PoL Physics of Life TU Dresden

BIOTEC Biotechnology Center TU Dresden

CENTER FOR SYSTEMS BIOLOGY DRESDEN

Bio-image analysis, bio-statistics, programming and machine learning for computational biology

Anna Poetsch, Marcelo Zoccoler, Johannes Müller, Robert Haase

@haesleinhuepf

April 2021

The team

Anna Poetsch, Marcelo Zoccoler, Johannes Müller, Robert Haase

Credits: Hagen Gebauer

Lecture overview: Programming

- We will focus on Python programming.
- “Python is the most wanted language for the third year in a row, meaning that developers who do not yet use it say they want to learn it.”
- Goal: **Allow you to do things automatically instead of suffering long time when doing it manually.**

```
b = 3
c = a + b
```

```
print(c)
```

```
8
```

```
d = 6
e = 7
f = a * d
g = f / e
h = 1 + g
```

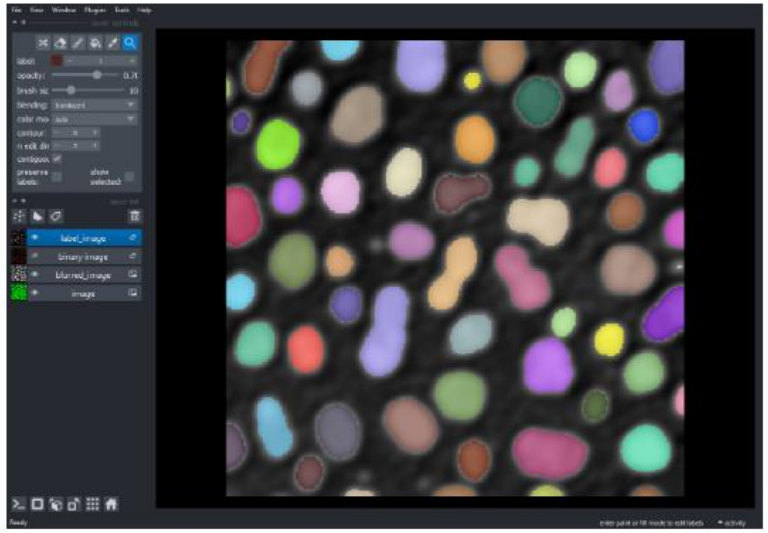
```
print(h)
```

```
5.285714285714286
```

```
from skimage.measure import label
label_image = label(binary_image)

# add labels to viewer
label_layer = viewer.add_labels(label_image)
```

You can visualize labelled objects as overlay (per default)

```
napari.utils.nbscreenshot(viewer)
```



Most Loved, Dreaded, and Wanted

Most Loved, Dreaded, and Wanted Languages

Loved | Dreaded | Wanted

Python	25.7%
JavaScript	17.8%
Go	15.0%
TypeScript	14.6%
Kotlin	11.1%
Rust	9.5%
C++	9.1%
WebAssembly	8.9%
Java	8.3%
SQL	7.6%

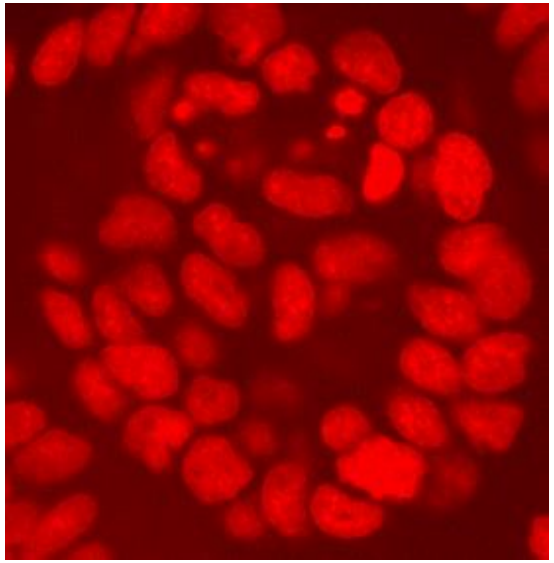
Source: https://insights.stackoverflow.com/survey/2019#technology

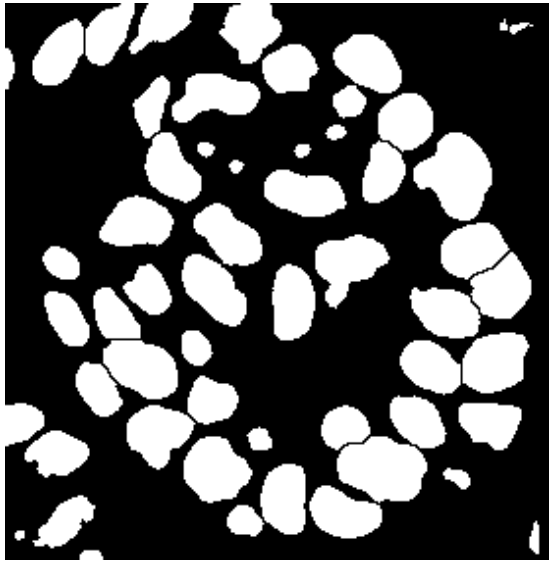
Lecture overview: Bio-image analysis

PoL Physics of Life TU Dresden

- Quantitative bio-image analysis
  - Transformations, Projections and Filtering
  - Image segmentation
  - Tracking objects
  - Measuring in images
- Goal: **Allow you to quantify observations.**

- Software
  - Python for automation
  - Napari for interactive image analysis





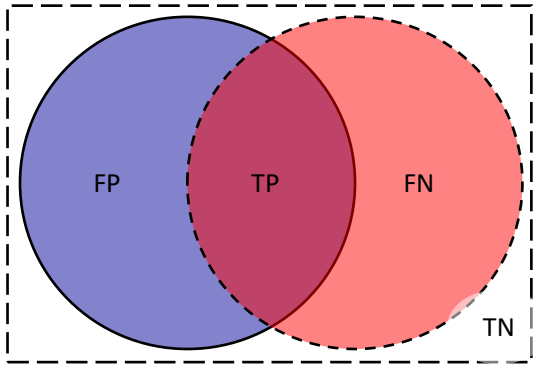
	label	maximum	mean	median	minimum	sigma
0	1	224.0	137.526132	136.0	112.0	13.360739
1	2	232.0	193.014354	200.0	128.0	28.559077
2	3	224.0	179.846995	184.0	128.0	21.328889
3	4	248.0	207.082171	216.0	120.0	27.772832
4	5	248.0	223.146402	232.0	128.0	30.246515
5	6	248.0	214.906725	224.0	128.0	26.386796
6	7	248.0	211.565891	224.0	136.0	30.197236
7	8	200.0	166.171429	168.0	136.0	16.466894
8	9	224.0	176.932331	176.0	128.0	24.022064
9	10	240.0	191.598174	200.0	128.0	28.239851

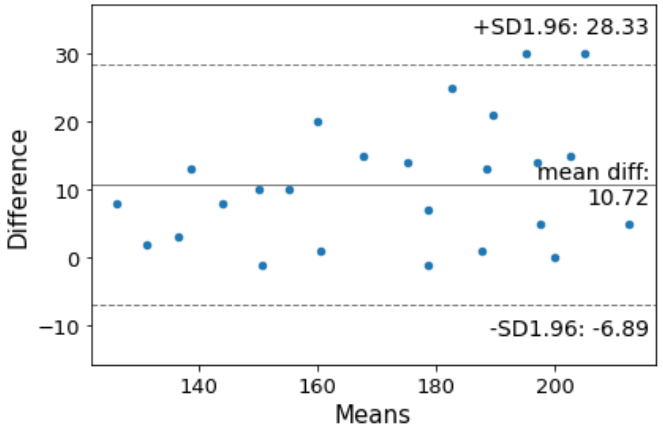
Image data source: Mauricio Rocha Martins, Norden lab, MPI CBG

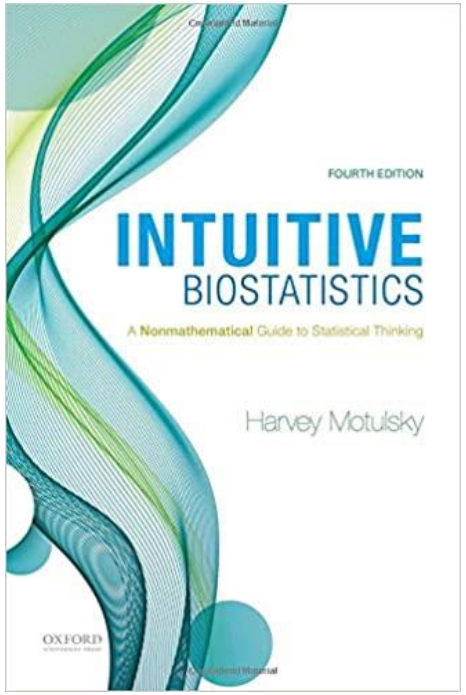
Lecture overview: Bio-statistics

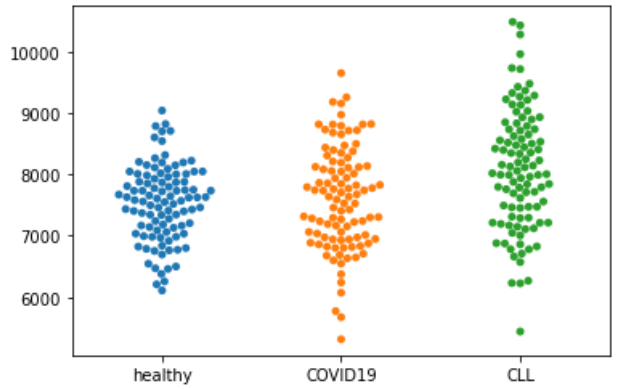
- Descriptive statistics
  - Distributions
  - Method comparison: Bland-Altman analysis
- Inferential statistics
  - hypothesis testing
- Multiple comparisons and correlations
- Big data, clustering, dimensionality reduction

- Goal: **Allow you to draw conclusions from quantified observations.**



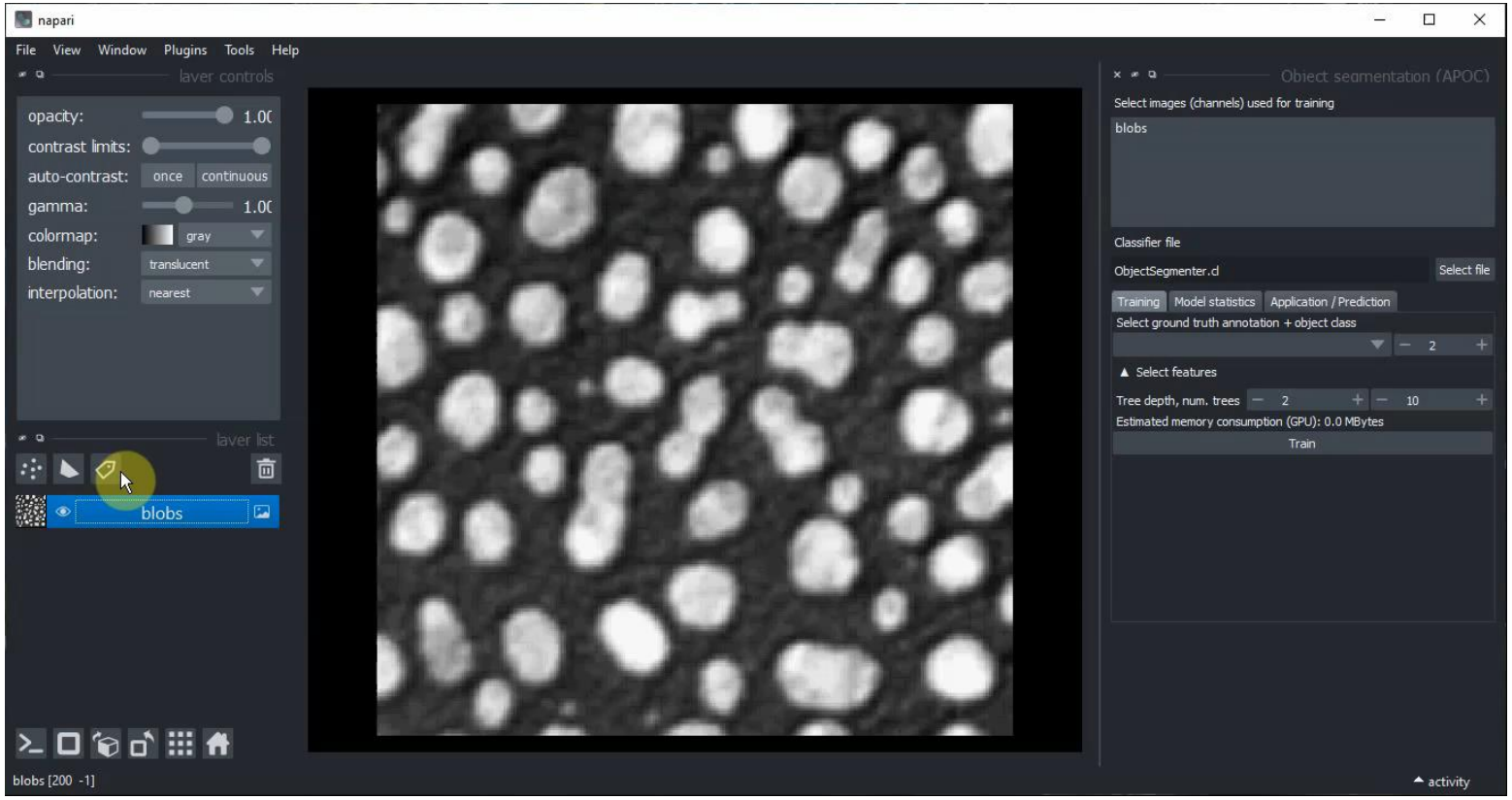


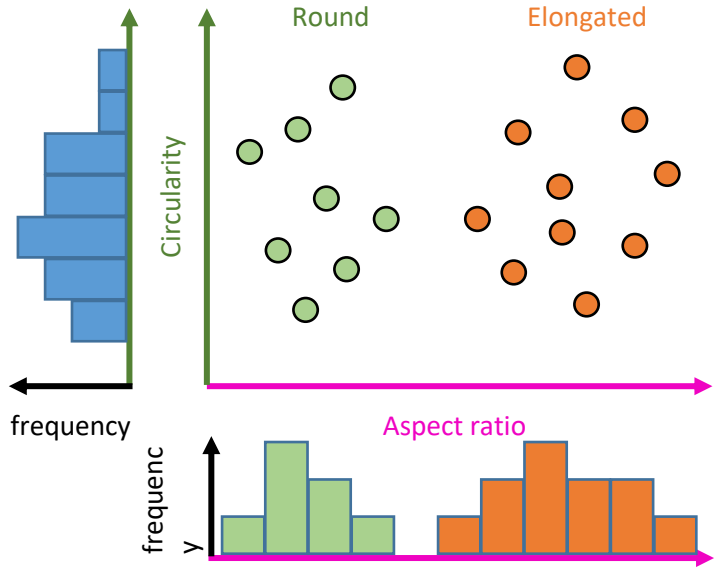


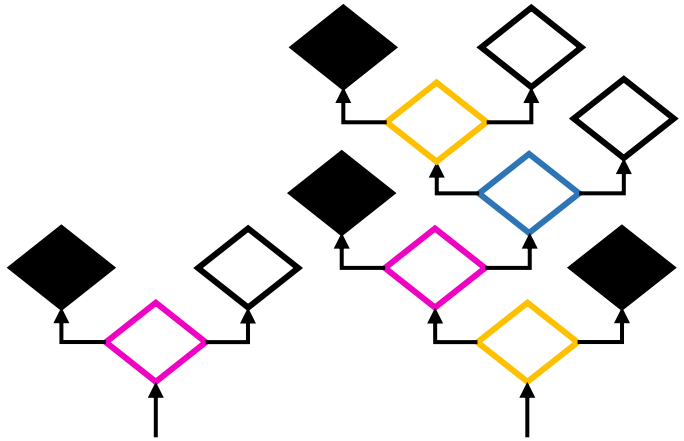


Lecture overview: Machine learning

- Machine learning
  - in the context of image analysis
- Computers can *learn* where, what objects in images are…
- Goal: **Give you an insight into state-of-the-art methods**







Random forest classifiers

Lecture overview

- 5.4.2022 - General introduction, setting up Conda and Python
- 12.4.2022 - Introduction to python programming, data structures and image data
- 19.4.2022 - Introduction to python algorithms: loops, conditions, functions, image processing
- 26.4.2022 - Image filtering + segmentation
- 3.5.2022 - Quantitative image analysis / feature extraction
- 10.5.2022 - Machine learning for bio-image analysis
- 17.5.2022 - Introduction to Biostatistics
- 24.5.2022 - Descriptive statistics
- 31.5.2022 - Method Comparison - Bland-Altman analysis
- 14.6.2022 - Hypothesis testing
- 21.6.2022 - Multiple comparisons and correlations
- 28.6.2022 - Big data, clustering, dimensionality reduction
- 5.7.2022 - Deep Learning
- 12.7.2022 – Summary, exam preparation

Lecture overview

PoL Physics of Life TU Dresden

- Every week will follow the same rough scheme
  - 13:00 - 13:30: Discussion about exercise last week
  - 13:30 : 1 h lecture
  - < 1h demonstration
  - < 1h homework
- Questions can be asked in the zoom meetings any time by speaking up. Or you post them anonymously: (see link to etherpad)

- Exam will cover the semester content accordingly
  - Theory of
    - image analysis,
    - statistics and
    - machine learning
  - Basics of programming
    - write simple < 10 line programs and
    - read code and describe what it does

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Introduction to Bio-Image Analysis

