MANUAL

PySpecTrace

Table of Contents

I. PySpecTrace Requirement and Setup

PySpecTrace has been tested with **Python 3.13** on **Windows 10 and later**. Before running the software, ensure that all required libraries listed in `requirements.txt` are installed.

An **API module** is required for PySpecTrace to read spectra from spectrometers. Commercial spectrometer usually provide the API module upon request.

- **Ocean Optics**: place the API module `OceanDirect.dll` in `Model\OceanSpectrometer\Oceandirect\lib`
- **Avantes**: place the API module `avaspecx64.dll` in `Model\AvantesSpectrometer\SDK`
- **CNILaser**: place the API module `SpectrometersControlLTG.dll` in `Model\CNILaserSpectrometer\SDK`

For further support or to obtain the API modules, please contact the manufacturers directly:

- Ocean Optics: https://www.oceanoptics.com/
- Avantes: https://www.avantes.com/
- CNI Laser: https://www.cnilaser.com/

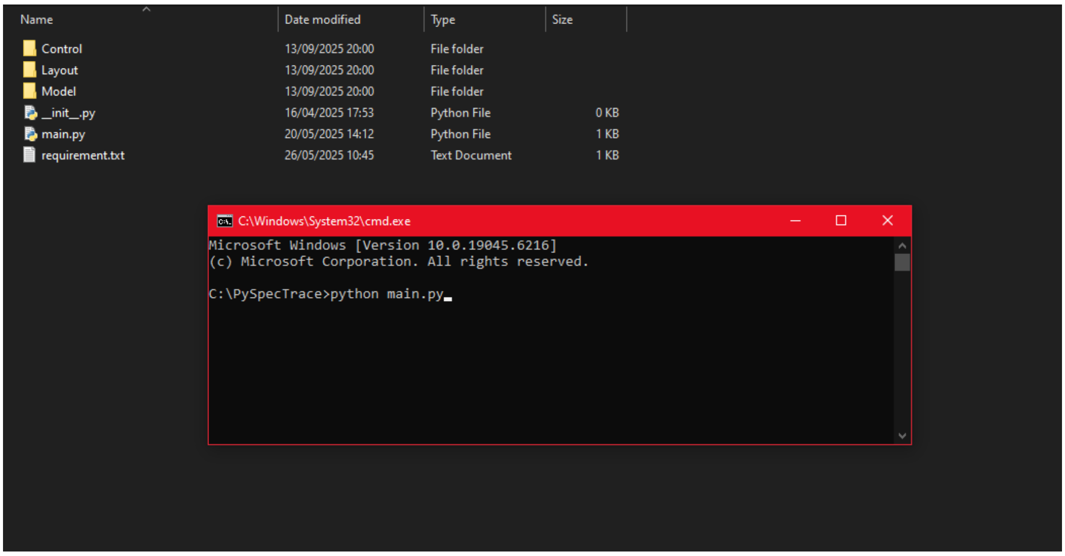
Other brand may also support to PySpecTrace with some compliance.

II. Running The Application

In the directory where `main.py` is located, open a terminal and run (Figure 1):

```
python main.py
```

This will launch the PySpecTrace window. Please note that features will remain unavailable until you select a spectrum source (Figure 2).

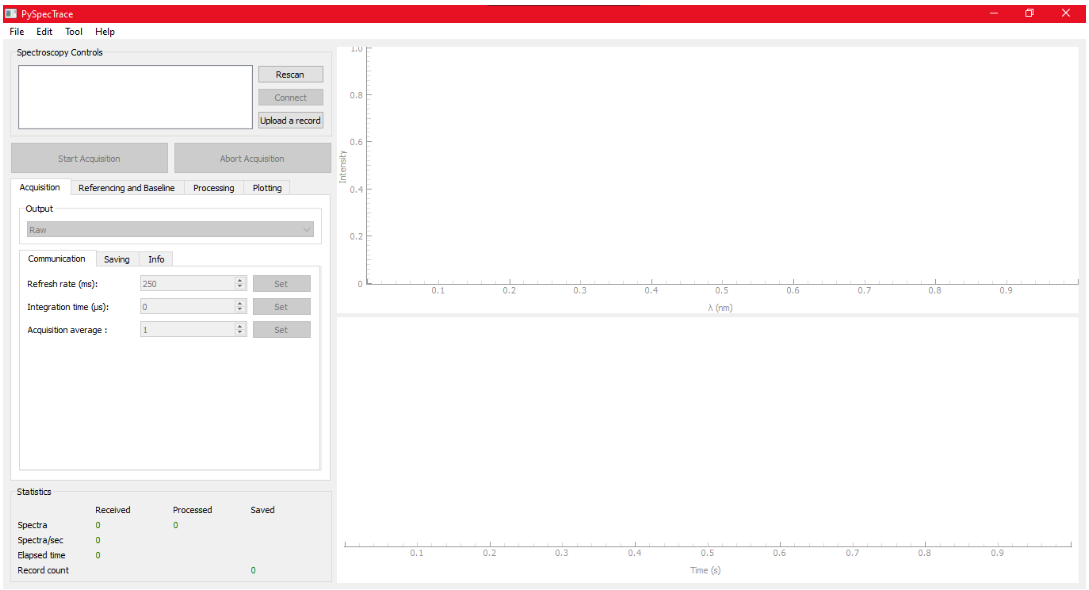


Name	Date modified	Type	Size
Control	13/09/2025 20:00	File folder	
Layout	13/09/2025 20:00	File folder	
Model	13/09/2025 20:00	File folder	
init.py	16/04/2025 17:53	Python File	0 KB
main.py	20/05/2025 14:12	Python File	1 KB
requirement.txt	26/05/2025 10:45	Text Document	1 KB

```
Microsoft Windows [Version 10.0.19045.6216]
(c) Microsoft Corporation. All rights reserved.

C:\PySpecTrace>python main.py_
```

Figure 1 Running PySpecTrace via terminal.



Output: Raw

Refresh rate (ms): 250

Integration time (µs): 0

Acquisition average : 1

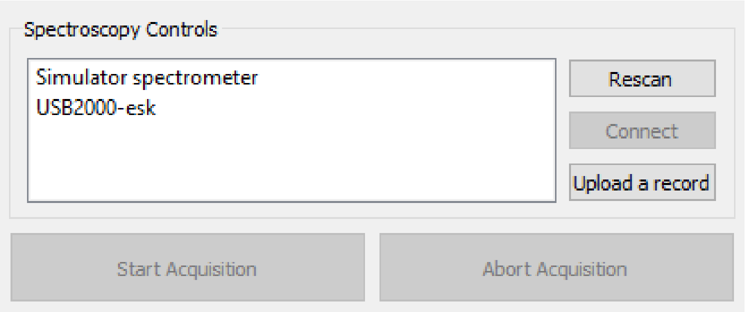
Statistics

	Received	Processed	Saved
Spectra	0	0	
Spectra/sec	0		
Elapsed time	0		
Record count			0

Figure 2 PySpecTrace interface.

III. Selection of Spectrum Source

Available spectrum sources will be listed every time the **Rescan** button is pressed (Figure 3). After selecting the spectrum source, click **Connect**, then click **Start Acquisition** to begin.

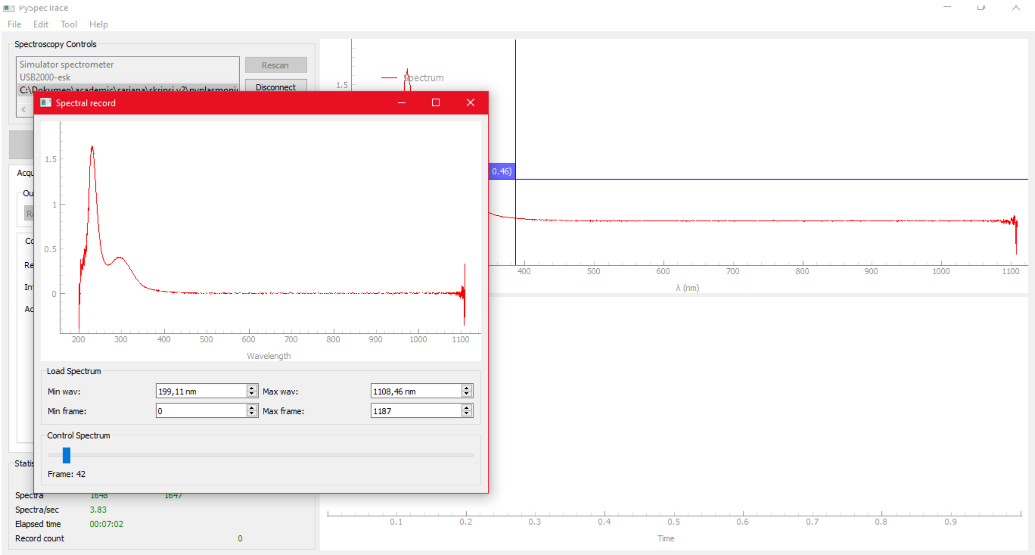


Spectroscopy Controls

Simulator spectrometer
USB2000-esk

Figure 3 Spectrum source panel.

