

# FCS Analysis

This document provides a detailed guide with step-by-step instructions for performing FCS analysis using a Python-based GUI.

Open **Spyder** in the lab computer and then open python script '**FCS\_data\_analysis.py**' that is in drive D (highlighted red in Image 1). Now, run the python script by clicking the highlighted yellow button shown in **Image 1**.

\*The python script can be downloaded from

GitHub: [https://github.com/ShivaniSemwal/FCS\\_analysis/blob/main/FCS\\_data\\_analysis.py](https://github.com/ShivaniSemwal/FCS_analysis/blob/main/FCS_data_analysis.py) and modify it according to your requirements.

**\*\*Important:** Python IDE used is Spyder and the script is compatible with Python version 3.11. Tkinter is used for developing GUI, make sure tkinter is installed properly on your computer/laptop.

Go to Tools → Preferences → IPython console → Graphics, change **Backend to Qt5** (image below) and apply changes. This will plot graphs in a separate window and not in the terminal.

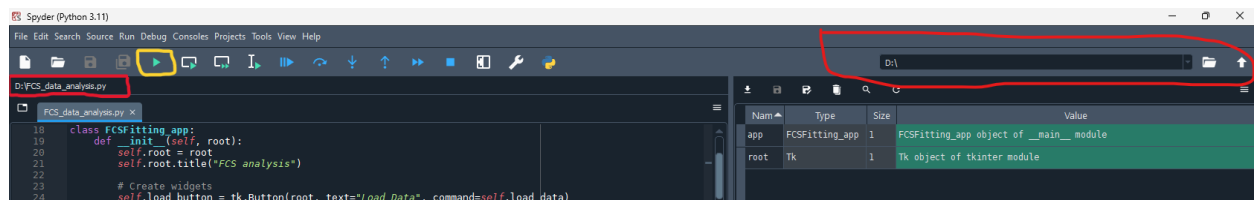
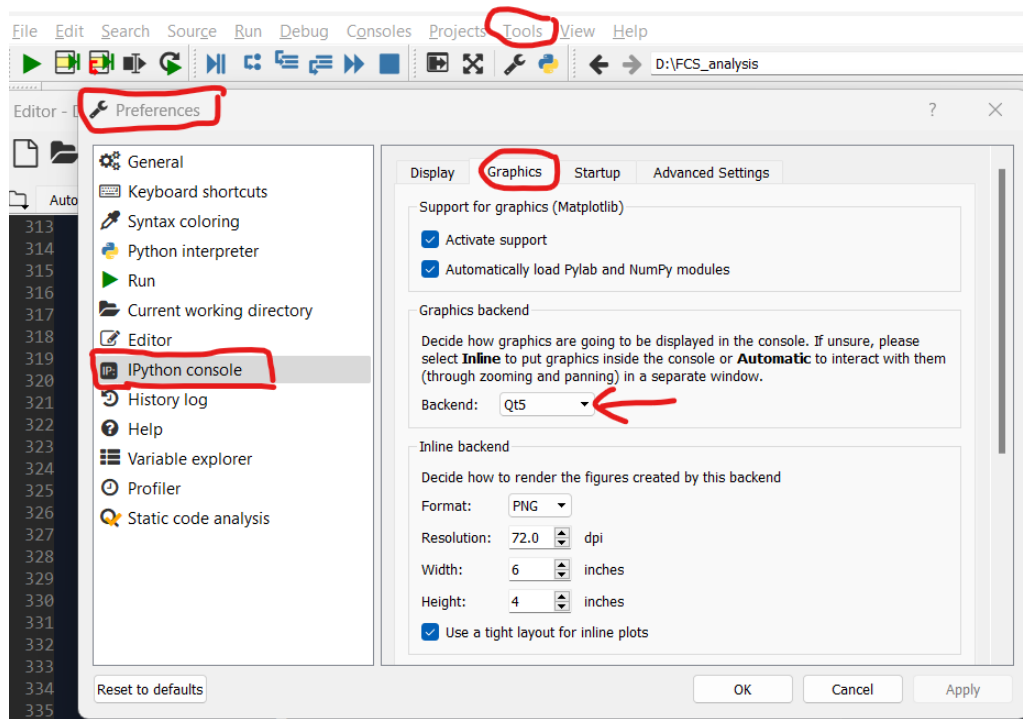
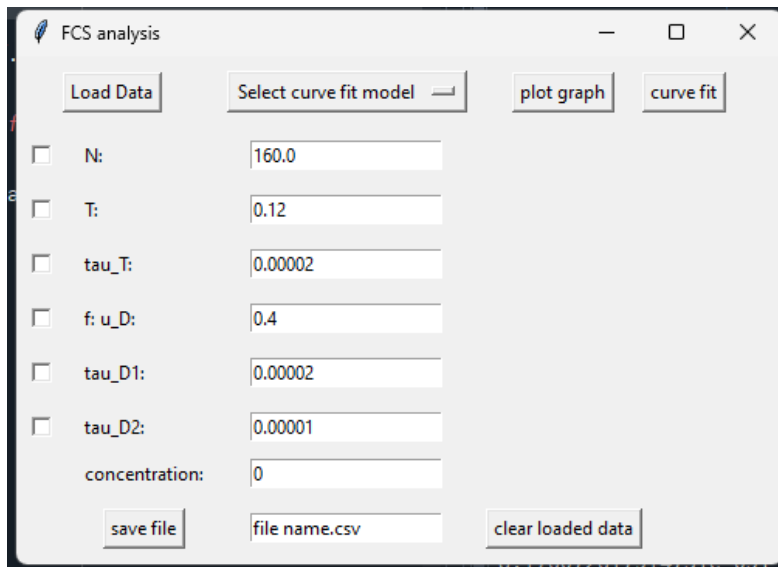


