timsTOF Visualization App Tutorial

Software Installation
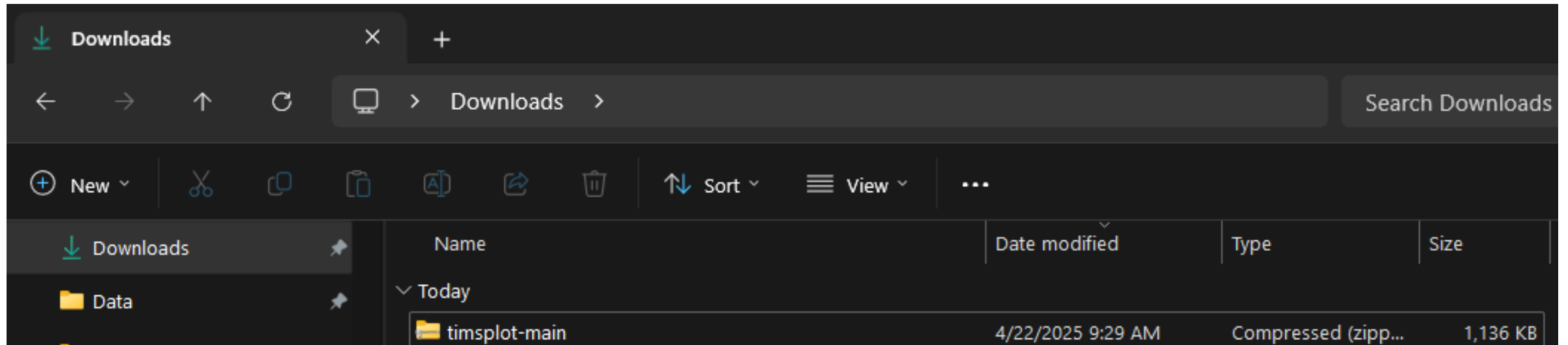
Necessary Software

- Install Python
  - https://www.python.org/downloads/release/python-3128/
    - *NOTE* alphatims is only supported on Python versions prior to 3.13

