FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

DIE HOCHSCHULE,
DIE MEHR KANN.

matplotlib

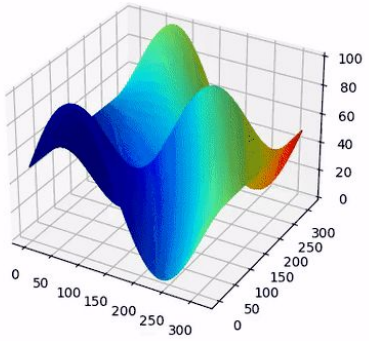
Leona Vračar
Markus Eder
Nicholas Wedige
Sonja Vukotić

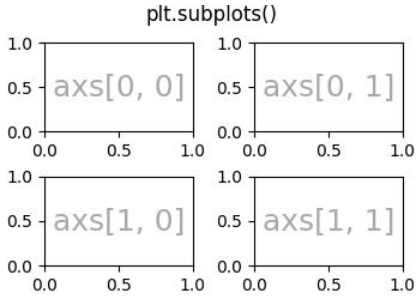
APPLIED LIFE SCIENCES | MASTERSTUDIENGANG BIOINFORMATIK

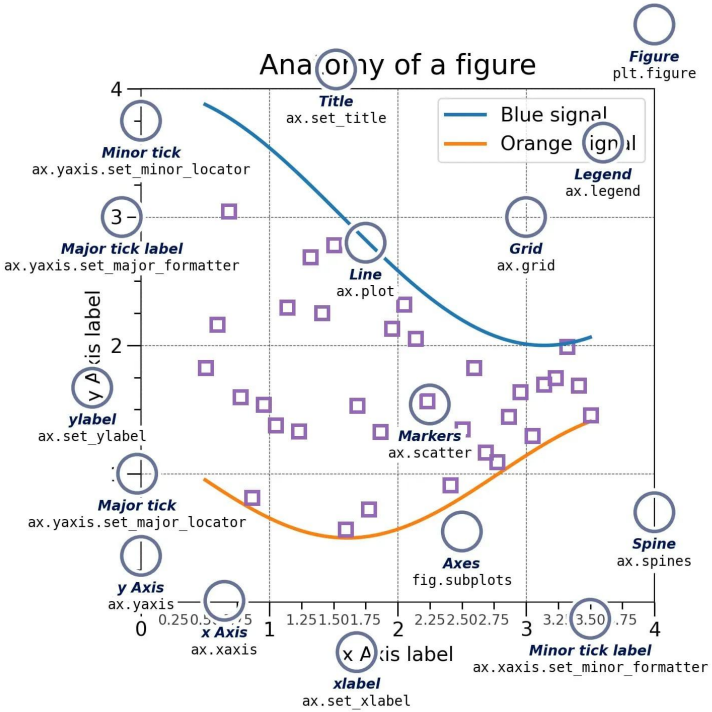
Motivation und Überblick

- Python-Library zur Erstellung statischer und interaktiver Datenvisualisierungen
- Sehr customizable
- Gut mit numpy integriert (und pandas)
  - Arbeiten mit Arrays
- Python-Alternative zu MatLab
- Animation von Daten möglich

```
import matplotlib.pyplot as plt
import numpy as np
```








```python
fig = plt.figure() # an empty figure with no Axes

fig, ax = plt.subplots() # a figure with a single Axes

fig, axs = plt.subplots(2, 2) # a figure with a 2x2 grid of Axes

# a figure with one Axes on the left, and two on the right:
fig, axs = plt.subplot_mosaic([['left', 'right_top'],
                               ['left', 'right_bottom']])
```

```python
fig, ax = plt.subplots(figsize=(5, 2.7), layout='constrained')
categories = ['turnips', 'rutabaga', 'cucumber', 'pumpkins']

ax.bar(categories, np.random.rand(len(categories)))
```

```python
fig, ax = plt.subplots()
ax.set_title("This is the title")
ax.set_xlabel("This is the x-Axis label")
ax.grid(True)
plt.show()
```

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

matplotlib Plot types User guide Tutorials Examples Reference Contribute Releases 3.9 (stable)

Section Navigation

- Quick start guide
- Frequently Asked Questions
- Figures and backends
- Axes and subplots
- Artists
- Customizing Matplotlib with style sheets and rcParams
- Colors
- Text
- Animations using Matplotlib
- User Toolkits
- Getting started
- Installation