Robert Haase

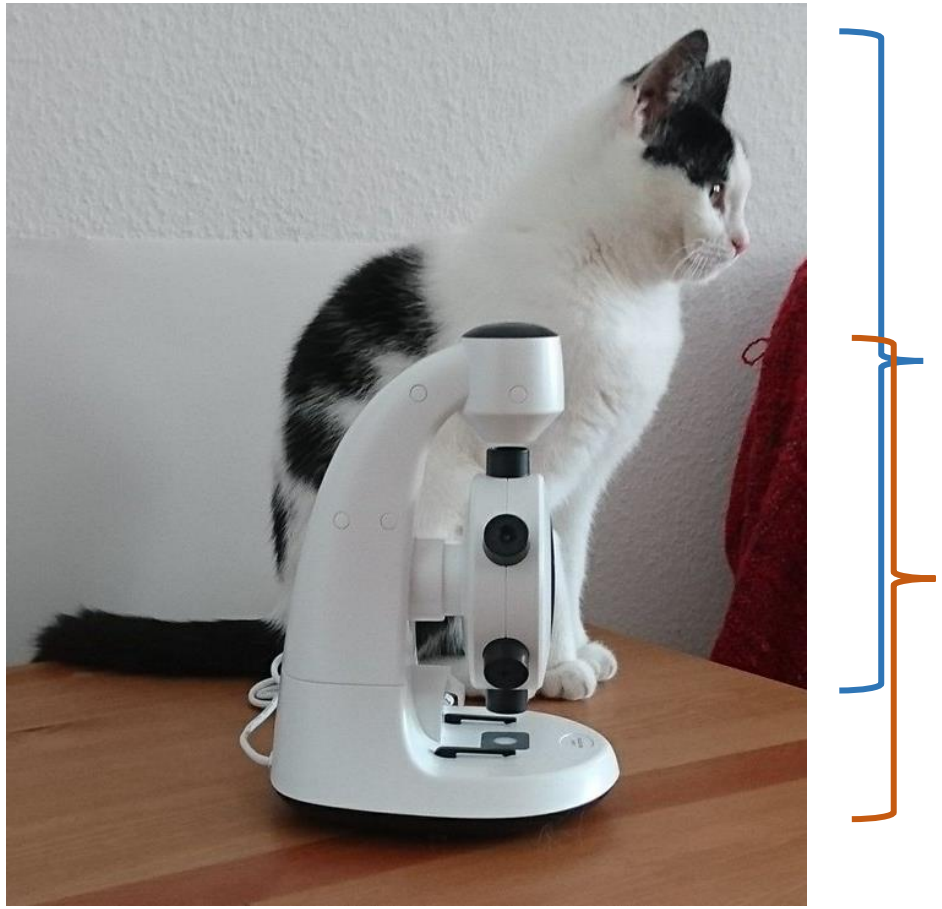
@haesleinhuepf

April 2022

Quantitative bio-image analysis

PoL Physics of Life TU Dresden

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

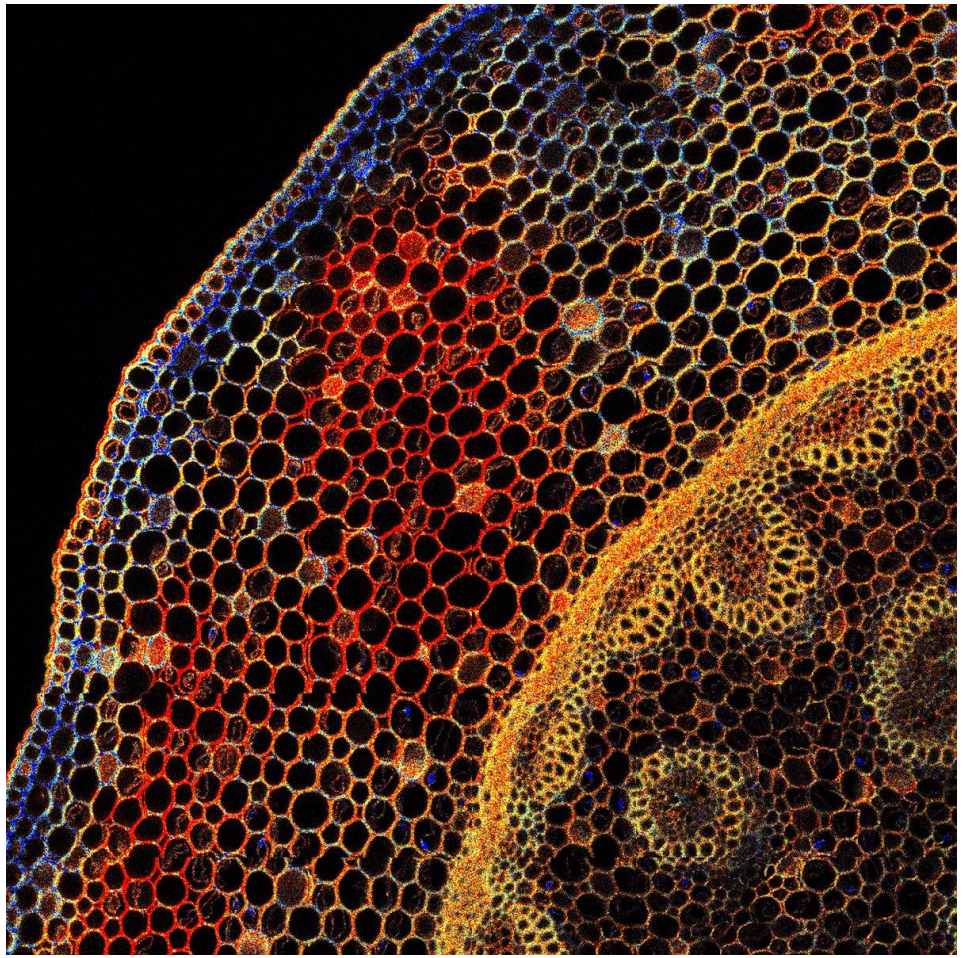


cat height = 1.5 x microscope height

Quantitative bio-image analysis

PoL Physics of Life TU Dresden

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



How many cells are there in the outer ring?

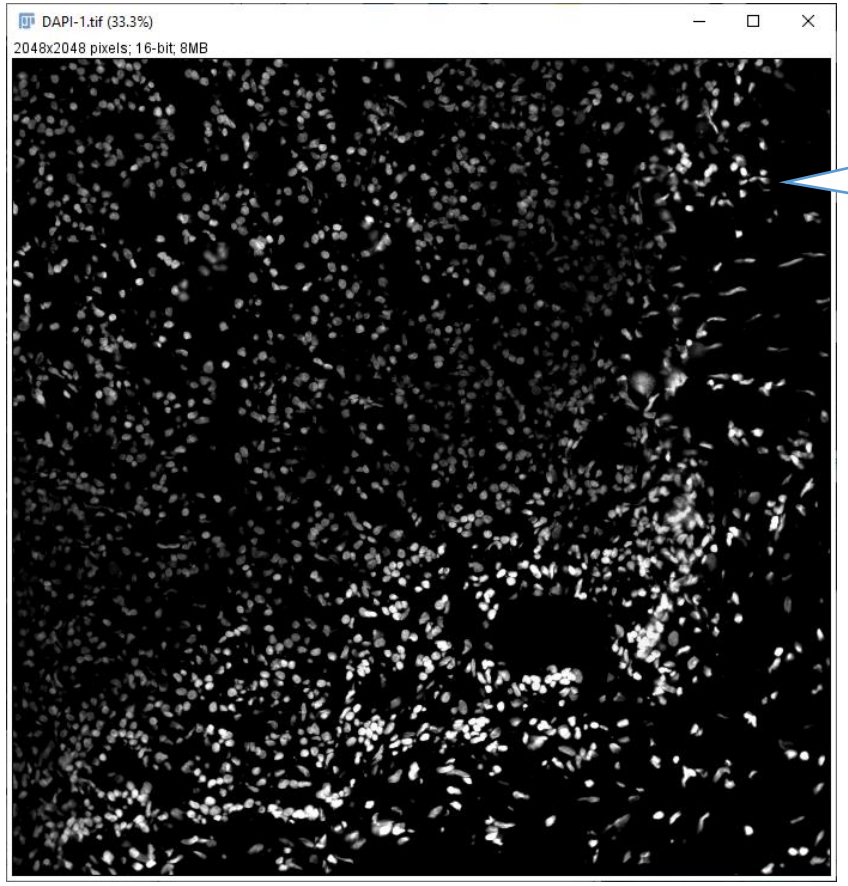
How high is the intensity in the inner ring?

How is the signal in the blue channel distributed?

Image data source: Lorenzo Scipioni, U. California / Irvine
@LorenzoScipion3

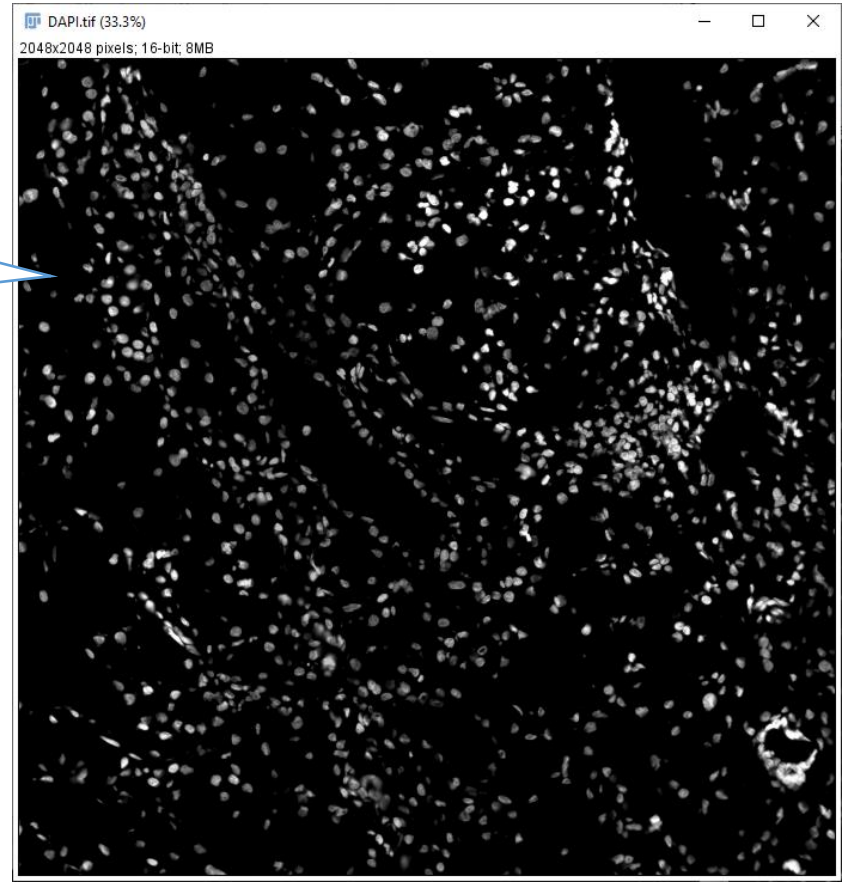
Objective bio-image analysis

- Measurements should be objective, not influenced by human interpretation



Nuclei in this image are …

… more dense than in this image.



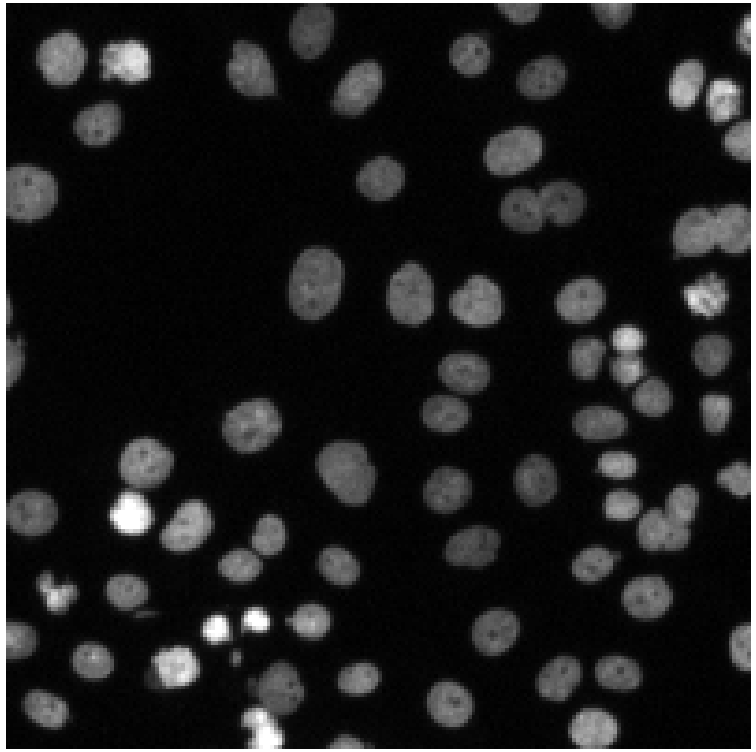
Use automation for less subjective analysis.

Reliable bio-image analysis

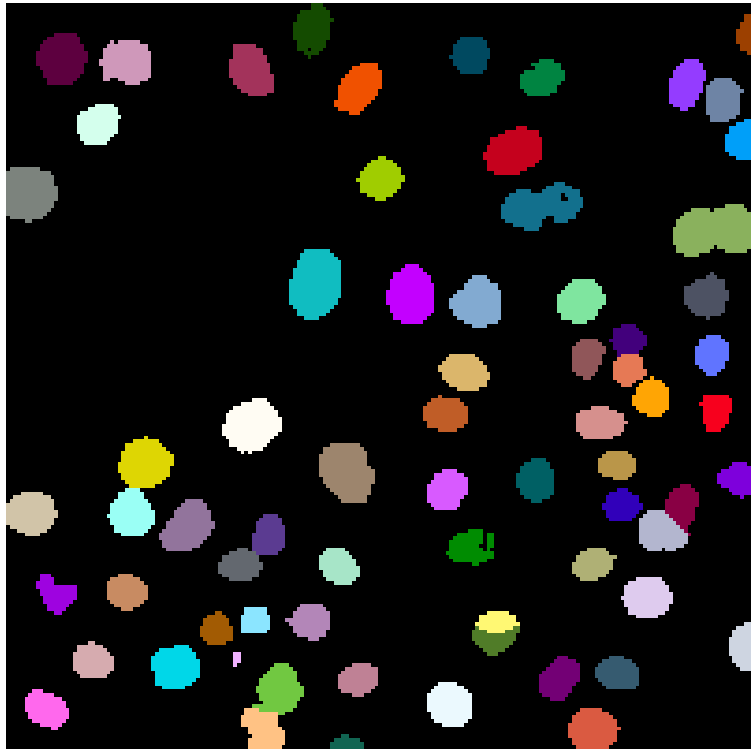
PoL Physics of Life TU Dresden

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

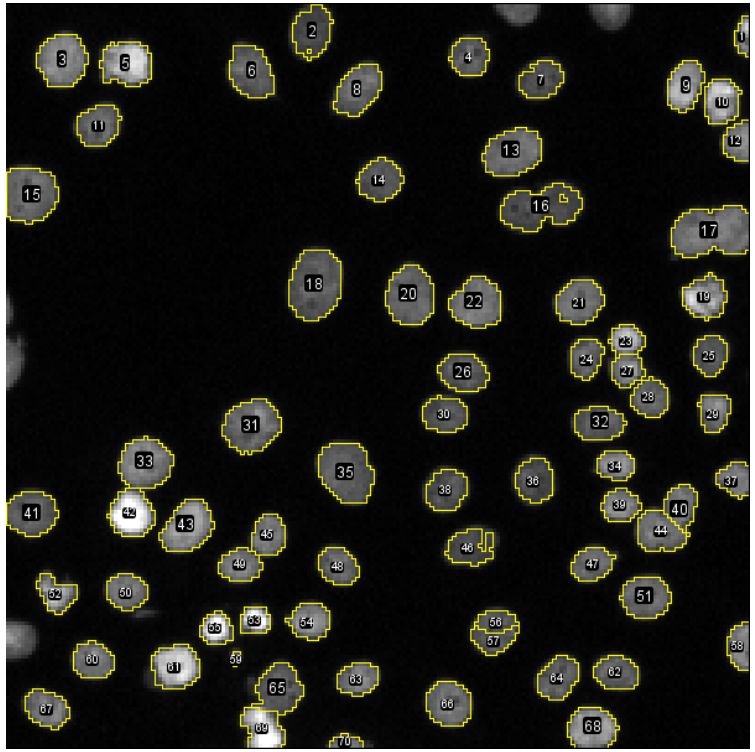
Original image



Label image



Overlay

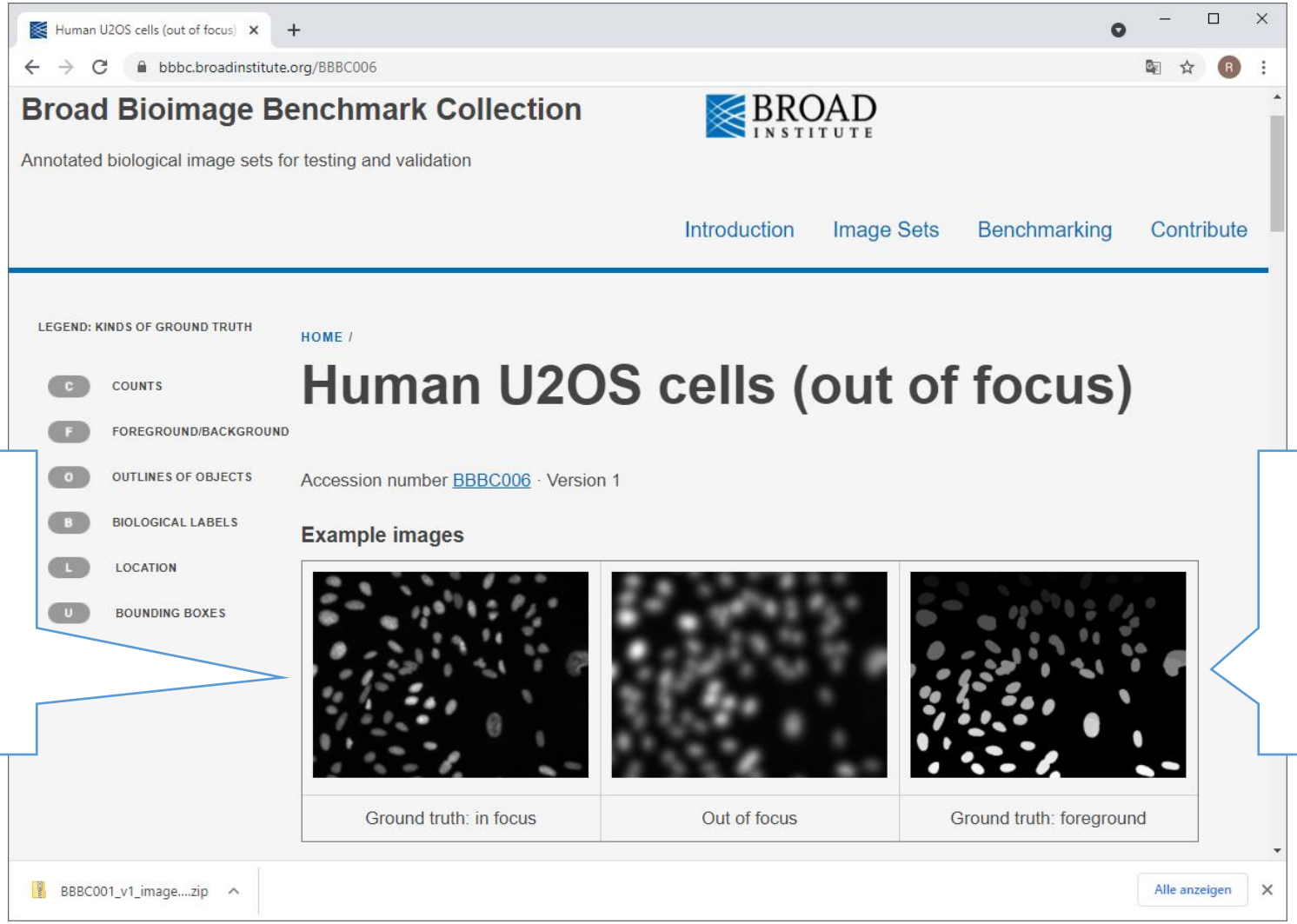


There are 70 nuclei in this image.

