Technologies for microscopic image analysis
显微图像分析技术

第5课

Takao, Daisuke 高尾大辅

dtakao@mail.hzau.edu.cn

Overview of this course

第一章　光学显微术　Optical microscopy

1. History of microscopy
2. Principles of optics
3. Bright field microscopy
4. Fluorescence microscopy

第二章　图像数据处理与分析　Image data processing and analysis

1. Processing of digital image data
2. Quantitative image analysis
3. **Application of ImageJ/Fiji software**
4. **Image processing and analysis with the Python coding environment**
5. Bioimage informatics

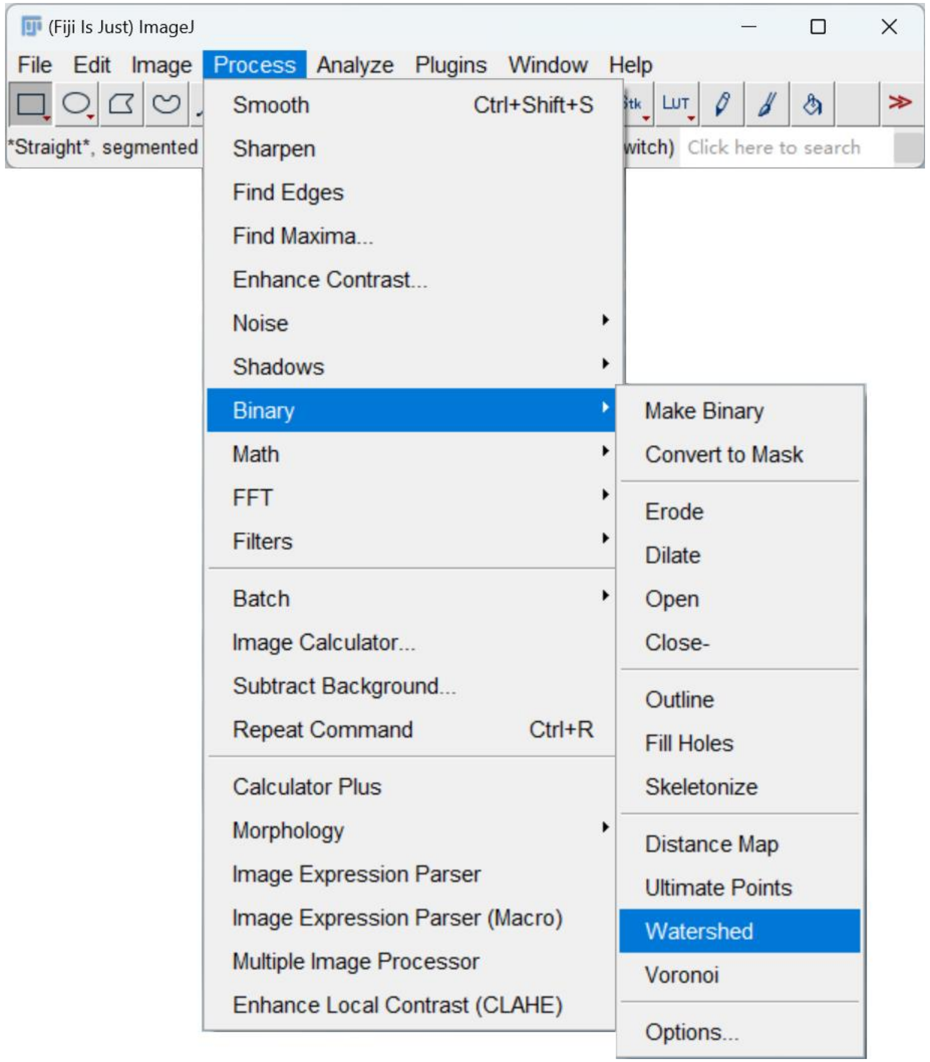
第三章　先进的显微术　Advanced microscopy

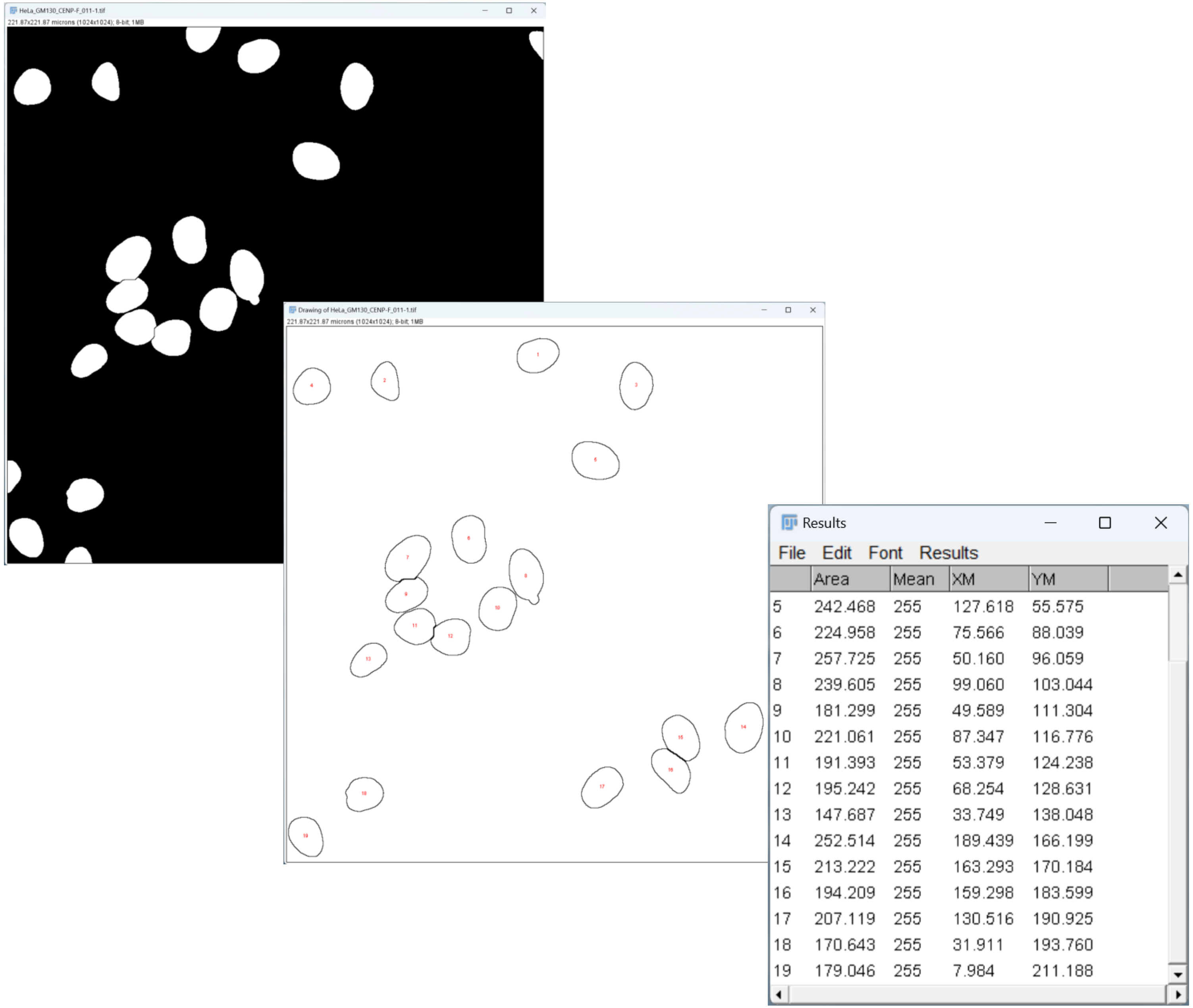
1. Advanced imaging technology: from single molecules to living animals
2. Super-resolution microscopy
3. Other bioimaging techniques