- Install Visual Studio Code
  - https://code.visualstudio.com/

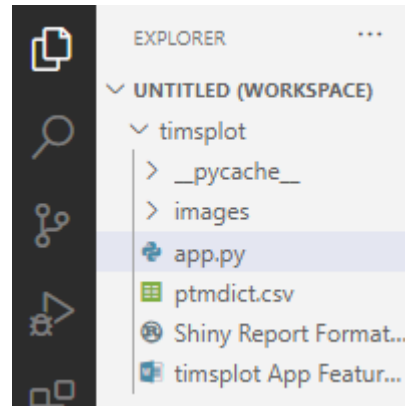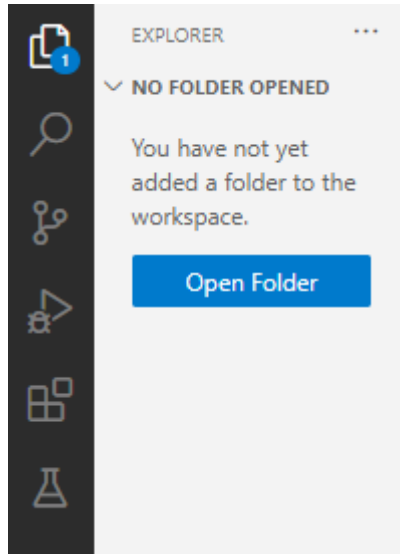# Download folder as a .zip from GitHub
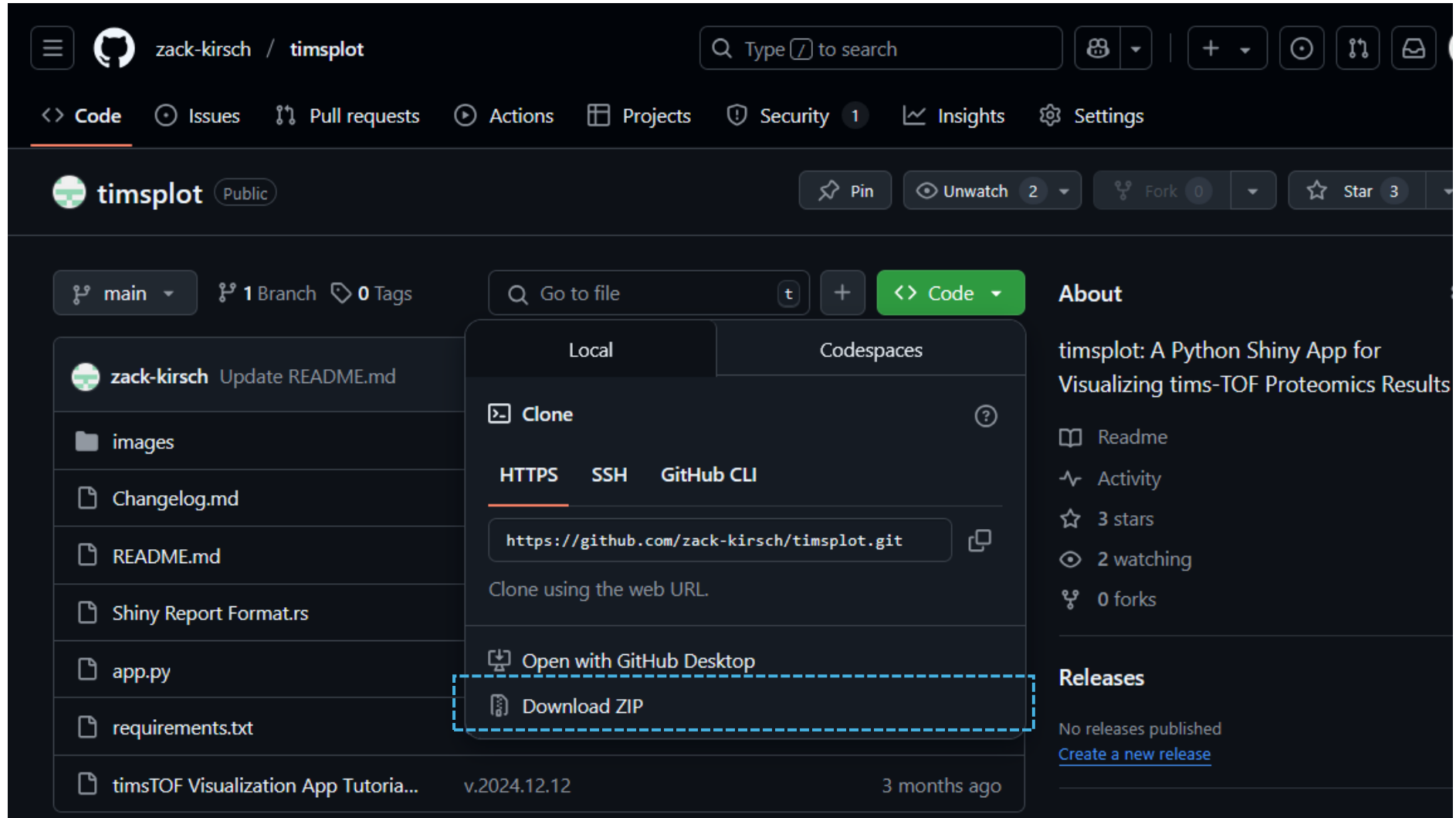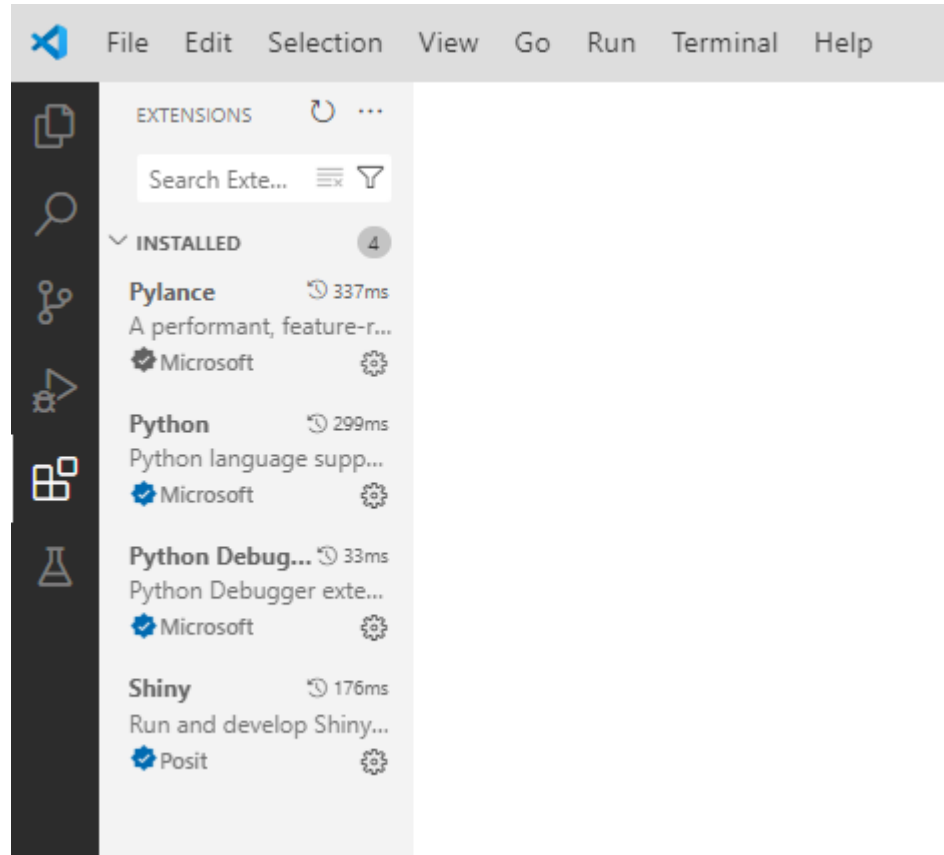
Extract .zip file