One can choose the spectrum source, and there are three possibilities:(1) a recorded spectrum (via the **upload a record** button) (Figure 4), (2) a built-in spectrometer simulator (Figure 5), and (3) a spectrometer (Figure 6).



Spectroscopy Controls

Simulator spectrometer
USB2000-esk

Spectral record

Load Spectrum

Min wav: 199,11 nm
Max wav: 1108,46 nm
Min frame: 0
Max frame: 1187

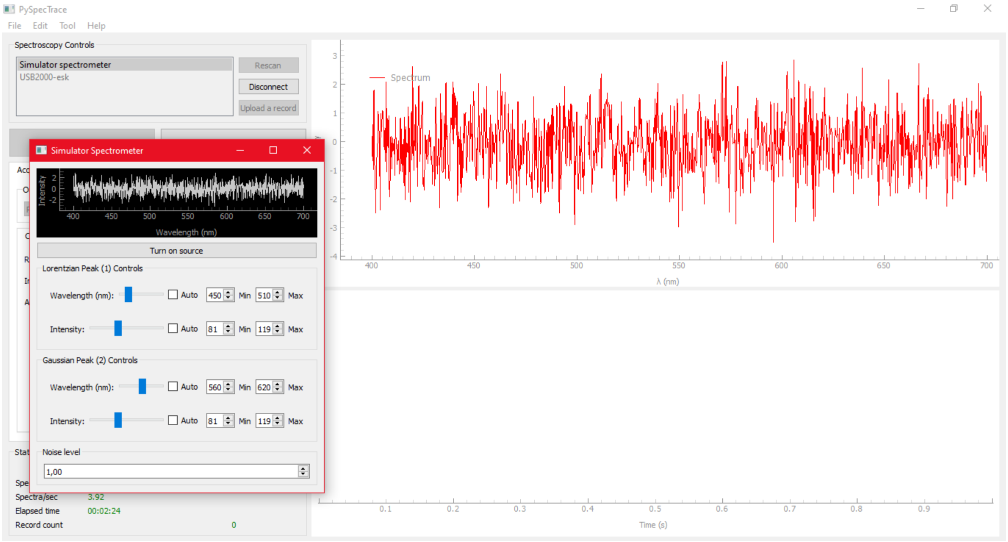
Control Spectrum

Frame: 42

Spectra/sec 3.83
Elapsed time 00:07:02
Record count 0

Figure 4 Spectrum display from the recorded spectrum source.

The recorded spectrum feature is intended for post-processing previously stored spectra. In this mode, users can establish playback boundaries and conduct a quick review of the spectrum by adjusting the provided control bar. The timestamp associated with each spectrum precisely reflects the moment the experiment was conducted, thereby maintaining the synchronization and accuracy of the sensorgram. Prior to engaging in post-processing, the spectrum file must first be converted into the .npz format, as detailed in the menus bar section.



Lorentzian Peak (1) Controls

Wavelength (nm): □ Auto 450 Min 510 Max

Intensity: □ Auto 81 Min 119 Max

Gaussian Peak (2) Controls

Wavelength (nm): □ Auto 560 Min 620 Max

Intensity: □ Auto 81 Min 119 Max

Noise level: 1,00

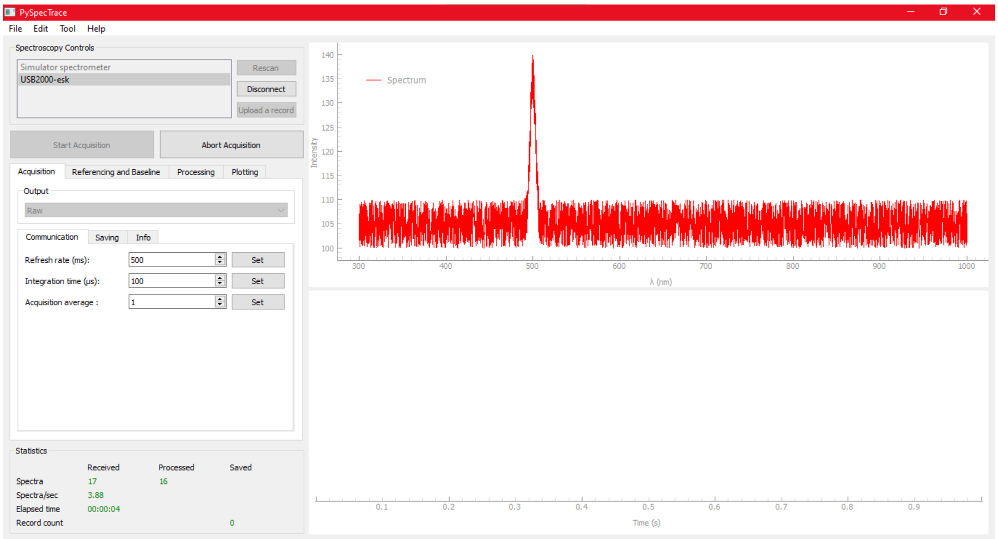
Spectra/sec 3.92

Elapsed time 00:02:24

Record count 0

Figure 5 Spectrum display from the spectrometer simulator source.

In the simulator spectrometer mode (Figure 5), users possess comprehensive authority to generate customized spectra. They can create Gaussian and Lorentzian peaks, implement dynamic peak shifts, modify intensity, and set dynamic range boundaries. The simulator further permits the addition of noise to the spectrum and allows toggling the light source on or off, which is advantageous for simulating a dark reference.



Output: Raw

Refresh rate (ms): 500

Integration time (µs): 100

Acquisition average : 1

Statistics

	Received	Processed	Saved
Spectra	17	16	
Spectra/sec	3.88		
Elapsed time	00:00:04		
Record count			0

Figure 6 Spectrum display from the spectrometer source.

IV. Features and Configuration

1. Spectrum Output

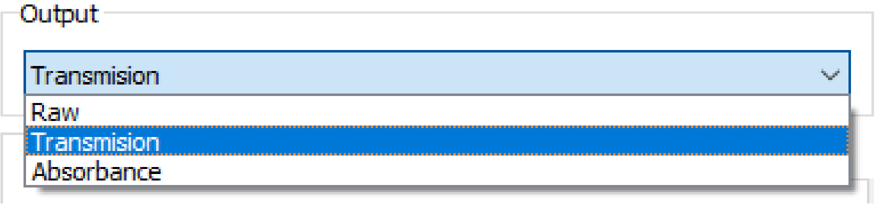
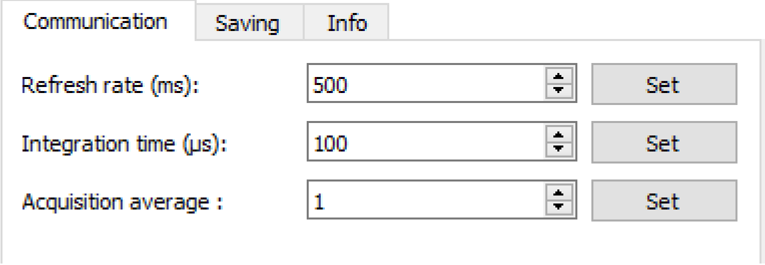


Figure 7 Drop-down menu to select processed spectrum output mode.

