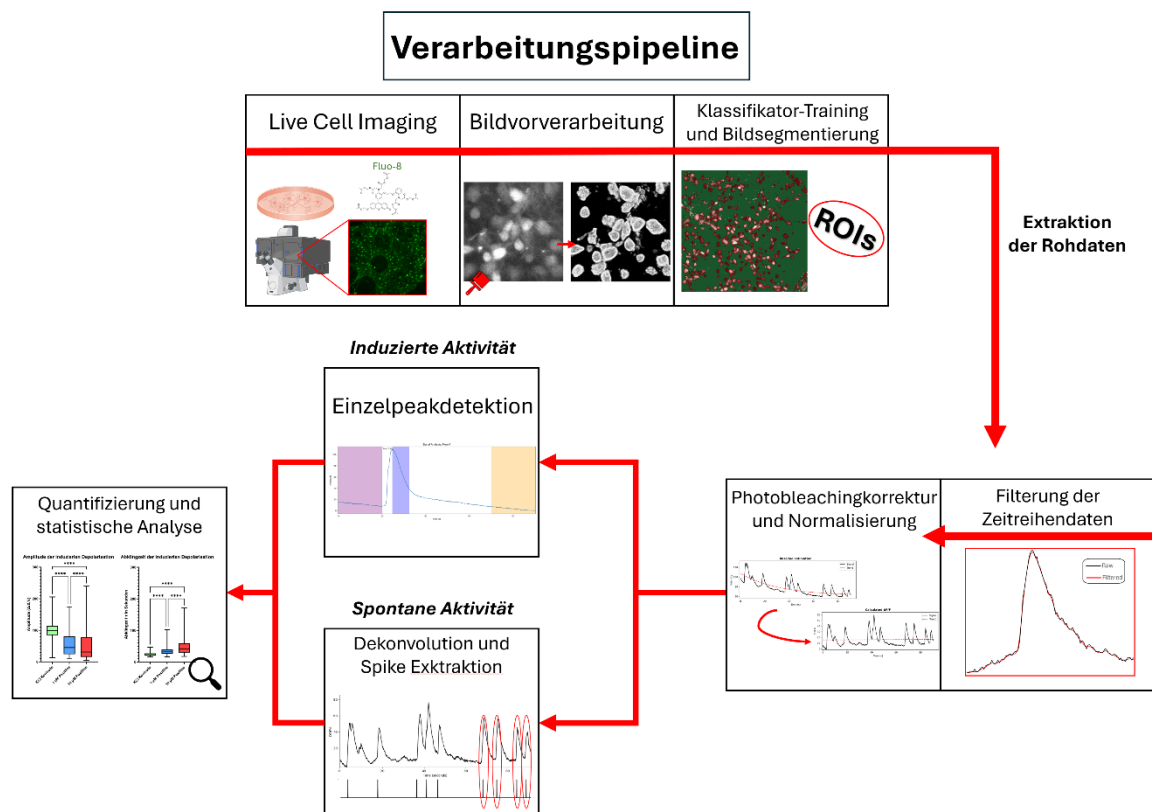


# Bedienungsanleitung Ca<sup>2+</sup>-Processing Pipeline

Diese Bedienungsanleitung führt die wesentlichen Schritte zur Installation von Python und einer Pythonumgebung zur Verwendung der Ca<sup>2+</sup>-Processing Pipeline. Dabei wird neben der Einrichtung der Pipeline, auch die Eingabe von Parametern der einzelnen Verarbeitungsschritte kurz aufgeführt. Diese Anleitung kann zukünftig weiter verfeinert und durch eine Troubleshooting/FAQ-Sektion erweitert werden.



Bei Problemen oder Fragen bezüglich der Installation oder Verwendung kann eine Anfrage an [robinfriedrich1704@gmail.com](mailto:robinfriedrich1704@gmail.com) gesendet werden.

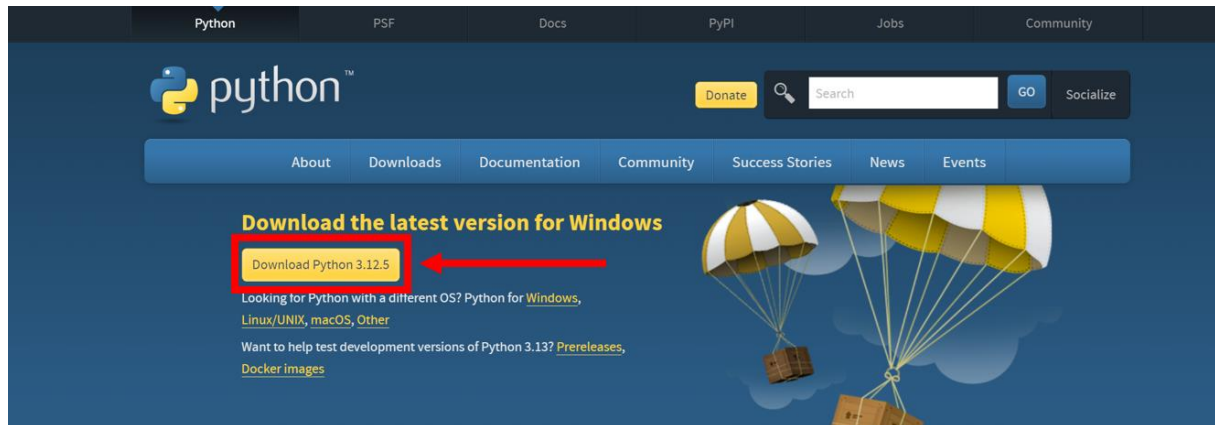
# Inhaltsverzeichnis

1. Installation von Python unter Windows.....	3
2. Installation von PyCharm.....	4
3. Hinzufügen von Abhängigkeiten .....	7
4. Ausführung des Codes .....	8
1. Preprocessing.....	9
2. Stackerstellung.....	10
3. WEKA Segmentierung .....	10
4. Filterung.....	13
5. Photobleaching Korrektur.....	14
6. Dekonvolution für spontane Aktivität .....	14
7. Single Peak Detektion für induzierte Aktivität.....	15

# 1. Installation von Python unter Windows

Für die Installation von Python muss zunächst eine aktuelle Version (mind. 3.12.2 – zur Entwicklung der Pipeline verwendete Version) von der Python-Webseite <https://www.python.org/downloads/> runtergeladen werden.

Über die Webseite wird eine Downloadschaltfläche für die aktuelle Version gelb hinterlegt angezeigt.



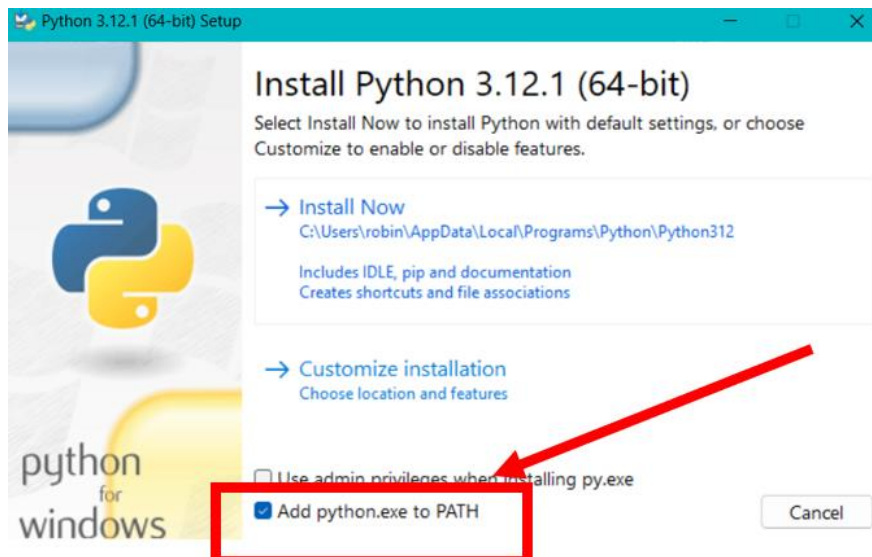
Falls die aktuelle Version Probleme beim Ausführen des Codes verursacht, empfiehlt es sich Version 3.12.2 runterzuladen.