Image Data source: BBBC001v1 [Carpenter et al., *Genome Biology*, 2006] from the Broad Bioimage Benchmark Collection [Ljosa et al., *Nature Methods*, 2012]

Reliable bio-image analysis

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



Original image data

"Ground truth" label images

Reproducible bio-image analysis

- “The image data was analyzed with ImageJ.”

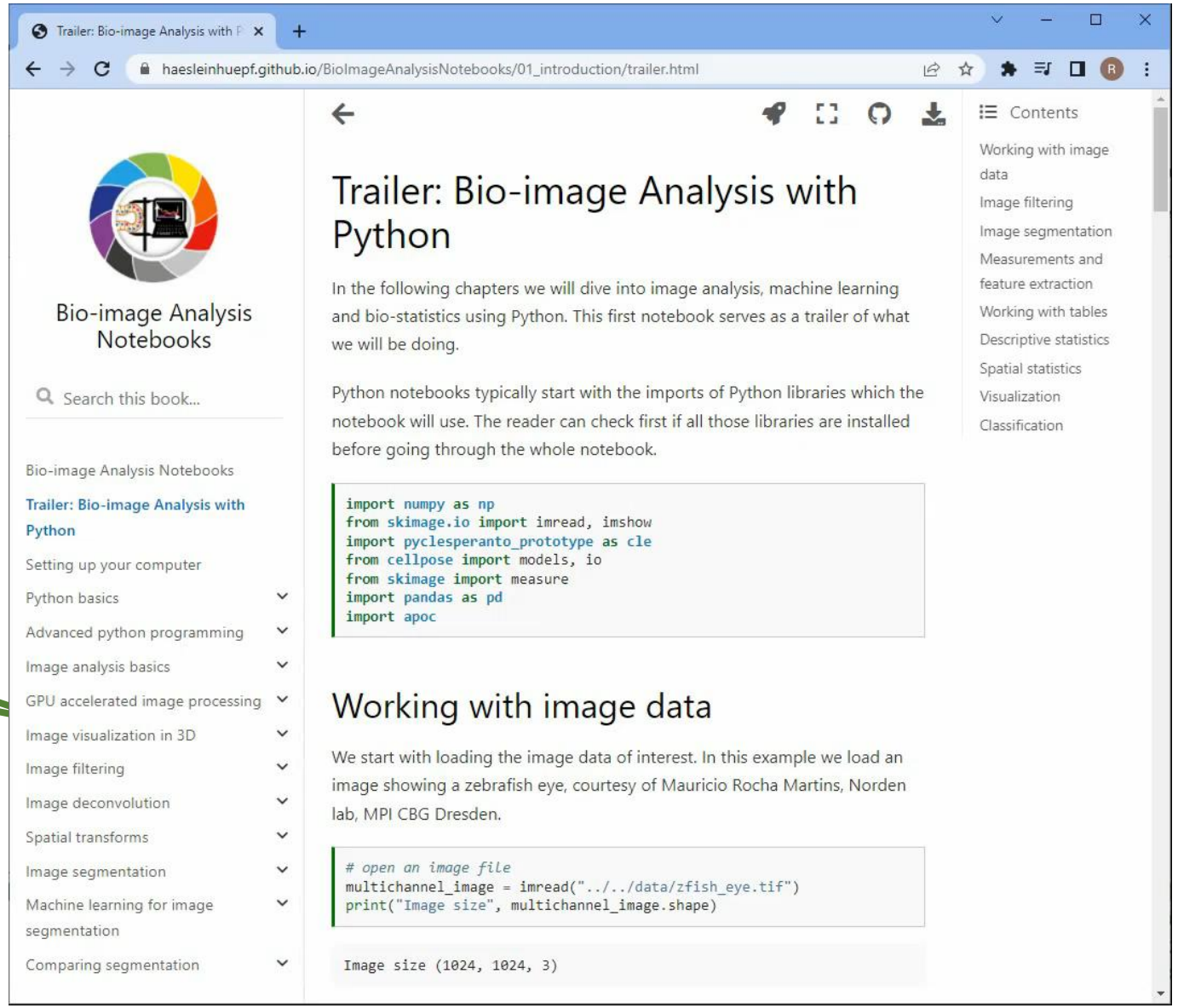
Can you reproduce what they did?

Reproducible bio-image analysis

- “The image data was analyzed with ImageJ.”

Can you reproduce what they did?

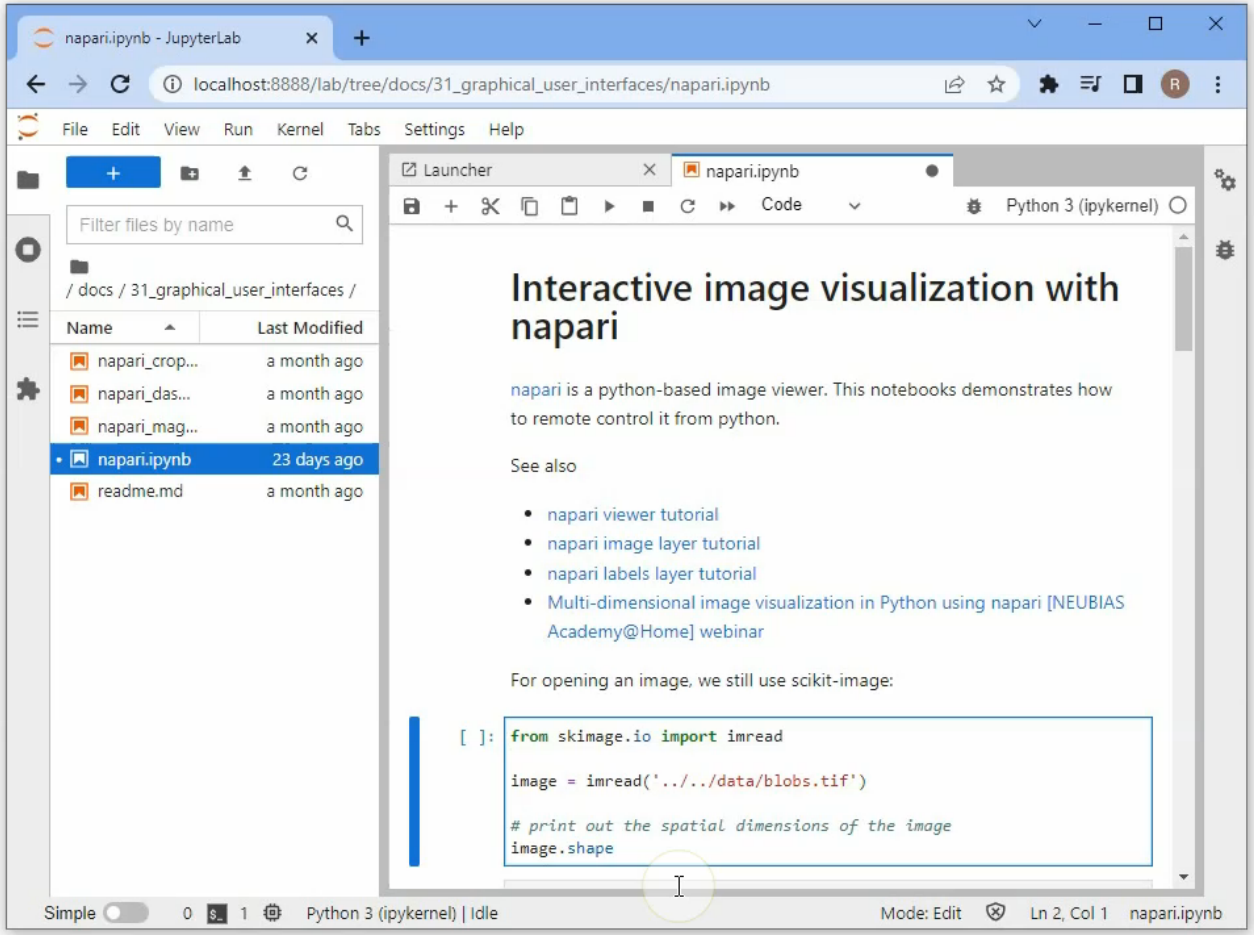
Can you reproduce what they did?



Interactivity versus reproducibility

PoL Physics of Life TU Dresden

- Remote controlling graphical user interfaces

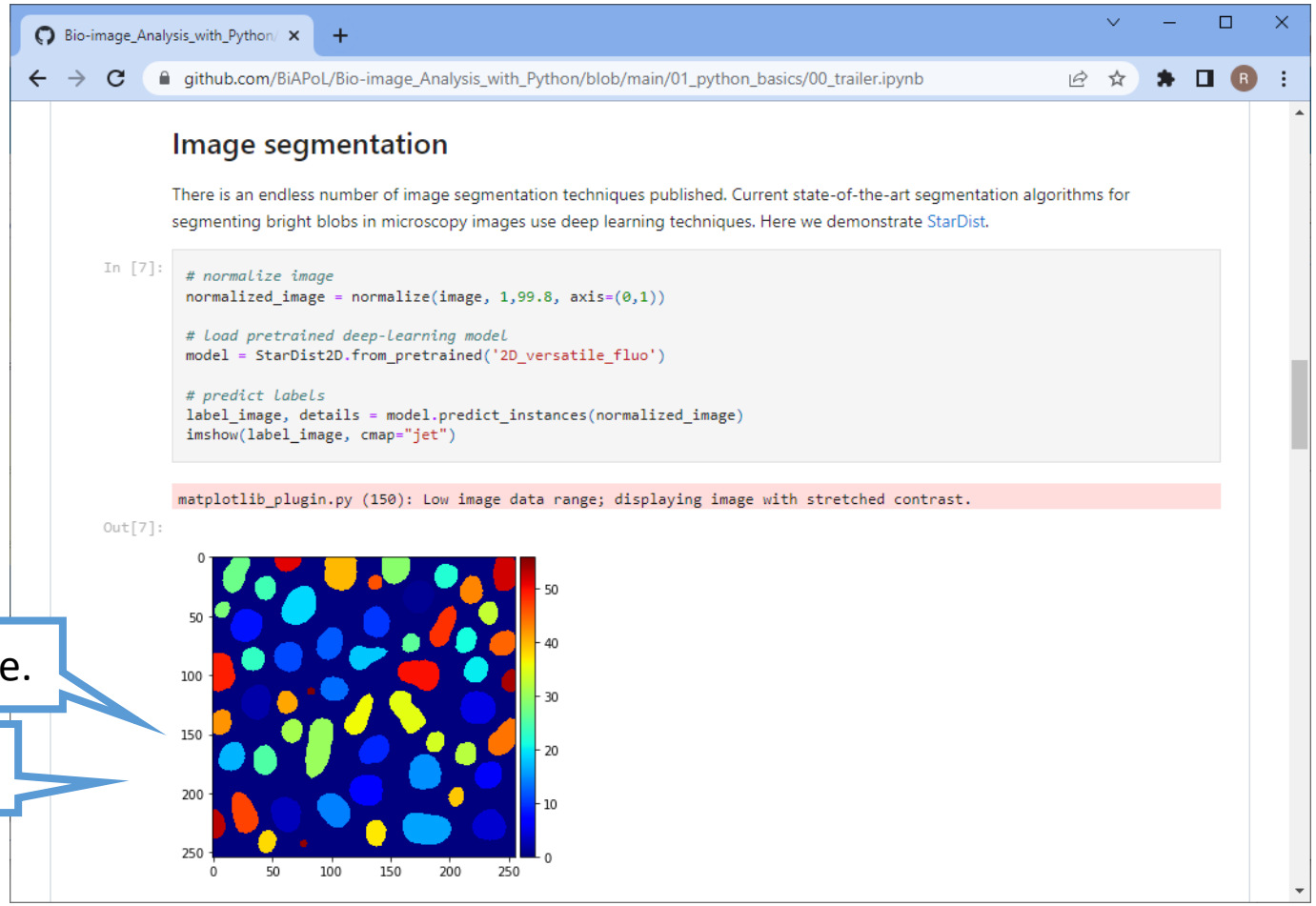


Repeatable Bio-image analysis

PoL Physics of Life TU Dresden

- Compared to wet-lab experiments, image analysis (in-silico) experiments are typically repeatable.
- In wet-lab experiments, samples may get destroyed while executing the experiment.
- Repeatability is a property of the experiment. You cannot improve repeatability by better documentation.

- However, you need to pay some extra attention to have repeatable image processing
  - Scripts need to be written in a way that their execution is repeatable.
  - Test it!



Run it once.

Run it twice.

Are results identical?

Introduction to bio-image analysis

PoL Physics of Life TU Dresden

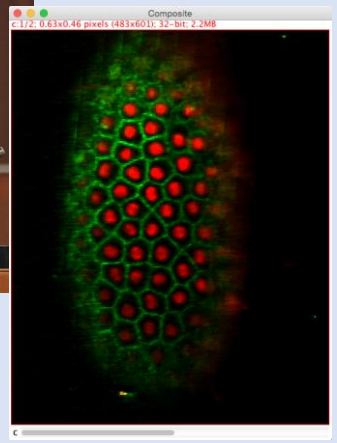
- Bio-image analysis is supposed to be
  - **Quantitative**
    - We derive numbers from images which describe physical properties of the observed sample.
  - **Objective**
    - The derived measurement does not depend on who did the measurement. The measurement is free of interpretation.
  - **Reliable (trustworthy / validated)**
    - We are confident that the measurement is describing what it is supposed to describe.
  - **Reproducible**
    - Somebody else can do the experiment under *different conditions* and gets similar measurements. For this, documentation is decisive!
  - **Repeatable**
    - We can do the same experiment twice under the *same conditions* and get similar measurements.

Image analysis is part of the experiment

Observation

$$p_{ij}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{j=1}^{n} [\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}$$

Modeling



Imaging

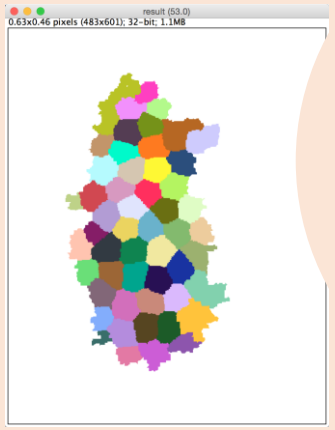
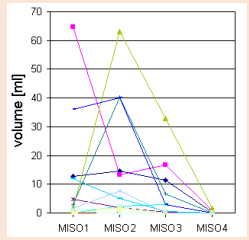


Image processing



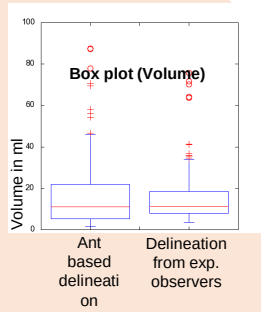
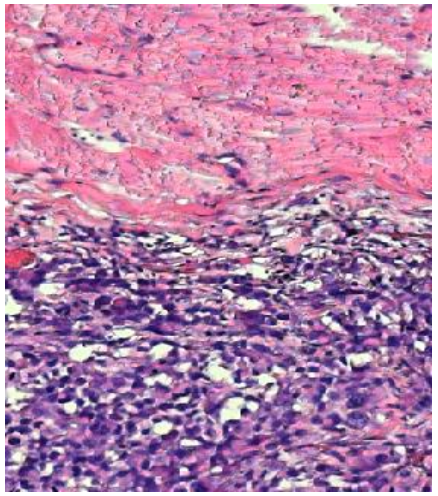


Image analysis
Bio-statistics

Common questions

- 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
  - How are different tissues characterized?
    - ➢ Machine learning



muscle, normal tissue

squamous-cell carcinoma

- 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?