After acquiring both the dark spectrum and the bright spectrum, the user can select between different output modes. The available options are transmission and absorbance, which depend on the type of the aimed analysis (Figure 7).

2. Data acquisition parameter



Refresh rate (ms): 500

Integration time (µs): 100

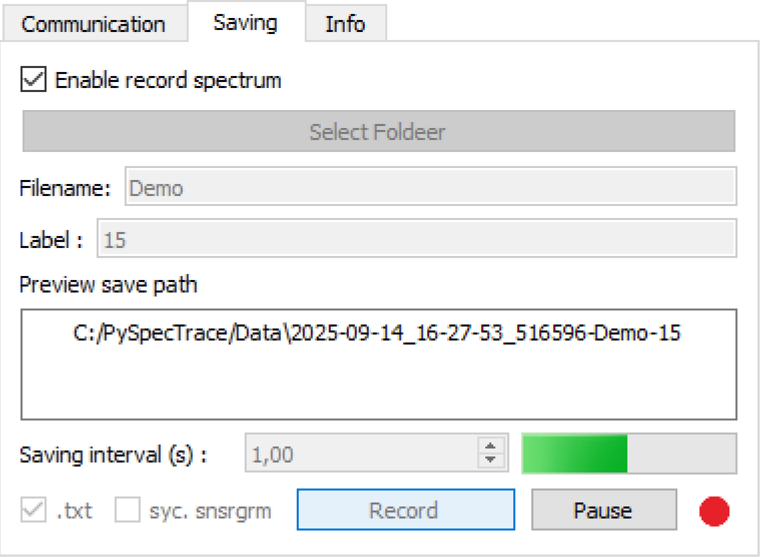
Acquisition average : 1

Figure 8 Setup for hardware-to-software data acquisition parameter.

The communication between the software and the spectrometer can be configured in this section (Figure 8). The **refresh rate** determines how frequently the GUI retrieve the data. The **integration time** specifies how long the sensor accumulates incoming photons i.e. higher values leads to stronger signals. The **acquisition average** defines the number of spectra that are collected to compute an averaged spectrum, which can be expressed mathematically as follows:

$$Spectrum = \frac{1}{Acquisition\ average} \sum_{i=1}^{Acquisition\ average} Spectrum_i$$

3. Data Saving



☑ Enable record spectrum

Filename: Demo

Label : 15

Preview save path

C:/PySpecTrace/Data\2025-09-14_16-27-53_516596-Demo-15

Saving interval (s) : 1,00

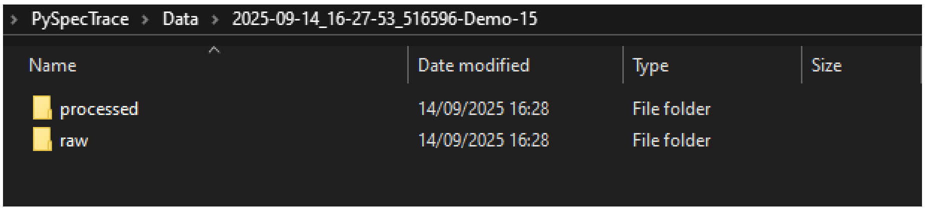
☑ .txt □ syc. snsrgrm

Figure 9 Saving the recorded spectrum acquired from the spectrometer.

The saving feature allows users to record spectra from a selected source (Figure 9). This is initiated by ticking the **Enable Record Spectrum** box, then choosing a destination folder, filename, and label. The selected save directory will be displayed in the **Preview Save Path** box.

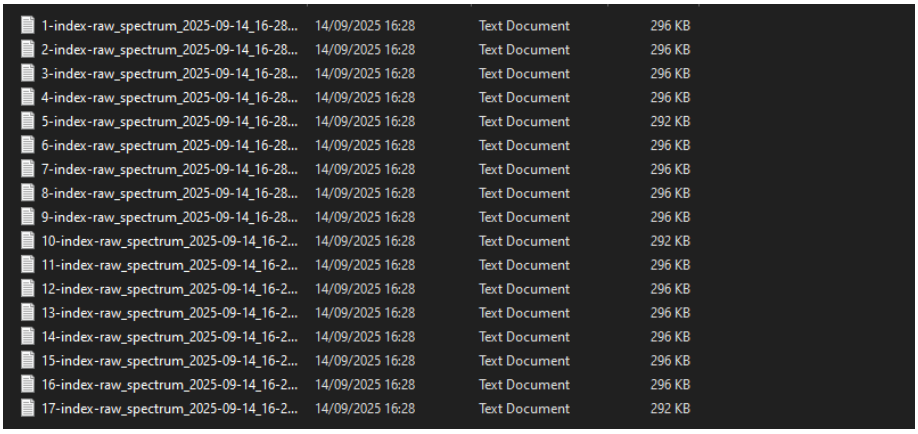
The saving interval can be configured according to the user's needs. For example, if the interval is set **to** 1 second**,** PySpecTrace will save the captured spectrum every second. Recording starts once the **Record** button is pressed. During recording, you can pause the process if necessary (for instance, when performing adjustments to the sample) to ensure that your data remains seamless.

The **Sys. Snsrgm (Synchronization Sensorgram)** checkbox provides a useful feature for resetting the sensorgram right when the recording is started. This is particularly helpful when the user prefer to synchronize sensorgram data with spectrum data. The saving interval is displayed as a green progress bar, and the active recording is indicated by the red marker.



› PySpecTrace › Data › 2025-09-14_16-27-53_516596-Demo-15

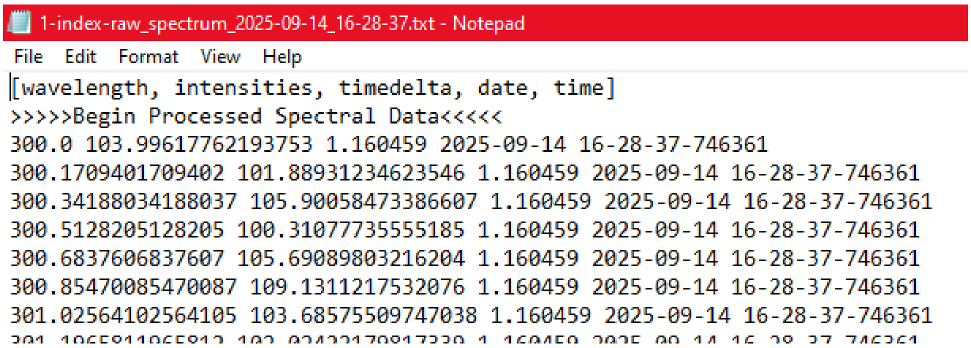
Name	Date modified	Type	Size
processed	14/09/2025 16:28	File folder	
raw	14/09/2025 16:28	File folder	



1-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
2-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
3-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
4-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
5-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	292 KB
6-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
7-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
8-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
9-index-raw_spectrum_2025-09-14_16-28...	14/09/2025 16:28	Text Document	296 KB
10-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	292 KB
11-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
12-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
13-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
14-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
15-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
16-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	296 KB
17-index-raw_spectrum_2025-09-14_16-2...	14/09/2025 16:28	Text Document	292 KB

Figure 10 The saving automatically create two folders (upper panel) that contain txt. files (bottom panel).

The recording process yields two types of output: the raw spectrum and the processed spectrum (Figure 10). The raw spectrum is preserved without any bright/dark correction, nor baseline correction. Conversely, the processed spectrum incorporates these corrections. Both outputs are stored as .txt files, with each file containing the spectrum data recorded at the respective time.



1-index-raw_spectrum_2025-09-14_16-28-37.txt - Notepad
File Edit Format View Help

```
[wavelength, intensities, timedelta, date, time]
>>>>>Begin Processed Spectral Data<<<<<
300.0 103.99617762193753 1.160459 2025-09-14 16-28-37-746361
300.1709401709402 101.88931234623546 1.160459 2025-09-14 16-28-37-746361
300.34188034188037 105.90058473386607 1.160459 2025-09-14 16-28-37-746361
300.5128205128205 100.31077735555185 1.160459 2025-09-14 16-28-37-746361
300.6837606837607 105.69089803216204 1.160459 2025-09-14 16-28-37-746361
300.85470085470087 109.1311217532076 1.160459 2025-09-14 16-28-37-746361
301.02564102564105 103.68575509747038 1.160459 2025-09-14 16-28-37-746361
301.1965811965812 103.02422170817339 1.160459 2025-09-14 16-28-37-746361
```

Figure 11 Structure of each spectra txt. File.