Getting Started

Getting started in Visual Studio Code

- Set the directory containing the app.py file as the working directory
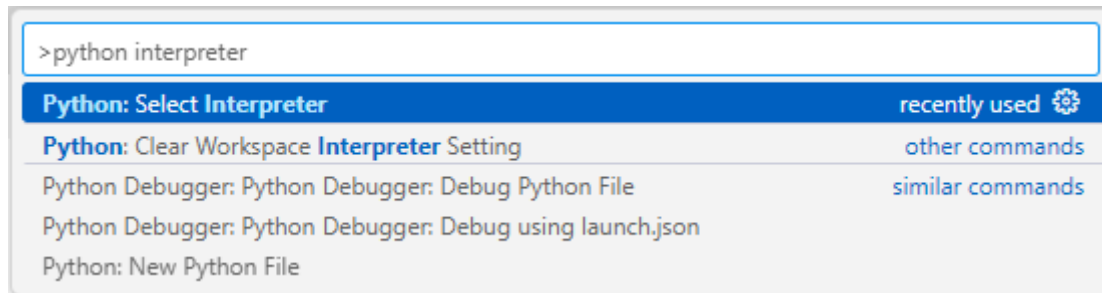- Open the app.py file by double-clicking on the file
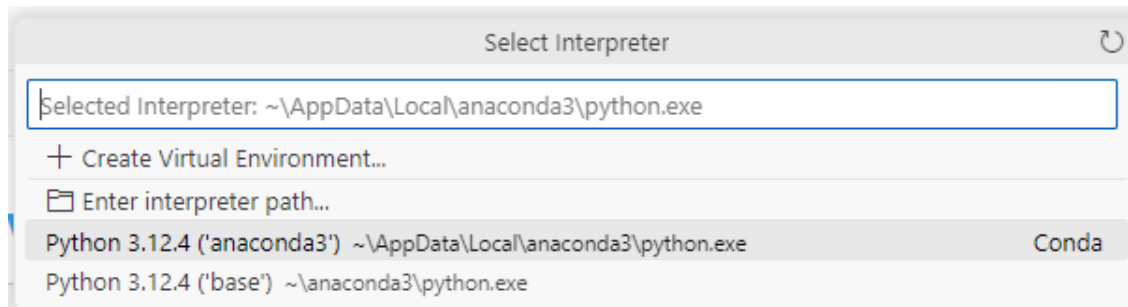
Install necessary extensions