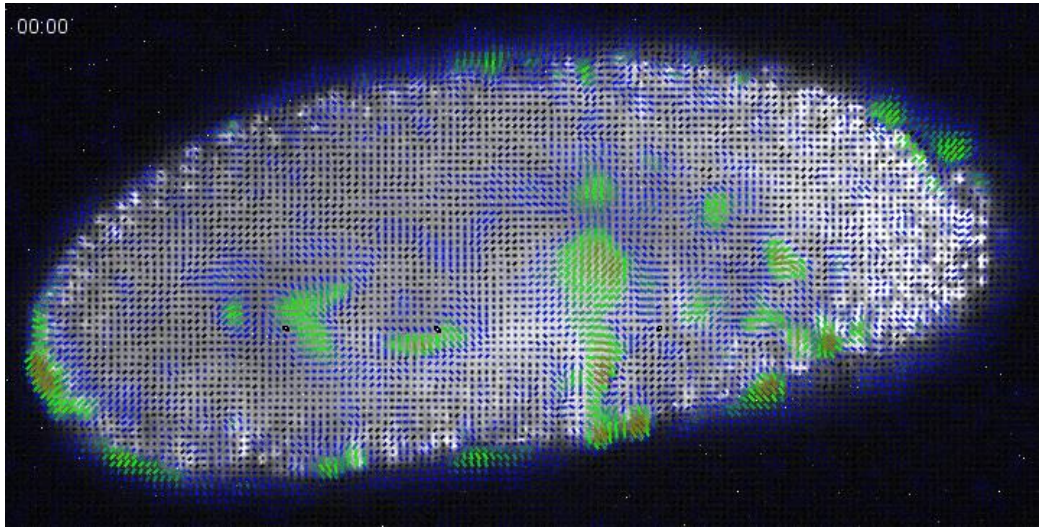


Image analysis is part of the experiment

PoL
Physics of Life
TU Dresden

- Think about how to analyze your images before starting the experiment.
  - Consider adapting your experiment so that quantitative image analysis can be performed easily.

- Think about controls, counter-proves, an easy to falsify null-hypothesis.
  - Be a lazy scientist. Do simple experiments.

- How can you exclude yourself from the experiment?
  - Think of blinding yourself or fully automate analysis.

- One experiment usually answers just one or less questions.

Image analysis is part of the experiment

Talk to image-analysis / data-science experts early during your project.

Not one week before the submission deadline.

The command line

Robert Haase

April 2022

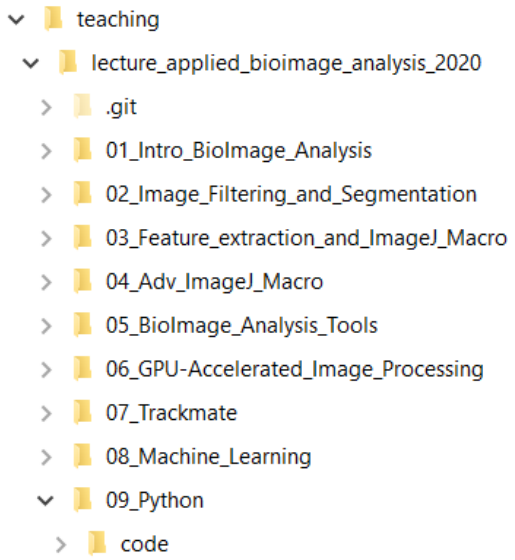
@haesleinhuepf

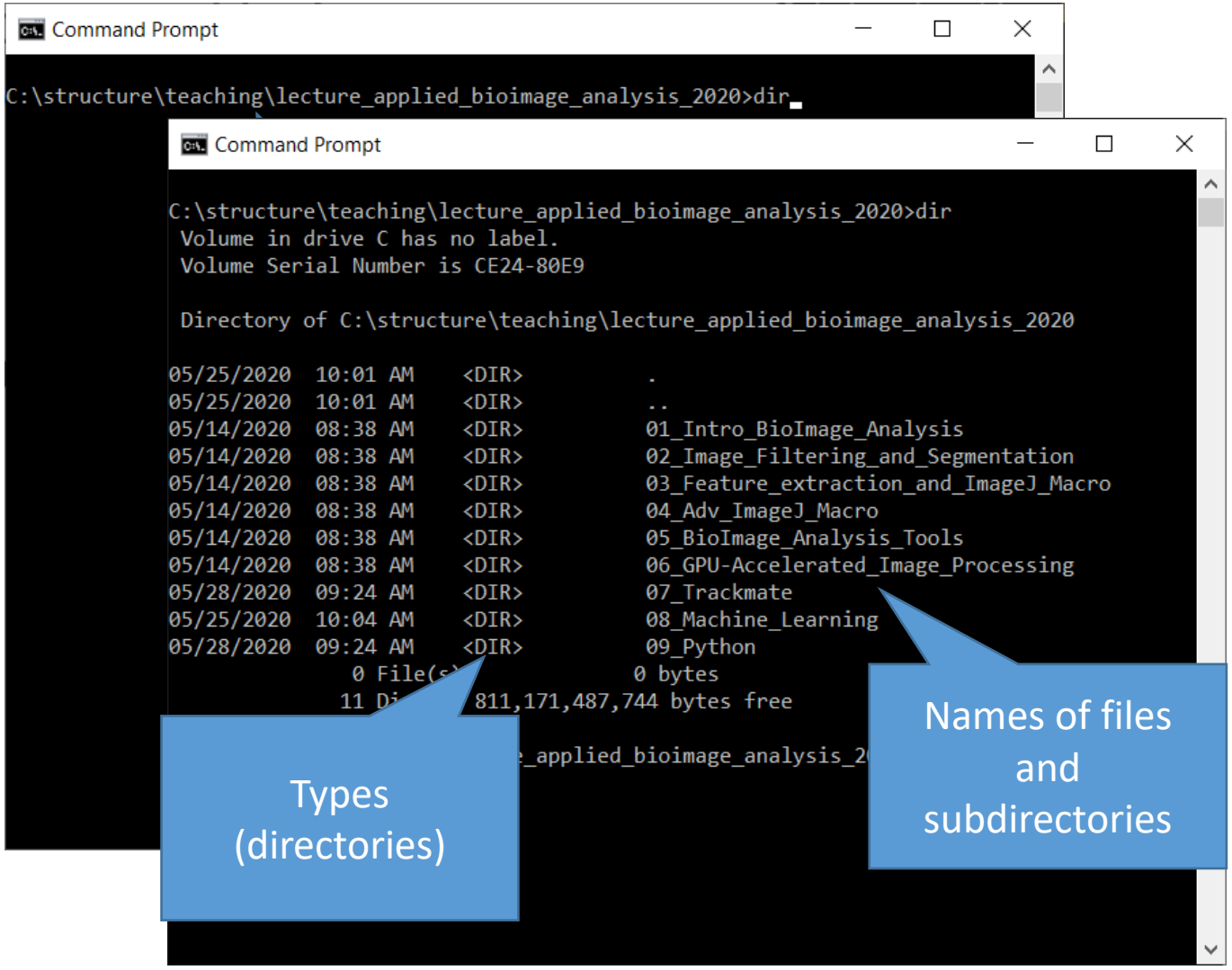
The command line

- A.k.a. the Terminal or Eingabeaufforderung: Welcome to the 20th century!

- The `dir` command tells you what's in the current directory

- On Mac and Linux the command is called `ls -l`



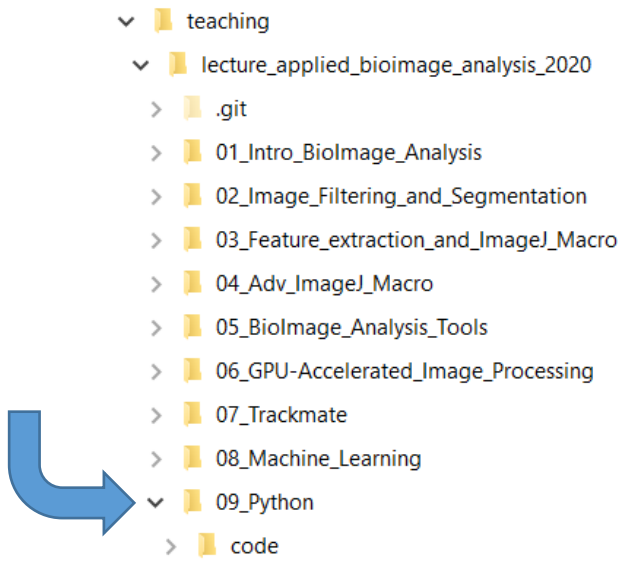


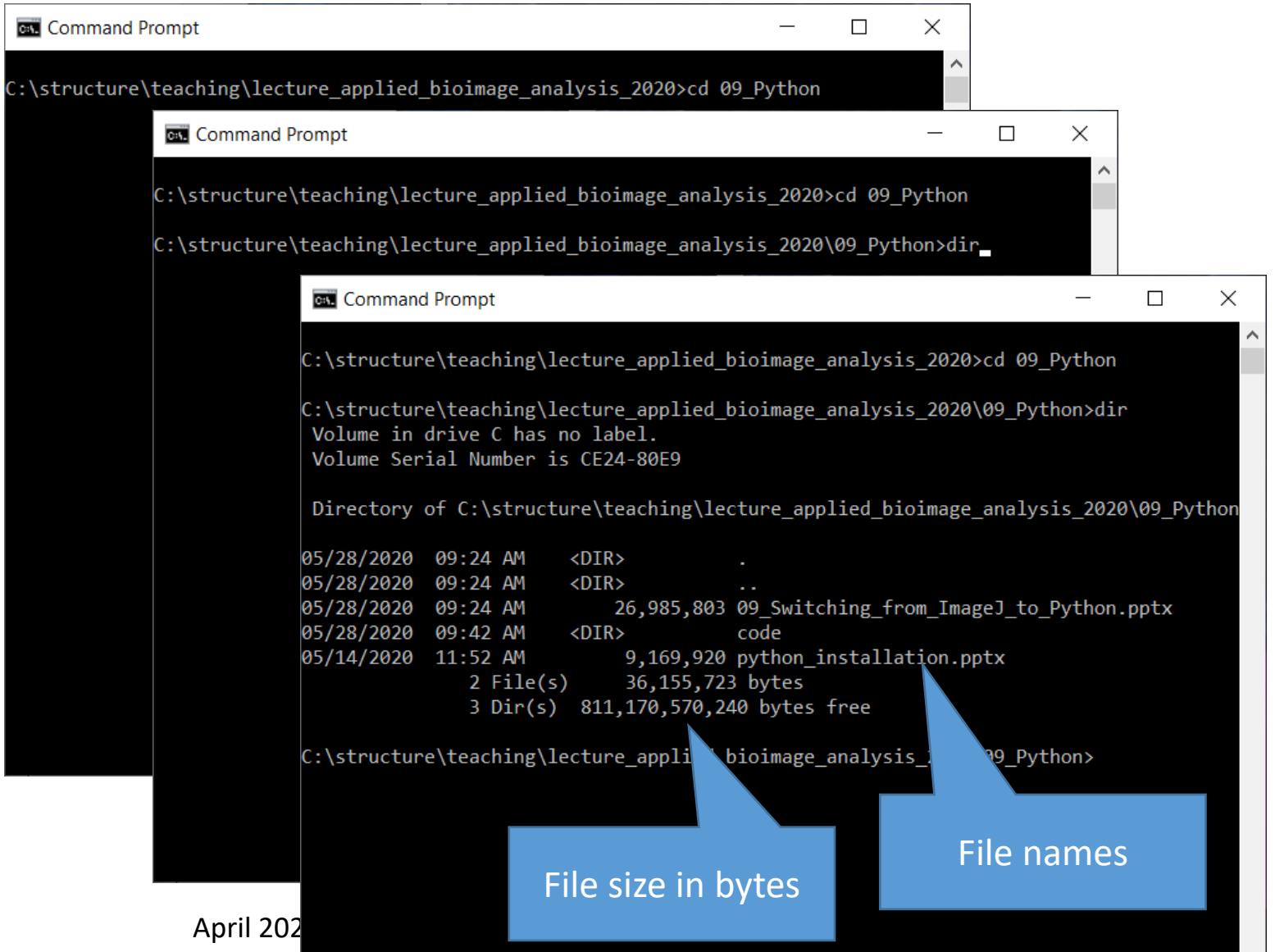
The command line

- A.k.a. the Terminal or Eingabeaufforderung: Welcome to the 20th century!

- The `cd` command let's you move between different directories.

- With `cd <pathname>` you go into a sub-directory

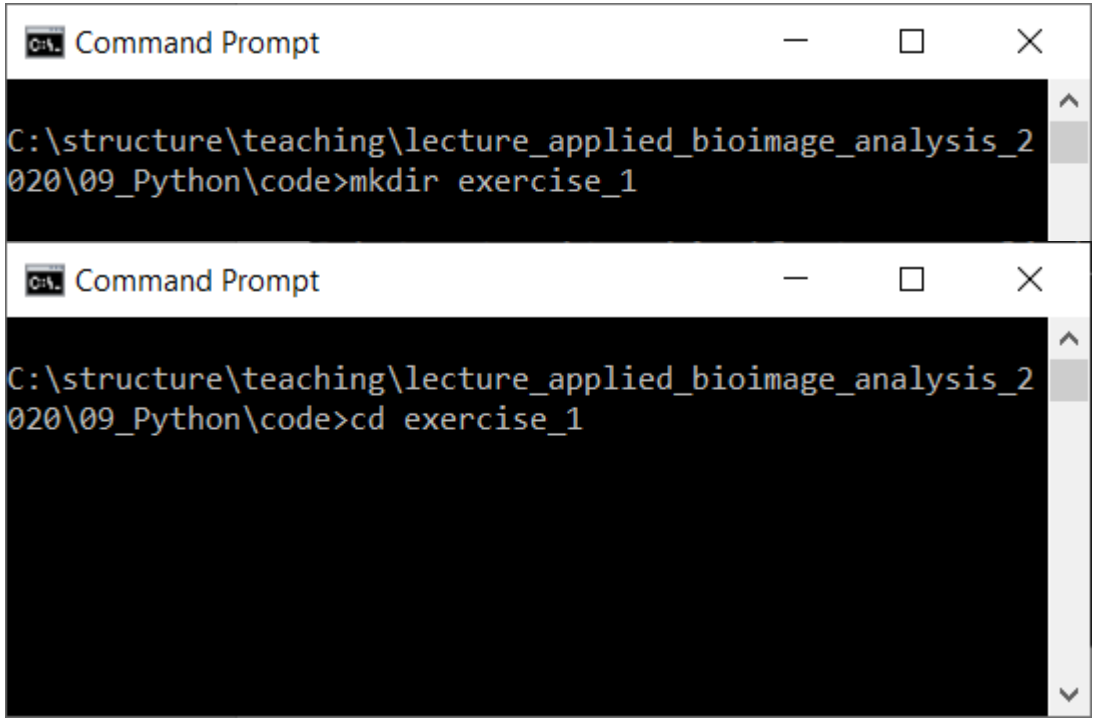


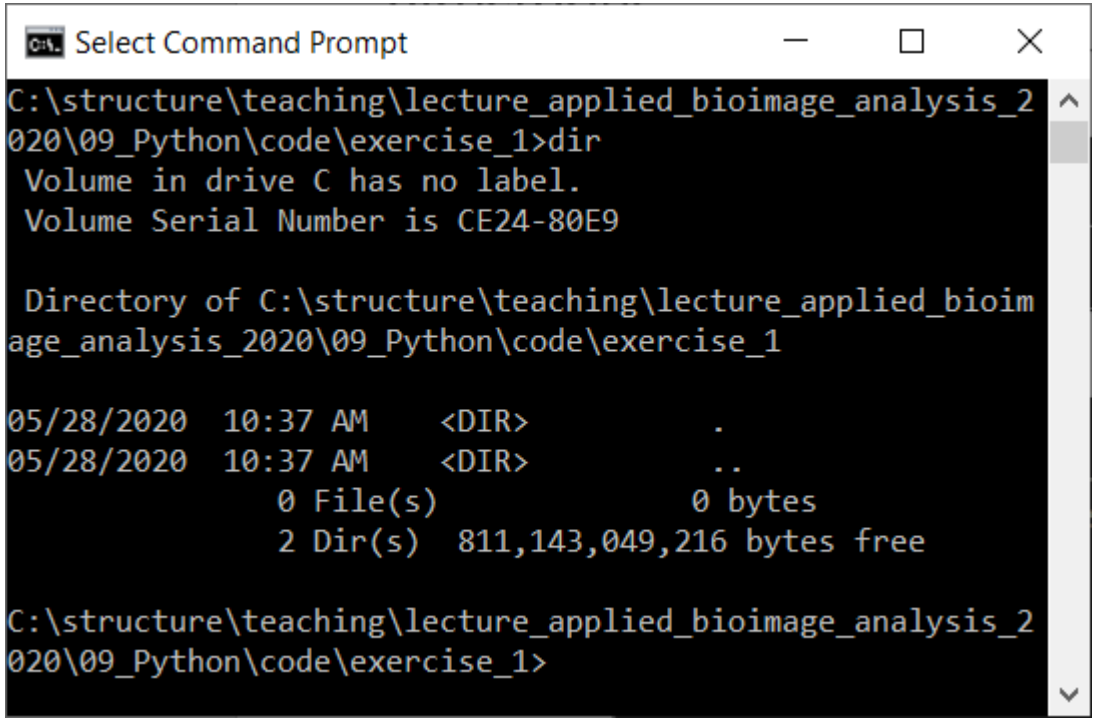


The command line

- A.k.a. the Terminal or Eingabeaufforderung: Welcome to the 20th century!