Image 1

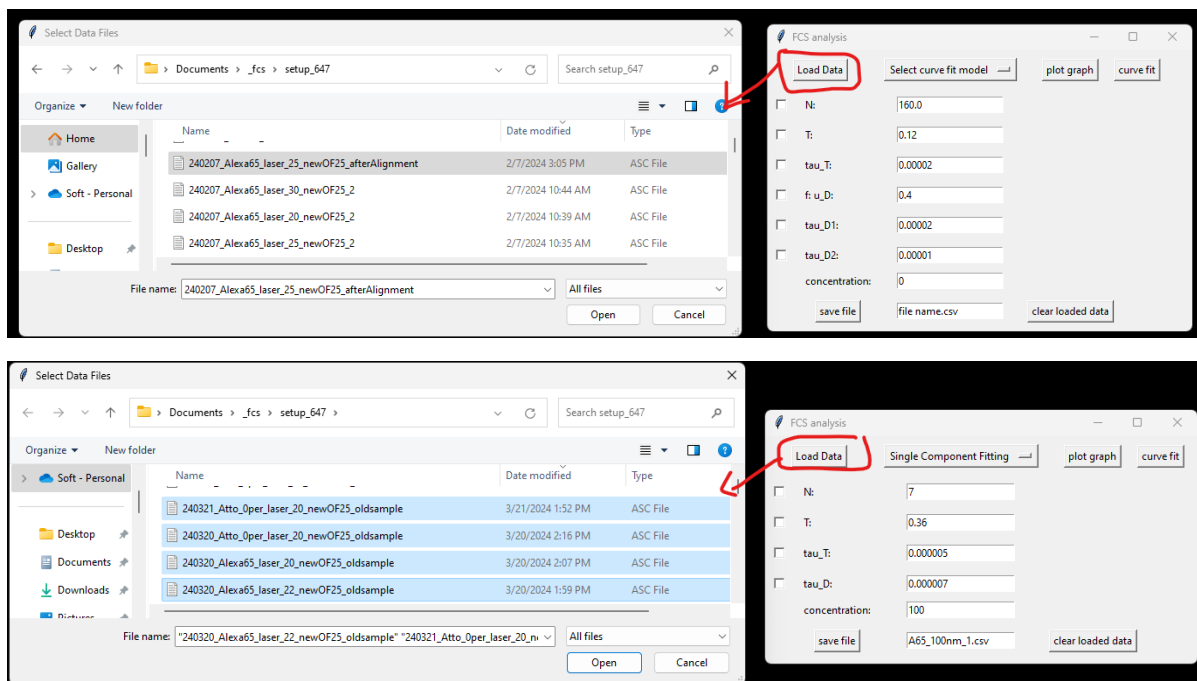
After running the script, you will see a GUI shown in **Image 2**.