第四章　人工智能的应用　Application of AI

1. Overview of AI technology
2. AI for bioimaging

Image processing and analysis using Fiji/ImageJ




HeLa_GM130_CENP-F_011-1.tif
221.87x221.87 microns (1024x1024); 8-bit; 1MB
Drawing of HeLa_GM130_CENP-F_011-1.tif
221.87x221.87 microns (1024x1024); 8-bit; 1MB
Results
File
Edit
Font
Results

	Area	Mean	XM	YM
5	242.468	255	127.618	55.575
6	224.958	255	75.566	88.039
7	257.725	255	50.160	96.059
8	239.605	255	99.060	103.044
9	181.299	255	49.589	111.304
10	221.061	255	87.347	116.776
11	191.393	255	53.379	124.238
12	195.242	255	68.254	128.631
13	147.687	255	33.749	138.048
14	252.514	255	189.439	166.199
15	213.222	255	163.293	170.184
16	194.209	255	159.298	183.599
17	207.119	255	130.516	190.925
18	170.643	255	31.911	193.760
19	179.046	255	7.984	211.188

Tutorial: Image processing and analysis with Python

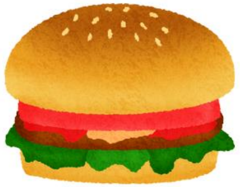
- Use image data from a published paper (Nagao et al., MBoC, 2020)
- Learn methods of segmentation, measurements, drawing graphs, etc

Preparation of the Python coding environment

Install and setup Anaconda (Python + other powerful packages for data science, informatics, and AI)

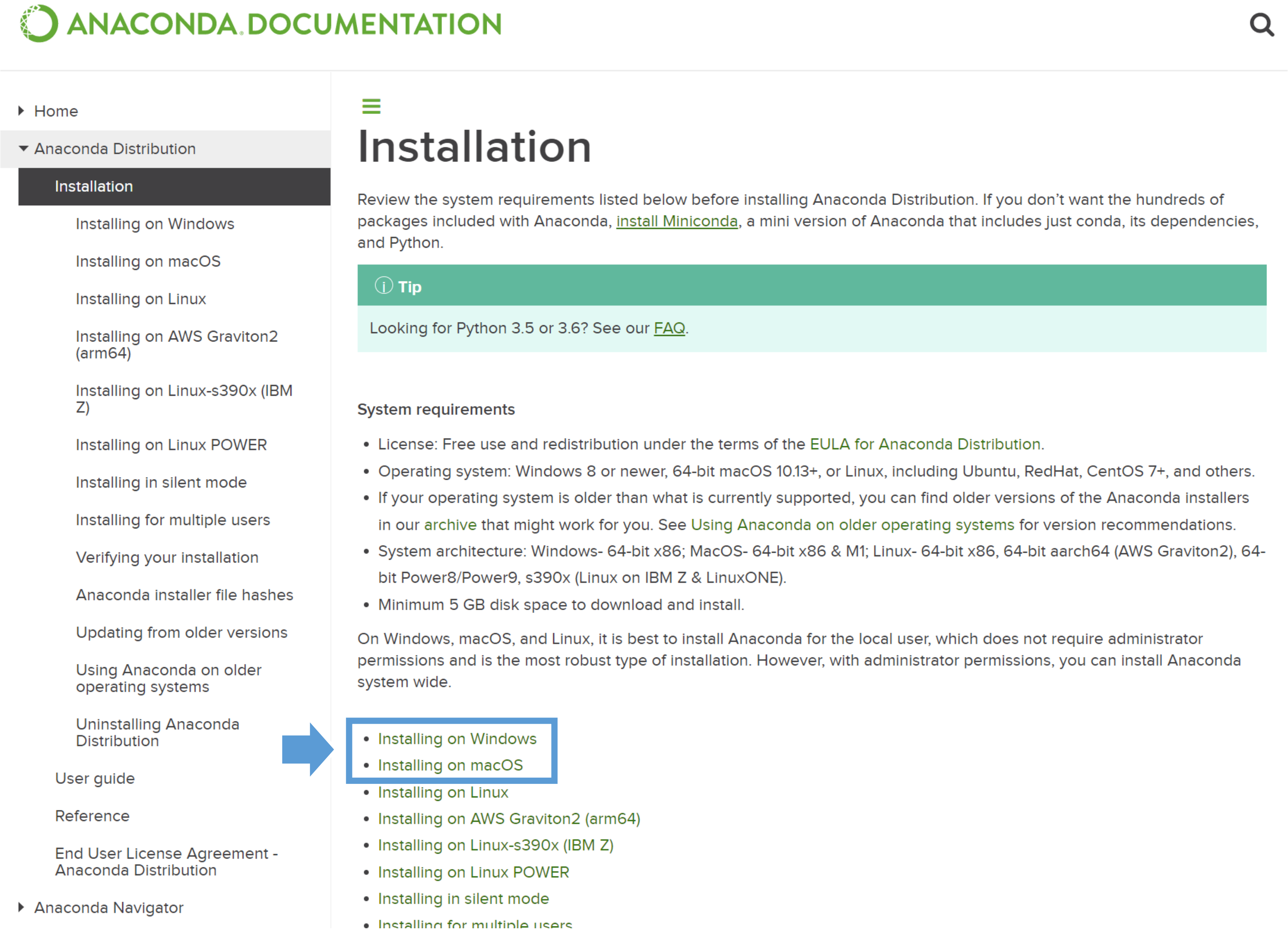
Not sure which one to use? – Anaconda is recommended for most of biologists