- The `mkdir` command creates new directories.



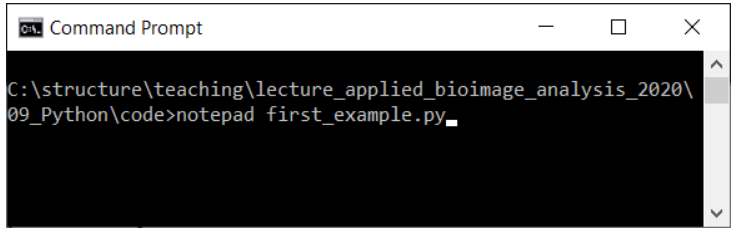


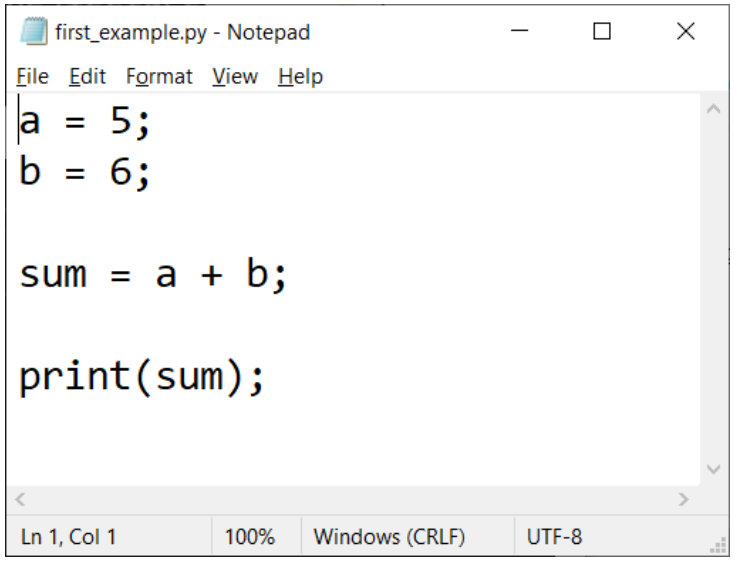
The command line

- Windows specific

- Notepad text editor

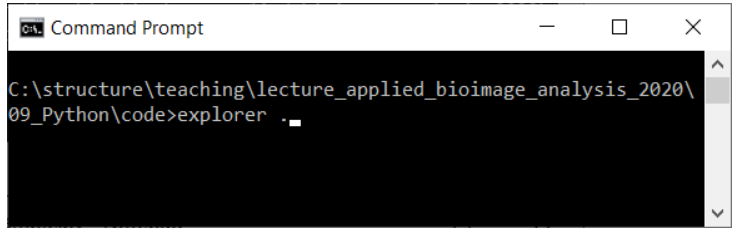
```
notepad <filename>
```

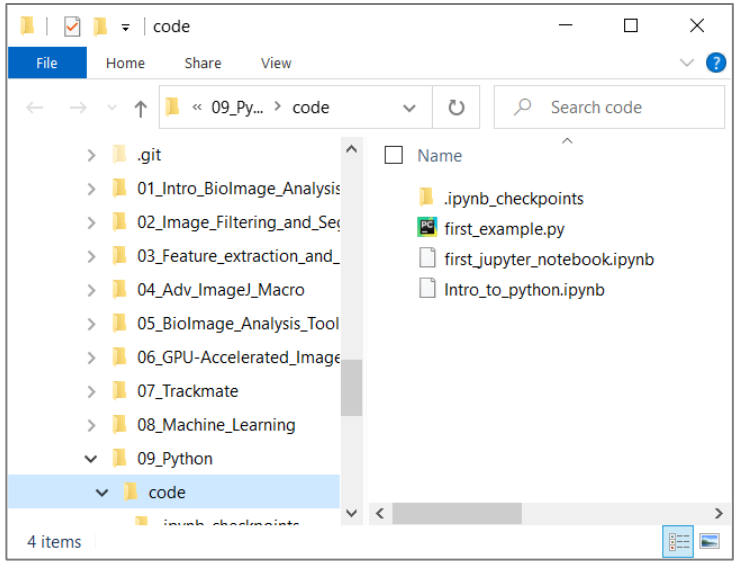




- Windows Explorer

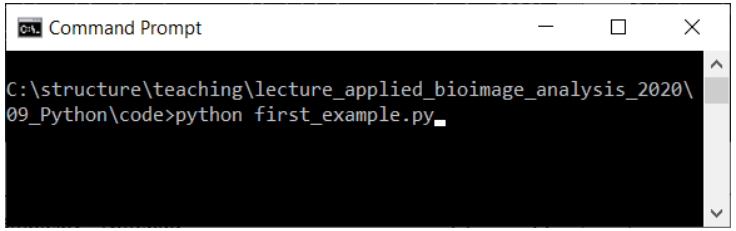
```
explorer .
```

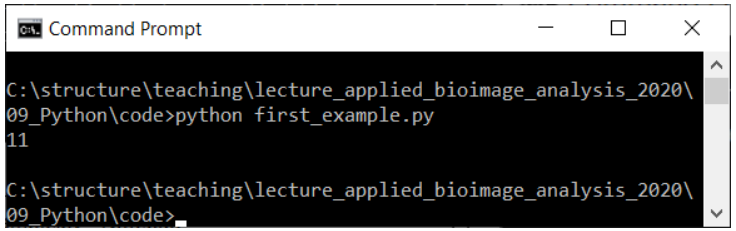




- Execute Python script

```
python <filename>
```





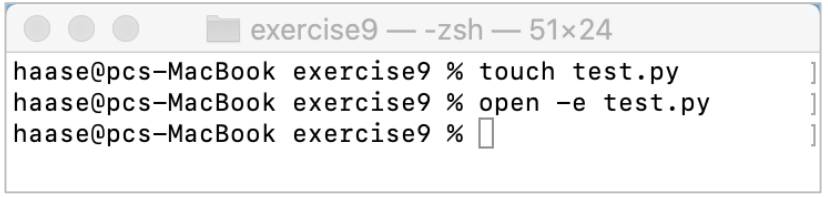
The command line

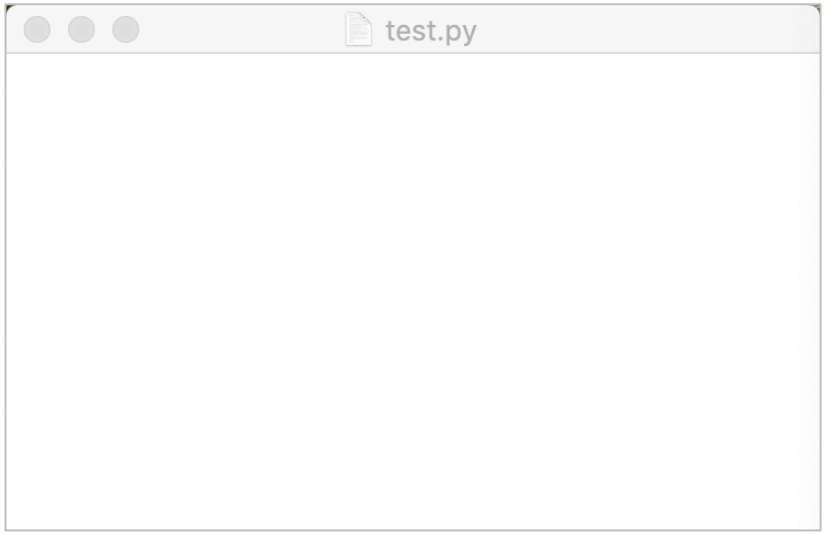
- Mac OS specific

- Text editor

```
touch <filename>
open -e <filename>
```

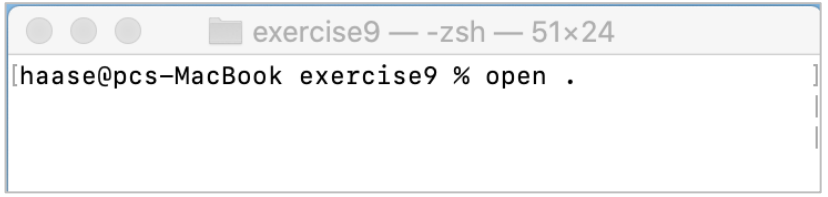
Create a new file

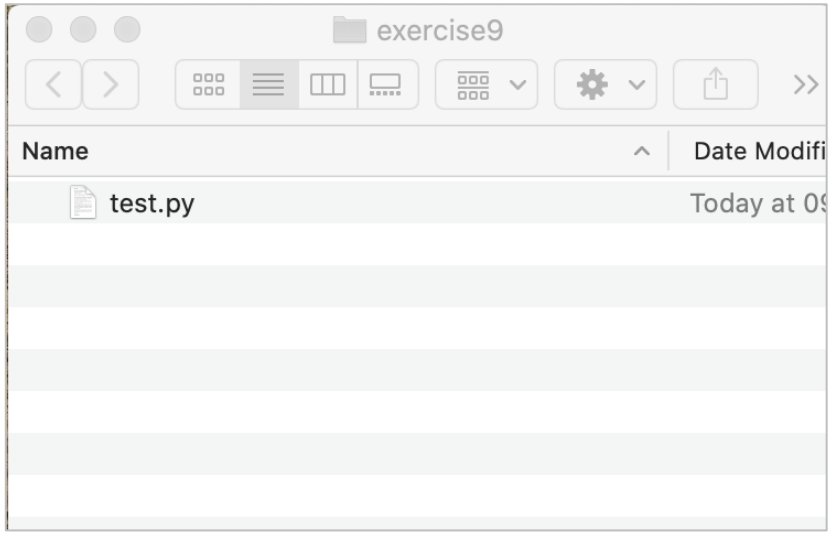




- Finder

```
open .
```





- Execute Python script

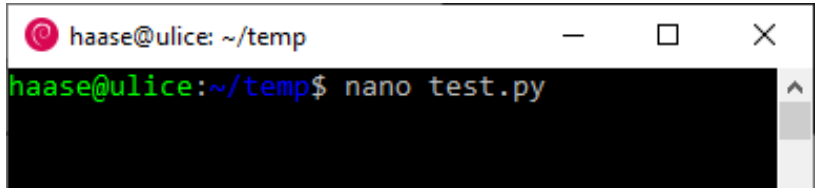
```
python <filename>
```

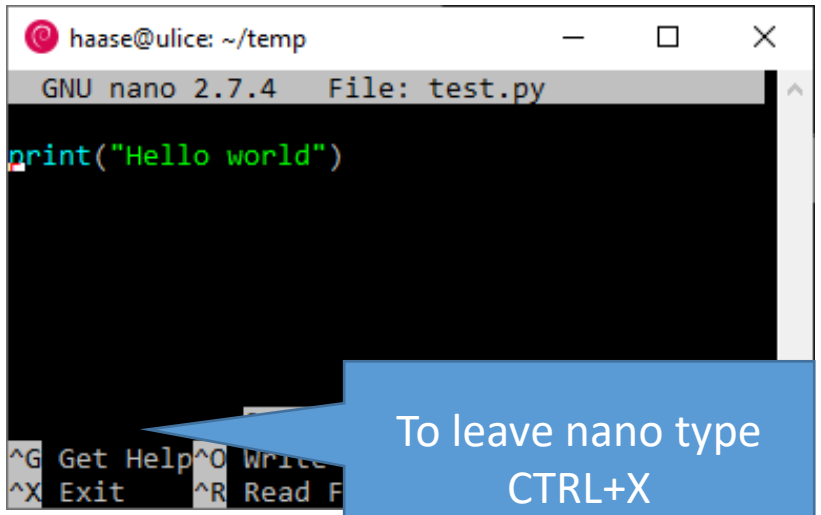
The command line

- Linux specific

- Nano text editor

`nano <filename>`

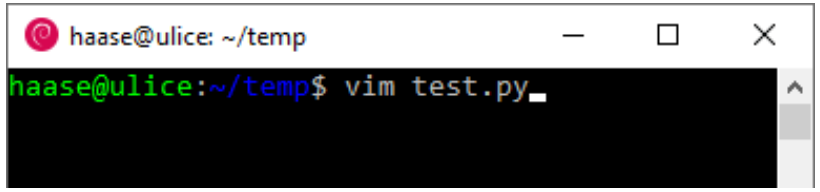


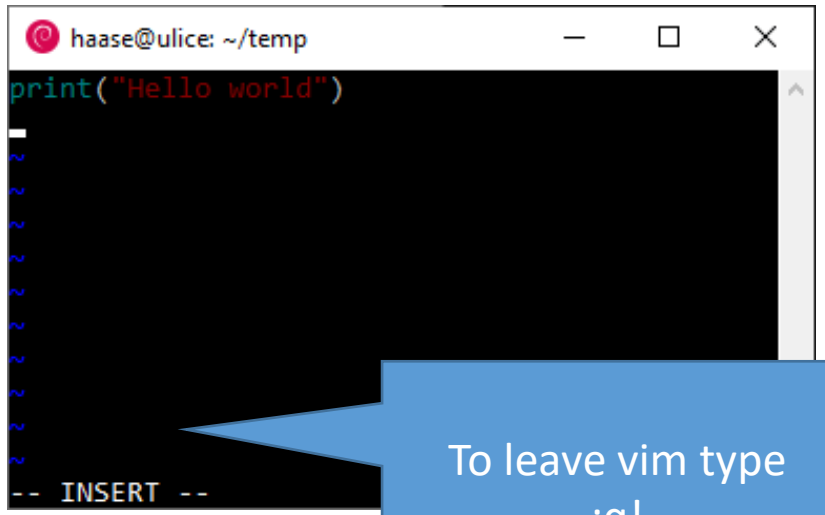


To leave nano type
CTRL+X
N

- vim

`vim <filename>`

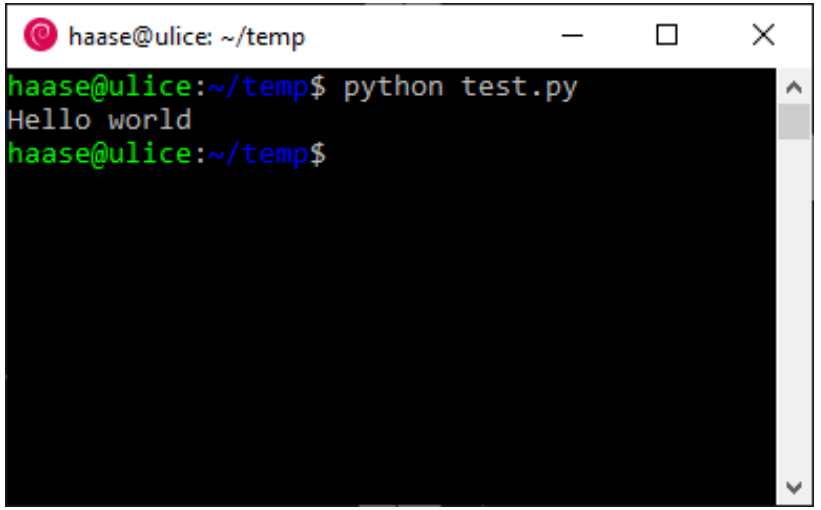




To leave vim type
:q!