**Image 2**

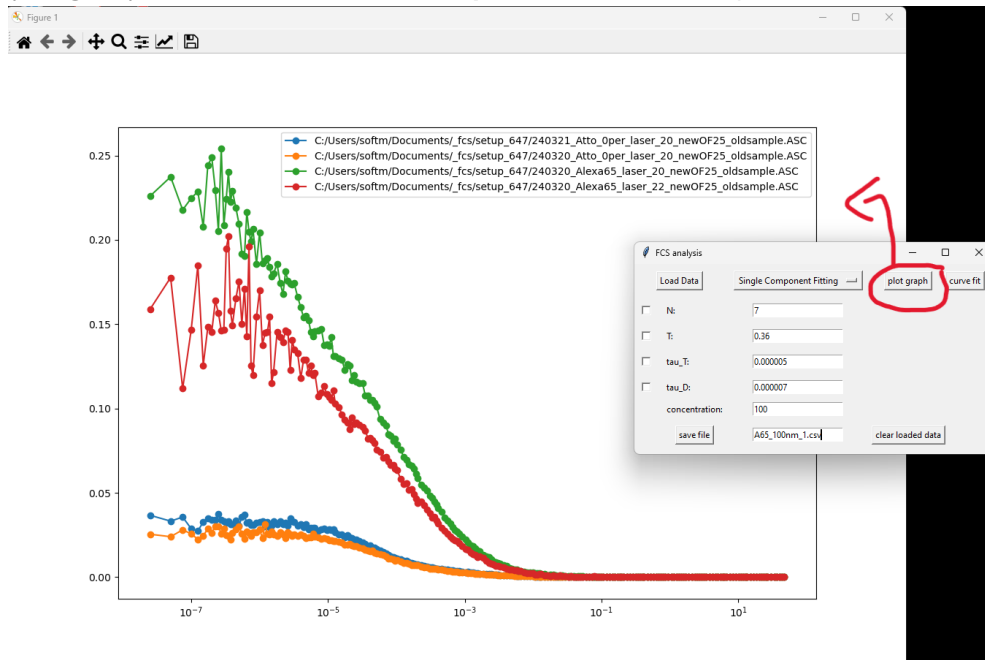
Each feature of the GUI is discussed below:

1. **Load Data:** This feature loads the file to be analysed. It can load single as well as multiple files together (**Image 3**).



**Image 3**

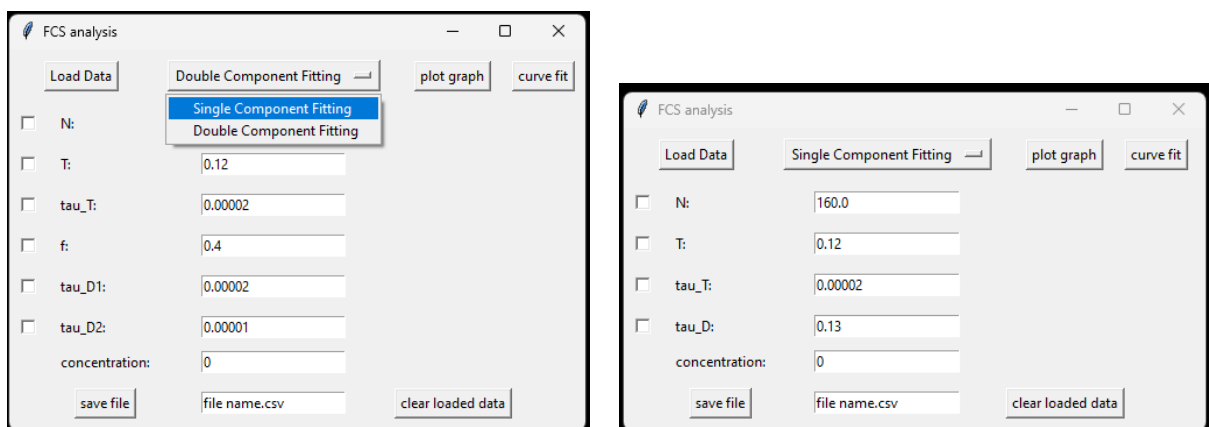
2. **Plot Graph:** If you just want to visualize the data first (after experiments), first load data (**Image 3**) and then click **Plot Graph**. You will see a plot of selected data (**Image 4**)



**Image 4**

**Note:** To plot different data, close the plot and **load data** again >> **plot graph**. All the stored variables or files are cleared after graphs are plotted.