Python



Anaconda

Preparation of the Python coding environment



- Home
- Anaconda Distribution
  - Installation
    - Installing on Windows
    - Installing on macOS
    - Installing on Linux
    - Installing on AWS Graviton2 (arm64)
    - Installing on Linux-s390x (IBM Z)
    - Installing on Linux POWER
    - Installing in silent mode
    - Installing for multiple users
    - Verifying your installation
    - Anaconda installer file hashes
    - Updating from older versions
    - Using Anaconda on older operating systems
    - Uninstalling Anaconda Distribution
  - User guide
  - Reference
  - End User License Agreement - Anaconda Distribution
- Anaconda Navigator

Installation

Review the system requirements listed below before installing Anaconda Distribution. If you don't want the hundreds of packages included with Anaconda, install Miniconda, a mini version of Anaconda that includes just conda, its dependencies, and Python.

> ⓘ **Tip**
>
> Looking for Python 3.5 or 3.6? See our FAQ.

System requirements

- License: Free use and redistribution under the terms of the EULA for Anaconda Distribution.
- Operating system: Windows 8 or newer, 64-bit macOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 7+, and others.
- If your operating system is older than what is currently supported, you can find older versions of the Anaconda installers in our archive that might work for you. See Using Anaconda on older operating systems for version recommendations.
- System architecture: Windows- 64-bit x86; MacOS- 64-bit x86 & M1; Linux- 64-bit x86, 64-bit aarch64 (AWS Graviton2), 64-bit Power8/Power9, s390x (Linux on IBM Z & LinuxONE).
- Minimum 5 GB disk space to download and install.

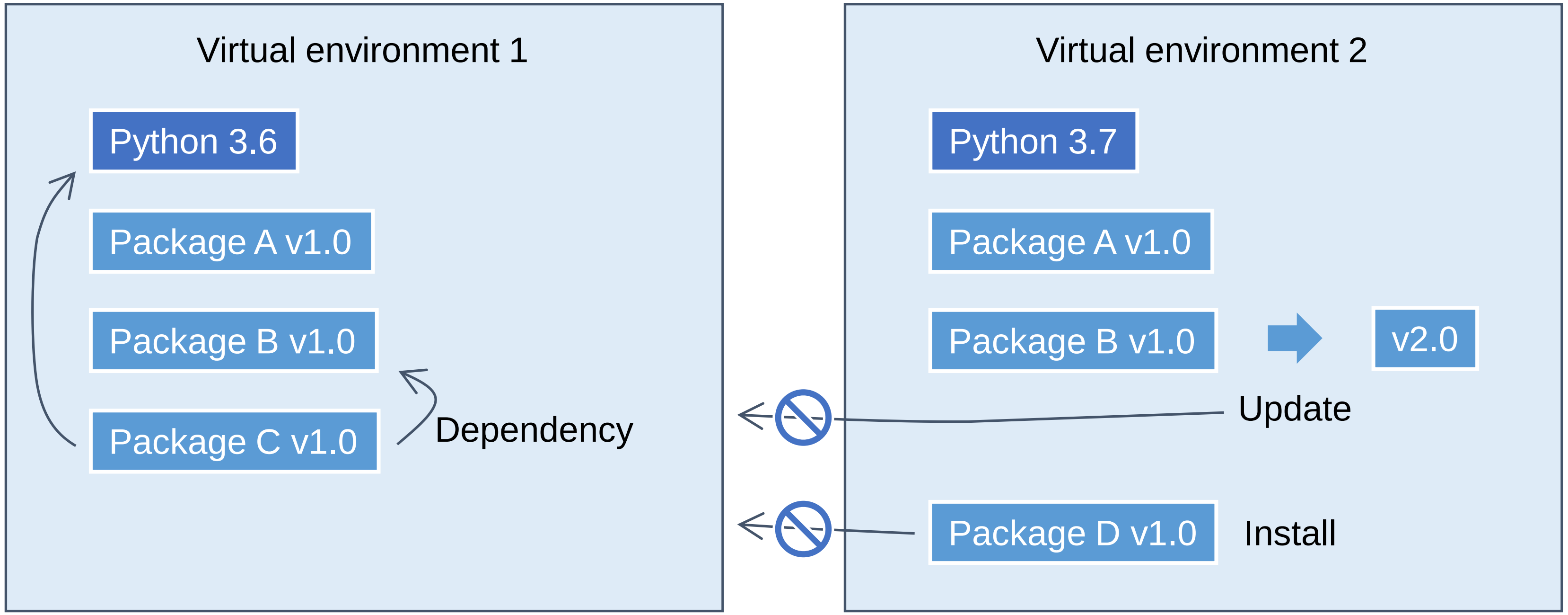
On Windows, macOS, and Linux, it is best to install Anaconda for the local user, which does not require administrator permissions and is the most robust type of installation. However, with administrator permissions, you can install Anaconda system wide.