- Execute Python script

`python <filename>`



PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

jupyter lab

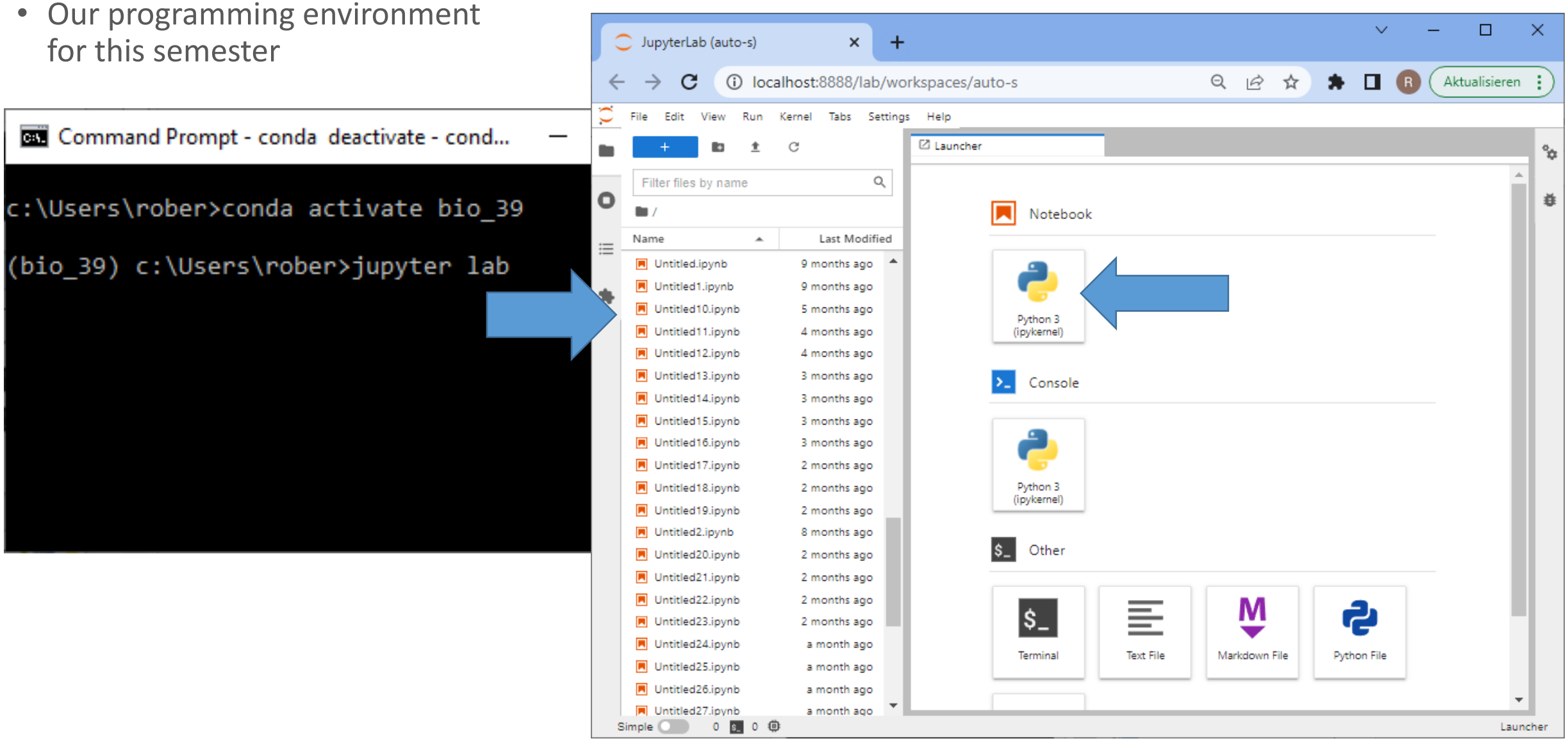
Robert Haase

April 2022

@haesleinhuepf

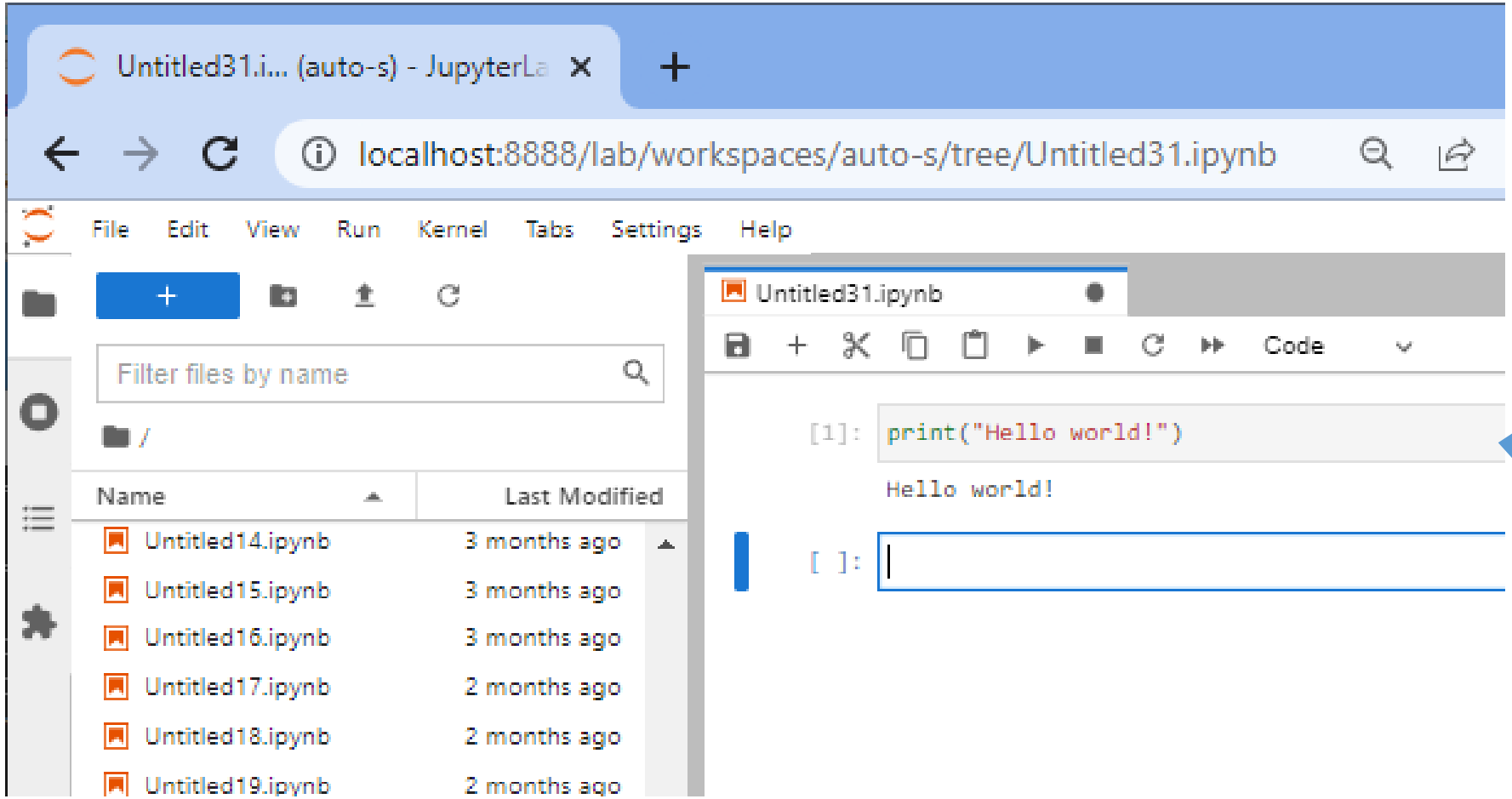
Jupyter lab

- Our programming environment for this semester



Jupyter lab

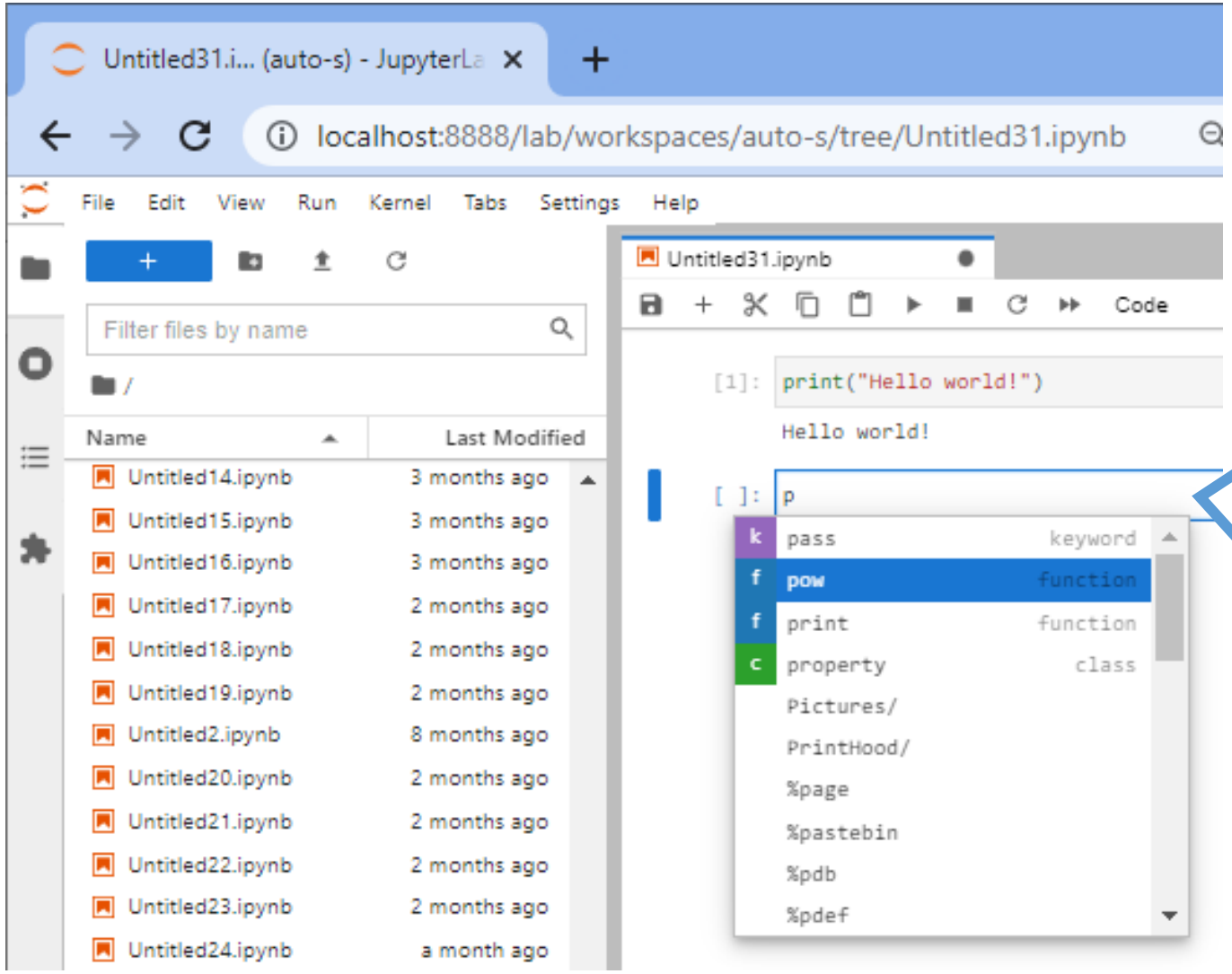
- Execute code cell-by-cell and see results instantaneously



SHIFT + ENTER to execute a code cell

Jupyter lab

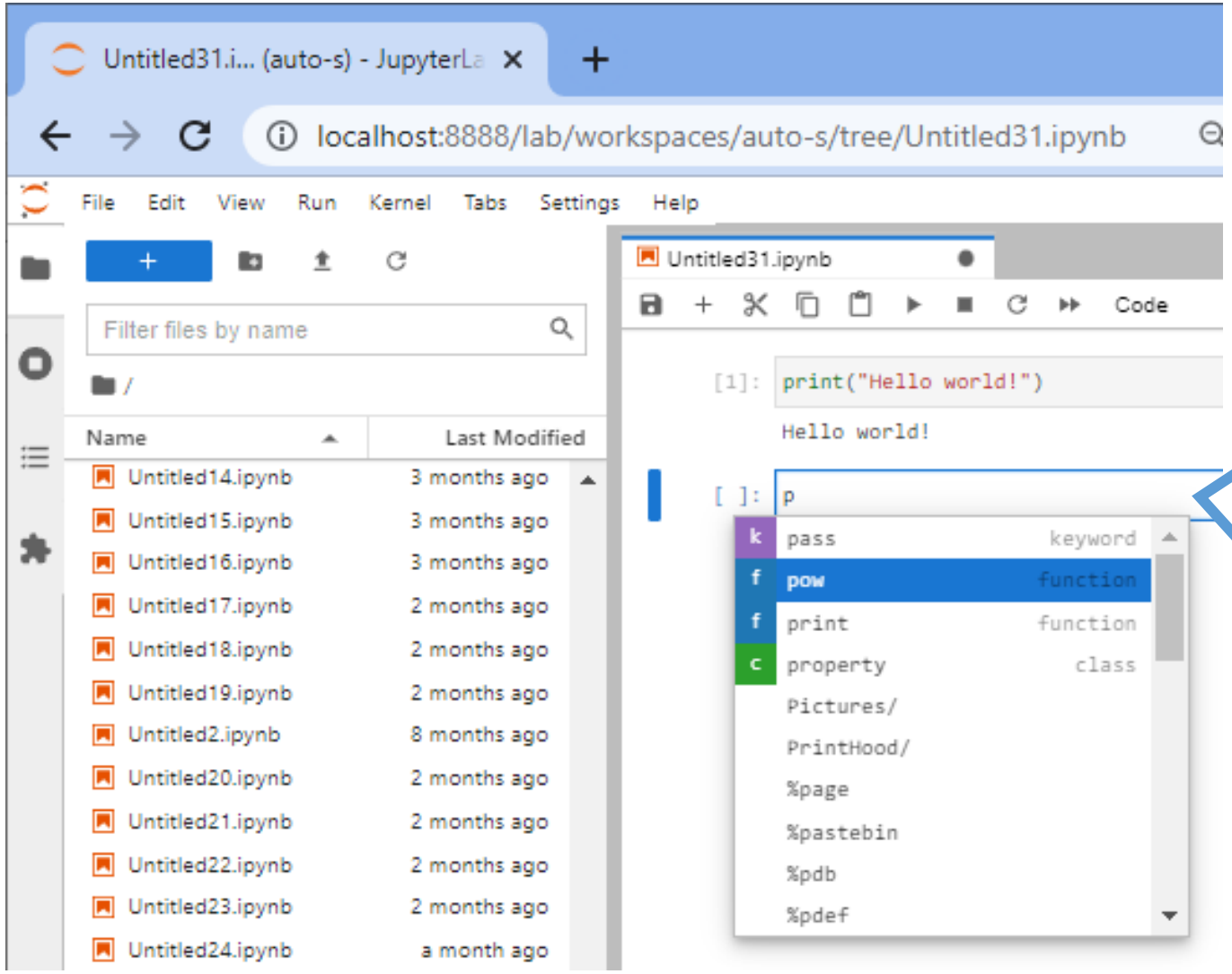
- Context-specific help, auto-completion



TAB
to open auto-completion

Jupyter lab

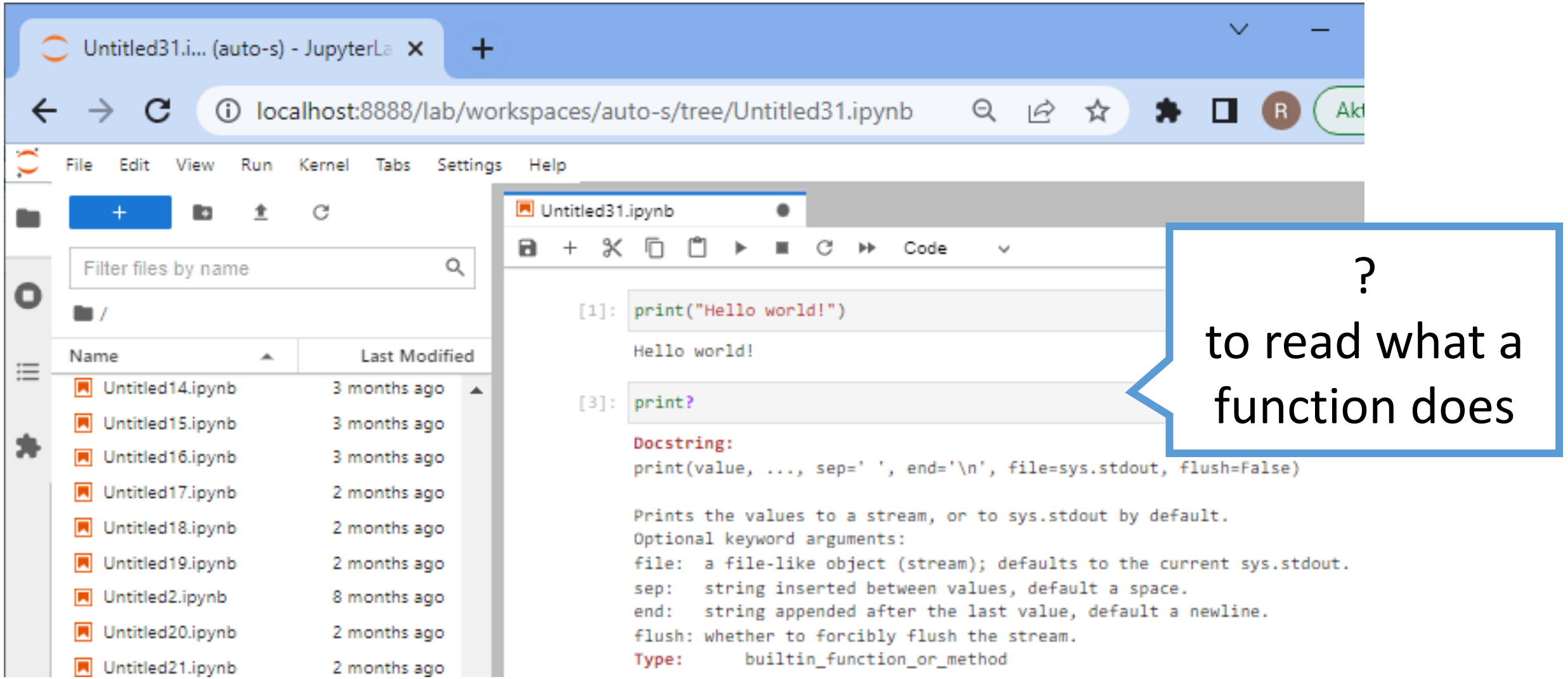
- Context-specific help, auto-completion