Each recorded txt. file contains several parameters, including wavelength, intensity, date, time, and delta-time (Figure 11). The delta-time indicates the time difference or the storage time interval between consecutive spectra.Referencing and baseline configuration

4. Background corrections

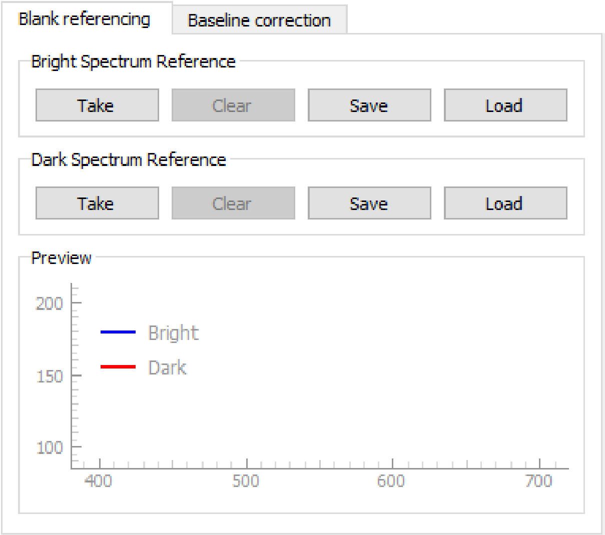


Figure 12 Panel for configuration of bright and dark backgrounds.

To acquire both dark and bright background, press the **Take** button (Figure 12). To remove it, press **Clear**. The acquired background spectrum can also be saved for future use with the **Save** button and later reloaded using the **Load** button. Figure 13 and Figure 14 display examples of bright and dark background spectra, respectively.

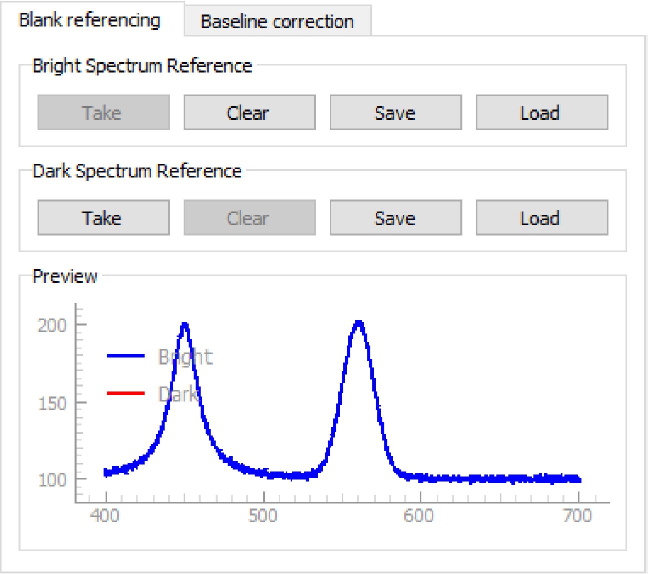


Figure 13 Example spectrum of bright background.

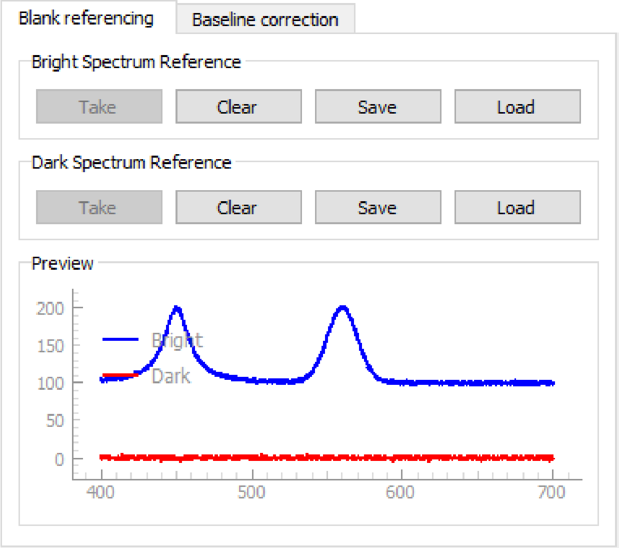
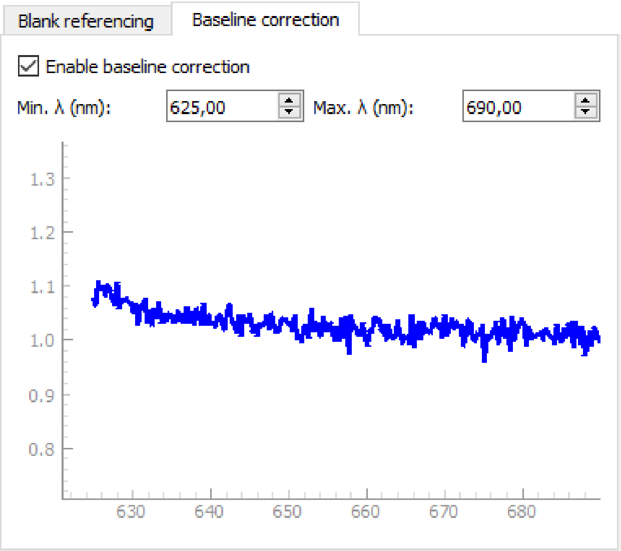


Figure 14 Example spectrum of dark background.

5. Baseline Correction



☑ Enable baseline correction

Min. λ (nm): 625,00 Max. λ (nm): 690,00

Figure 15 Selection of wavelength range for baseline correction.

The baseline correction feature adjusts a selected portion of the spectrum to zero level, aiming to eliminate artifacts such as signal drift and fluctuations (Figure 15). To enable this feature, tick the **Enable Baseline Correction** box and define the wavelength range for baseline. The software calculates the average value within this range and substracts the spectrum by this value.

6. Spectrum Fitting and Tracing.

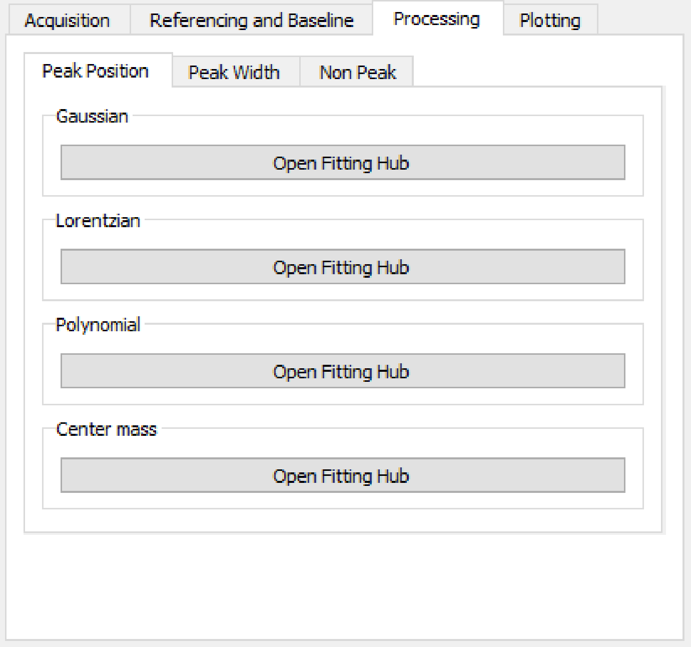
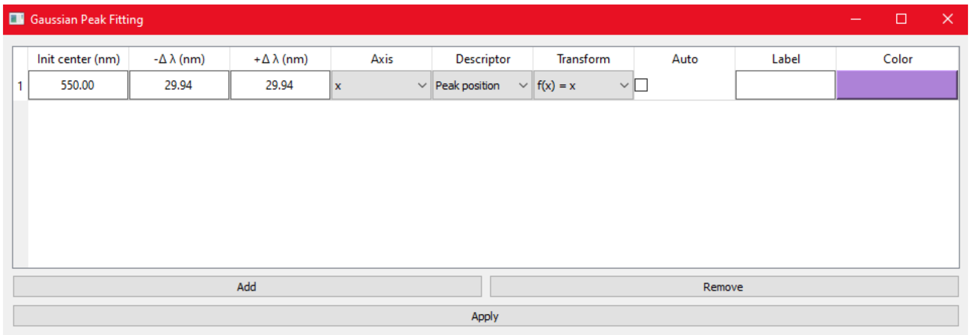


Figure 16 Three options for spectrum tracing: Peak position, peak width, and non-peak (intensity change tracing).