3. **Select Model for curve fitting:** Load the data first, and then select the appropriate fitting model as shown in **Image 5 (left)**. Default initial guess for fitting parameters is already set that can be changed depending on your experimental data (**Image 6**).



**Image 5**

### Single Component Fitting:

$$\left(1 + \left(\frac{T}{1-T}\right)\right) e^{-\frac{\tau}{\tau_T}} * \frac{1}{N} \left[ \left(\frac{1}{1 + \frac{\tau}{\tau_D}}\right) \left(\sqrt{\frac{1}{1 + \frac{\tau}{R^2 \tau_D}}}\right) \right]$$

### Double Component Fitting:

$$\left(1 + \left(\frac{T}{1-T}\right)\right) e^{-\frac{\tau}{\tau_T}} * \frac{1}{N} \left[ f * \left(\frac{1}{1 + \frac{\tau}{\tau_{D1}}}\right) \left(\sqrt{\frac{1}{1 + \frac{\tau}{R^2 \tau_{D1}}}}\right) + (1-f) * \left(\frac{1}{1 + \frac{\tau}{\tau_{D2}}}\right) \left(\sqrt{\frac{1}{1 + \frac{\tau}{R^2 \tau_{D2}}}}\right) \right]$$

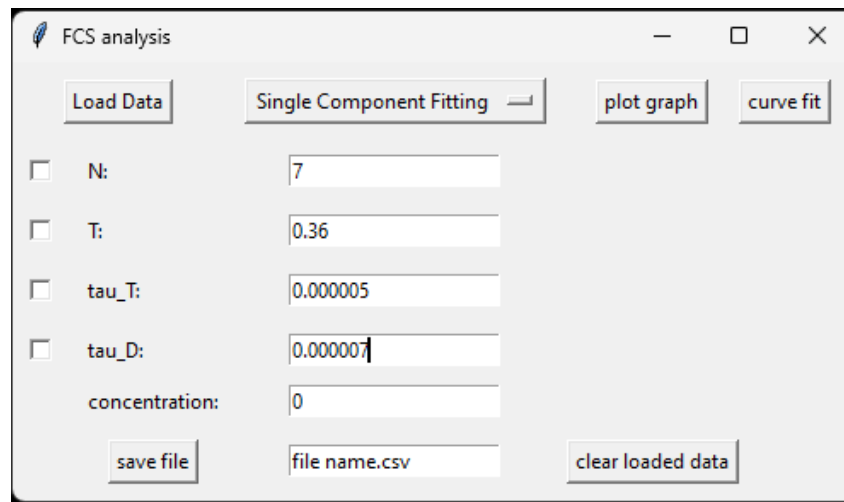


Image 6

4. **Curve Fit:** After loading data, selecting curve-fit model, and setting initial guesses for curve fit parameters, click **curve fit (Image 7)** to see how good the curve fitting looks. If you are happy with the curve fit and fitting parameters, save the graph (**Image 8, Left**) and save results (**Image 10**). Fitting parameters can be viewed at python terminal (**highlighted in Red, image below**).

```

Console I/A x
index: 2
index: 3

In [12]: runfile('D:/FCS_data_analysis.py', wdir='D:')
index: 0
initial_params (10.0, 0.36, 5e-06, 7e-05)
best val, error [7.31515468e+00 3.62080671e-01 4.96342512e-06 7.04555303e-05]
[0.1960255628020183, 0.016543011236634382, 4.327132674511454e-07, 4.4547921973714905e-06]
initial_params [10.0, 0.36, 5e-06, 7e-05]
fitted_params: 7.315154675626391 0.3620806705069867 4.963425123866883e-06
7.045553032636322e-05

In [13]:

```

If fitting is not good, set different initial guesses and click **curve fit**. Keep repeating it till you get the satisfactory results. Once you get the good results, close the graph, click **clear loaded data** (bottom right of the GUI) and re-load the same data again and then **curve fit**. Now, you can save the plot and fitting parameters.