- Install Python and Shiny extensions using the Extensions:Marketplace tab

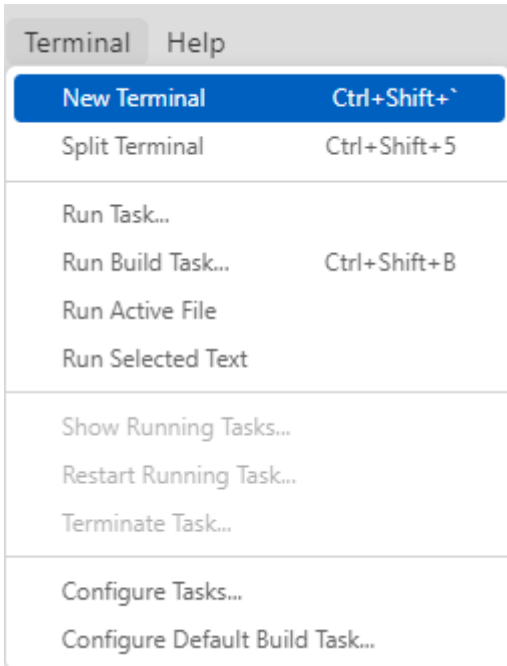# Set Python interpreter to current installation of Python/Anaconda

- Ctrl+Shift+P and search Python interpreter
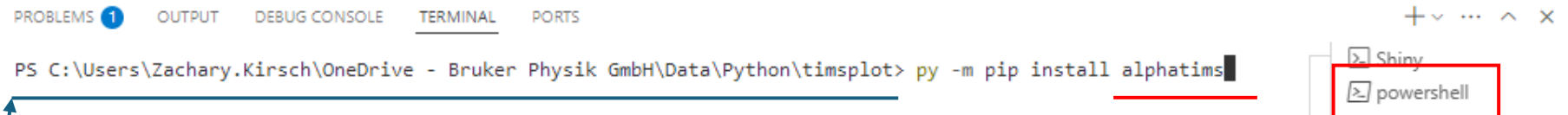

- Select current Python installation

Installing necessary Python libraries

- Open a new powershell terminal and use pip to install
  - alphatims, colorcet, faicons, hvplot, logomaker, matplotlib-venn, pyarrow, scikit-learn, shiny, shinyswatch, and upsetplot
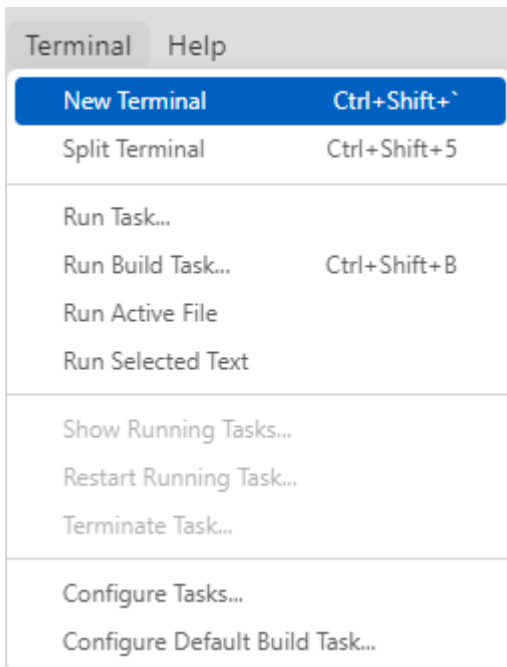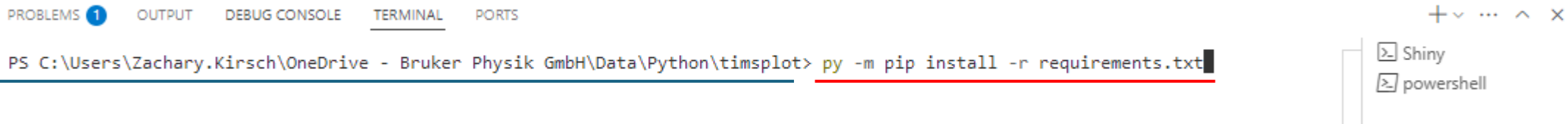
- Replace the part underlined in red with the library to install