- Installing on Windows
- Installing on macOS
- Installing on Linux
- Installing on AWS Graviton2 (arm64)
- Installing on Linux-s390x (IBM Z)
- Installing on Linux POWER
- Installing in silent mode
- Installing for multiple users

https://docs.anaconda.com/anaconda/install/index.html

Preparation of the Python coding environment

Concept of virtual environments



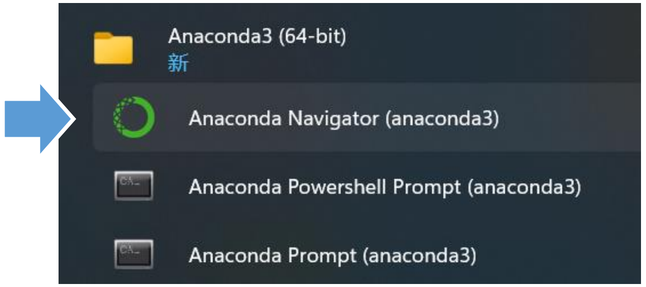
Package C only works with Python 3.6
and requires Package B v1.0

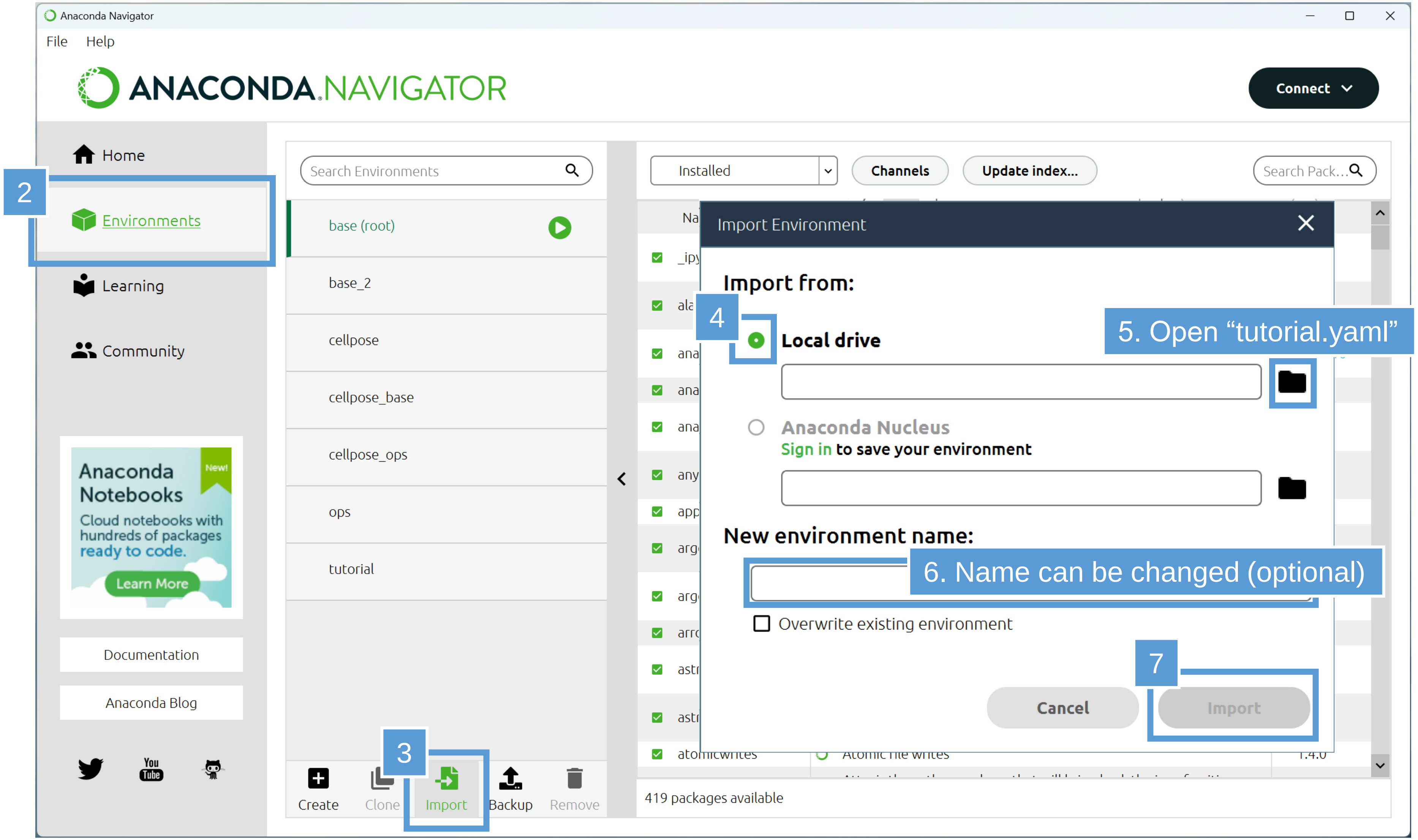
Multiple virtual environments can be constructed in one computer

Virtual environment 1: Cell segmentation-related packages installed
Virtual environment 2: Multivariate analysis-related packages installed
⋮

Preparation of the Python coding environment

1. Open Anaconda Navigator


Anaconda3 (64-bit)
新
Anaconda Navigator (anaconda3)
Anaconda Powershell Prompt (anaconda3)
Anaconda Prompt (anaconda3)