Looking for a specific release?

Python releases by version number:

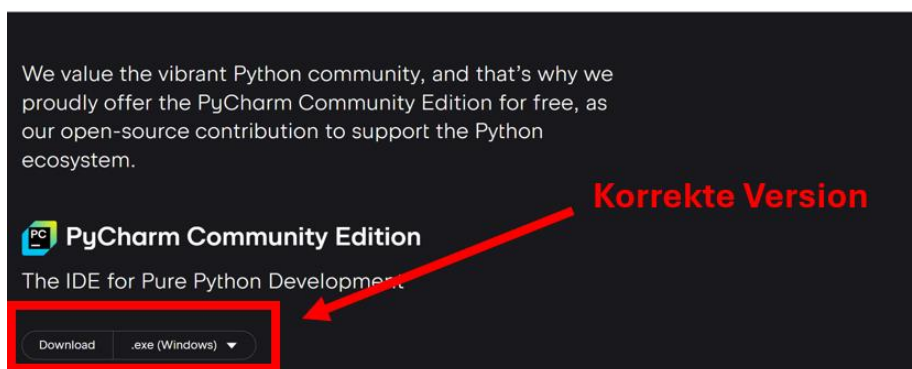
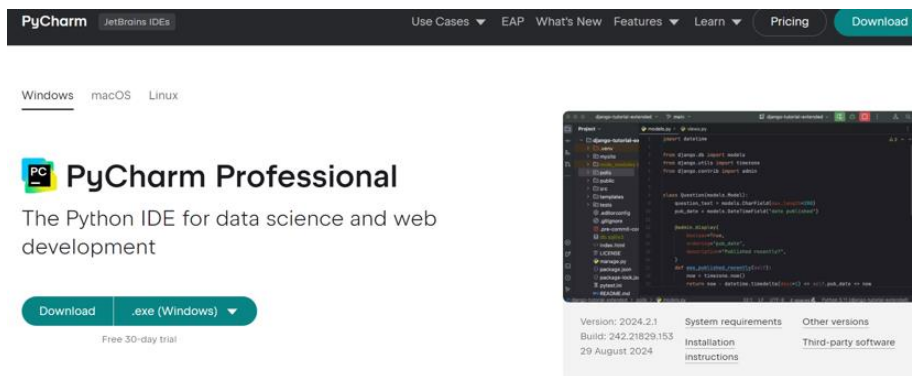
Release version	Release date		Click for more
<a href="#">Python 3.11.9</a>	April 2, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.10.14</a>	March 19, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.9.19</a>	March 19, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.8.19</a>	March 19, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.11.8</a>	Feb. 6, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.12.2</a>	Feb. 6, 2024	<a href="#">Download</a>	<a href="#">Release Notes</a>
<a href="#">Python 3.12.1</a>	Dec. 8, 2023	<a href="#">Download</a>	<a href="#">Release Notes</a>

Bei der Installation von Python ist es wichtig, die Option „Add python.exe to PATH“ auszuwählen. Anschließend kann mit der Installation begonnen werden.

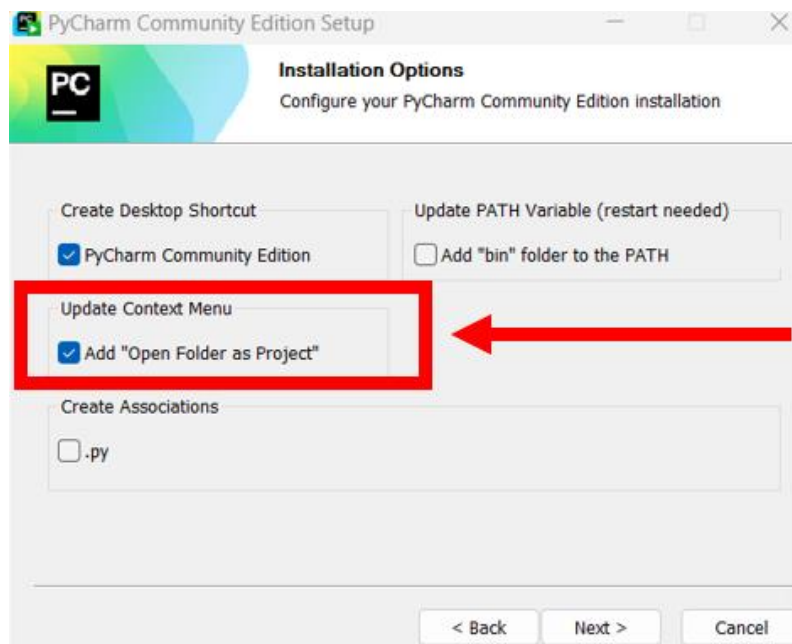


## 2. Installation von PyCharm

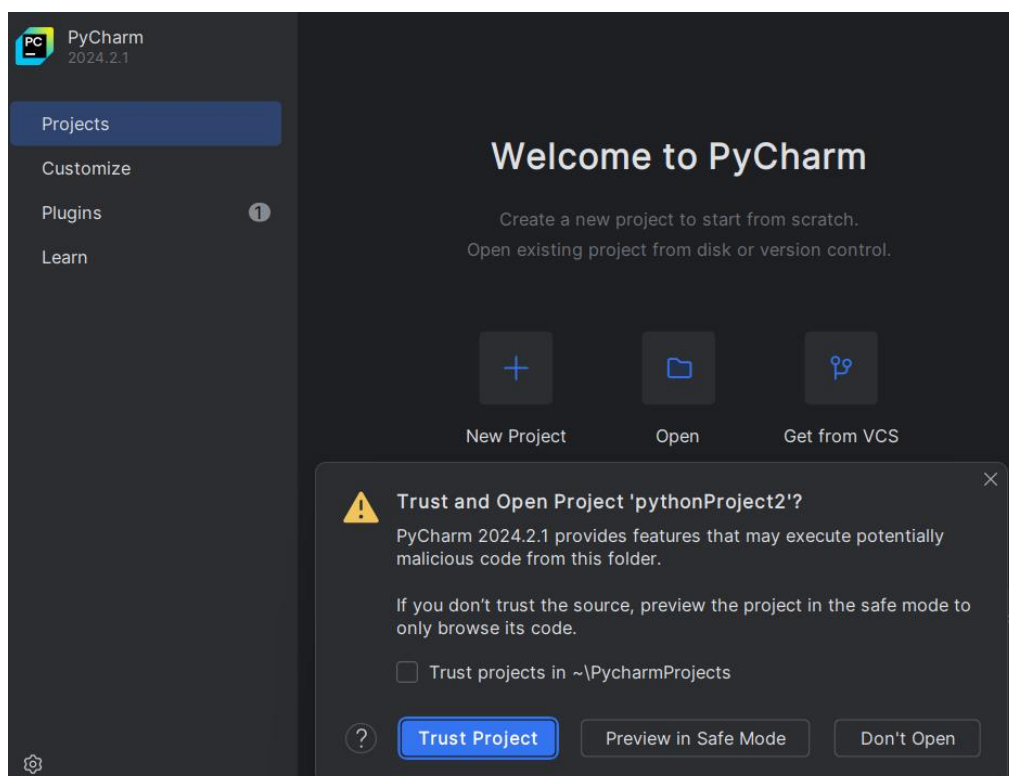
Nach der erfolgreichen Installation von Python wird eine beliebige IDE (Integrierte Entwicklungsumgebung) zum Ausführen oder Anpassen des Codes installiert werden. In diesem Fall wird die frei verfügbare IDE „PyCharm“ von JetBrains verwendet. Die aktuelle Installationsdatei ist unter <https://www.jetbrains.com/pycharm/download/?section=windows> verfügbar. Hierbei ist es wichtig, die kostenfreie PyCharm Community Edition auszuwählen.