Several key features are available for tracing spectra. These include tracing of **peak position**, **peak width**, and **non-peak (**Figure 16**)**. Multiple tracings can be applied simultaneously, and any additions or removals the tracing are confirmed by pressing the **Apply** button (see the detail explanation below).

a) Peak Position

(1) Gaussian

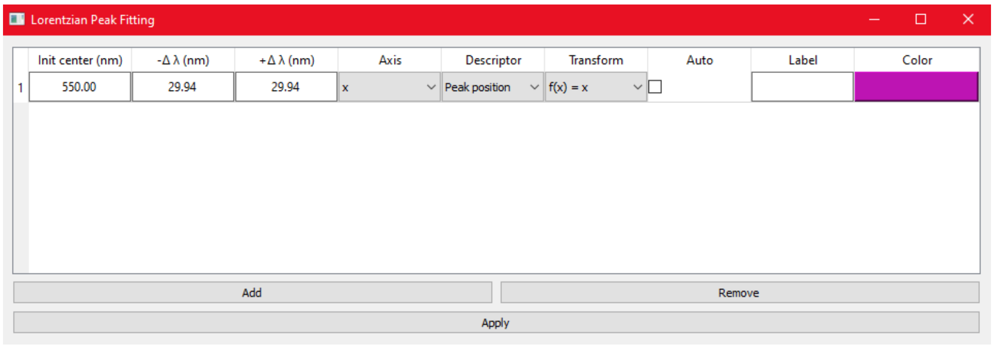


	Init center (nm)	-Δ λ (nm)	+Δ λ (nm)	Axis	Descriptor	Transform	Auto	Label	Color
1	550.00	29.94	29.94	x	Peak position	f(x) = x	□		🟪

Figure 17 Peak position tracing with Gaussian fitting.

The **init. center** represents the initial peak guess for the Gaussian fit (Figure 17). The parameters **Δ-** and **Δ+** define the fitting range to the left and right of the **init. center**, respectively. Tracing can be configured to track either the x-axis or the y-axis. Available descriptors include the **peak position** and **inflection point**, which can be directly transformed into various mathematical equations. The **Auto** checkbox enables a dynamic peak guess that follows peak shifts i.e. the n-th peak guess is obtained from the (n-1)-th peak fitting output. Additionally, the **Label** field enables user to assign a label name for display, while the **Color** field lets the user to specify a color for the trace.

(2) Lorentzian

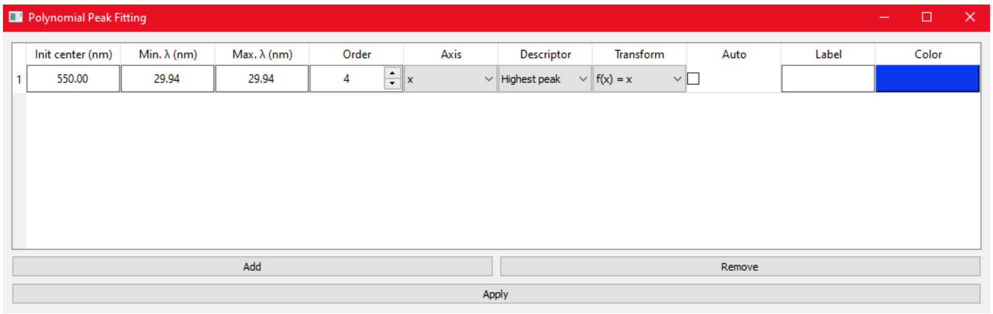


	Init center (nm)	-Δ λ (nm)	+Δ λ (nm)	Axis	Descriptor	Transform	Auto	Label	Color
1	550.00	29.94	29.94	x	Peak position	f(x) = x	□		🟪

Figure 18 Peak position tracing with Lorentzian fitting.

Explanation is similar to that of Gaussian fitting above.

(3) Polynomial

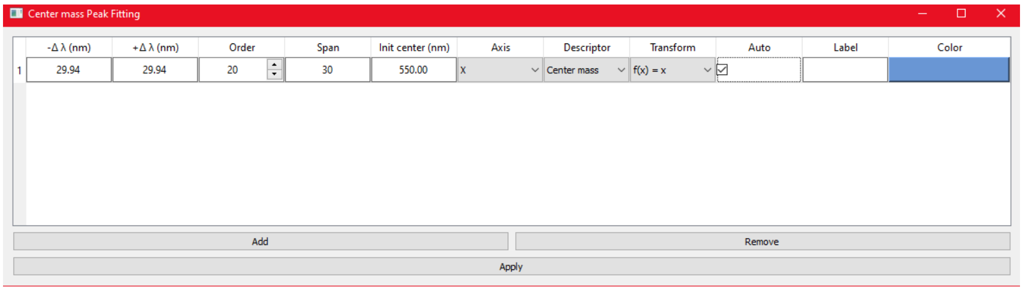


	Init center (nm)	Min. λ (nm)	Max. λ (nm)	Order	Axis	Descriptor	Transform	Auto	Label	Color
1	550.00	29.94	29.94	4	x	Highest peak	f(x) = x	□		🟦

Figure 19 Peak position tracing with polynomial fitting.

Parameter setting for polynomial fitting is displayed in Figure 19. The minimum supported polynomial order is **3**. The tracing output can be opted as the x or the y-coordinate. Available descriptors include the **highest peak position** and the **lowest peak position**.

(4) Peak Center of Mass (Centroid)



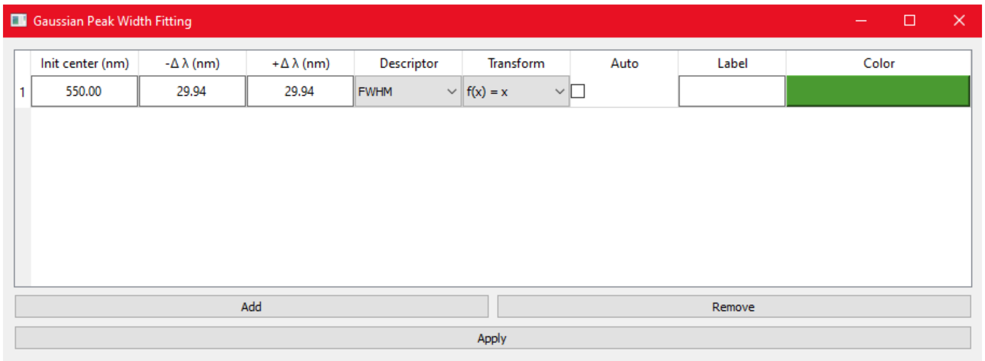
	-Δ λ (nm)	+Δ λ (nm)	Order	Span	Init center (nm)	Axis	Descriptor	Transform	Auto	Label	Color
1	29.94	29.94	20	30	550.00	x	Center mass	f(x) = x	☑		🟦

Figure 20 Peak position tracing with center mass (centroid) fitting.