2

3

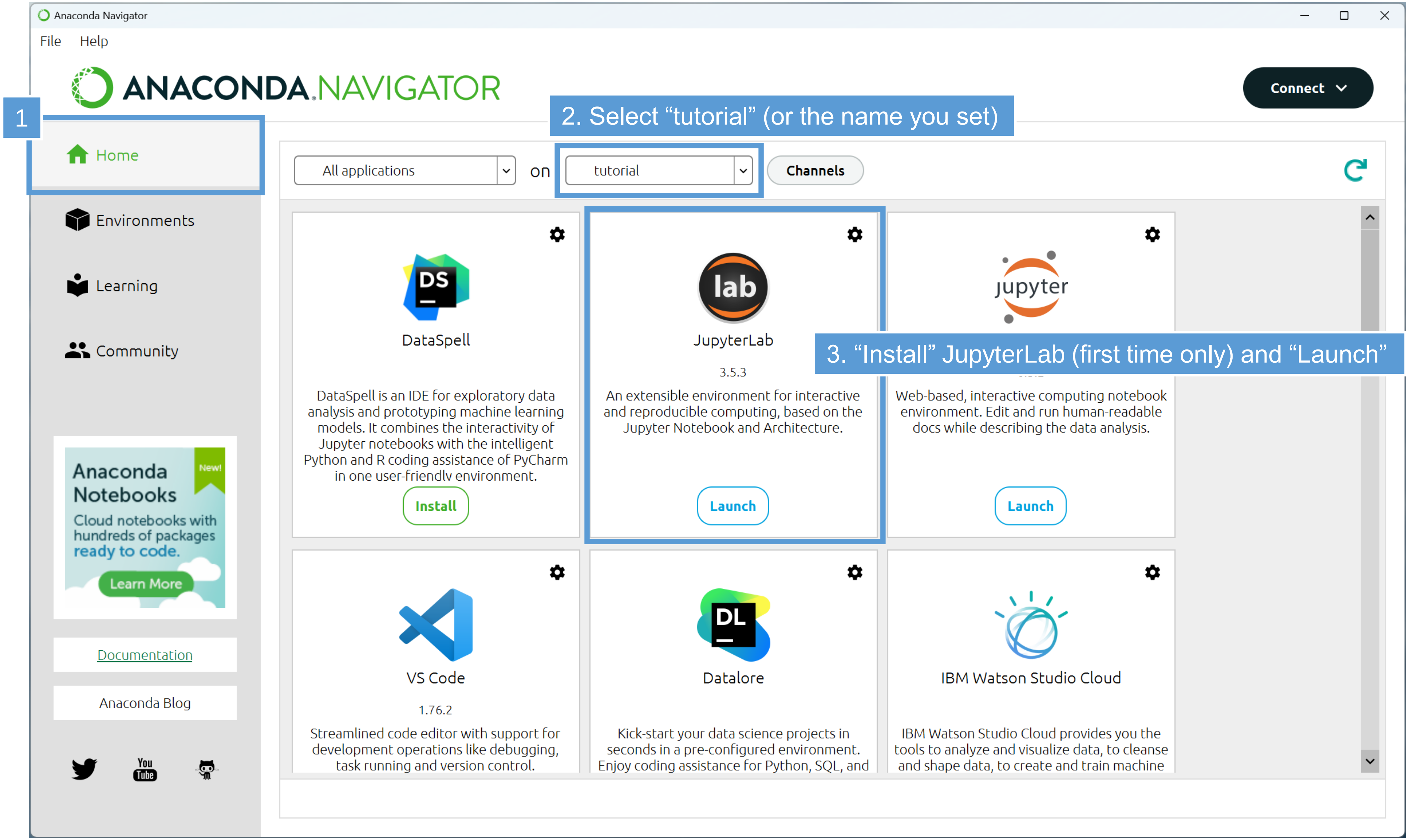
4

5. Open “tutorial.yaml”

6. Name can be changed (optional)

7

Preparation of the Python coding environment



1

2. Select “tutorial” (or the name you set)

3. “Install” JupyterLab (first time only) and “Launch”

DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

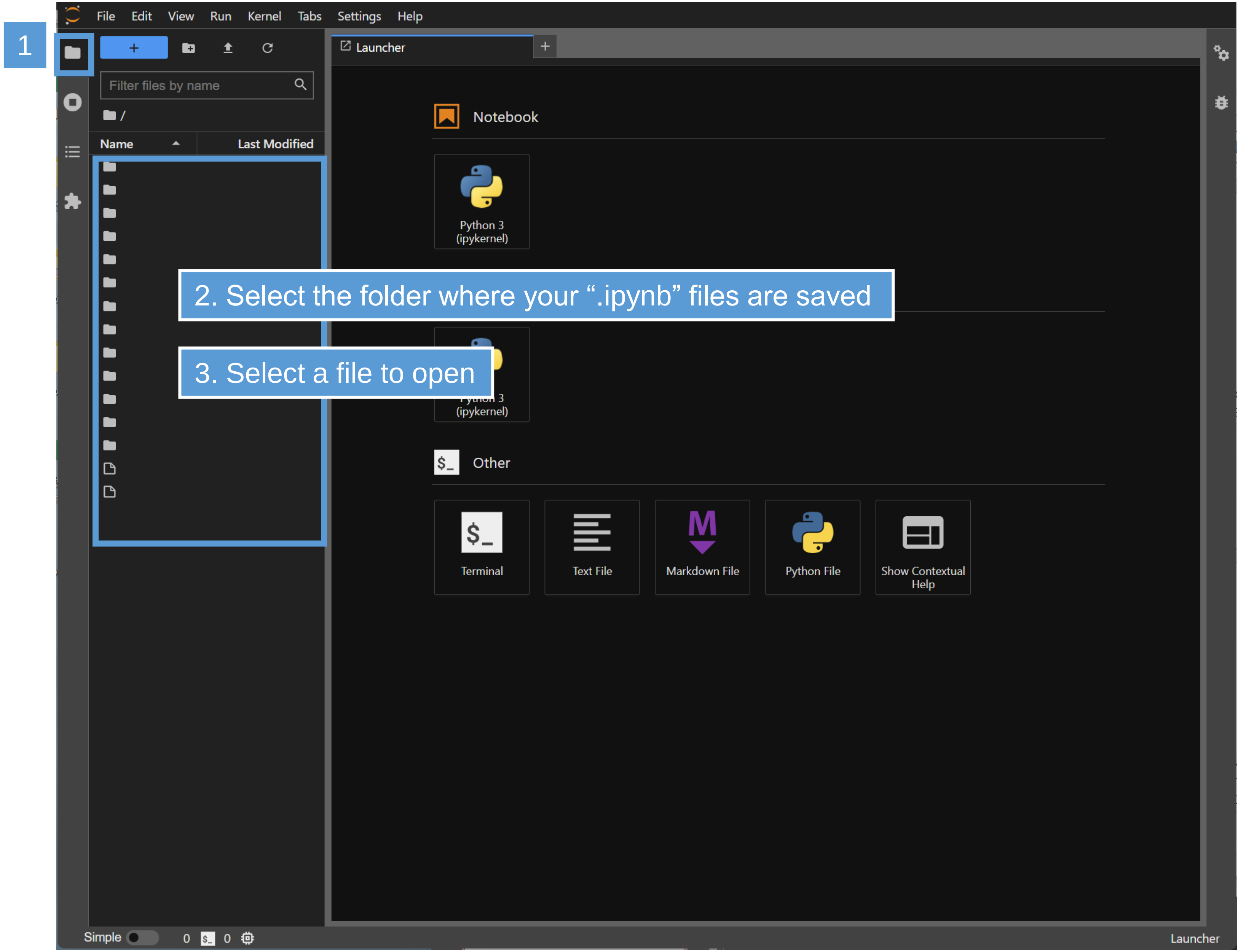
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Streamlined code editor with support for development operations like debugging, task running and version control.

Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine

Preparation of the Python coding environment



1

2. Select the folder where your ".ipynb" files are saved

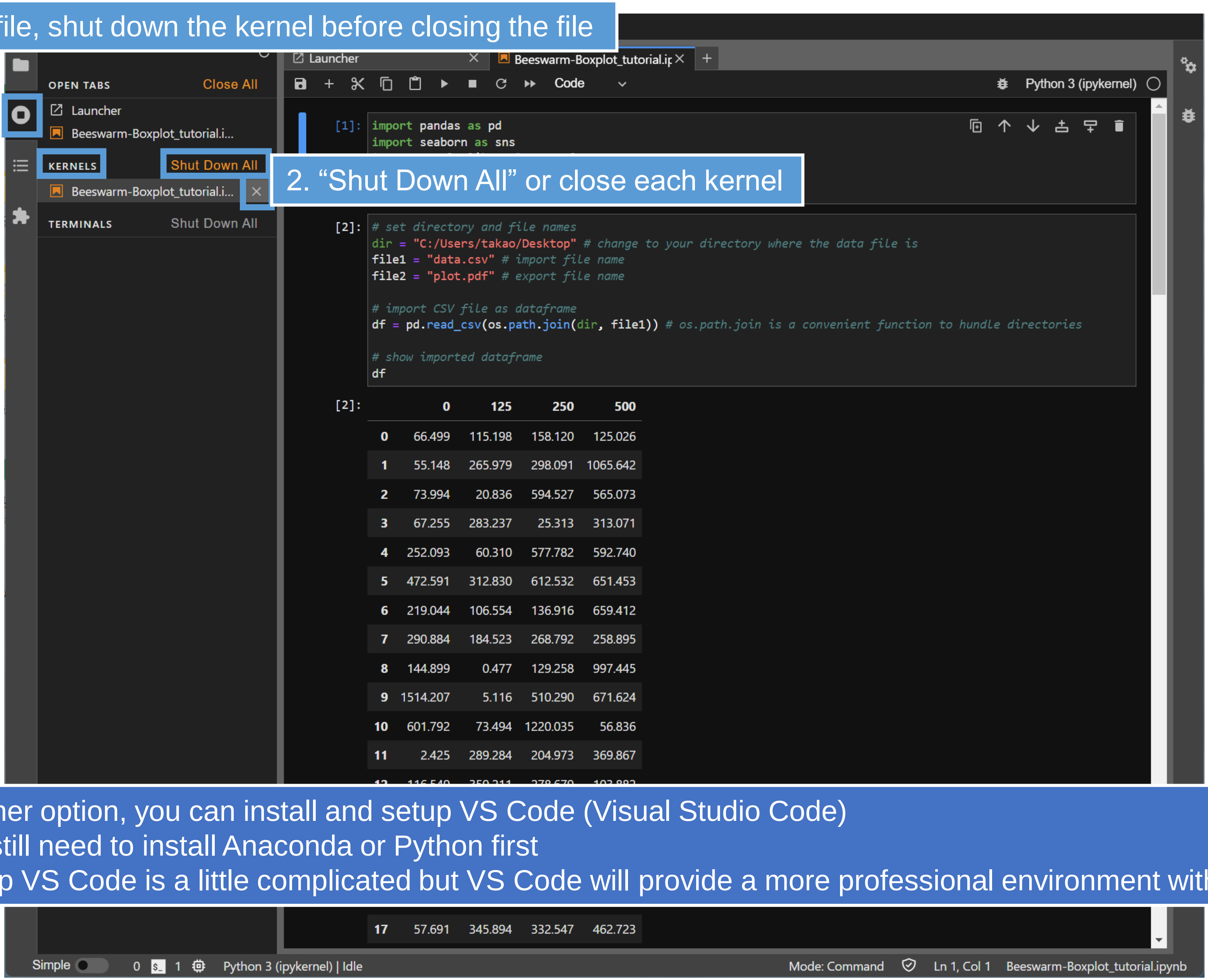
3. Select a file to open

Preparation of the Python coding environment

If you opened a file, shut down the kernel before closing the file

1

2. “Shut Down All” or close each kernel


OPEN TABS
Close All
Launcher
Beeswarm-Boxplot_tutorial.i...
KERNELS
Shut Down All
Beeswarm-Boxplot_tutorial.i...
TERMINALS
Shut Down All
Launcher
Beeswarm-Boxplot_tutorial.ip
Code
Python 3 (ipykernel)
Simple
0
1
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 1
Beeswarm-Boxplot_tutorial.ipynb