TAB
to open auto-completion

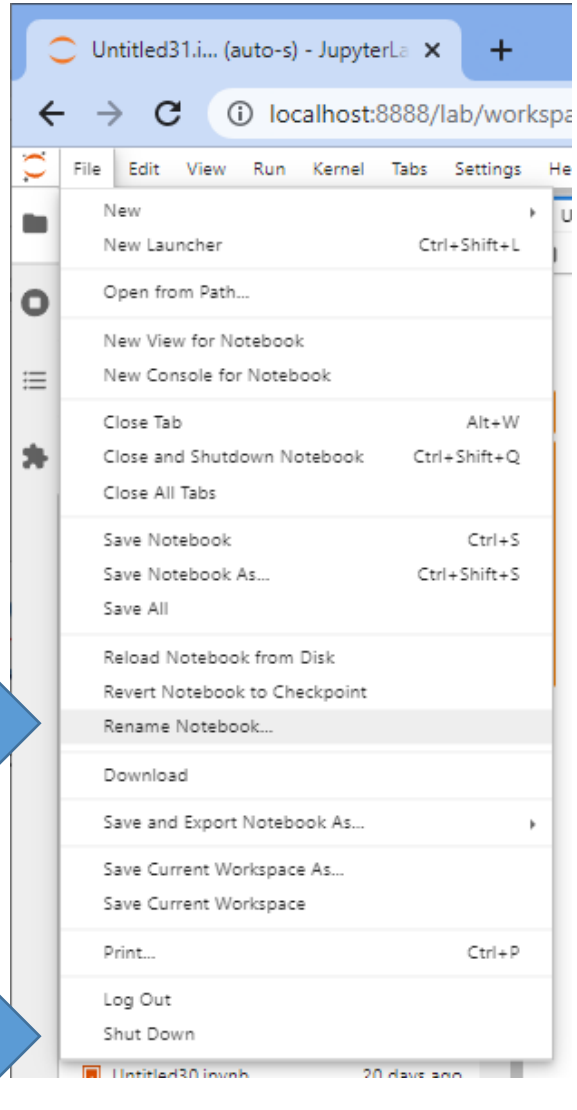
Jupyter lab

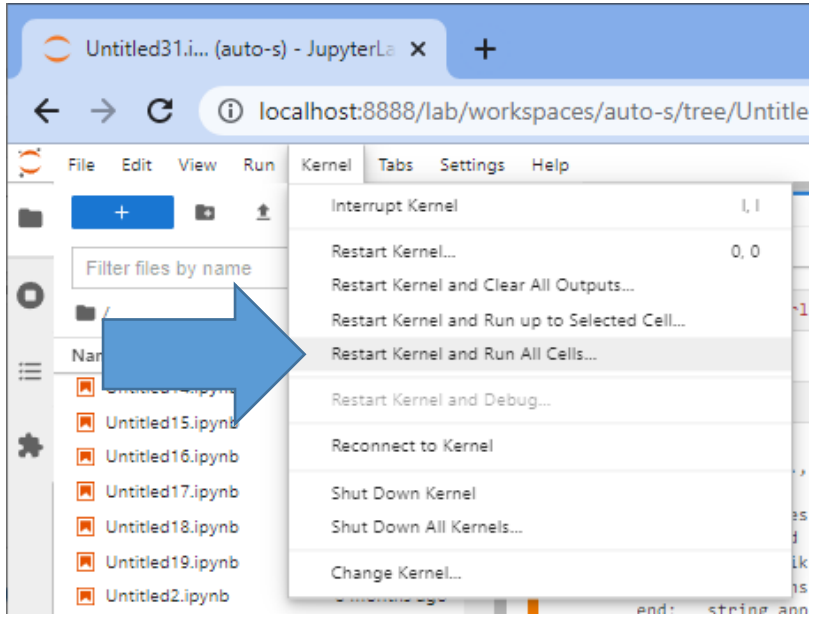
- Help / “docstrings”



Jupyter lab

- Saving / renaming / closing





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

PoL
Physics of Life
TU Dresden

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Python Programming

Robert Haase

April 2022

@haesleinhuepf

Data science with python

PoL Physics of Life TU Dresden

- 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 - C

Star-convex

This repository contain

object detection for 2D

- Uwe Schmidt, Ma
  Broaddus, and Ge
  Cell Detection wit
  International Conf
  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`

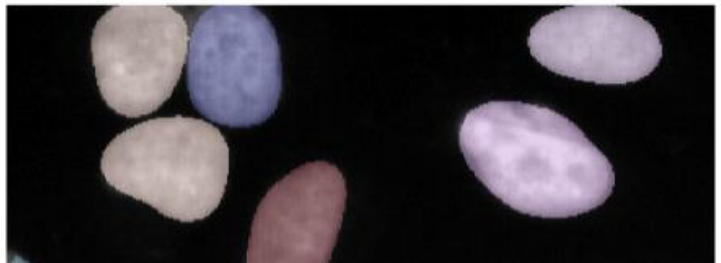
```
from s
# pri
StarDi
# crea
model
```

stardist/3_prediction.ipynb at ma

github.com/stardist/stardist/blob/master/examples/2D/3_prediction.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');
```



Annot

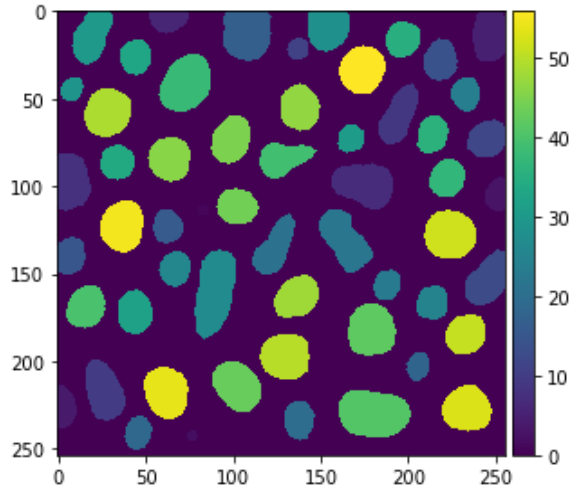
To train
annotati
correspo
labeled
with 0).
among t
options:
having a

```
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
```

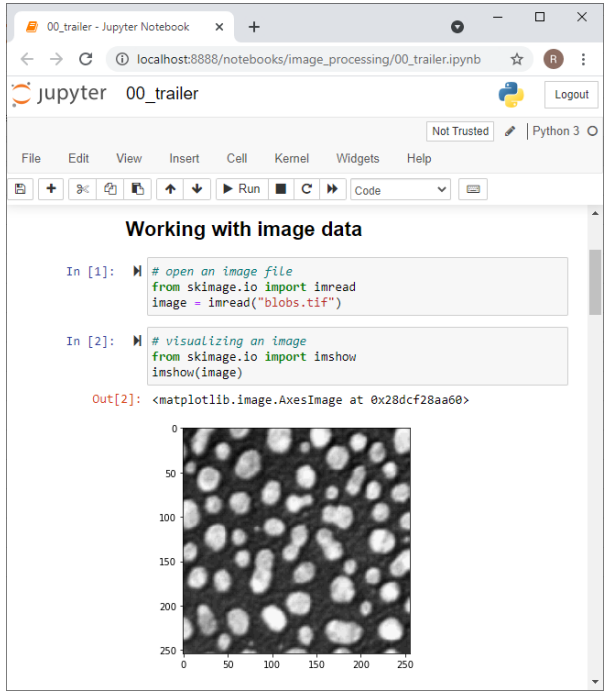
```
]: <matplotlib.image.AxesImage at 0x28dd9f991c0>
```

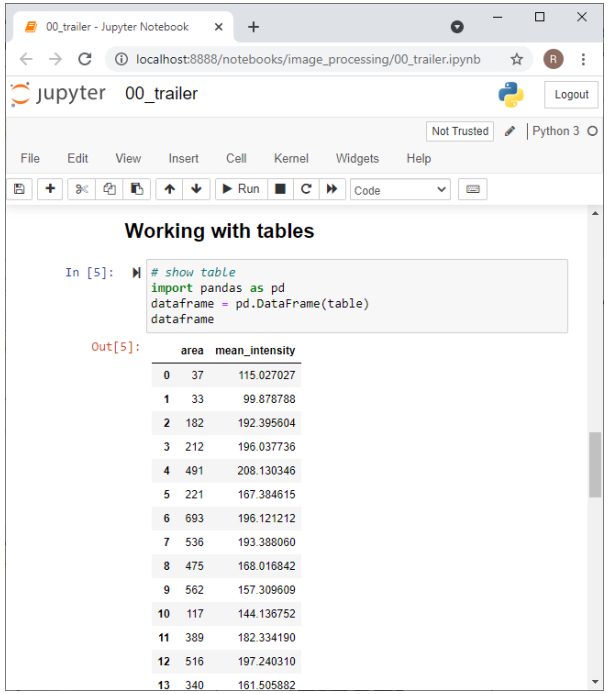


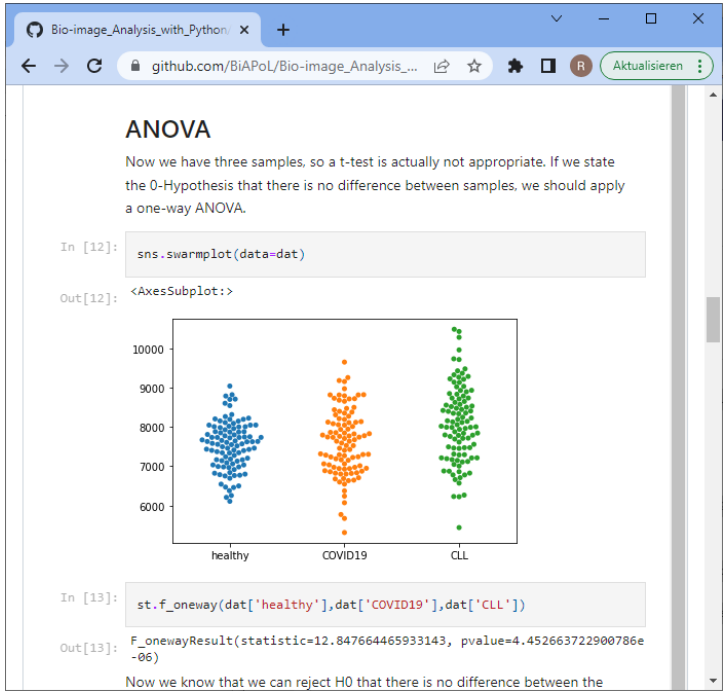
https://github.com/stardist/stardist

Python

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



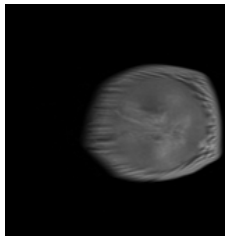




How to eat a banana the computational way using Python

PoL Physics of Life TU Dresden

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

Python programming basics

Robert Haase

April 2022

@haesleinhuepf

Working with variables

- Variables can hold numeric values and you can do math with them

```
# initialize program
a = 5
b = 3

# run algorithm on given parameters
sum = a + b

# print out result
print (sum)
```

```
8
```

Mathematical operations

- Math commands supplement operators to be able to implement any form of calculations

- Power

```
In [ ]: pow(3, 2)
Out[ ]: 9
```

- Absolute

```
In [ ]: abs(-8)
Out[ ]: 8
```

- Rounding

```
In [ ]: round(4.6)
Out[ ]: 5
```

Be careful with some of them!

```
In [ ]: round(4.5)
Out[ ]: 4
```

https://en.wikipedia.org/wiki/Rounding#Round_half_to_even

Comments

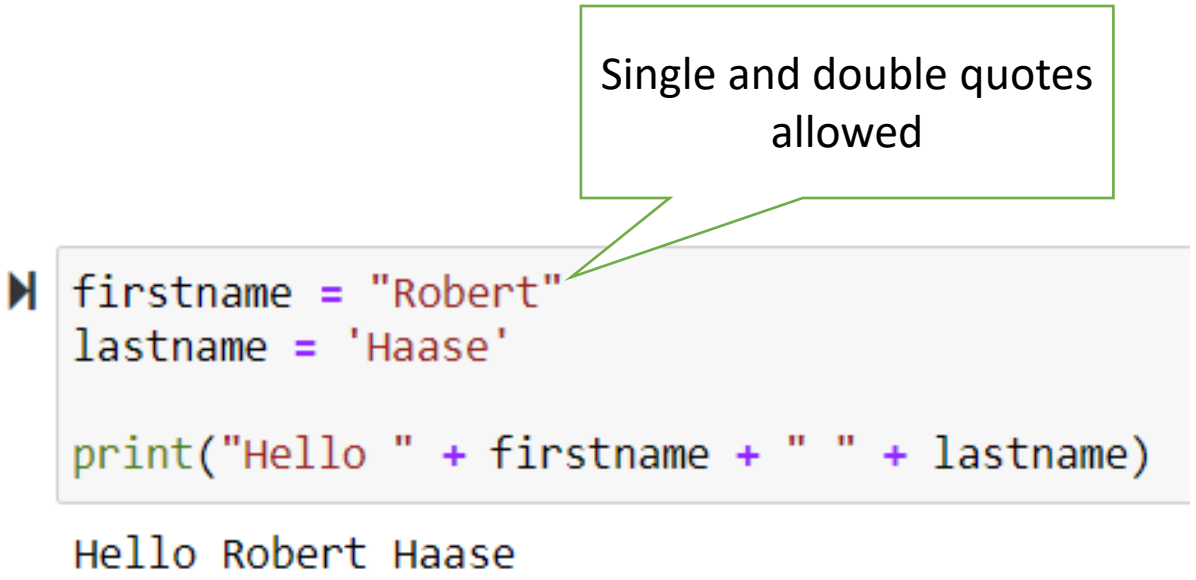
Comments should contain additional information such as

- 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 )
```

Working with variables and string values

- Also strings as values for variables are supported



Working with variables and string values

PoL Physics of Life TU Dresden

- Also strings as values for variables are supported
- When combining strings and numbers, you need to explicitly define what you want to do.

```
# mixing types

a = 5
b = "2"

print (a + b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-4-51629e6a285f> in <module>
      4 b = "2"
      5
----> 6 print (a + b)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

```
# mixing types

a = "5"
b = 2

print (a + b)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-85ae49867097> in <module>
      4 b = 2
      5
----> 6 print (a + b)

TypeError: can only concatenate str (not "int") to str
```

```
# mixing types to make numbers

a = 5
b = "2"

print (a + int(b))
```

```
7
```

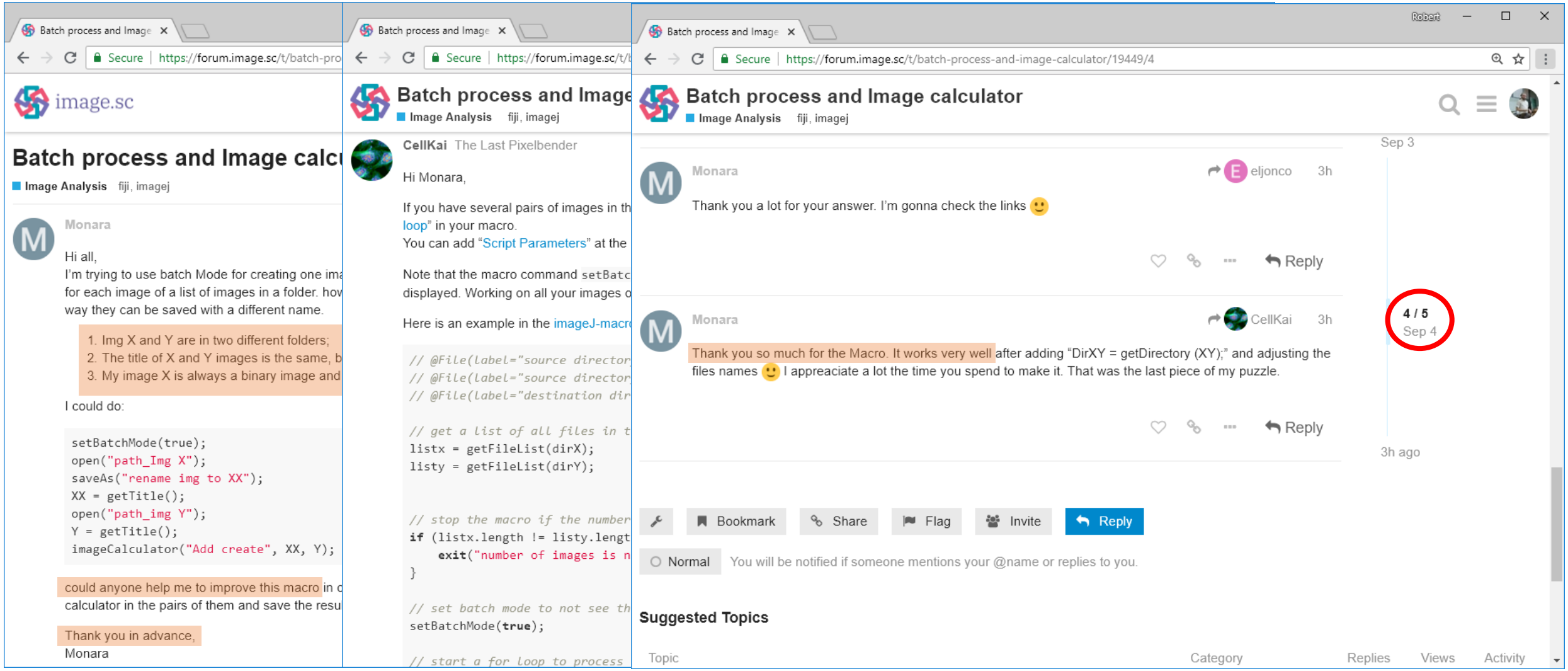
```
# mixing types to make strings