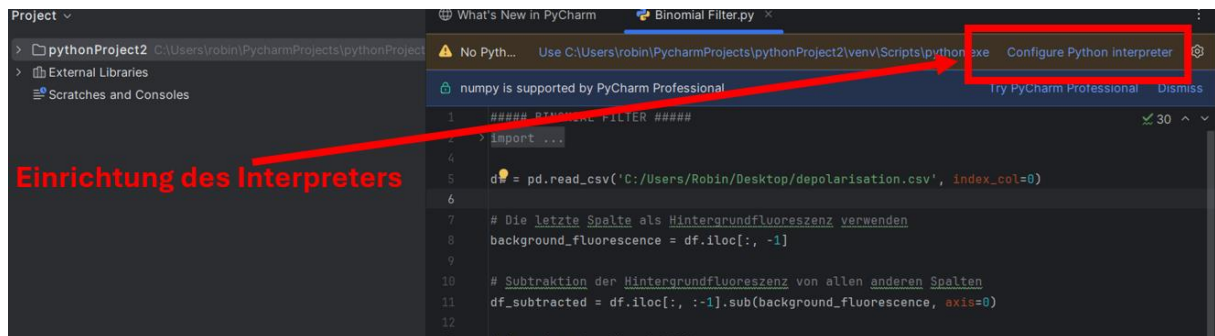
Bei der Installation von PyCharm wird ein Häkchen bei „64-bit launcher“ gesetzt. Zusätzlich kann die Option „add Open Folder as Project“ ausgewählt werden, um den Import der Pipeline zu erleichtern.



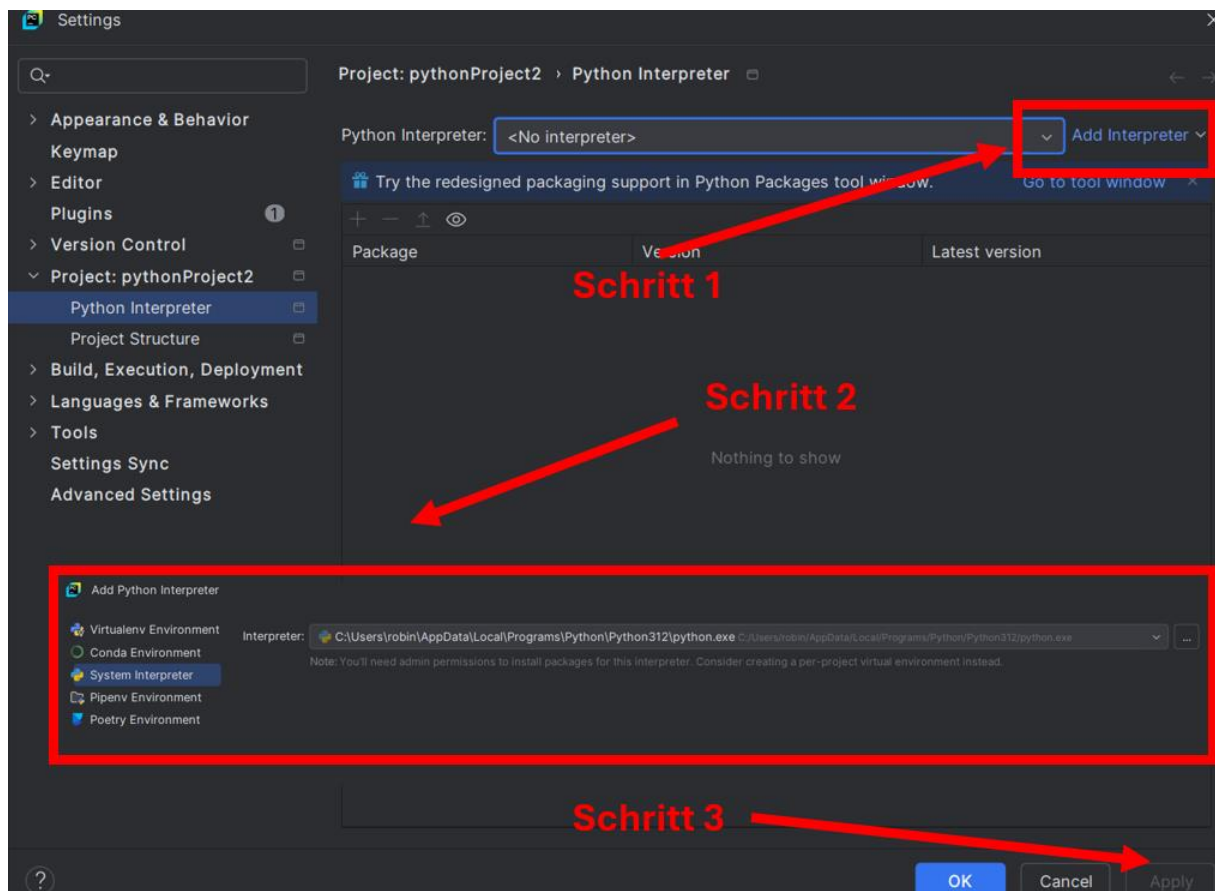
Nach erfolgreicher Installation muss PyCharm für die weitere Verwendung konfiguriert werden. Zunächst wird der Ordner des Python Projekts mit einem Klick auf „Open“ hinzugefügt. Sofern das korrekte Projekt ausgewählt wurde, kann das Projekt durch Bestätigung mittels „Trust Project“ der Oberfläche hinzugefügt werden.



Nach dem Einlesen des Projekts muss vor Verwendung ein Interpreter ausgewählt werden. Hierfür kann ein beliebiges Skript aus dem Ordner des Python Projekts geöffnet werden. Oben rechts in der Oberfläche erscheint die Option „Configure Python Interpreter“.