Quick start guide

Quick start guide

This tutorial covers some basic usage patterns and best practices to help you get started with Matplotlib.

```
import matplotlib.pyplot as plt
import numpy as np
```

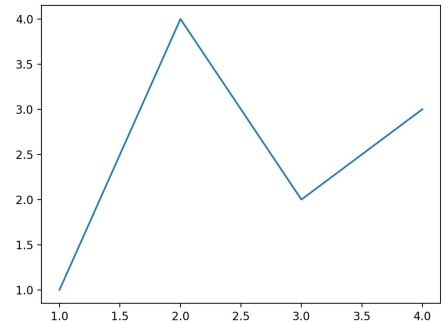
A simple example

Matplotlib graphs your data on `Figure`s (e.g., windows, Jupyter widgets, etc.), each of which can contain one or more `Axes`, an area where points can be specified in terms of x-y coordinates (or theta-r in a polar plot, x-y-z in a 3D plot, etc.). The simplest way of creating a Figure with an Axes is using `pyplot.subplots`. We can then use `Axes.plot` to draw some data on the Axes, and `show` to display the figure:

```
fig, ax = plt.subplots()             # Create a figure containing a single Axes.
ax.plot([1, 2, 3, 4], [1, 4, 2, 3])  # Plot some data on the Axes.
plt.show()                           # Show the figure.
```

On this page

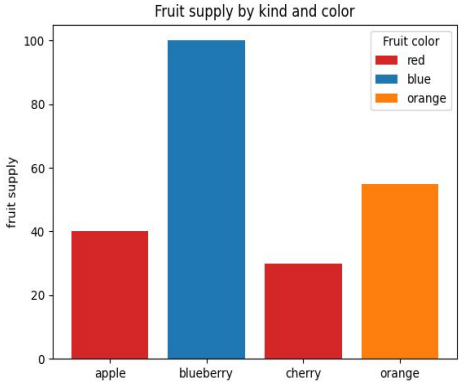
- A simple example
- Parts of a Figure
- Types of inputs to plotting functions
- Coding styles
- Styling Artists
- Labelling plots
- Axis scales and ticks
- Color mapped data
- Working with multiple Figures and Axes
- More reading



https://matplotlib.org/stable/users/explain/quick_start.html

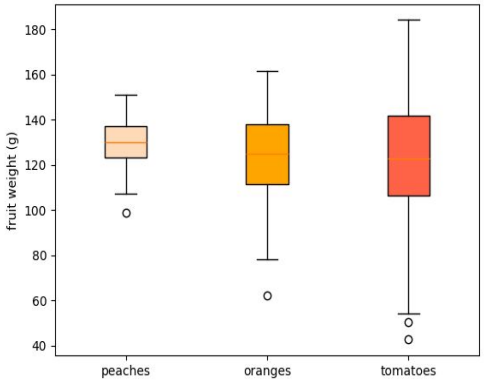
FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Diagramm- und Datentypen

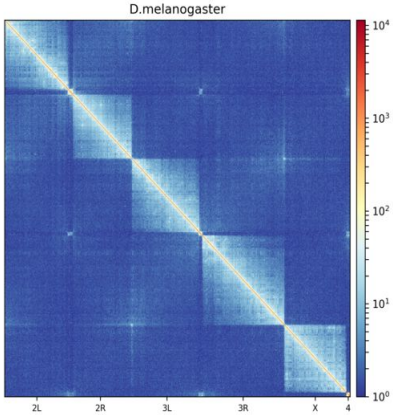


Pairwise data

(x,y)

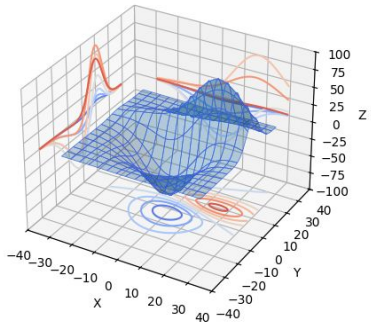


Statistical distributions



Gridded data

(arrays and images plotted on coordinate grids)