This path will just be the file path to the app folder

Installing necessary Python libraries (Alternative)

- Open a new powershell terminal and use pip to install

PROBLEMS **1**   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS   + ∨ ··· ∧ ✕

PS C:\Users\Zachary.Kirsch\OneDrive - Bruker Physik GmbH\Data\Python\timsplot> py -m pip install -r requirements.txt

⯈ Shiny
⯈ powershell

Terminal Help

New Terminal   Ctrl+Shift+`
Split Terminal   Ctrl+Shift+5

Run Task...
Run Build Task...   Ctrl+Shift+B
Run Active File
Run Selected Text

Show Running Tasks...
Restart Running Task...
Terminate Task...

Configure Tasks...
Configure Default Build Task...

- Make sure the requirements.txt file is located in the same folder as the app.py file
  - The path shown in the figure here (underlined in blue) will vary to wherever you moved the downloaded folder from GitHub

Installing necessary Python libraries

```
# ============================================================================== Library Imports (all others needed for calculations)
#region
import alphatims.bruker as atb
import alphatims.plotting as atp
from collections import OrderedDict
from datetime import date
import io
import itertools
from itertools import groupby
import logomaker as lm
import math
import matplotlib
import matplotlib.pyplot as plt
from matplotlib.pyplot import cm
import matplotlib.patches as mpatches
import matplotlib.colors as mcolors
from matplotlib.patches import Rectangle
from matplotlib.ticker import MaxNLocator,MultipleLocator
from matplotlib_venn import venn2,venn2_circles,venn3,venn3_circles
import numpy as np
import os
import pandas as pd
import pathlib
import re
from scipy.cluster.hierarchy import dendrogram, linkage
import scipy.stats as stats
from scipy.stats import norm
import seaborn as sns
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from tkinter import *
from upsetplot import *
from zipfile import ZipFile
```

- If any libraries are underlined in yellow, it means they aren't recognized by the IDE

