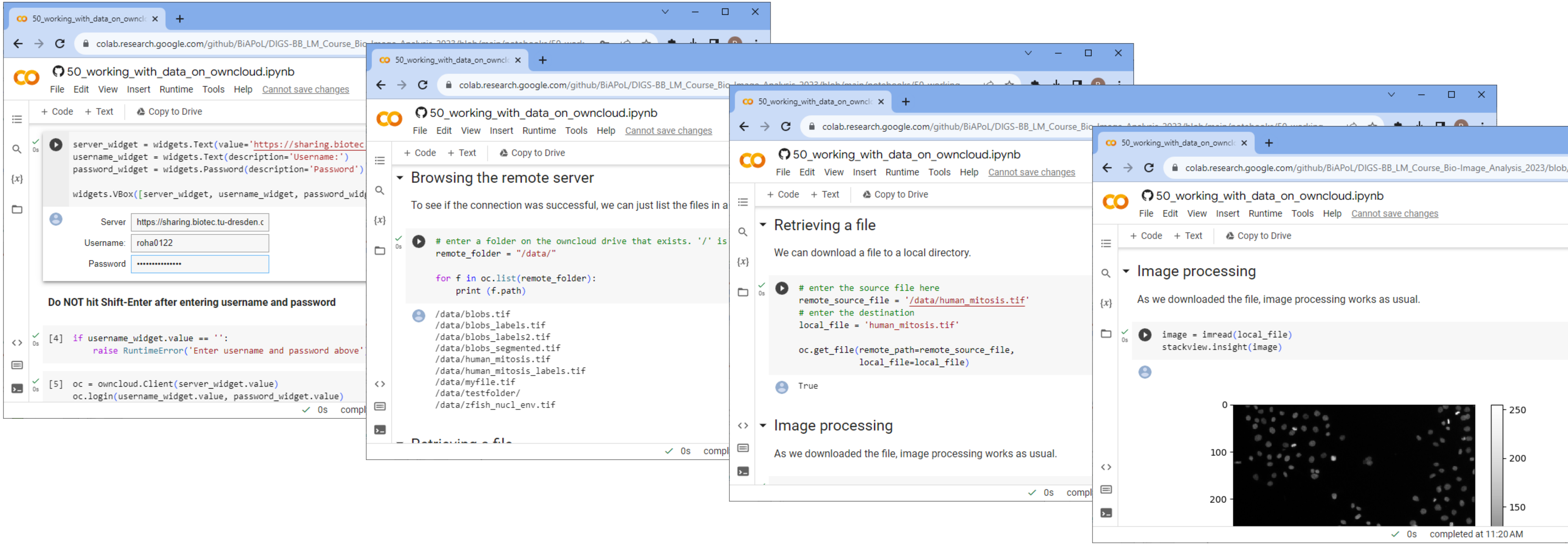
Learn how Google helps you share data safely.

See Google Drive for desktop's Privacy Policy and Terms of Service.

Cancel | Allow

Connecting to Owncloud

- You can also work with data on TU Dresden’s owncloud from Google Colab (that’s better because of data-protection issues.


50_working_with_data_on_owncl
colab.research.google.com/github/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023/blob
50_working_with_data_on_owncloud.ipynb
File Edit View Insert Runtime Tools Help Cannot save changes
+ Code + Text Copy to Drive
✓ 0s completed at 11:20 AM

```
server_widget = widgets.Text(value='https://sharing.biotec
username_widget = widgets.Text(description='Username:')
password_widget = widgets.Password(description='Password')

widgets.VBox([server_widget, username_widget, password_widg
```

Server: https://sharing.biotec.tu-dresden.c
Username: roha0122
Password: •••••••••••••••

Do NOT hit Shift-Enter after entering username and password

```
[4] if username_widget.value == '':
        raise RuntimeError('Enter username and password above'
```

```
[5] oc = owncloud.Client(server_widget.value)
    oc.login(username_widget.value, password_widget.value)
```