Parameter setting for centroid tracing is displayed in Figure 20. The **order** refers to the degree of the polynomial used for fitting. In this mode, the output descriptor is limited to tracking the **x-axis at the center of mass**. The **Span** represents the wavelength range in which the peak base is defined.

b) Peak Width

(1) Gaussian

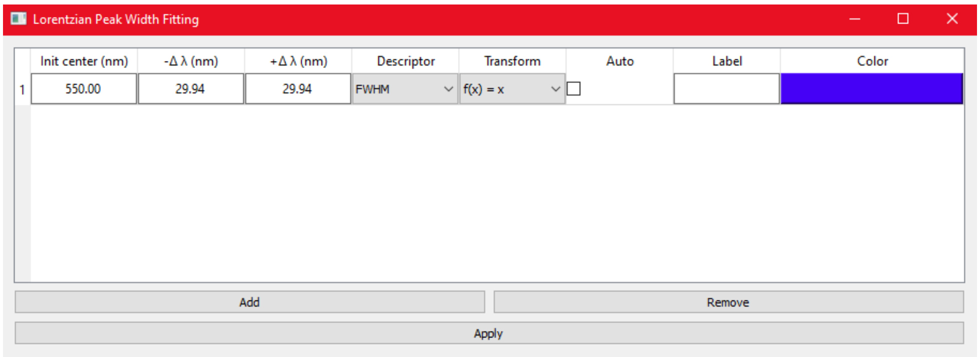


	Init center (nm)	-Δ λ (nm)	+Δ λ (nm)	Descriptor	Transform	Auto	Label	Color
1	550.00	29.94	29.94	FWHM	f(x) = x	□		🟩

Figure 21 Peak width tracing with Gaussian fitting.

The **init. center** represents the initial peak guess for the Gaussian fit (Figure 21). The parameters **Δ-** and **Δ+** define the fitting range to the left and right of the init center, respectively. Tracing can be configured to track either the x-axis or the y-axis. The peak width is defined as FWHM (full-width at half-maximum), refer to Supporting Information for details. The **Auto** checkbox enables a dynamic peak guess that follows peak shifts i.e. the n-th peak guess is obtained from the (n-1)-th peak fitting output. Additionally, the **Label** field enables user to assign a label name for display, while the **Color** field lets the user to specify a color for the trace.

(2) Lorentzian

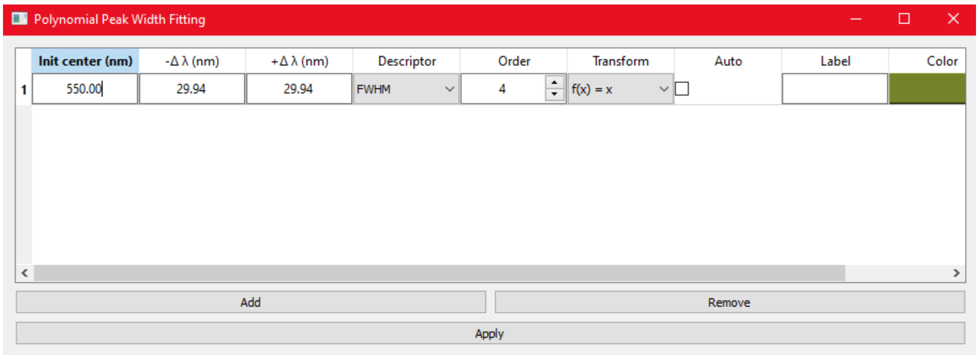


	Init center (nm)	-Δ λ (nm)	+Δ λ (nm)	Descriptor	Transform	Auto	Label	Color
1	550.00	29.94	29.94	FWHM	f(x) = x	□		🟦

Figure 22 Peak width tracing with Lorentzian fitting

Same explanation as the Gaussian fit above.

(3) Polynomial

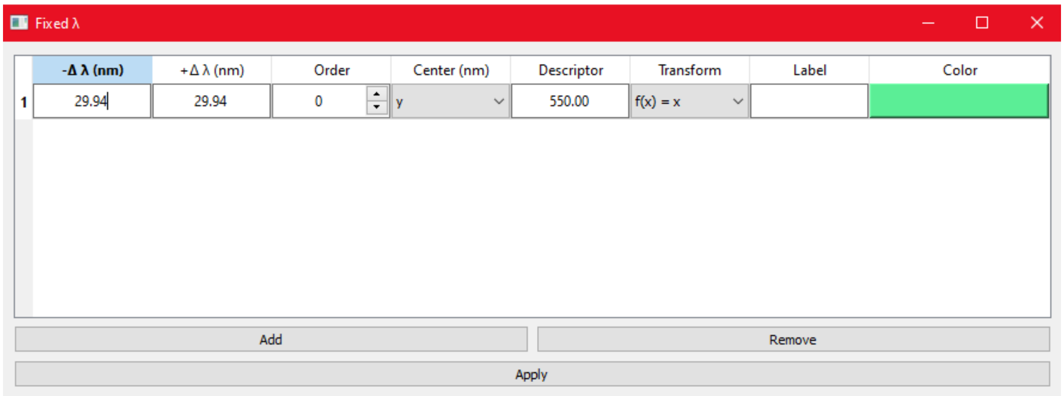


	Init center (nm)	-Δ λ (nm)	+Δ λ (nm)	Descriptor	Order	Transform	Auto	Label	Color
1	550.00	29.94	29.94	FWHM	4	f(x) = x	□		🟩

Figure 23 Peak width tracing with polynomial fitting

Setting panel for peak width tracing using polynomial is presented in Figure 23. The peak width tracing with polynomial fitting is defined as the **peak width at half height**.

c) *Tracing Intensity Change (Non-peak tracing)*



	-Δ λ (nm)	+Δ λ (nm)	Order	Center (nm)	Descriptor	Transform	Label	Color
1	29.94	29.94	0	y	550.00	f(x) = x		🟩

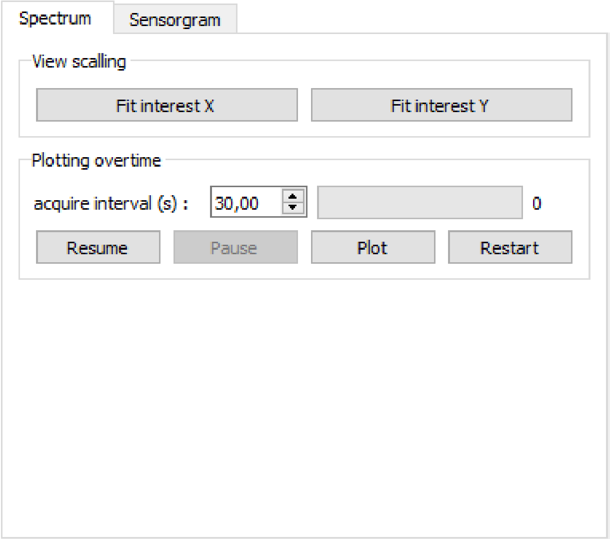
Figure 24 Tracing spectrum intensity change at fixed wavelength.

PySpecTrace also provides tracing functionality for a spectrum in which a peak does not exist (or is too weak) (Figure 24). In this case, the user may track the **intensity change at a fixed wavelength (λ)**. This feature essentially employs polynomial fitting. When the order is set to **0**, tracing is output is the actual intensity at the target wavelength of the spectrum. For orders greater than **0**, tracing is applied after a polynomial fit is applied.

7. Data Visualization

Data visualization consists of visualization for the spectrum and sensorgram, with details explained as follows.

a) *Spectrum Display Panel*

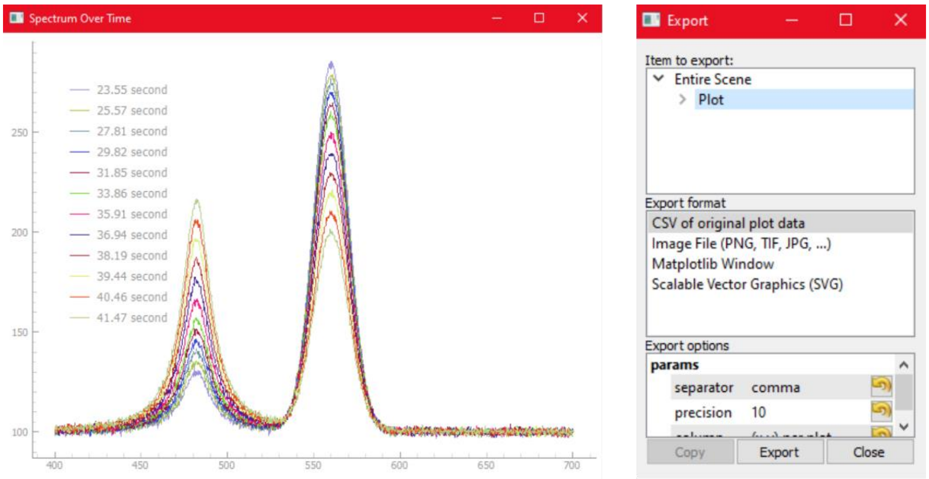


acquire interval (s) : 30,00

0

Figure 25 Spectrum configuration panel to configure the spectrum plotting panel.

