**Introduction to UV-vis spectroscopy and Fitting Data in Python**

Background

Watch introductory video on UV-vis spectroscopy here:

<https://www.youtube.com/watch?v=zuUvQN8KXOk>

Overview

In this lab, you will be experimentally determining the molar extinction coefficient of Atto550, a fluorescent molecule often used for labeling biological samples. You will be measuring the absorbances of different dilutions of Atto 550 using UV-vis spectroscopy utilizing plotting and fitting functions in Python to analyze the data.

Data Collection

Materials:

- Pipettes

- (1) 5-uM aliquot of Atto 550

- (5) 0.6-mL microtubes

Instructions:

1. Using the 5-uM aliquot of Atto 550, prepare 200uL dilutions of 100 nM, 250 nM, 500 nM, 750 nM, and 1 uM in DI-water. Set aside for UV-vis measurements.

To do this, you’ll be using the formula C1V1 = C2V2 where C1 is the original concentration of your Atto 550 aliquot, V1 is the volume of the aliquot that you will be adding to water, C2 is your desired final concentration, and V2 is your desired final volume.

1. What is C1 for each dilution calculation?
2. What is V2 for all dilution calculations?
3. Complete the following table before preparing your solutions.

*\*Note:*

*u = micro = x10-6*

*n = nano = x10-9*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sample concentration**  (C2) | **Volume of 5-uM Atto 550 Added**  (V1) | **Volume of Water Added** | **Total Volume**  (V2) |
| **100 nM** |  |  |  |
| **250 nM** |  |  |  |
| **500 nM** |  |  |  |
| **750 nM** |  |  |  |
| **1 uM** |  |  |  |

1. Add 70 uL of each sample into individual cuvettes\*. Load cuvettes into the UV vis machine and begin measurement.

*\*Note: one of the cuvettes has a larger window than the other four – add 140 uL of your sample to that cuvette.*

1. Once the measurements are complete, save your spectrograph as a CSV file. It will save to the downloads folder of the UV-vis computer. Move the file to a USB drive to analyze in Python on your computer

Data Analysis

1. Open Python IDE (I recommend downloading Thonny linked here: <https://thonny.org>)
2. In the coding window, start with the following lines of code to import relevant libraries:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import curve\_fit

1. Read the dat file to a data frame in python using the following lines of code:

#determine the file path to the .dat file that contains the data from your spectrograph

fp = r’/Your/file/path/here/**name\_of\_your\_file1.dat**’

#read the contents of the file to a data frame that python can work with

df1 = pd.read\_csv(fp, delimiter="\t", skiprows=1, encoding='unicode\_escape')

1. Convert the data frame to a numerical array using the following lines of code:

**Var1** = df1.to\_numpy()

#Converting the data frame to numerical array makes it easier to manipulate the data to create graphs and makes it easier to fit the data.

1. Open your .dat file in a separate window. You’ll notice that for each concentration of Atto550, there will be a column of wavelengths in units of nm and a column of Absorbances. The variable, **Var1** (or whatever name you gave the array) has all of this information stored, but we’re going to split the individual columns of data into new variables to make graphing the data easier to understand.

The columns of wavelengths are all the same so we only need to pull the first column of data. These values will be plotted on the x axis of our plot. To extract just the wavelengths from the array that is storing all of the data use the following line of code in your python script:

x = Var1[:,0]

#The 0 index means that we want just the data in the 1st column. The “:” means that we want all of the rows in that 1st column. Now the variable x is storing only the 1st column of wavelengths.

1. The absorbances depend on which concentration of our sample is plotted, so we are going to store the absorbance information from each sample separately. Identify which column each absorbance information is in and define them as separate variables. You will need to modify the following code:

Y100nM = Var1[:,\_]

Y250nM = Var1[:,\_]

Y500nM = Var1[:,\_]

Y750nM = Var1[:,\_]

Y1uM = Var1[:,\_]

#Remember that the 1st column has an index of 0, so the following columns are 1, 2, 3, etc. Place the correct column index in the code above to store the absorbances for the appropriate concentrations.