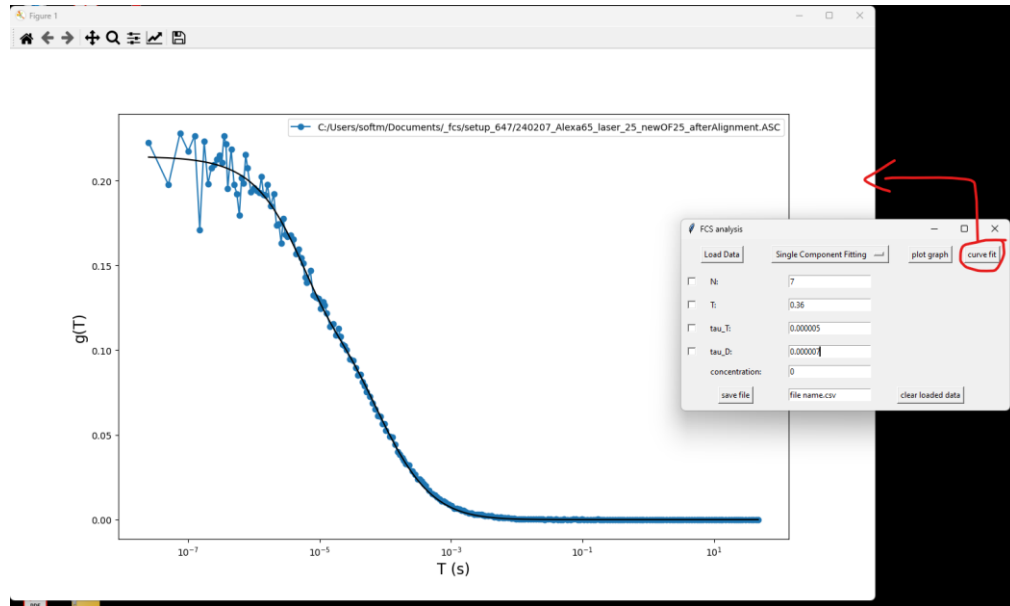


Image 7

**Note:** Always close the plot and click **clear loaded data** before loading a different or new dataset for the analysis.

- Save and Close the plot:** Plots can be saved directly from the toolbar as shown in **Image 8 (Left)** and closed by clicking "X" at the top right of the figure **Image 8(Right)**.

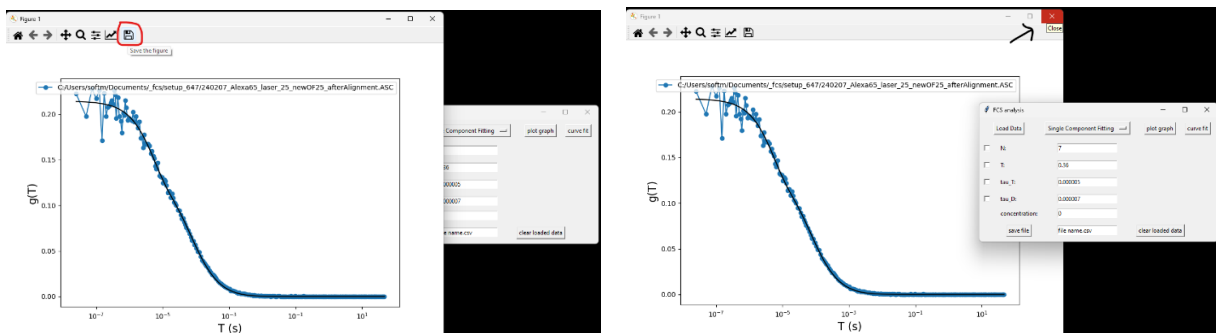


Image 8

- Save Results to one file:** If you are saving results from single dataset, enter the concentration of your sample. As an example, I have entered 100 because concentration of my sample is 100nM. Let say, you want to save results from 4 datasets having same concentration (100nM), then either enter 100,100,100,100 or 1,2,3,4 in the editable **concentration** field of the GUI (**Image 9**).