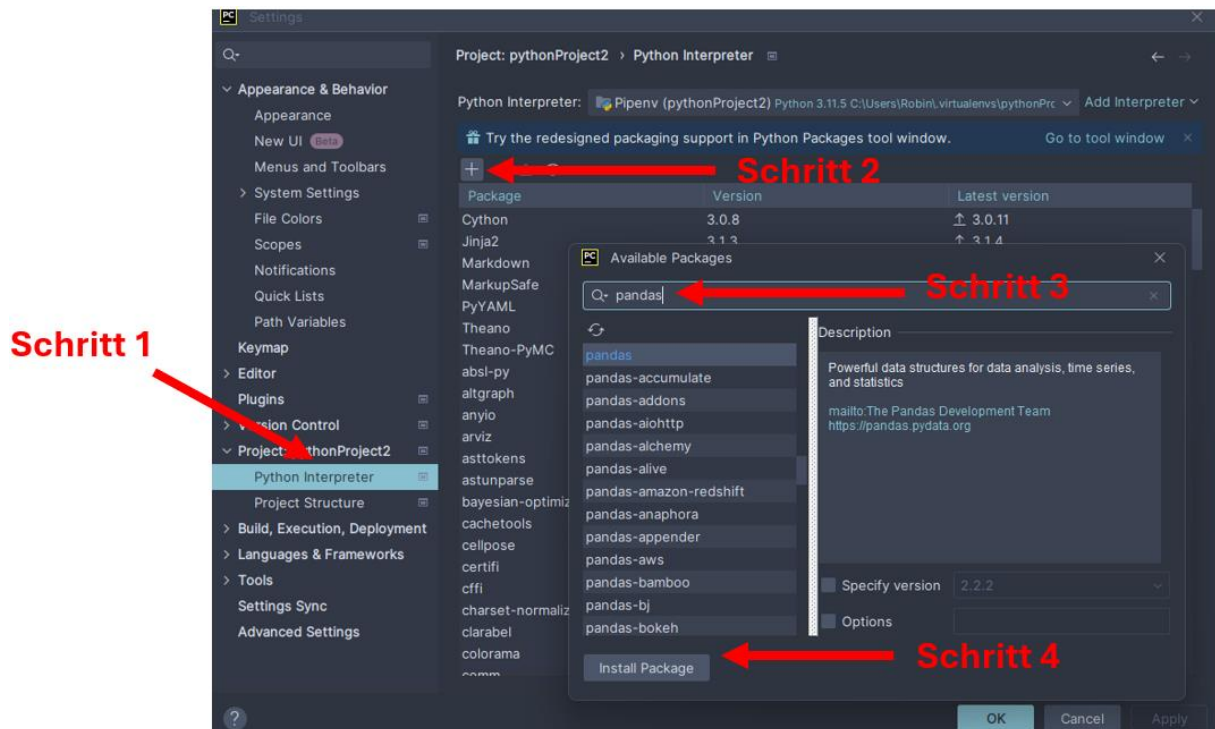


Nach Auswahl der Schaltfläche öffnet sich ein neues Fenster. Falls kein Interpreter angezeigt wird, kann ein neuer Interpreter über die Schaltfläche „Add Interpreter“ hinzugefügt werden. Daraufhin öffnet sich ein weiteres Fenster. Der zuvor installierte Python-Interpreter lässt sich unter der Option „System Interpreter“ finden. Wenn mehrere Python-Versionen auf dem System installiert sind, sollte die gewünschte Version ausgewählt werden. Nach Auswahl des Interpreters kann das Fenster mit „OK“ geschlossen werden. Um den Interpreter für das Projekt festzulegen, genügt ein Klick auf „Apply“, und das Fenster kann anschließend geschlossen werden.



### 3. Hinzufügen von Abhängigkeiten

Bevor die Pipeline verwendet werden kann, müssen die Abhängigkeiten installiert werden. Diese Abhängigkeiten sind Pakete, die für den Code importiert und verwendet werden. Die Installation kann entweder über den Code selbst erfolgen, indem die entsprechenden Befehle vor dem Import-Abschnitt hinzugefügt werden, oder man kann die Pakete manuell über die Benutzeroberfläche installieren. Dazu navigiert man zu File > Settings... > Project: „PROJEKTNAME“ > Python Interpreter. Im nächsten Schritt wird über die „+“-Schaltfläche ein Suchfenster geöffnet. Dort kann das gewünschte Paket durch Eingabe des Namens gesucht und ausgewählt werden. Mit einem Klick auf „Install Package“ wird das Paket installiert. Nachdem alle benötigten Pakete installiert wurden, kann das Suchfenster geschlossen werden. Die installierten Pakete werden nun in einer Liste angezeigt. Mit „Apply“ wird die Installation auf das aktuelle Projekt angewendet.



Nachfolgend werden alle Abhängigkeiten für die Verwendung der Pipeline aufgelistet:

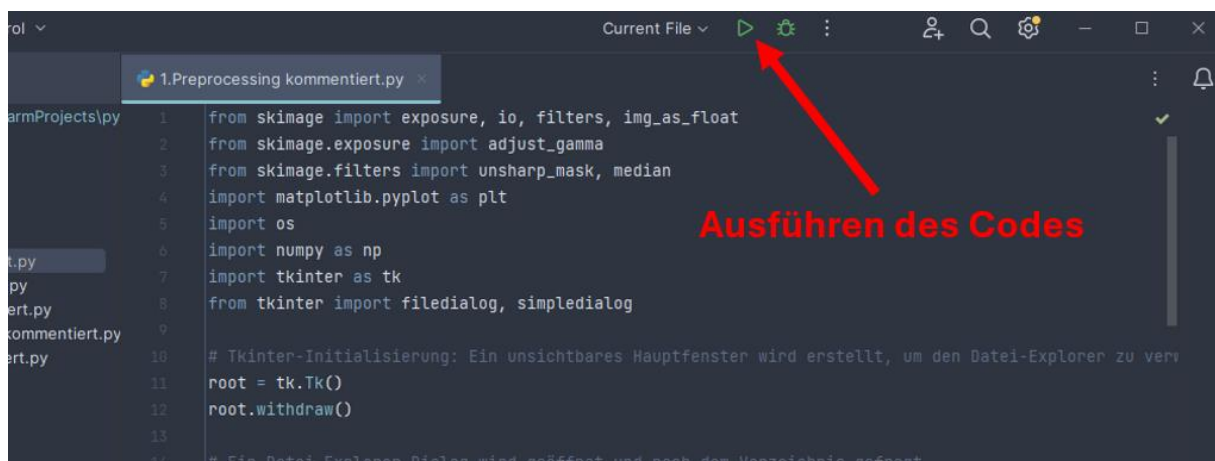
Bezeichnung	Versionsnummer
filterpy	1.4.5
matplotlib	3.8.2
numpy	1.26.4
oasis	0.1.3



pandas	2.2.0
Pillow	10.2.0
scikit_learn	1.4.0
scipy	1.12.0
skimage	0.22.0
tifffile	2024.1.30
tqdm	4.66.1

## 4. Ausführung des Codes

Nach dem erfolgreichen Importieren aller Abhängigkeiten können die Skripte der Pipeline ausgeführt werden. Dafür muss zunächst ein Skript geöffnet und mit einem Klick auf die „Start“-Schaltfläche initialisiert werden.



The screenshot shows a code editor window with a dark theme. The toolbar at the top right contains several icons, including a green play button (Run). A red arrow points from the text 'Ausführen des Codes' to this play button. The code in the editor is a Python script for image preprocessing, starting with imports from skimage, matplotlib, os, numpy, and tkinter. The script is titled '1.Preprocessing kommentiert.py'.