- If you've already installed a library and it's not being recognized, either restart Visual Studio Code or make sure the correct Python interpreter is selected
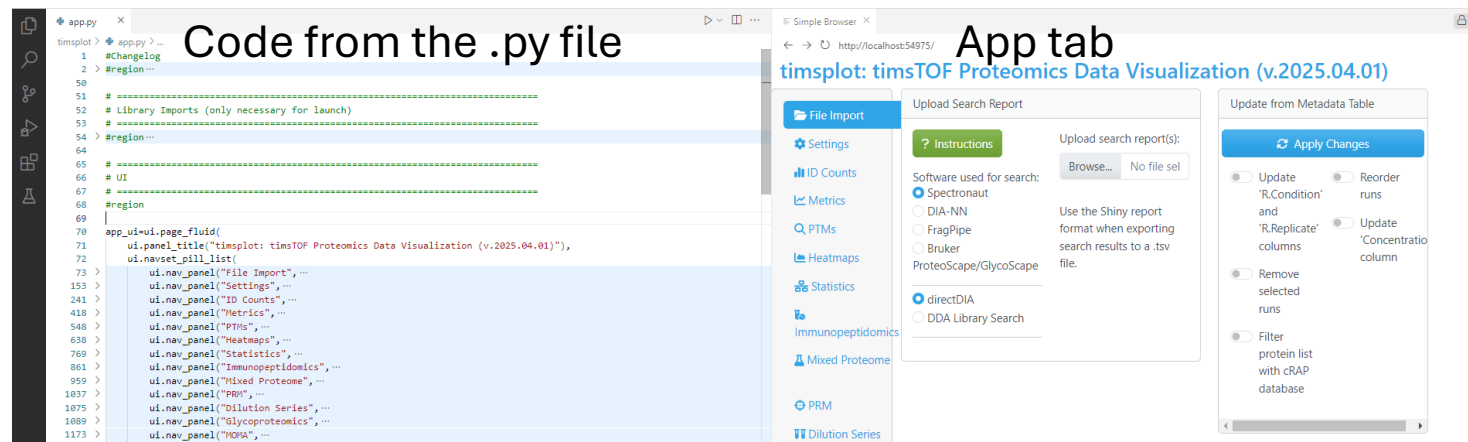
Starting the Shiny app
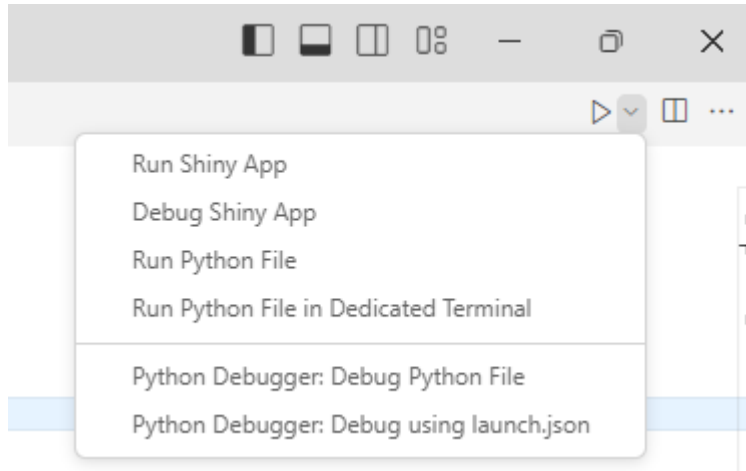
- With the extensions and libraries installed, you should be able to run the app with Visual Studio Code recognizing it as a Shiny app

Starting the Shiny app

- The app interface will launch in a tab within VSCode, which can be maximized to fill the screen by dragging the tab out of VSCode

Code from the .py file

App tab

- It can also be opened in a browser window by clicking this icon in the top right of the app window
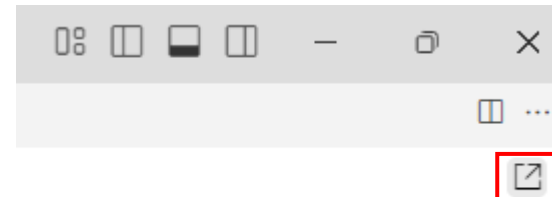
Using the App

File import

Select search software

Upload search report (.tsv, .parquet, .zip)

timsplot: timsTOF Proteomics Data Visualization (v.2025.04.01)

- 📁 File Import
- ⚙ Settings
- 📊 ID Counts
- 📈 Metrics
- 🔍 PTMs
- 📉 Heatmaps
- 🔀 Statistics
- 🔬 Immunopeptidomics
- ⚗ Mixed Proteome
- ⊕ PRM
- 🧪 Dilution Series
- 🔬 Glycoproteomics
- 👤 MOMA
- ⚛ De Novo
- 🔀 Two-Software Comparison
- 📤 Export Tables
- 🖥 Raw Data

Upload Search Report

? Instructions

Software used for search:
- ◉ Spectronaut
- ○ DIA-NN
- ○ FragPipe
- ○ Bruker ProteoScape/GlycoScape

- ◉ directDIA
- ○ DDA Library Search

Upload search report(s):

| Browse... | No file selected |

Use the Shiny report format when exporting search results to a .tsv file.

Update from Metadata Table

🔄 Apply Changes

- ○ Update 'R.Condition' and 'R.Replicate' columns
- ○ Remove selected runs
- ○ Filter protein list with cRAP database

- ○ Reorder runs
- ○ Update 'Concentration' column

Metadata Tables

? Instructions

(Optional) Upload filled metadata table:

| Browse... | No file selected |

📥 Download metadata table as shown

○ Use uploaded metadata table

R.FileName	R.Condition	R.Replicate	remove		R.Condition	order	Concentration

Updating result metadata

This is necessary when condition names are not native to the search report or if changes are needed for the order, presence, or names of samples

If changes are made to the metadata table, use these switches to specify the change and then click the "apply changes" button

R.FileName: (required, auto-filled)
Will be pre-filled from the report file.