Browsing the remote server

To see if the connection was successful, we can just list the files in a

```
# enter a folder on the owncloud drive that exists. '/' is
remote_folder = "/data/"

for f in oc.list(remote_folder):
    print (f.path)
```

```
/data/blobs.tif
/data/blobs_labels.tif
/data/blobs_labels2.tif
/data/blobs_segmented.tif
/data/human_mitosis.tif
/data/human_mitosis_labels.tif
/data/myfile.tif
/data/testfolder/
/data/zfish_nucl_env.tif
```

Retrieving a file

We can download a file to a local directory.

```
# enter the source file here
remote_source_file = '/data/human_mitosis.tif'
# enter the destination
local_file = 'human_mitosis.tif'

oc.get_file(remote_path=remote_source_file,
            local_file=local_file)
```

```
True
```

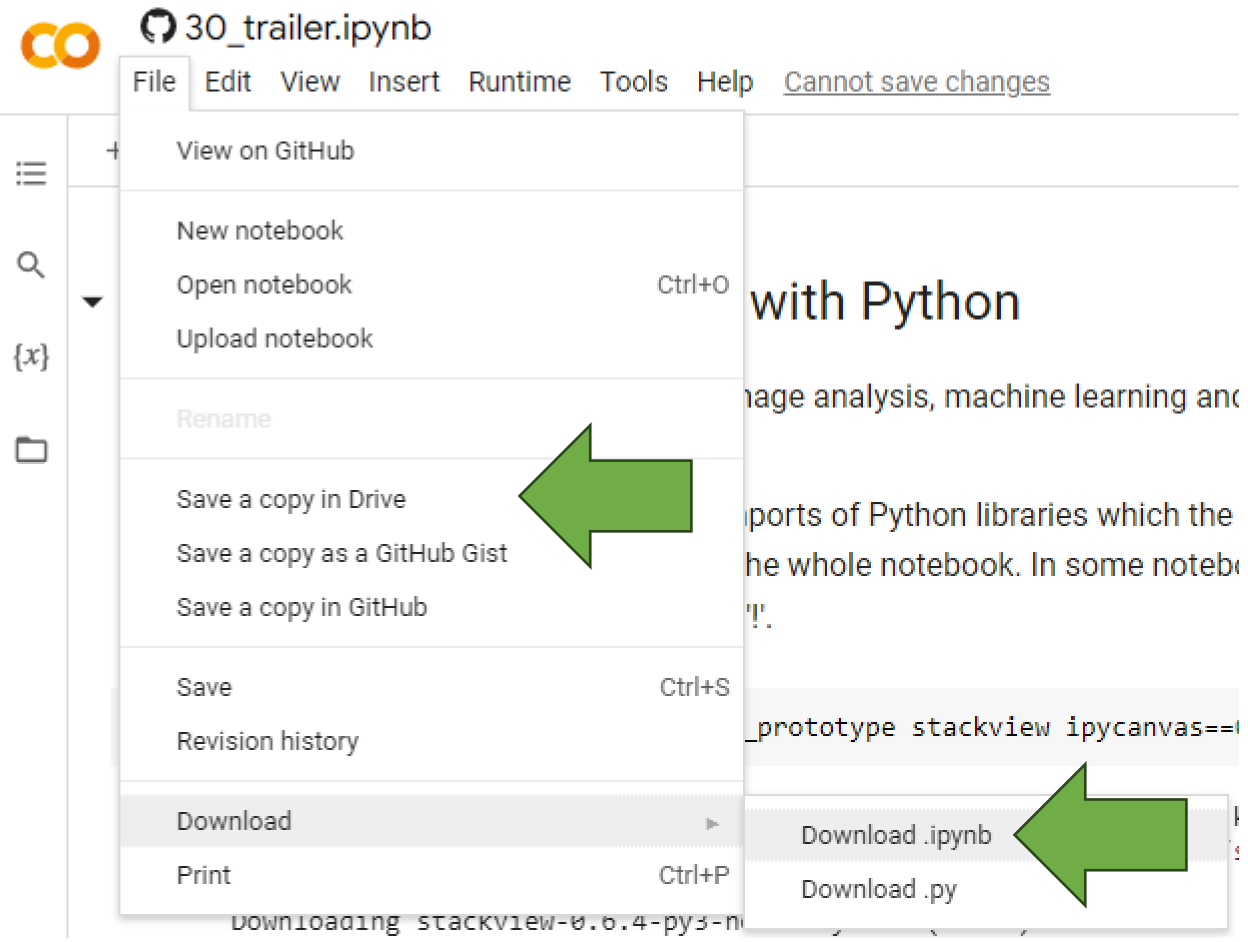
Image processing

As we downloaded the file, image processing works as usual.

```
image = imread(local_file)
stackview.insight(image)
```

Don’t forget to save!

- If you close the browser, your notebook may be gone.
- Save your changes occasionally.



Summary

Today, you learned

- *Bio-image analysis*
  - Quantitative
  - Objective
  - Reproducible
  - Repeatable
  - Reliable
- Google Colab
  - Working with Notebooks
  - Image Processing Workflows
  - Google Drive
  - Owncloud
- [Very] basic Python programming