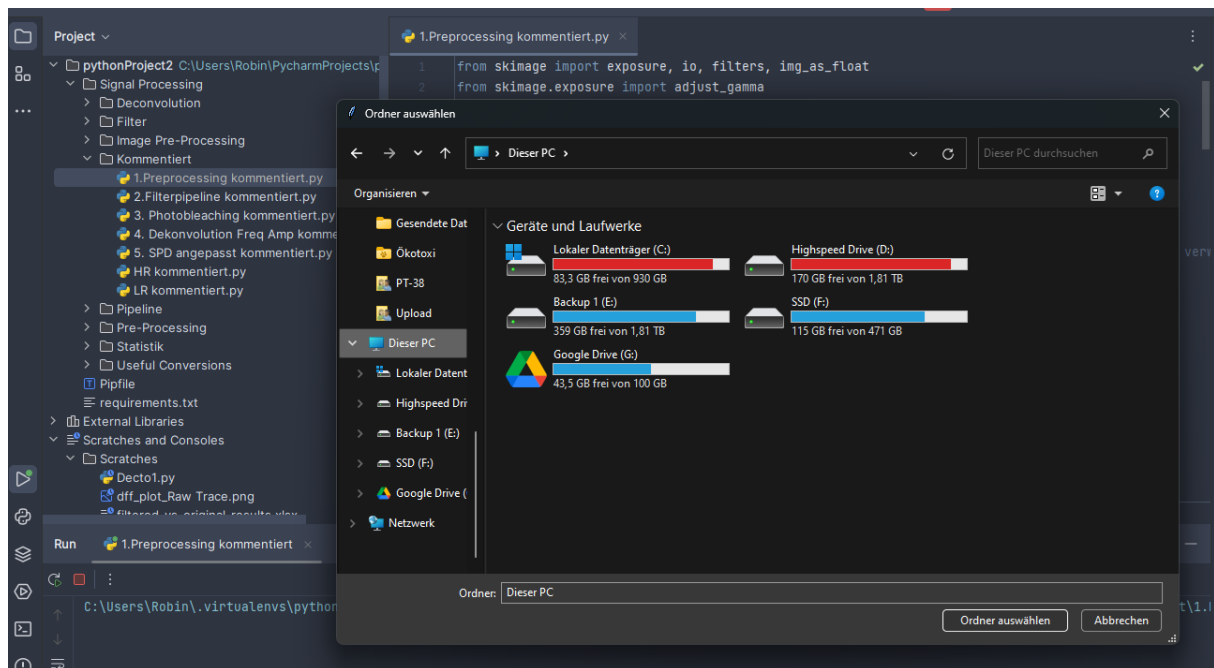
```

1 from skimage import exposure, io, filters, img_as_float
2 from skimage.exposure import adjust_gamma
3 from skimage.filters import unsharp_mask, median
4 import matplotlib.pyplot as plt
5 import os
6 import numpy as np
7 import tkinter as tk
8 from tkinter import filedialog, simpledialog
9
10 # Tkinter-Initialisierung: Ein unsichtbares Hauptfenster wird erstellt, um den Datei-Explorer zu ver
11 root = tk.Tk()
12 root.withdraw()
13
14 # Ein Datei-Explorer Dialog wird geöffnet und nach dem Verzeichnis gefragt

```

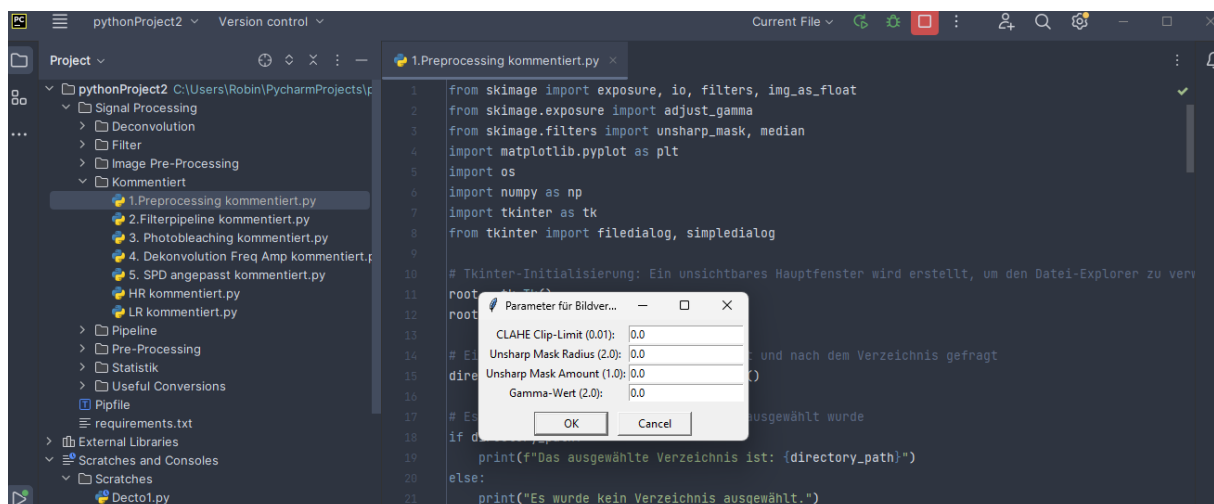
Zu Beginn wird über eine Abfrage der Speicherort der Eingangsdatei erfragt. Es empfiehlt sich, ein Ordner je Versuch anzulegen und die Dateien mit einem eindeutig identifizierbaren Namen zu beschriften.



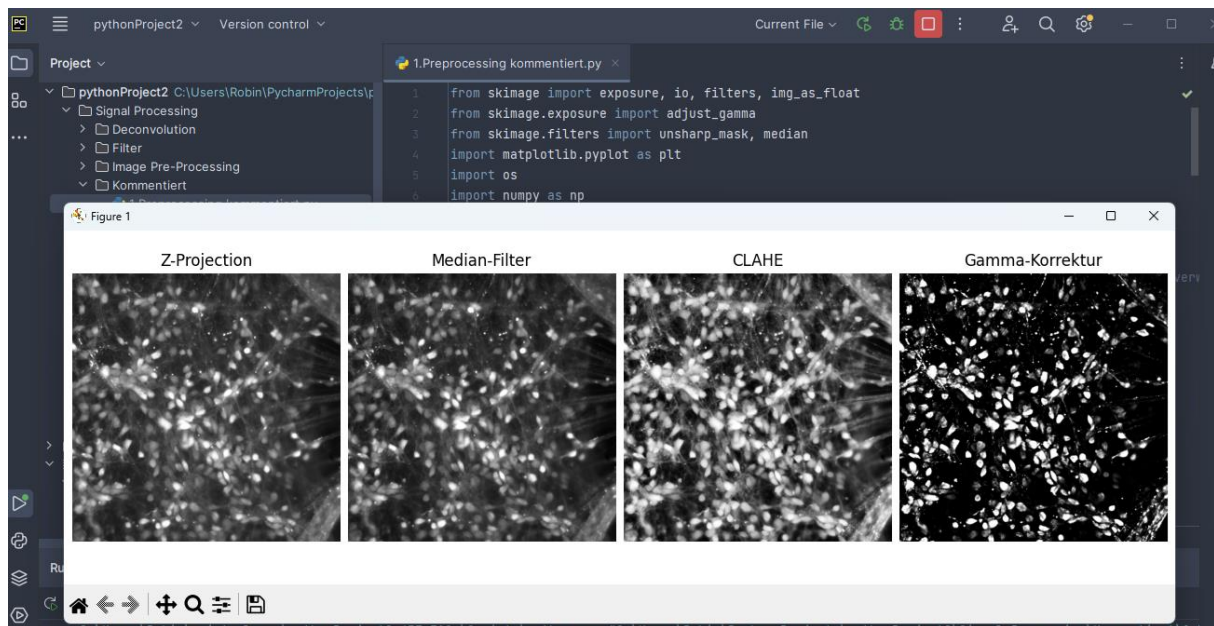


## 1. Preprocessing

Der erste Schritt der Pipeline besteht in der Vorverarbeitung zur Erstellung einer Bilddatei, die für die Weka Segmentierung zur ROI-Detektion verwendet wird (hierfür wird das Skript „1. Preprocessing kommentiert“ verwendet). Durch Abfrage wird der Ort der Bilddateien erfragt. Die Bilddateien sollten zuvor mit der LAS X Software exportiert worden sein. Durch eine weitere Abfrage können die Parameter für die Vorverarbeitung eingegeben werden.