1. Ask Cam for help using the pyplot library to visualize the data. Plot all 5 curves on the same plot and choose a different color for each curve. Include axis labels and a legend. You can get as creative as you like. Screenshot your figure to present in group meeting.
2. Now we’re going to use the data to determine the extinction coefficient of Atto 550. Atto 550 absorbs light most efficiently near a wavelength of 550 nM. Determine the highest absorbance for each sample concentration and complete the table below:

|  |  |
| --- | --- |
| Concentration (m) | Max Absorbance |
| 100 x 10-9 | 0.0254 |
| 250 x 10-9 | 0.0434 |
| 500 x 10-9 | 0.0734 |
| 750 x 10-9 | 0.1034 |
| 1 x 10-6 | 0.1334 |

1. Create a new .py file and repeat Step 2) of the instructions.
2. Store the concentrations and max absorbances in two different array variables using the following code:

Conc = (100\*10\*\*-9, 250\*10\*\*-9, 500\*10\*\*-9, 750\*10\*\*-9, 1\*10\*\*-6)

Abs = ( \_\_\_\_, \_\_\_\_\_, \_\_\_\_, \_\_\_\_, \_\_\_\_ )

1. The concentration, absorbance, and extinction coefficient are related by the equation:

A = εbC

where A is the absorbance, ε is the extinction coefficient, b is the path length, and C is the concentration. The path length (the distance traveled by the light as it passes through the sample) is equal to the side length of the cuvette which is 1cm, therefore we can simplify the relationship between variables as:

A=εC

To plot absorbance vs concentration, we’ll use the plotting function in the matplotlib library. You can use the following code:

fig, axs = plt.subplots()

s=1

c=’blue’

m=’o’

fs1=12

fs2=16

axs.scatter(Conc, Abs, s=s, c=c, marker=m)

axs.set\_xlabel("Concentration (M)", fontsize= fs1)

axs.set\_ylabel("Absorbance (a.u.)", fontsize= fs1)

axs.set\_title(‘\_\_\_\_\_\_\_\_\_', fontsize=fs2)

plt.show()

In this code the “s” variable controls the size of the markers, the “c” variable controls the color, “m” is the type of marker, and the “fs1” and “fs2” variables control the sizes of your axes and title labels respectively. You can modify these values as you see fit.

1. Now that we have our data plotted, we want to use a fit of the data to experimentally determine the extinction coefficient of Atto550. The extinction coefficient, ε, is a measure of how strongly a molecule absorbs light and is a property of the molecule itself. There are two fit parameters that can be determined from fitting data to a linear equation – slope and y intercept. Which of these fit parameters for our data represents the extinction coefficient (circle)?

slope y-intercept

1. To fit the data to a linear function we need to supply Python with the general form of the function that we want to fit.

def linfunc(x, m, b):

y = mx + b

return y

Here we’ve defined a function called “linfunc” that accepts values of “x” as input, fit parameters m and b, and returns an output, y.

Next, rather than calculate output values for our function using a known slope and y intercept, we want to fit our absorbance vs concentration data to this function and determine the slope and y intercept. To do this we’ll use the following code:

params, \_ = curve\_fit(linfunc, Conc, Abs)

print(params)

The print function will output the slope and y intercept that best fits the concentration and absorption data. Based on these parameters what is the extinction coefficient of Atto550?

\_\_\_\_\_\_\_\_\_\_\_\_\_ M-1cm-1

1. Lastly let’s add our fit to our original scatter plot. Insert the bolded code into the code used to create the scatter plot:

fig, axs = plt.subplots()

s=1

c=’blue’

**c2=’black’**

m=’o’

fs1=12

fs2=16

**lw = 1**

axs.scatter(Conc, Abs, s=s, c=c, marker=m)

**axs.plot(Conc, linfunc(Conc, \*params), c=c2, linewidth=lw, label = ‘y=mx+b’)**

axs.set\_xlabel("Concentration (M)", fontsize= fs1)

axs.set\_ylabel("Absorbance (a.u.)", fontsize= fs1)

axs.set\_title(‘\_\_\_\_\_\_\_\_\_', fontsize=fs2)

plt.show()

In the code for the plot, replace m and b in the label description with the actual values determined in the fit for m and b. Make sure that the code for the figure is written at the end of the script to ensure that all of the relavant functions and variables are defined prior to plotting the figure.

1. Lastly, look up the reported value of the extinction coefficient of Atto550. What is the percent difference before the value determined and the literature value. You can use the following equation:

|(Your value – Literature value)/(Literature value)| \* 100% =\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Comment on some of the sources of variation between what the extinction coefficient is reported to be versus what you determined the extinction coefficient to be from your data?