Coming up next

- Image Filtering
- Image Segmentation
- Feature Extraction

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

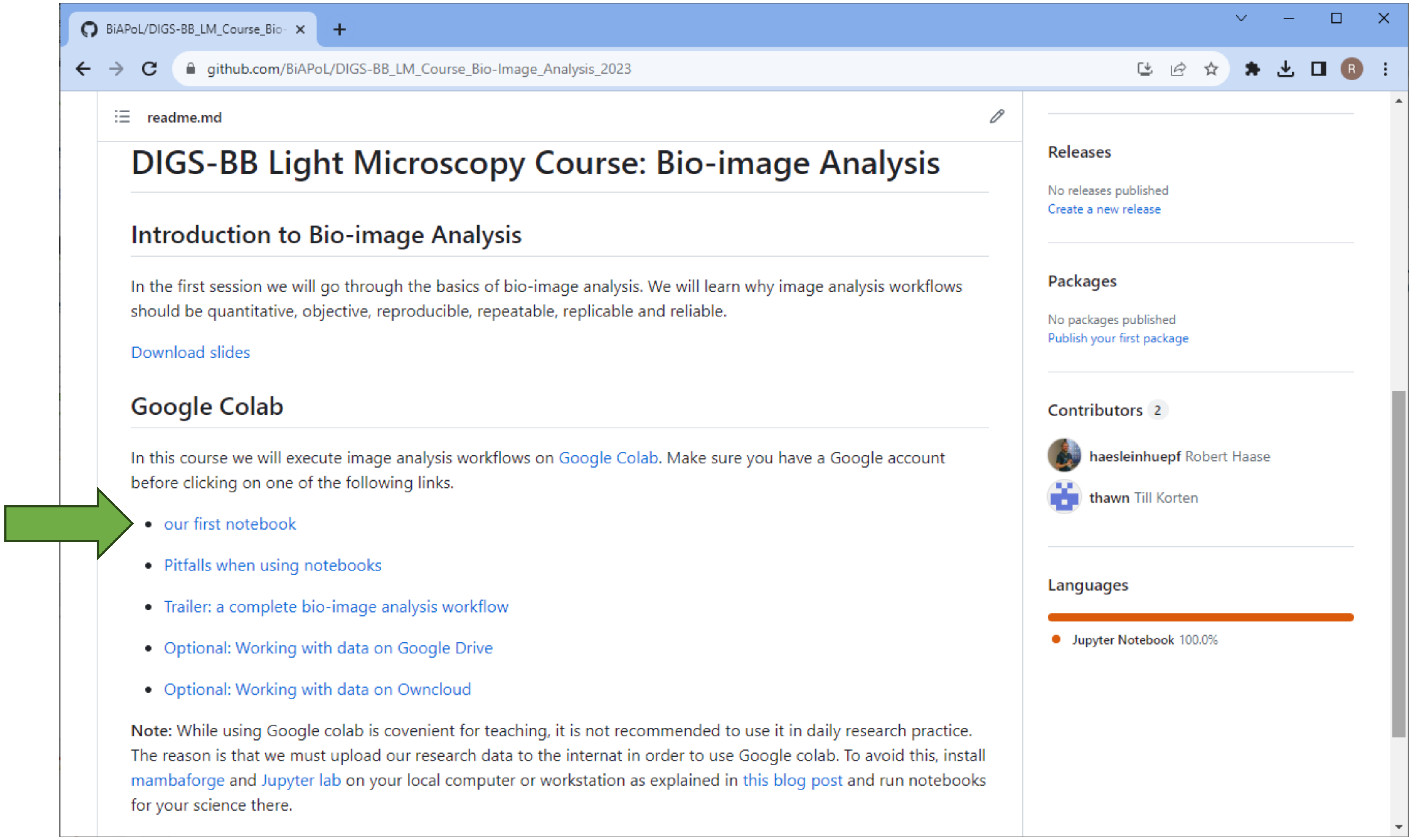
Exercises

Robert Haase

September 2023

Google Colab: Demo notebooks

- Run the provided notebooks in Google Colab and take care of the exercises on the bottom of [some] notebooks.
- https://github.com/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023



BiAPoL/DIGS-BB_LM_Course_Bio
github.com/BiAPoL/DIGS-BB_LM_Course_Bio-Image_Analysis_2023
readme.md

DIGS-BB Light Microscopy Course: Bio-image Analysis

Introduction to Bio-image Analysis

In the first session we will go through the basics of bio-image analysis. We will learn why image analysis workflows should be quantitative, objective, reproducible, repeatable, replicable and reliable.

Download slides

Google Colab

In this course we will execute image analysis workflows on Google Colab. Make sure you have a Google account before clicking on one of the following links.

- our first notebook
- Pitfalls when using notebooks
- Trailer: a complete bio-image analysis workflow
- Optional: Working with data on Google Drive
- Optional: Working with data on Owncloud

Note: While using Google colab is covenient for teaching, it is not recommended to use it in daily research practice. The reason is that we must upload our research data to the internat in order to use Google colab. To avoid this, install mambaforge and Jupyter lab on your local computer or workstation as explained in this blog post and run notebooks for your science there.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package