[1]:

```python
import pandas as pd
import seaborn as sns
```

[2]:

```python
# set directory and file names
dir = "C:/Users/takao/Desktop" # change to your directory where the data file is
file1 = "data.csv" # import file name
file2 = "plot.pdf" # export file name

# import CSV file as dataframe
df = pd.read_csv(os.path.join(dir, file1)) # os.path.join is a convenient function to hundle directories

# show imported dataframe
df
```

[2]:

	0	125	250	500
0	66.499	115.198	158.120	125.026
1	55.148	265.979	298.091	1065.642
2	73.994	20.836	594.527	565.073
3	67.255	283.237	25.313	313.071
4	252.093	60.310	577.782	592.740
5	472.591	312.830	612.532	651.453
6	219.044	106.554	136.916	659.412
7	290.884	184.523	268.792	258.895
8	144.899	0.477	129.258	997.445
9	1514.207	5.116	510.290	671.624
10	601.792	73.494	1220.035	56.836
11	2.425	289.284	204.973	369.867
17	57.691	345.894	332.547	462.723

- For another option, you can install and setup VS Code (Visual Studio Code)
- You will still need to install Anaconda or Python first
- Setting up VS Code is a little complicated but VS Code will provide a more professional environment with a similar look

The contents of shared files

Data

- Hoechst_mean.csv
- HeLa_GM130_CENP-F (20 TIFF files of #011–030)

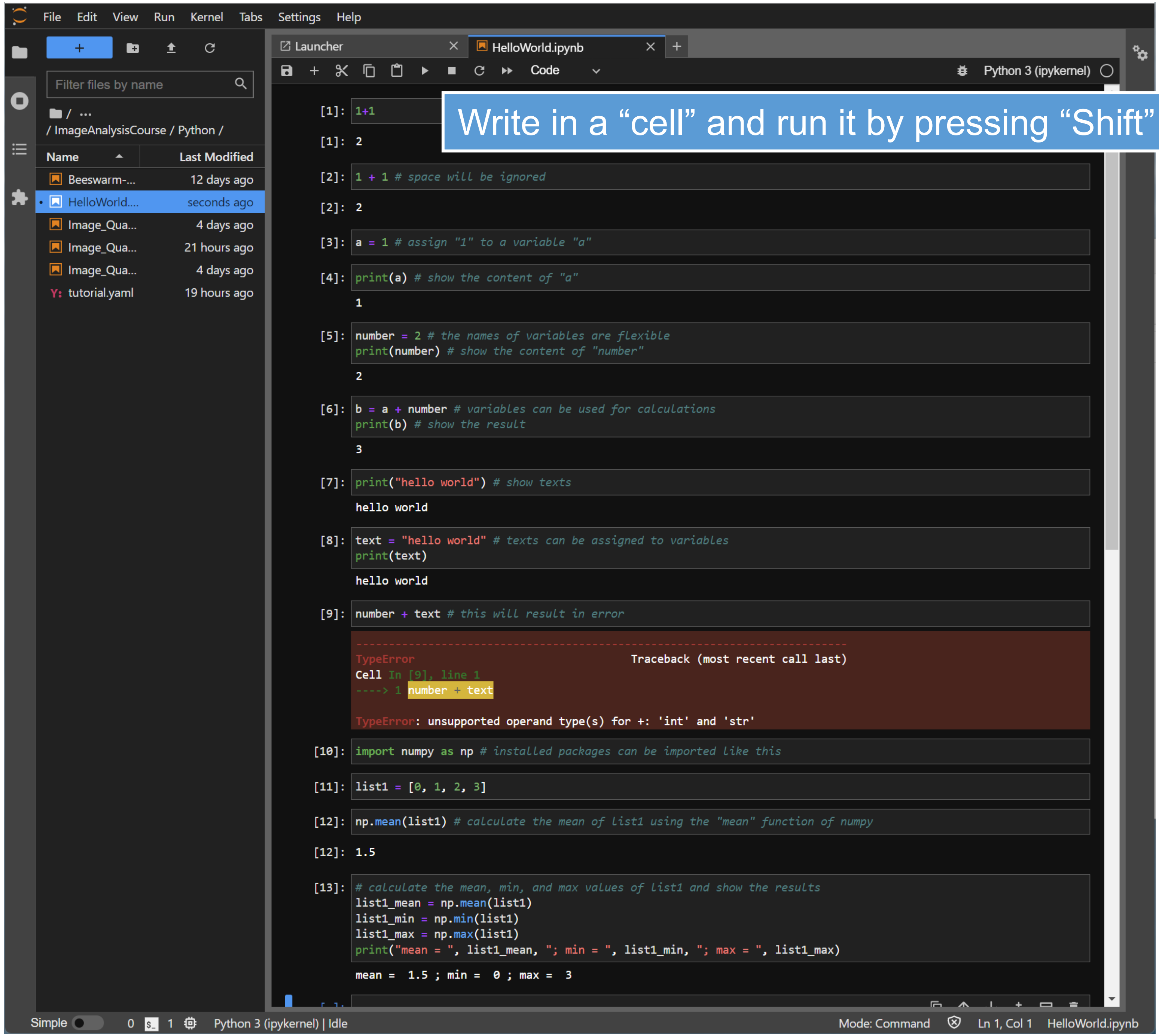
Jupyter notebook (ipynb) and related files

- HelloWorld.ipynb
- Beeswarm-Boxplot_tutorial.ipynb
- Image_Quantification_tutorial_1.ipynb
- Image_Quantification_tutorial_2.ipynb
- Image_Quantification_tutorial_3.ipynb

Proceed through the files in this order

- tutorial.yaml (used for setting up the environment; slide 8, step 5)