R.Condition: (required)
Experimental condition names. If Condition was specified in Spectronaut, it will be added here. Otherwise, it needs to be filled out.

R.Replicate: (required)
Replicate numbers. Should be unique in each condition. . If Replicate was specified in Spectronaut, it will be added here. Otherwise, it needs to be filled out.

Concentration: (optional)
Sets the value for each condition.

Order: (optional)
Specify the numerical order you want the conditions shown in.

Remove: (optional)
Adding an 'x' means that you want to remove the run from the subsequent figures.

Update from Metadata Table

- ⦿ Update 'R.Condition' and 'R.Replicate' columns
- ⦿ Remove selected runs
- ⦿ Reorder runs
- ⦿ Update 'Concentration' column

[Apply changes to search report / reinitialize search report]

Metadata Tables

-Double click on any cell to update its contents

-To remove runs, add an 'x' to the 'remove' column

R.FileName	R.Condition	R.Replicate	remove
20240613_1-1-3_K562-Yeast_EColi_10ng_OFF_1_Slot1-50_1_3848	1-1-3_10ng	1	
20240613_1-1-3_K562-Yeast_EColi_10ng_OFF_2_Slot1-50_1_3849	1-1-3_10ng	2	
20240613_1-1-3_K562-Yeast_EColi_10ng_OFF_3_Slot1-50_1_3850	1-1-3_10ng	3	
20240613_1-2-2_K562-Yeast_EColi_10ng_OFF_1_Slot1-48_1_3844	1-2-2_10ng	1	
20240613_1-2-2_K562-Yeast_EColi_10ng_OFF_2_Slot1-48_1_3845	1-2-2_10ng	2	
20240613_1-2-2_K562-Yeast_EColi_10ng_OFF_3_Slot1-48_1_3846	1-2-2_10ng	3	

R.Condition	order	Concentration
1-1-3_10ng		
1-2-2_10ng		

General Navigation

Coloring Options

- Multiple color setting options for colors to be used in plotting
- Can mix and match from those listed when using custom colors or type in color Hex codes
- If there's only one sample condition, the "Pick for me" sections pick a color at random

Coloring Options

Choose coloring option for output plots:
- ○ Pick for me (rainbow)
- ● Pick for me (matplotlib tableau)
- ○ Custom

Input color names from the tables to the right, one per line:

Note: replicates of the same condition will have the same color

Run	Color per run
1-1-3_10ng	
1-2-2_10ng	

Colors by condition according to the matplotlib tableau colors

Choose coloring option for output plots:
- ● Pick for me (rainbow)
- ○ Pick for me (matplotlib tableau)
- ○ Custom

Input color names from the tables to the right, one per line:

Note: replicates of the same condition will have the same color

Run	Color per run
1-1-3_10ng	
1-2-2_10ng	

Colors by condition according to a rainbow color series split based on the number of conditions

Choose coloring option for output plots:
- ○ Pick for me (rainbow)
- ○ Pick for me (matplotlib tableau)
- ● Custom

Input color names from the tables to the right, one per line:

lightcoral
cornflowerblue

Note: replicates of the same condition will have the same color

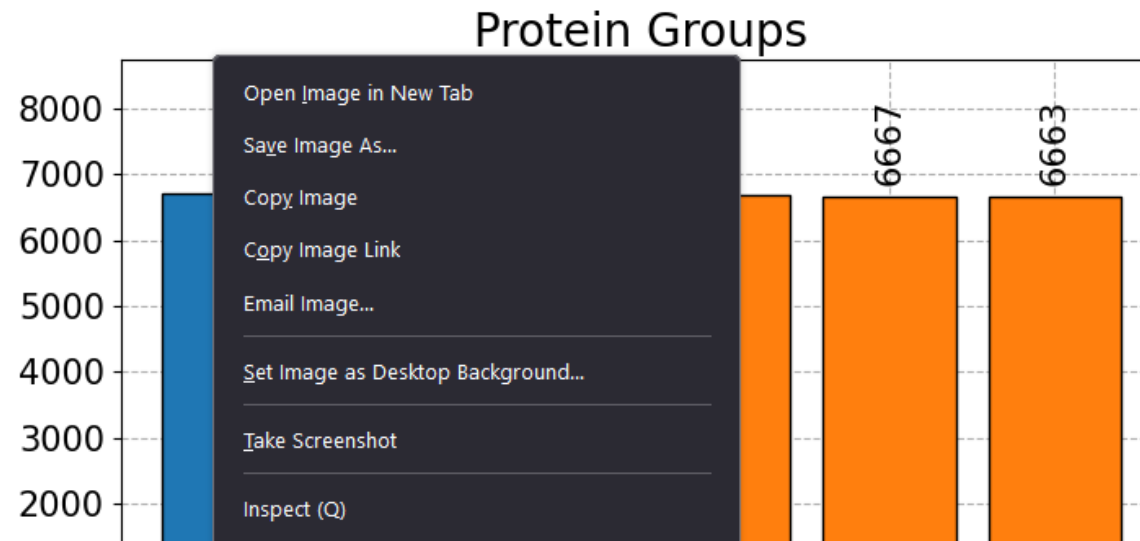| Run	Color per run
1-1-3_10ng	
1-2-2_10ng	

Colors by condition according to specified colors in the text area

Saving rendered plots

- Saving images is easiest when the app is being run in an external browser

- Once a plot is generated and the desired dimensions are set, you can right-click on the plot and save it to a .png file

File Import Formats

Uploading DIA windows to overlay

- CSV Format:

	A	B	C	D	E	F	G
1	#MS Type	Cycle Id	Start IM [1/K0]	End IM [1/K0]	Start Mass [m/z]	End Mass [m/z]	CE [eV]
2	MS1	0	-	-	-	-	-
3	PASEF	1	0.965	1.12	725.13	736.24	-
4	PASEF	1	0.805	0.965	559.8	567.5	-
5	PASEF	1	0.6	0.805	350.68	382.04	-
6	PASEF	2	0.977	1.15	746.34	758.38	-
7	PASEF	2	0.87	0.977	574.21	582.01	-
8	PASEF	2	0.6	0.87	412.39	425.4	-
9	PASEF	3	0.986	1.19	769.41	782.64	-
10	PASEF	3	0.89	0.986	588.81	596.81	-
11	PASEF	3	0.6	0.89	437.42	447.23	-
12	PASEF	4	0.995	1.23	794.87	808.65	-
13	PASEF	4	0.9	0.995	603.8	612.07	-
14	PASEF	4	0.6	0.9	456.05	464.9	-

Uploading peptide list to track

File Import

Settings

ID Counts

Metrics

PTMs

PCA

Heatmaps

Mixed Proteome

PRM

Dilution Series

PRM List   PRM Table   PRM Peptide

Upload Peptide List:

Browse...   No file selected

- CSV Format:

	A	B
1	PG.ProteinGroups	EG.ModifiedPeptide
2	P00359	_TASGNIIPSSTGAAK_
3	P0A853	_KYDIPVVMDSAR_
4	P00330	_[Acetyl (Protein N-term)]SIPETQK_
5	P68104;Q5VTE0	_IGGIGTVPVGR_
6	P00924	_SIVPSGASTGVHEALEM[Oxidation (M)]R_
7	P00549	_GVFPFVFEK_
8	P00924	_IEEELGDNAVFAGENFHHGDK_

Referencing raw data files (individual files)

- Paste the full path of the .d folder for each file you want to plot into the text area

- Should look like the example text shown

- Avoid spaces, use a new line for each file

Referencing raw data files (directory)

- Paste the path to the directory containing raw data files
- The app will search for any .d files in the directory and import them