FCS analysis

Load Data    Single Component Fitting    plot graph    curve fit

☐ N: 7

☐ T: 0.36

☐ tau\_T: 0.000005

☐ tau\_D: 0.000007

concentration: 100

save file    file name.csv    clear loaded data

Image 9

Now, enter the appropriate filename in the **save file** editable field for saving the results and then click **save file** (Image 10, Top). File will be saved in a .csv format in the selected directory (highlighted red, Image 1) and saved file can be viewed in excel or python. It saves all the fitting parameters including concentration, total countrate from both the channels and uncertainties (Image 10, Bottom).

FCS analysis

Load Data    Single Component Fitting    plot graph    curve fit

☐ N: 7

☐ T: 0.36

☐ tau\_T: 0.000005

☐ tau\_D: 0.000007

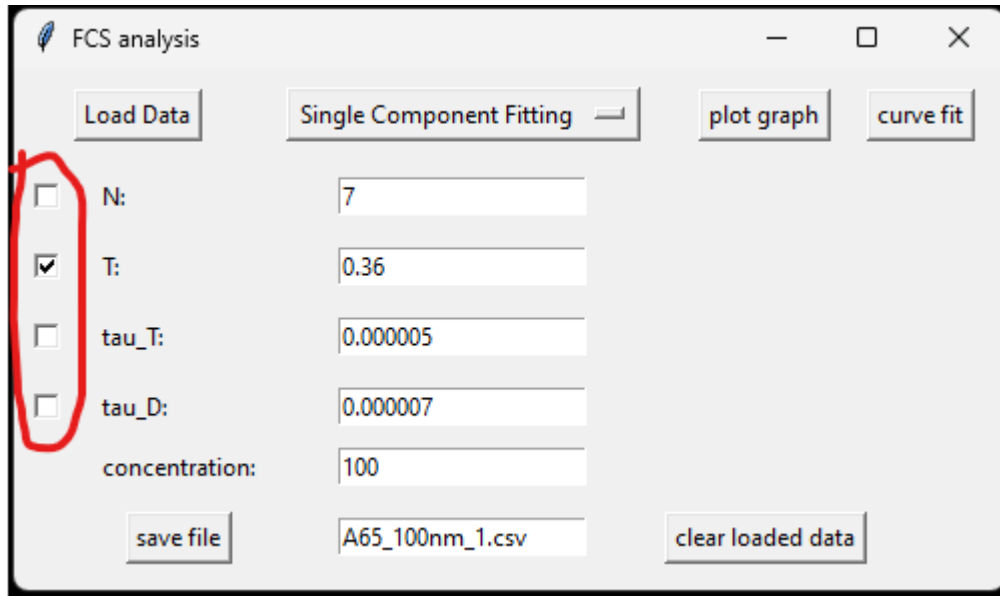
concentration: 100

**save file**    A65\_100nm\_1.csv    clear loaded data

conc(nM)	Total_countrate (KHz)	N	T	tau_T	tau_D	errN	errT	errtau_T	errtau_D
100	62.99	7.3	0.36	4.96e-06	7.05e-05	0.2	0.02	4.33e-07	4.45e-06

Image 10

7. **Hold values:** If you want to fix the value of a particular fitting parameter, checkmark the square boxes (highlighted red, **Image 11**). In the image below, T is held to a value 0.36. Multiple parameters can be held to a value by checking the square boxes next to the parameters.



**Image 11**

**Note:** To edit **Start** and **End** points of the datasets, make changes in the lines# shown below,

**For .ASC files:** Line#556, Start index#0 and End is 500.

```
556 self.start_end_point.append([0,500]) # set start and end point of the dataset to be analysed;
```

**For .SIN files:** Line#585, Start index#85 and End is 450. Modify it according to your data.

```
585 self.start_end_point.append([85,450]) # set start and end point of the dataset to be analysed;
```