3D and volumetric data

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

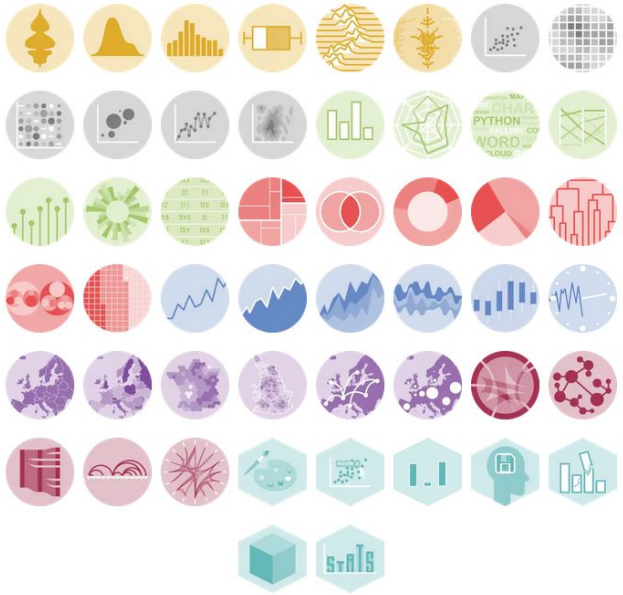
Diagramm- und Datentypen (2)

Python graph gallery

- Nützliche Ressource für mögliche Darstellungsarten

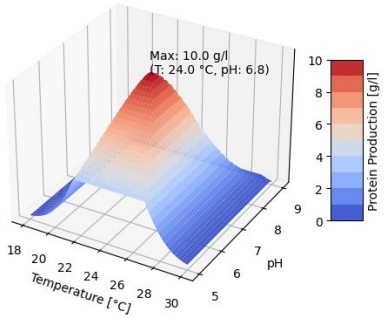
Matplotlib Cheat-Sheets

- Schnelle Hilfestellung zum Formatieren & Erstellen von Plots



FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Bioinformatische Anwendungen

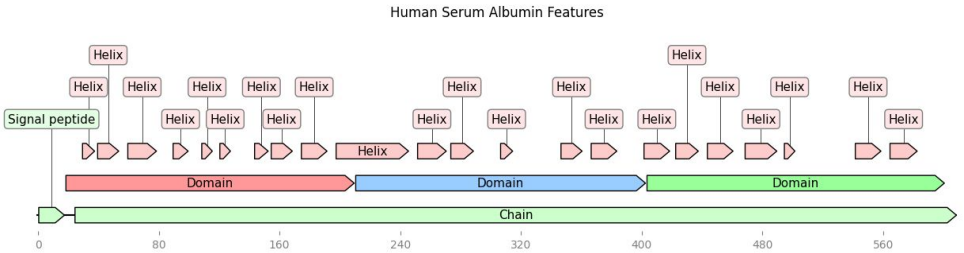


- Datenvisualisierung allgemein
- Verwendung in Kombination mit Biopython
- Phylogenetische Bäume
- Operative taxonomische Einheit (OTE) Visualisierung
- Genome/Sequenz Visualisierung (z.B. GC content, duplication rates)
- Genexpression Heatmaps
- Design of Experiment (DoE)
- Proteinstruktur
- … und viel mehr

FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Beispiel: DNA Features Viewer Package

```python
import matplotlib.pyplot as plt
from dna_features_viewer import GraphicFeature, GraphicRecord

#-------------------------------------------------------------------------------------------------------

# Function to filter out features shorter than a specified length
def filter_short_features(features, min_length):
    return [feature for feature in features if feature.end - feature.start + 1 >= min_length]

#-------------------------------------------------------------------------------------------------------

def set_feature_color(feature_type, domain_index, domain_colors):
    if "domain" in feature_type.lower():
        color = domain_colors[domain_index % len(domain_colors)]
        domain_index += 1
    elif "helix" in feature_type.lower():
        color = "#ffcccc"
    elif "beta" in feature_type.lower():
        color = "#cceeff"
    else:
        color = "#ccffcc"
    return color, domain_index

#-------------------------------------------------------------------------------------------------------

# Create a graphic record
record = GraphicRecord(sequence_length=sequence_length, features=features)

# Create the plot
ax, _ = record.plot(figure_width=15, with_ruler=True)
plt.title("Human Serum Albumin Visualization (Capitalized Labels)")

# Save and show the plot
plt.savefig("human_serum_albumin.png")
plt.show()
```



FH CAMPUS WIEN
UNIVERSITY OF APPLIED SCIENCES

Beispiel: Quick Live Coding Session

Collab Link (download or view):

Copy of Input_Matplotlib.ipynb