Contributors 2

haesleinhuepf Robert Haase
thawn Till Korten

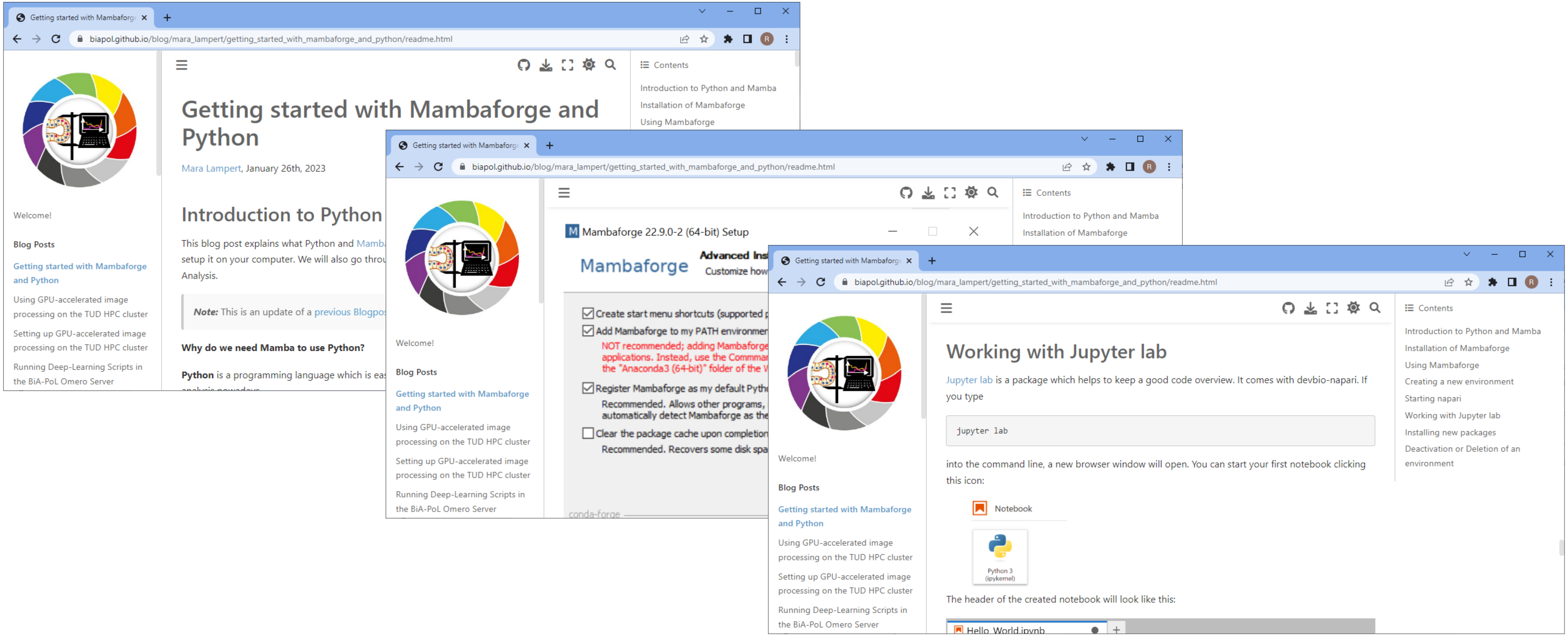
Languages

Jupyter Notebook 100.0%

Optional homework: Install mambaforge and test Python

Detailed instructions:

- https://biapol.github.io/blog/mara_lampert/getting_started_with_mambaforge_and_python/readme.html



Getting started with Mambaforge

biapol.github.io/blog/mara_lampert/getting_started_with_mambaforge_and_python/readme.html

Contents

Introduction to Python and Mamba

Installation of Mambaforge

Using Mambaforge

Getting started with Mambaforge and Python

Mara Lampert, January 26th, 2023

Introduction to Python

This blog post explains what Python and Mamb… setup it on your computer. We will also go throu… Analysis.

Note: This is an update of a previous Blogpos…

Why do we need Mamba to use Python?

Python is a programming language which is ea… analysis nowadays.

Welcome!

Blog Posts

Getting started with Mambaforge and Python

Using GPU-accelerated image processing on the TUD HPC cluster

Setting up GPU-accelerated image processing on the TUD HPC cluster

Running Deep-Learning Scripts in the BiA-PoL Omero Server

Mambaforge 22.9.0-2 (64-bit) Setup

Mambaforge

Advanced Ins…

Customize how…

☑ Create start menu shortcuts (supported p…

☑ Add Mambaforge to my PATH environmen…

NOT recommended; adding Mambaforge… applications. Instead, use the Comman… the "Anaconda3 (64-bit)" folder of the W…

☑ Register Mambaforge as my default Pytho…

Recommended. Allows other programs, … automatically detect Mambaforge as the…

☐ Clear the package cache upon completion

Recommended. Recovers some disk spa…

conda-forge

Contents

Introduction to Python and Mamba

Installation of Mambaforge

Using Mambaforge

Creating a new environment

Starting napari

Working with Jupyter lab

Installing new packages

Deactivation or Deletion of an environment

Working with Jupyter lab

Jupyter lab is a package which helps to keep a good code overview. It comes with devbio-napari. If you type

```
jupyter lab
```

into the command line, a new browser window will open. You can start your first notebook clicking this icon:

The header of the created notebook will look like this:

@haesleinhuepf

September 2023

Screenshots by Mara Lampert CC-BY 4.0