Die zuvor ermittelten Parameter sind: CLAHE Clip-Limit 0.01; Unsharp Mask Radius 10.0; Unsharp Mask Amount 1.0 und Gamma-Wert 4.0. Anpassung der Parameter können notwendig sein, falls das Ergebnis nicht zufriedenstellend ist. Hierzu wurde eine visuelle Kontrolle erstellt, die zur Bewertung vor dem Abspeichern der Datei dient.



## 2. Stackerstellung

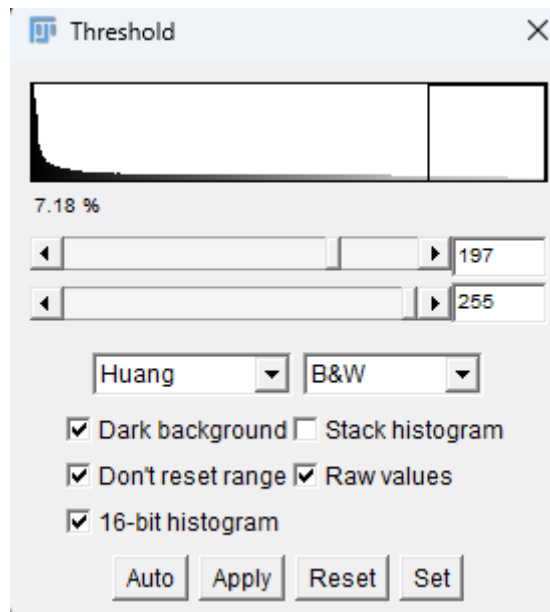
Mit dem Skript „LR kommentiert“ wird eine Bilderserie aus den einzelnen Bilddateien erstellt. Diese Bildserie wird nach erfolgreicher ROI-Detektion in Fiji verwendet, um die Fluoreszenzintensitätswerte auszulesen-

## 3. WEKA Segmentierung

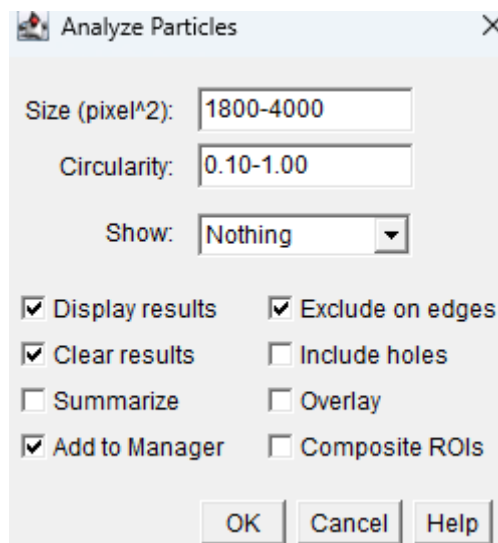
Das zuvor generierte, vorverarbeitete Bild kann nun in Fiji (ImageJ2) geöffnet werden. Für die Segmentierung muss das Segmentierungsfenster aufgerufen werden. Dies erfolgt über **Plugins > Segmentation > Trainable Weka Segmentation**. Sobald sich das Fenster geöffnet hat, können über „Load Data“ die Trainingsdaten des Klassifikators geladen werden. Nach erfolgreichem Import startet die Segmentierung durch Klicken auf „Train Classifier“. Sobald der Vorgang abgeschlossen ist, wird das Bild segmentiert dargestellt (**rot** = Region von Interesse, **grün** = Hintergrund oder Zellfortsätze). Für die weitere Verarbeitung kann über „Get probability“ eine Probability Map erstellt werden. Anschließend kann das Fenster der Weka-Segmentierung geschlossen werden.

Zur weiteren Verarbeitung müssen nun einige Schritte durchgeführt werden:

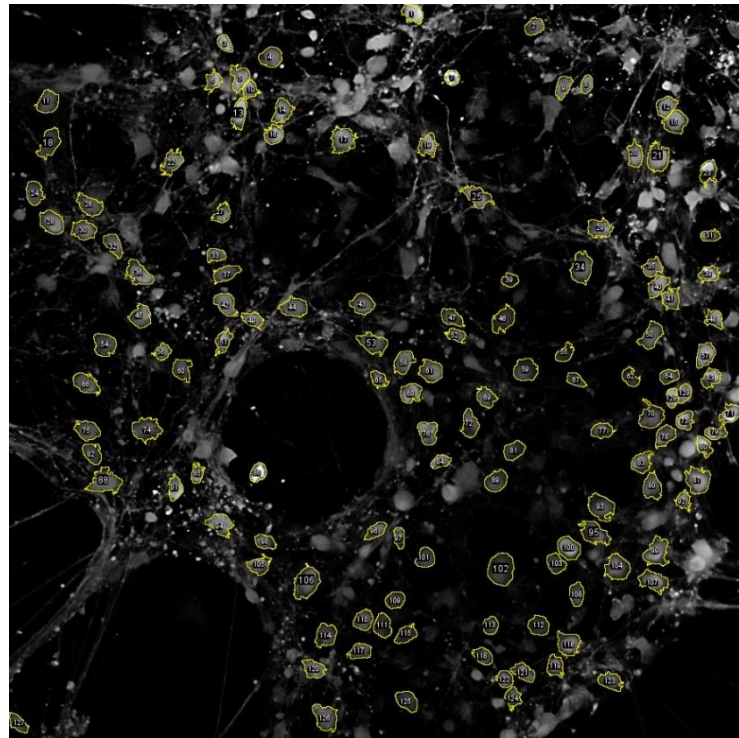
1. **Image > Type > 8-bit**
2. **Image > Adjust > Threshold**
3. Im Threshold-Fenster „**Huang**“ aus dem Drop-Down Menü auswählen (der optimale Wert liegt zwischen 180 und 205)



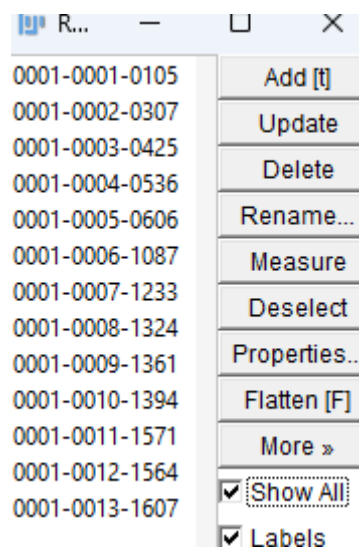
4. Zellkörper schließen und Löcher füllen: **Process > Binary > Close + Fill Holes + Watershed**. Die Zellkörper sollten nun vollständig gefüllt sein.
5. **Analyze > Analyze Particles**: Die Zellgröße in Pixel angeben (1800 bis 4000 Pixel für Aufnahmen mit einem 20x-Objektiv) und mit „OK“ bestätigen.