Setting panel for spectrum display is shown in Figure 25. The **Fit Interest X** feature adjusts the spectrum view along the x-axis to focus on a specific fitting, while **Fit Interest Y** scales the view along the y-axis for the same purpose. The **Plotting Over Time** feature allows spectra to be continuously plotted in sequence, providing a visual representation of spectral changes over time (Figure 26-left). The **Acquire Interval** setting defines the acquisition interval in seconds, and the number displayed beside the progress bar indicates how many spectra have been collected. The process can be paused with the **Resume** button or restarted as needed.



separator comma

precision 10

Figure 26 Spectrum plotting overlay acquired intermittently (left panel) and export option (right panel).

The plotting results can be exported into several formats, such as CSV data, matplotlib, or SVG images (Figure 26-right).

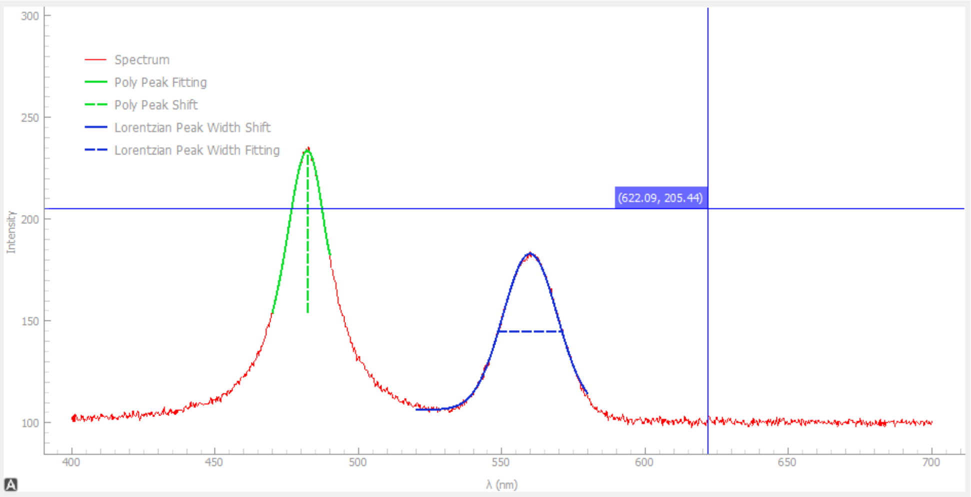


Figure 27 Display panel for acquired or processed spectrum.

The spectrum panel provides several interactive tools to help users analyze and adjust the display (Figure 27). A tooltip is available to identify spectrum coordinates. Some commands to modify the scale for view purpose are:

- Right-click and drag right → the X-axis scale gets smaller
- Right-click and drag left → the X-axis scale gets larger
- Right-click and drag down → the Y-axis scale gets larger
- Right-click and drag up → the Y-axis scale gets smaller

To shift the view, left-click and pull in the desired direction. Additionally, the button at the bottom-left corner, labeled "a," allows the user to fit the view to both the x-axis and y-axis automatically.

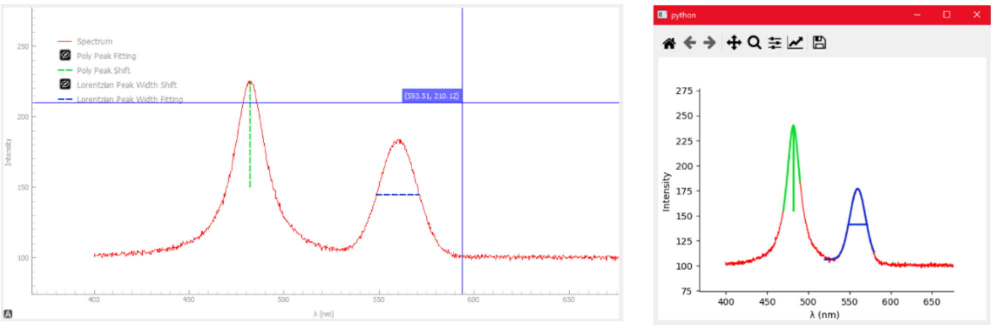
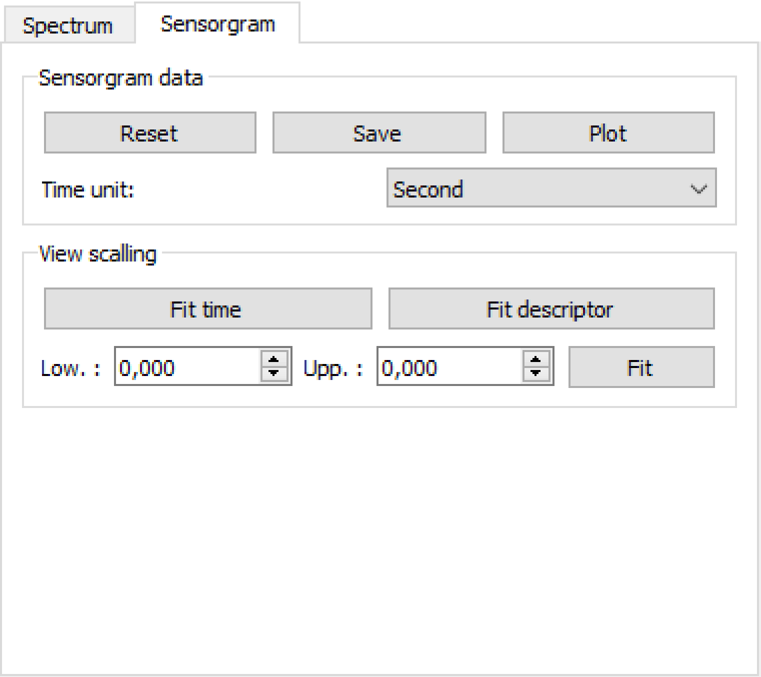


Figure 28 Left panel: demonstration of graph selection to display by left-clicking the respective legend. Right panel: Example of the graph export as a vector graphic.

Each plot in the spectrum panel can be hidden to simplify the view during analysis by clicking on its legend (Figure 28-left). Legends can also be repositioned as needed. In addition, the spectrum panel supports exporting data in multiple formats, including **CSV**, **Matplotlib**, and **SVG** (Figure 28-right).

b) Sensorgram Display Panel



Time unit: Second

Low. : 0,000 Upp. : 0,000

Figure 29 Sensorgram configuration panel to configure the sensorgram panel.

Figure 29 shows sensorgram display setting panel. To save a sensorgram, click the **Save** button, and it will be stored as a CSV file. This data can be replotted later using the sensorgram plot feature available in the menu bar. To clear the sensorgram, press the **Reset** button. To view individual sensorgrams, click the **Plot** button.

The **Fit Time** option adjusts the view to fit along the x-axis, while **Fit Descriptor** adjusts the view along the y-axis. Since user often need to compare shift values between two tracings. PySpecTrace provides a **scaling** feature that allows descriptors to be normalized to the same scale: Low and Upp are buttons used to apply the same scale to every descriptor.For example, suppose you have two descriptors with values 3 (descriptor 1) and 105 (descriptor 2). If you set Low = 5 and Upp = 8, then:

- For descriptor 1: the X-axis range is from (3 – 5) to (3 + 8).
- For descriptor 2: the X-axis range is from (105 – 5) to (105 + 8).

So now, all descriptors are shown at the same scale (1:1). This helps users compare differences between descriptors more quickly.