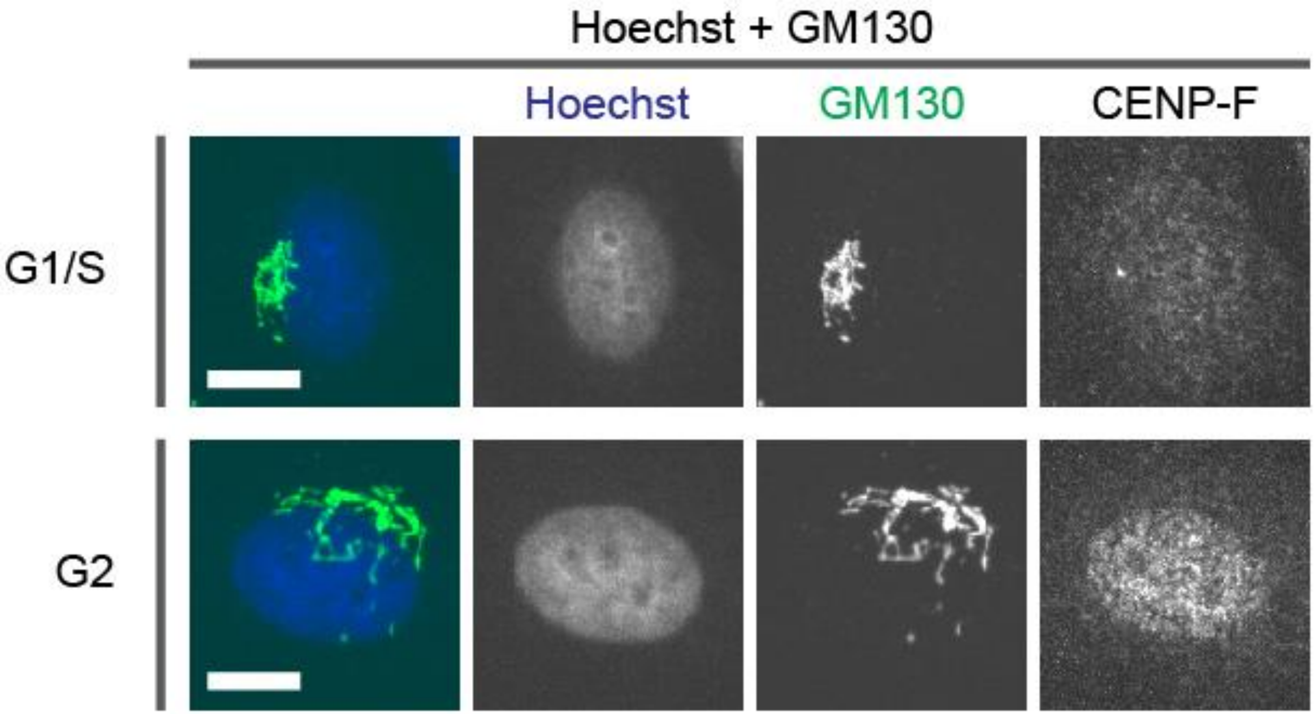
“hello world”

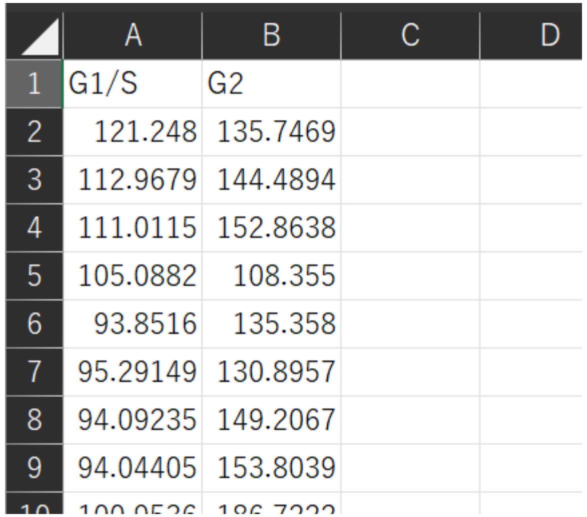

File Edit View Run Kernel Tabs Settings Help
Launcher
HelloWorld.ipynb
Code
Python 3 (ipykernel)
Filter files by name
/ ImageAnalysisCourse / Python /
Name
Last Modified
Beeswarm-... 12 days ago
HelloWorld.... seconds ago
Image_Qua... 4 days ago
Image_Qua... 21 hours ago
Image_Qua... 4 days ago
tutorial.yaml 19 hours ago
Simple 0 1 Python 3 (ipykernel) | Idle
Mode: Command Ln 1, Col 1 HelloWorld.ipynb

Write in a “cell” and run it by pressing “Shift” + “Enter/Return”

```
[1]: 1+1
[1]: 2

[2]: 1 + 1 # space will be ignored
[2]: 2

[3]: a = 1 # assign "1" to a variable "a"

[4]: print(a) # show the content of "a"
     1

[5]: number = 2 # the names of variables are flexible
     print(number) # show the content of "number"
     2

[6]: b = a + number # variables can be used for calculations
     print(b) # show the result
     3

[7]: print("hello world") # show texts
     hello world

[8]: text = "hello world" # texts can be assigned to variables
     print(text)
     hello world

[9]: number + text # this will result in error
     ---------------------------------------------------------------------------
     TypeError                                 Traceback (most recent call last)
     Cell In [9], line 1
     ----> 1 number + text

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

[10]: import numpy as np # installed packages can be imported like this

[11]: list1 = [0, 1, 2, 3]

[12]: np.mean(list1) # calculate the mean of list1 using the "mean" function of numpy
[12]: 1.5

[13]: # calculate the mean, min, and max values of list1 and show the results
      list1_mean = np.mean(list1)
      list1_min = np.min(list1)
      list1_max = np.max(list1)
      print("mean = ", list1_mean, "; min = ", list1_min, "; max = ", list1_max)
      mean =  1.5 ; min =  0 ; max =  3
```

Plot measured data





	A	B	C	D
1	G1/S	G2		
2	121.248	135.7469		
3	112.9679	144.4894		
4	111.0115	152.8638		
5	105.0882	108.355		
6	93.8516	135.358		
7	95.29149	130.8957		
8	94.09235	149.2067		
9	94.04405	153.8039		
10	100.0526	186.7222		

Nagao et al., MBoC, 2020

HeLa cells

- Mean fluorescence intensities of Hoechst (DNA) were measured and listed in Hoechst_mean.csv.
- Data were categorized according to cell cycle phases (G1/S and G2)