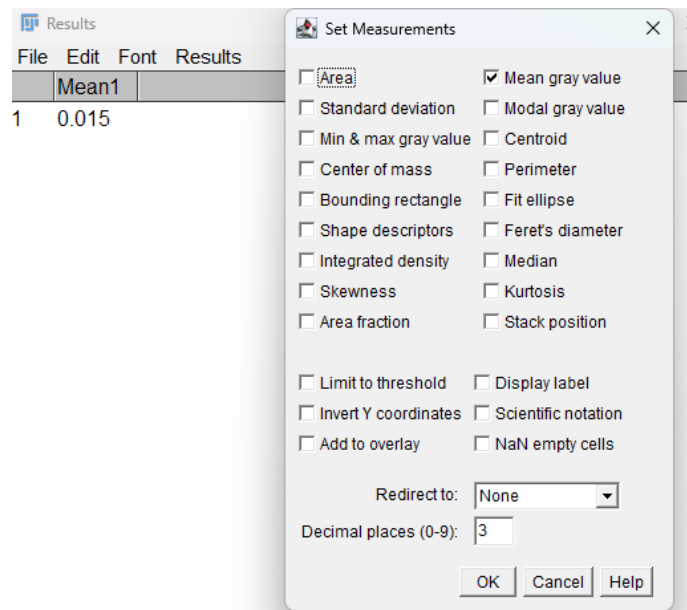
Nachdem die ROIs detektiert wurden, müssen vier Hintergrund-ROIs manuell ausgewählt und dem ROI Manager über „Add“ hinzugefügt werden. Diese werden zur anschließenden Entfernung der Hintergrundfluoreszenz benötigt.



6. Die ROIs können im ROI Manager über More> Save gespeichert werden.



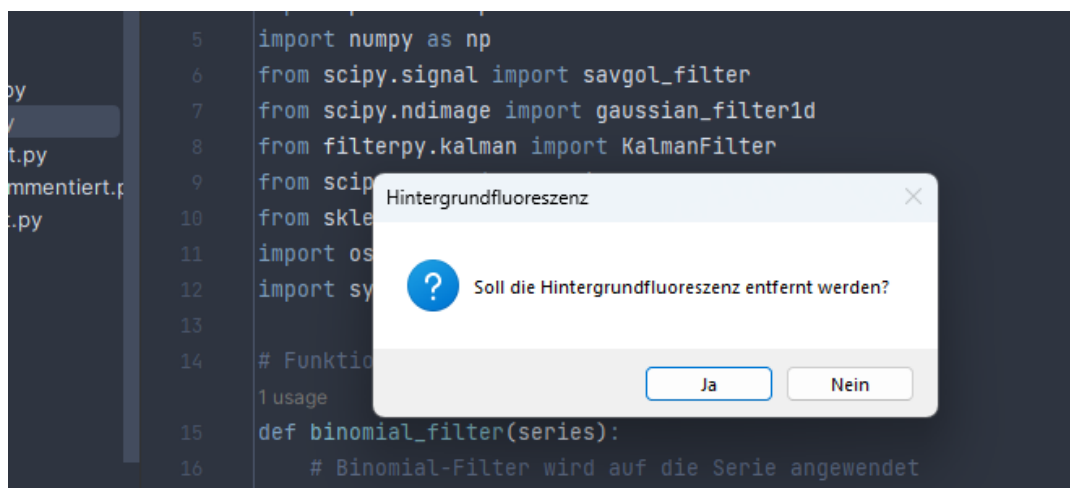
Nach der Speicherung der ROIs können alle Fenster bis auf das Hauptfenster von Fiji geschlossen werden. Nun kann die Bilderserie (Stack) der Aufnahme in Fiji geladen werden. Der ROI Manager wird über **Analyze > Tools > ROI Manager** erneut geöffnet. Die zuvor erstellten ROIs können über **More > Open** geladen werden. Mit der Funktion **More > Multi Measure** werden die Fluoreszenzintensitätswerte der einzelnen ROIs über die gesamte Aufnahme gemessen. Damit die anschließende Verarbeitung korrekt verläuft, sollte im Ergebnisfenster unter **Results > Set Measurements** nur „Mean gray value“ ausgewählt sein.



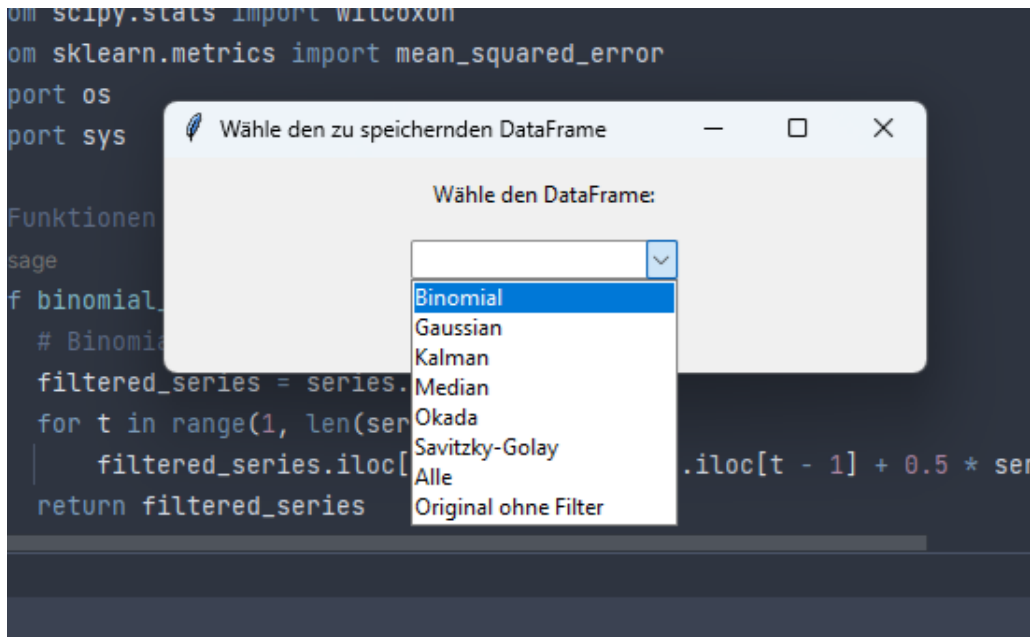
Die Messergebnisse werden mittels File > Save as in einem .csv-Format gespeichert.

#### 4. Filterung

Die gespeicherte .csv kann nun durch das Skript „2. Filterpipeline kommentiert“ verarbeitet werden. Durch Entfernung der Hintergrundfluoreszenz (Abfrage) werden die Zeitreihen zunächst korrigiert. Anschließend werden die verschiedenen Filterverfahren angewendet – statistische Kennzahlen werden in einer separaten Excel (Abfrage) abgespeichert und dienen als Qualitätskontrolle.



Nach erfolgreicher Filterung können spezifische Filterverfahren gespeichert werden oder alle Verfahren als ZIP-Datei gespeichert werden.

