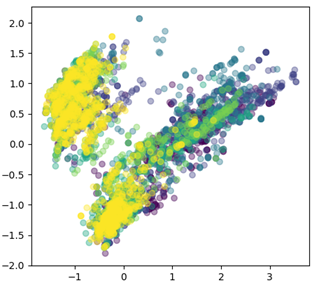
**Experiment 4: Chemoinformatics and Python**

Chemoinformatics is the use of computational techniques to solve problems in chemistry. These *in silico* methods can be used to transform data into information and aid in the process of drug discovery. Recently, a rise in computational power and increased availability of developed tools have turned chemoinformatics into an invaluable tool for research.



The purpose of this experiment is to introduce you to chemoinformatics using Python. By the end of this module you will have the ability to use Python to quickly compile and plot data.

During this lab, you will:

• Learn about data structures in Python

• Load compiled data that is suitable for sharing and later use

• Effectively parse through compiled data

• Perform mathematical operation on compiled data

• Plot data in a multitude of ways

**Google Colab**

This tutorial will be complemented by Google Colaboratory (Colab), which is an online python environment with preinstalled packages and executes code on the cloud provided for free by Google. Its primary advantages here are allowing a fast and simple way for new students to get started coding instantly, as it is agnostic of computer and operating system and requires no setup. In the first requirement of this tutorial, you will have to download ‘lab4\_notebook.ipynb’ from Canvas and upload it to your Google Drive. Open your file in the ‘colab app’. As explained in the video, code is separated into blocks called ‘cells’, which execute independently of each other. Hit control or command-enter to execute the code in a cell. In the following sections, you will be executing code cell-by-cell and editing code as outlined in the colab file. Also, ensure that you have downloaded the alldrugprops.json file from canvas.

**Notes:**

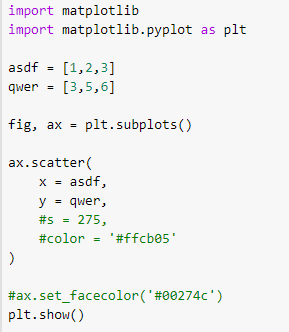
Before starting this module ensure that:

1. You have downloaded lab4\_notebook.ipynb and added it to your drive.
2. You have downloaded ‘alldrugprops.json’ from Canvas.

**Protocol:**

**Part 1: Simple plotting using MatplotLib**

**1A)** Type in the text below, then Run (ctrl+enter) (CMND+Return on Mac). Alternatively, you can press the play button. If you get a syntax error, you likely have a missing comma or bracket somewhere – proofread your code to identify any typos. In this script, we have defined two lists, their names are “asdf” and “qwer”.



**1B)** The plt.show() command from the matplotlib package opens your plot in a new window. Later we will save our plots as a .png, but for now we will see how things change in the plt.show() window. **Take a screen shot of your plot for your lab report.**

**1C)** Delete the first hashtag to activate the line: s = 275. Rerun your code (CTRL+S then ctrl+enter). What happened? Play with different values for s (size) until you are happy with the result. **Take a screen shot of your plot for your lab report.**

**1D)** Remove the next hashtag. This should change the color. #ffcb05 is a hex code, which is one of several ways to describe a color. **Take a screen shot of your plot for your lab report.**

**1E)** Remove the next two hashtags. Run the code. **Take a screen shot of your plot for your lab report.**

**1F)** Add in a fourth point at x = 1.5, y = 4. **Take a screen shot of your plot for your lab report.**

**Part 2. Plotting from a data file.**

-In Part 1, we worked from two lists named “asdf” and “qwer”. In reality, you would more often read in lists from a larger data set, which we will do here. The file alldrugsprops.json contains information for all of the 9,279 drugs in the DrugBank database, with appended chemoinformatics data which was calculated in RDKit (a free python package specifically for chemoinformatics) or Pipeline Pilot (a powerful chemoinformatics software which is not free). [TIP: The DrugBank database is an important source of free data for chemoinformatics. CHEMBL is another important free database for drug discovery which includes thousands of chemical structures (as their SMILES strings) and associated biochemical, *in vivo*, and pharmacokinetic data collected from patents and the scientific literature.]

**2A)** To start, you must have downloaded the alldrugprops.json file. To allow colab to read the file, open the file directory by clicking on the folder icon to the left. Drag the alldrugprops.json file into the opened side bar to upload the file.

Graphical user interface, application

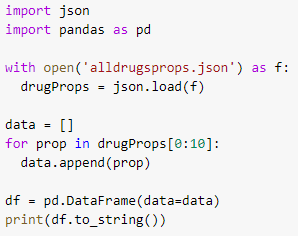
Description automatically generated

If it worked, you should see the file in the sidebar:

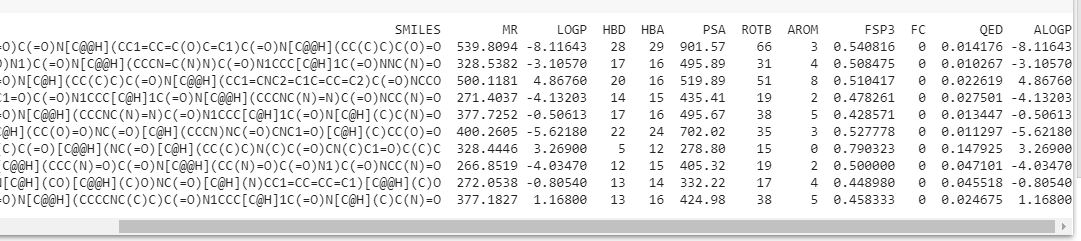
Graphical user interface, text, application

Description automatically generated

**2C)** Run the next block of code to get a sense of what the data looks like



(scroll the output all the way to the right)



**2D)** Now that you have a sense of what the data structure in the .json file looks like, let’s make a plot. Copy the following code in colab and run it to produce the plot. Recall that hashtag inactivates a line of code. The text with hashtags below is just instruction and does not need to be entered. If you get a syntax error, check for missing parentheses or commas or other typos. **Take a screenshot of your plot for your lab report.**



**Part 3:**

Create the same plot as in Part 2, but color by a different quantity.

Keys you can use from the JSON:

• MR

• LOGP

• HBD

• HBA

• PSA

• ROTB

• AROM

• FSP3

• QED

• ALOGP

Be sure to label/write down the quantity you chose. Use ax.set\_title(“title”) to add a title to your plot.

**Part 4:**

Create the same plot as in Part 3, but use different quantities for the x and y axis. Do not use logp and psa again. Change your x and y labels accordingly.

Use Google to find an answer in the python user community. Search “How do I \_\_\_\_\_\_\_\_\_ in matplotlib”. You can change the shape of the points, add in a plot title, switch the axes to a log scale, turn up the transparency on your plots for example. A big part of using python is learning how to extract information from the very large user community. There are many sophisticated things you can do once you’ve got some comfort in python. You could run a principal component analysis on several of the properties in the .json file. You could write a FOR loop to plot three different plots on the same page with different y-axes. Give your plot a personal touch.

**Take a screenshot of your plot for your lab report**

**For the laboratory report:**

Save all plots you made. Ensure that they are all labelled and titled accordingly. Answer the discussion questions.