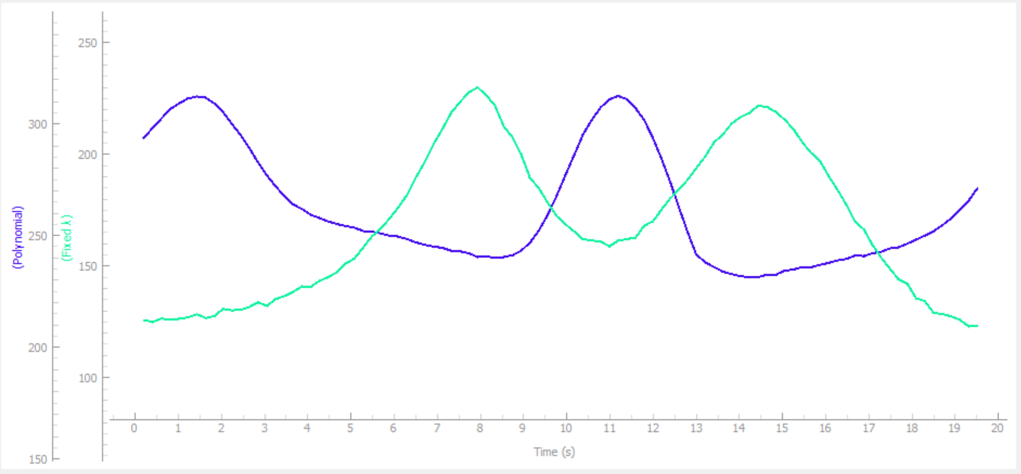
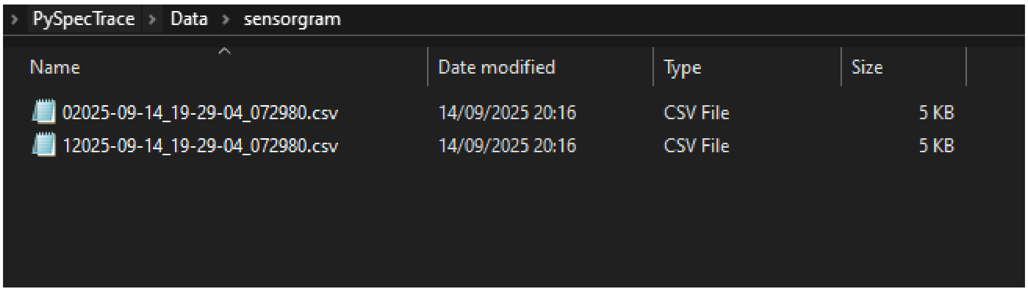


Figure 30 Display of sensorgram plotting panel.

The sensorgram panel shows descriptors from spectrum tracing (Figure 30). Each descriptor has its own y-axis scale, which users can adjust for easier viewing. To change the scale of a specific descriptor, move the cursor to the corresponding y-axis, right-click, and drag downward to increase the scale or upward to decrease it. To adjust the time scale, click on the plot and left-drag to the left to zoom in or to the right to zoom out.



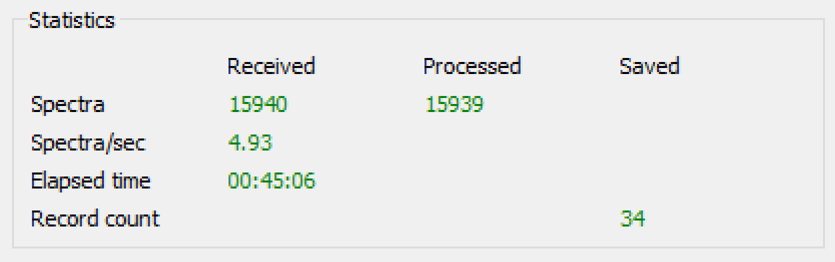
PySpecTrace › Data › sensorgram

Name	Date modified	Type	Size
02025-09-14_19-29-04_072980.csv	14/09/2025 20:16	CSV File	5 KB
12025-09-14_19-29-04_072980.csv	14/09/2025 20:16	CSV File	5 KB

Figure 31 Example of sensorgram output files.

To save a sensorgram, click the **Save** button in the sensorgram configuration panel. This will generate a CSV file for each descriptor, containing the time values, descriptor values, labels, and colors (Figure 31). These CSV descriptors can later be re-plotted using the plotting features available in the menu bar.

8. Statistics information



Statistics

	Received	Processed	Saved
Spectra	15940	15939	
Spectra/sec	4.93		
Elapsed time	00:45:06		
Record count			34

Figure 32 Statistics panel.

At the lower-left corner of the GUI section, the Statistics Panel shows performance data for PySpecTrace, including the number of spectra received, spectra per second (spectra/sec), elapsed time, and the number of files saved (Figure 32). The saved file count increases by two for each acquisition since both the raw spectrum and processed spectrum are stored. The spectra/sec value can indicate the computer's processing speed. If the $\frac{1}{refresh\ rate(second)}$ is greater than spectra/sec, the computer may not be able to process spectra quickly enough.

9. Extra Features in the Menu Bar

a) .txt to .npz File Converter

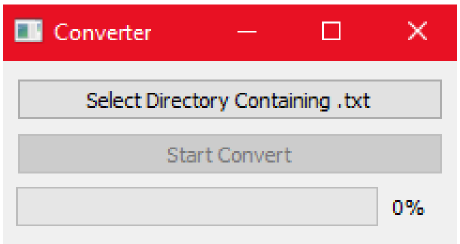


Figure 33 Converter feature to convert .txt into .npz data.

The converter is an essential tool for converting previously recorded spectra into **.npz** files, allowing them to be replayed or post-processed (Figure 33). The first step is to select the directory containing the `.txt` files (the recorded spectra), regardless of whether processed or raw. The converter then generates a new `.npz` file that can be uploaded as a spectrum source in the spectrum panel (Figure 3). This tool can also convert recordings from the **SpectraSuite** (OceanOptics) application, enabling them to be used in PySpecTrace.

b) Sensorgram Plotter

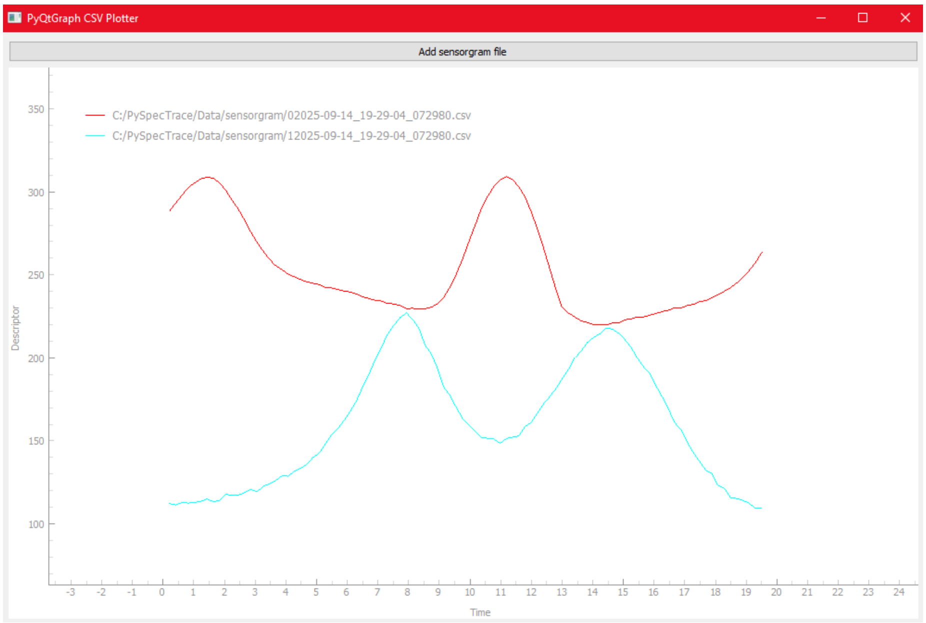

PyQtGraph CSV Plotter
Add sensorgram file
C:/PySpecTrace/Data/sensorgram/02025-09-14_19-29-04_072980.csv
C:/PySpecTrace/Data/sensorgram/12025-09-14_19-29-04_072980.csv
350
300
250
200
150
100
Descriptor
-3
-2
-1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
Time

Figure 34 Offline plotting tool for sensorgram.

The **Sensorgram Plotter** is an additional feature that allows plotting of CSV data generated by the sensorgram (Figure 34). By clicking **Add Sensorgram File**, the sensorgram is added to the panel, where it can also be exported in various file formats.