a = "5"
b = 2

print (a + str(b))
```

```
52
```

- Conversion to a floating point number: float()

Image science online community

- Visit http://image.sc, an online Q&A platform for the image sciences hosted by University Visconsin and the Broad institute of Harvard and MIT.



CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Demo

Robert Haase

April 2022

@haesleinhuepf

CENTER FOR
SYSTEMS BIOLOGY
DRESDEN

Homework

Robert Haase

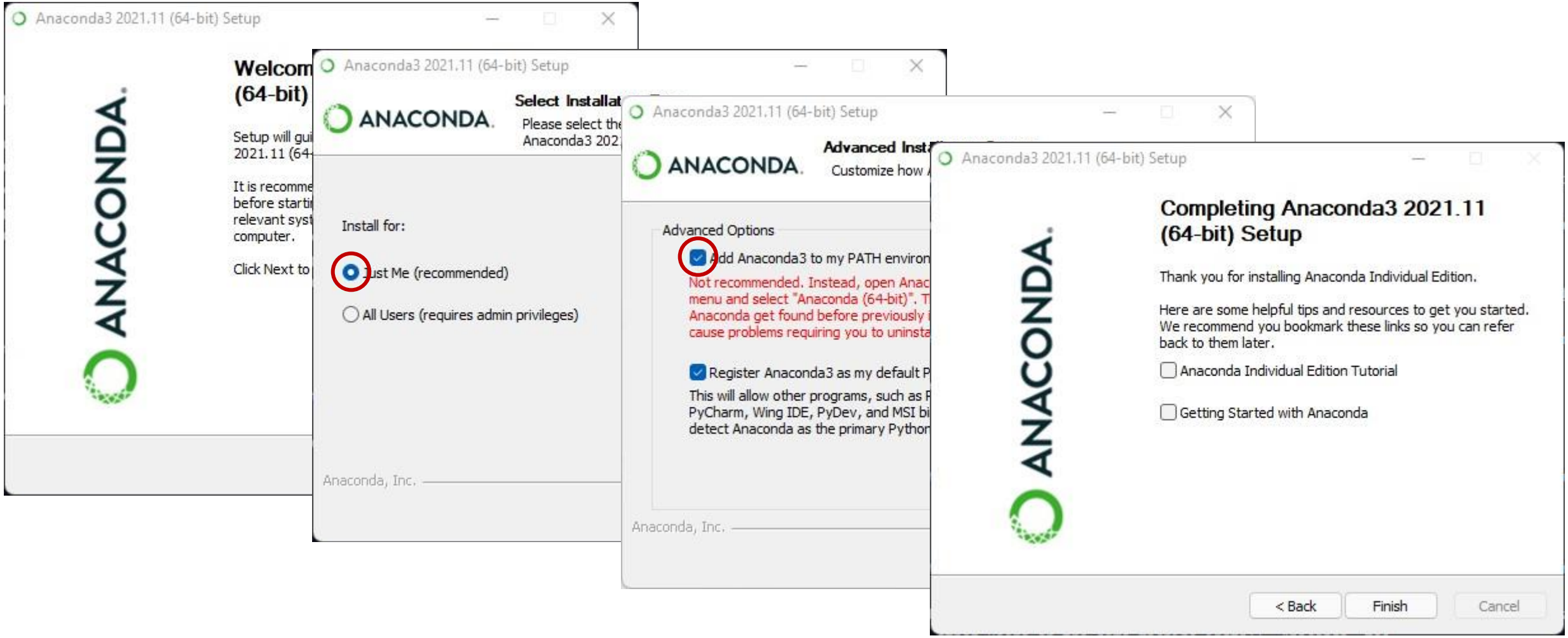
April 2022

@haesleinhuepf

Homework: Install anaconda and test python

Detailed instructions:

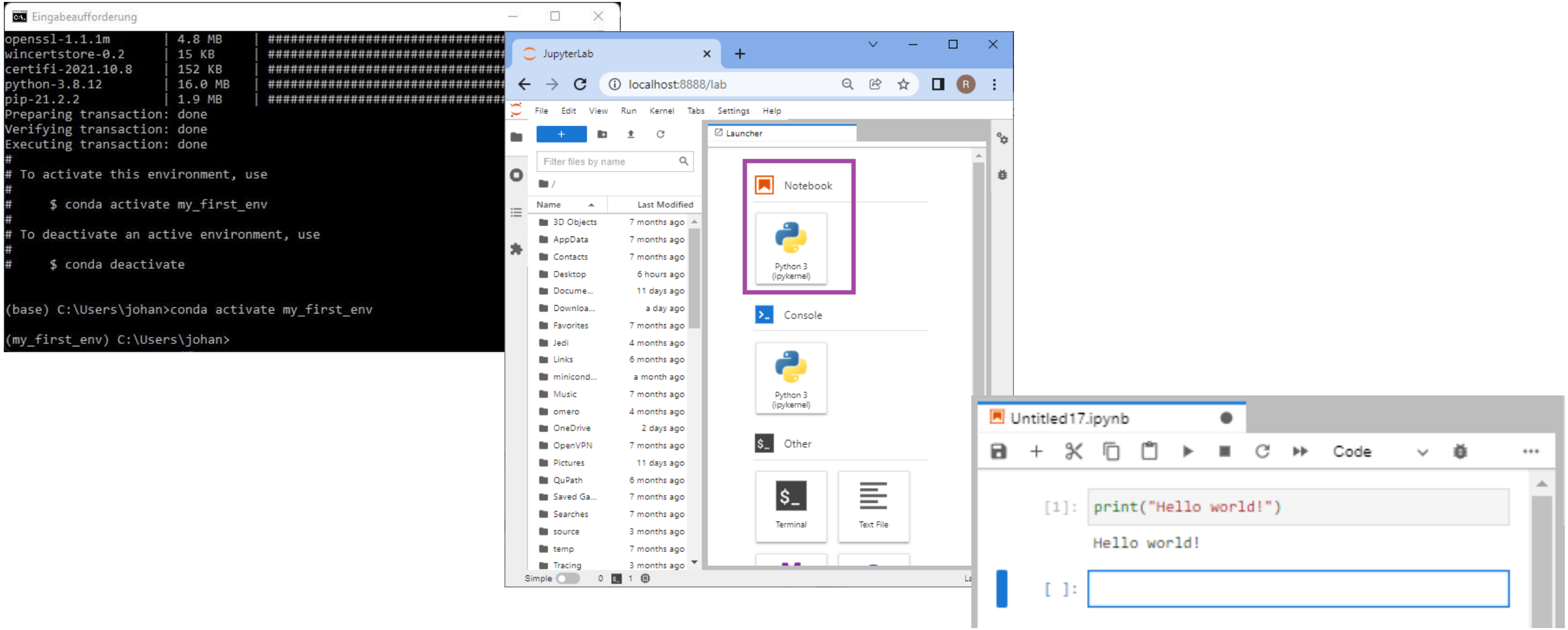
- https://biapol.github.io/blog/johannes_mueller/anaconda_getting_started/



Homework: Install anaconda and test python

Detailed instructions:

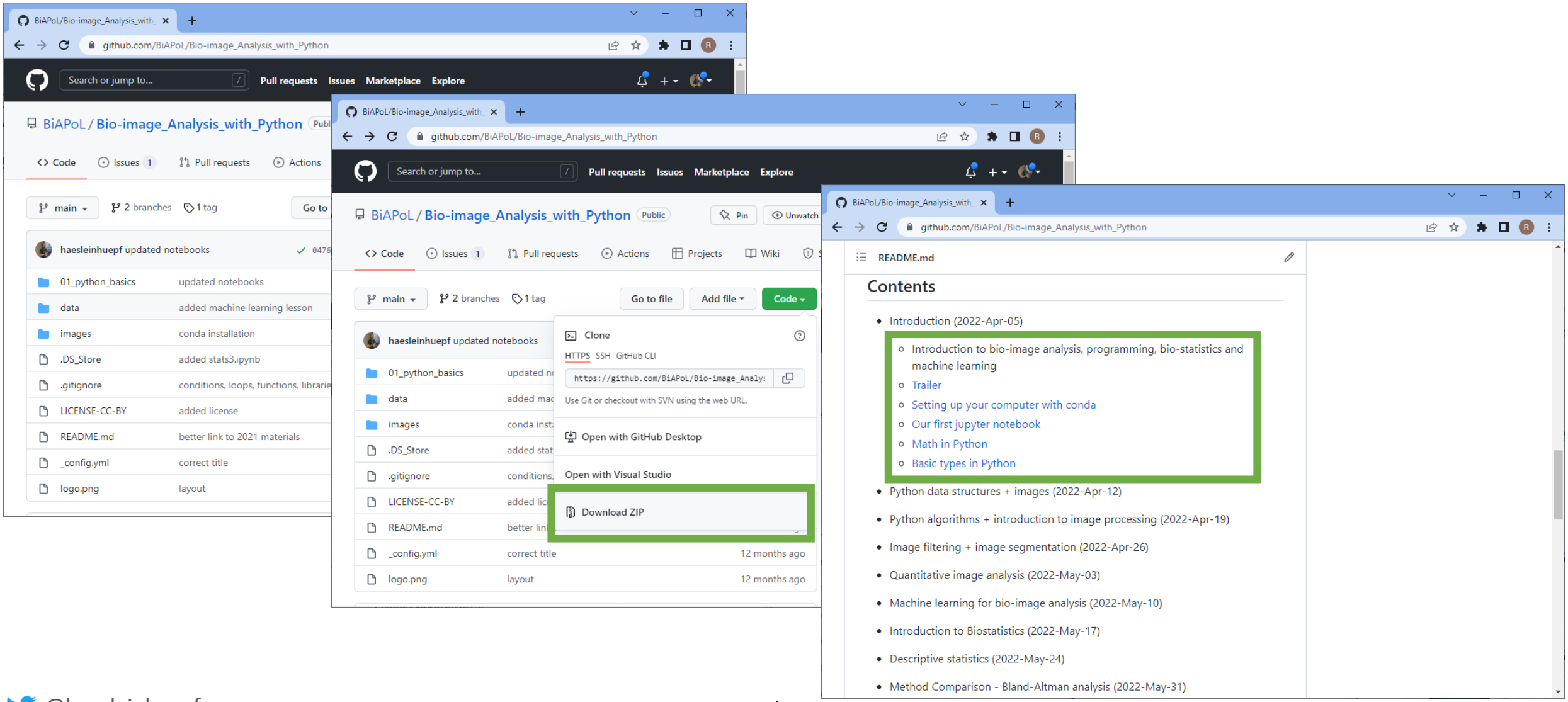
- https://biapol.github.io/blog/johannes_mueller/anaconda_getting_started/



Homework: Explore our github repository

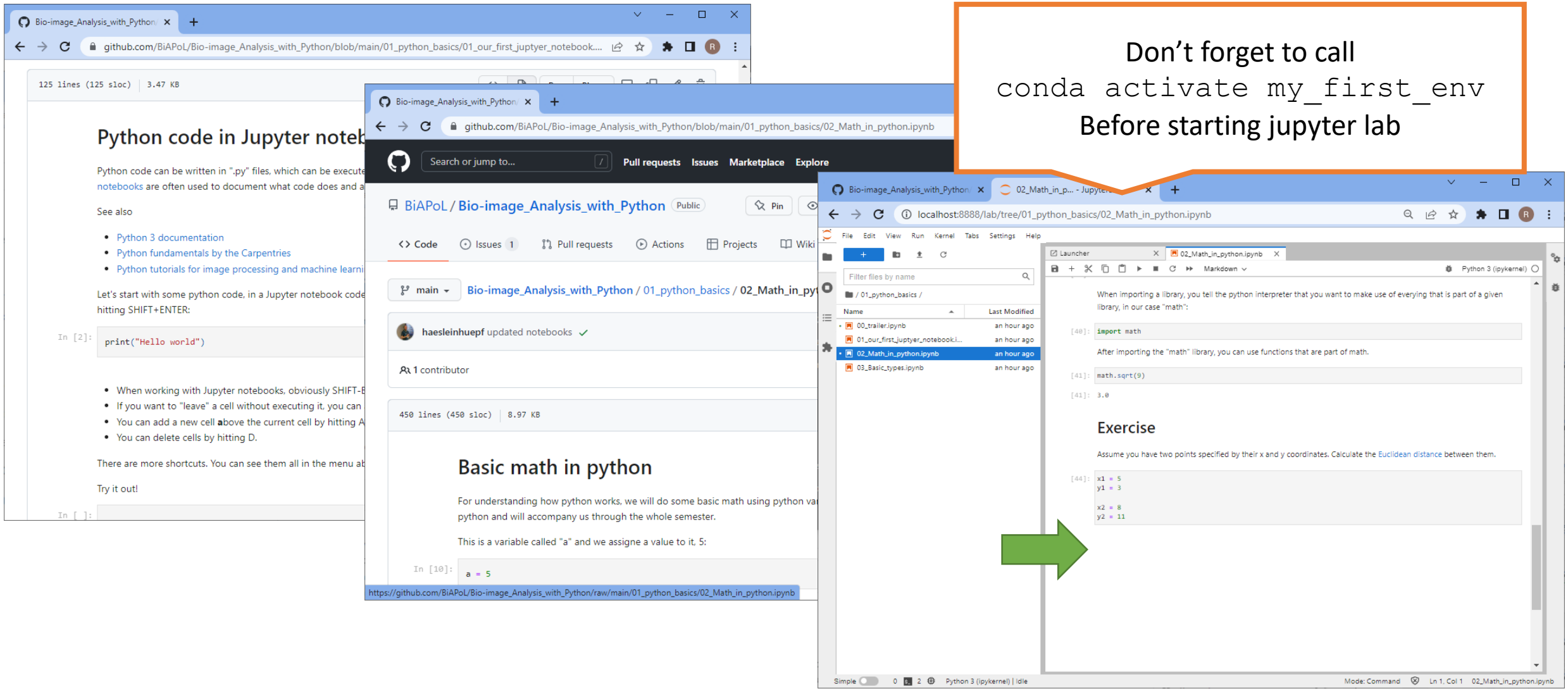
PoL Physics of Life TU Dresden

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

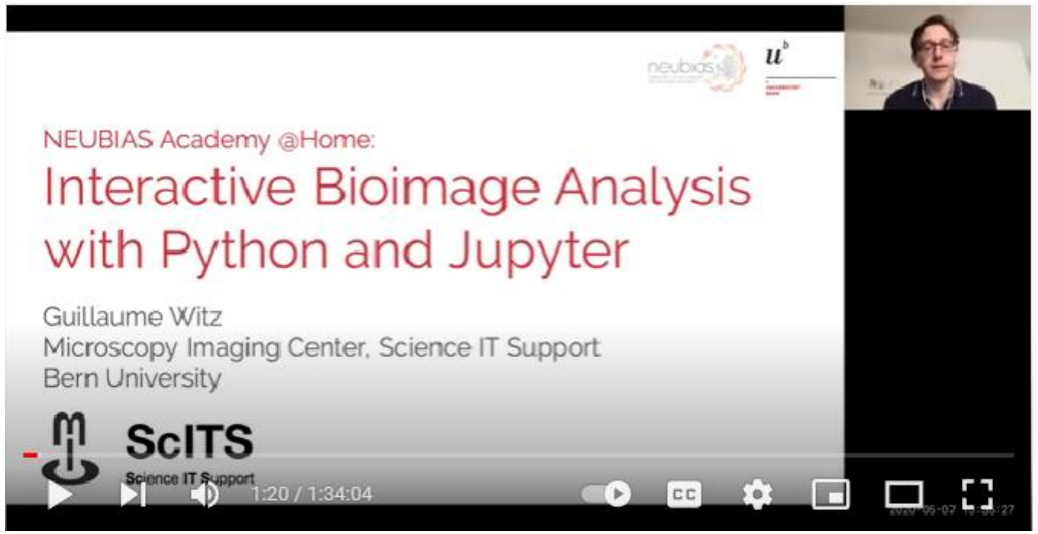


Homework: Basic python

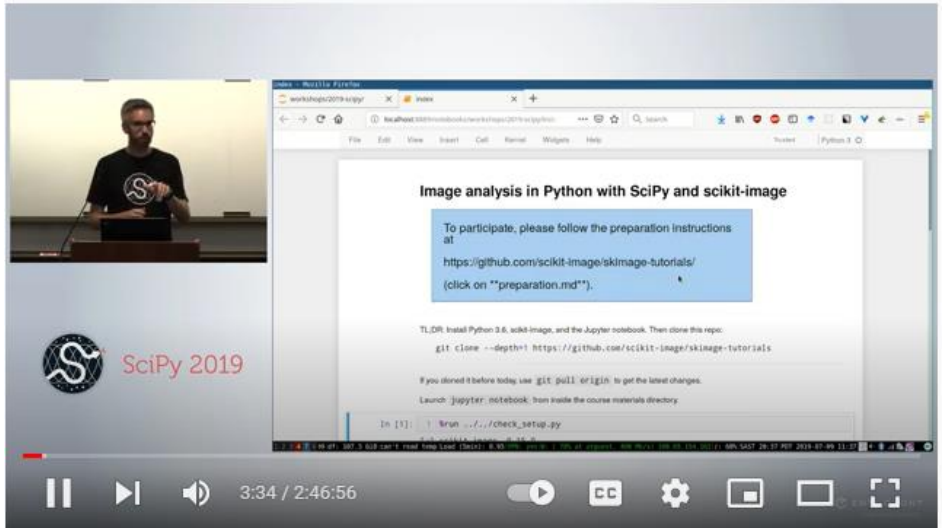
- https://github.com/BiAPoL/Bio-image_Analysis_with_Python



More resources



Guillaume Witz, NEUBIAS Academy 2020



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

Summary

PoL Physics of Life TU Dresden

Today, you learned

- *Bio-image analysis*
  - Quantitative
  - Objective
  - Reproducible
  - Repeatable
  - Reliable
- The command line
- Python basics

- Reminder:
  - Use the etherpad to ask questions anonymously!
  - Alternatively: Ask on https://image.sc in public.

Coming up next

- Python programming
  - Lists, tuples and dictionaries
  - Tables
  - Image data

```
# going through arrays pair-wise
measurement_1 = [1, 9, 7, 1, 2, 8, 9, 2, 1, 7, 8]
measurement_2 = [4, 5, 5, 7, 4, 5, 4, 6, 6, 5, 4]

for m_1, m_2 in zip(measurement_1, measurement_2):
    print("Paired measurements: " + str(m_1) + " and " + str(m_2))
```

```
Paired measurements: 1 and 4
Paired measurements: 9 and 5
Paired measurements: 7 and 5
Paired measurements: 1 and 7
Paired measurements: 2 and 4
Paired measurements: 8 and 5
Paired measurements: 9 and 4
Paired measurements: 2 and 6
Paired measurements: 1 and 6
Paired measurements: 7 and 5
Paired measurements: 8 and 4
```