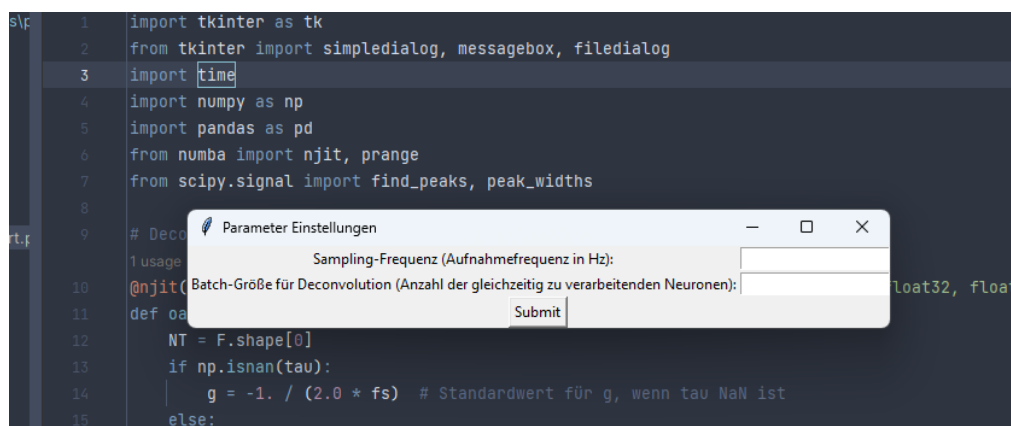


## 5. Photobleaching Korrektur

Nach der Filterung wird die Photobleaching Korrektur angewendet (Skript „3. Photobleaching Korrektur“). Hierfür wird die Excel-Datei der gefilterten Zeitreihen eingelesen – eine Eingabe von Parametern ist nicht erforderlich. Die korrigierten Zeitreihen werden in einer neuen Excel abgespeichert.

## 6. Dekonvolution für spontane Aktivität

**Für die Verarbeitung von Zeitreihendaten spontaner  $\text{Ca}^{2+}$ -Aktivität** wird die Dekonvolution (Skript „4. Dekonvolution Freq Amp kommentiert“) ausgeführt. Nach dem Einlesen der Datei erfolgt eine Abfrage. Hier muss der Nutzer die Aufnahmefrequenz in Hz eingeben (Bsp.: Aufnahmeintervall 100ms entspricht einer Aufnahmefrequenz von 10Hz → Sampling-Frequenz = 10). Zusätzlich kann noch die Batch-Größe angegeben werden, jedoch hat diese bei kleinen Datensätzen keinen signifikanten Einfluss auf die Rechenzeit.

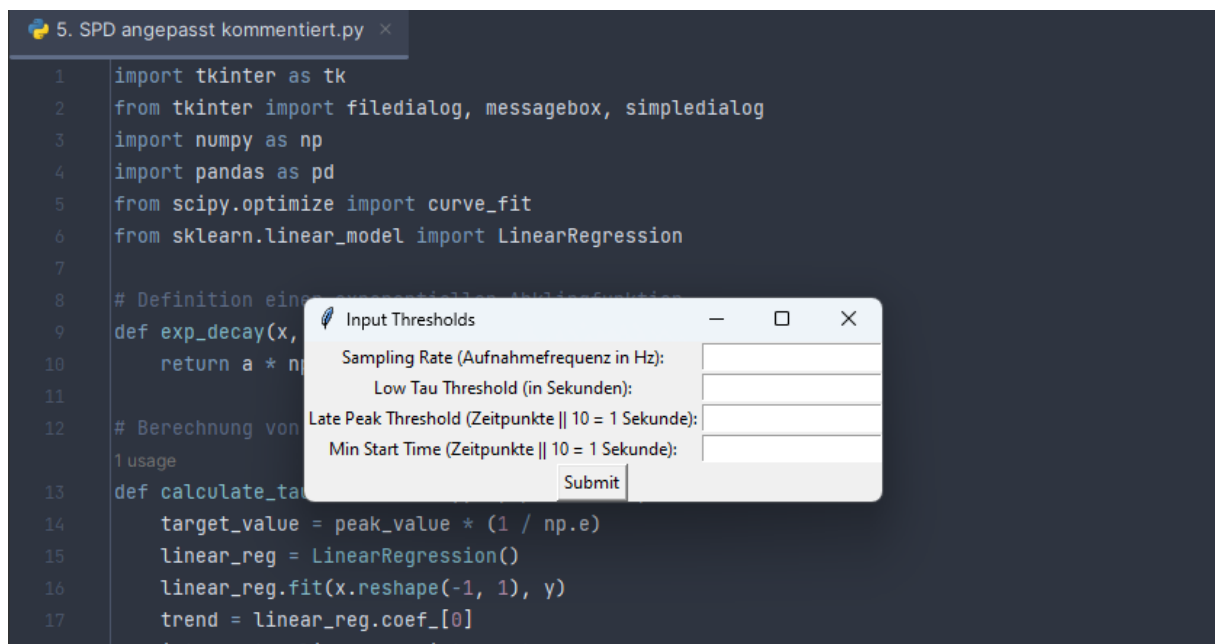


## 7. Single Peak Detektion für induzierte Aktivität

**Für die Verarbeitung von Zeitreihendaten mit einem einzelnen Peak** (z. B. durch induzierte Depolarisation von Nervenzellen) wird das Skript „5. SPD angepasst kommentiert“ ausgeführt. Nach dem Einlesen der Datei werden folgende Parameter abgefragt:

1. **Aufnahmefrequenz:** Wie bei der Dekonvolution (siehe Schritt 6) wird hier die Frequenz der Aufnahmen festgelegt.
2. **Low Tau Threshold:** Legt fest, wie lang die Abklingzeit eines Peaks mindestens sein muss.
3. **Late Peak Threshold:** Gibt an, ab welchem Zeitpunkt keine Peaks mehr berücksichtigt werden sollen. Dies ist besonders relevant bei induzierter Depolarisation, da ein Peak meist wenige Sekunden nach der Zugabe von Substanzen wie KCl auftritt. Später auftretende Peaks können in einem separaten Arbeitsblatt innerhalb der Excel-Datei gespeichert werden. (Hinweis: 10 Zeitpunkte entsprechen 1 Sekunde bei einer Aufnahmefrequenz von 100 ms/10 Hz.)
4. **Min Start Time:** Definiert den Zeitpunkt, ab dem der Algorithmus nach einem Peak sucht. Dieser Zeitpunkt könnte z. B. nach der Applikation von KCl liegen. Wenn die KCl-Zugabe 10 Sekunden nach Beginn der Aufnahme erfolgt, würde dies einer Min Start Time von 100 entsprechen.

Die Ergebnisse werden in einer Excel-Datei gespeichert und können anschließend zur statistischen Auswertung exportiert werden.



```
1 import tkinter as tk
2 from tkinter import filedialog, messagebox, simpledialog
3 import numpy as np
4 import pandas as pd
5 from scipy.optimize import curve_fit
6 from sklearn.linear_model import LinearRegression
7
8 # Definition einer Exponentialfunktion
9 def exp_decay(x, a):
10     return a * np.exp(-x)
11
12 # Berechnung von tau
13 def calculate_tau(target_value, peak_value):
14     target_value = peak_value * (1 / np.e)
15     linear_reg = LinearRegression()
16     linear_reg.fit(x.reshape(-1, 1), y)
17     trend = linear_reg.coef_[0]
```