. Lab 2 Protocol

**Lab 2: 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.

In this lab we will expand on what we learned in the previous lab. Specifically, we will

* filter out unusable data
* use data visualization to validate medicinal chemistry principles
* perform basic statistical analysis
* Simplify multidimensional data using Principal Component Analysis

**Notes:**

Before starting this module ensure that:

1. Ensure that you have downloaded the activity\_data\_with\_props.csv and lab2\_notebook.ipynb file from canvas.

**Protocol:**

**Module 1: Importing and Filtering Data in Pandas**

This csv is taken from diamond Xchem’s COVID moonshot project (<https://covid.postera.ai/covid/activity_data>) and is inhibition data against the SARS-COV2 main protease. The physical properties were calculated based on the SMILES strings using RDKit (a free python package specifically for chemoinformatics).

**1A)** We will begin by importing the packages that we need as well as uploading the data file into colab, just like in lab 1. Add lab2\_notebook.ipynb to your google drive, open in colab, and add “activity\_data\_with\_props.csv” to the file system as shown below.

Graphical user interface, application

Description automatically generated

**Figure S7.** Another visual example of uploading a given datafile to Colab via dragging and dropping the file onto the file manager.

**1B)** We are opening the activity\_data\_with\_props csv and calling it “data”. We will also use the “len” function to see we have 1699 data points

Graphical user interface, application

Description automatically generated

**Graphical user interface, application

Description automatically generated**

**Figure S8.** Inspection of the data from the provided datafile in tabular format using Pandas. The number of points in the datafile is revealed by using the len() function.

**1C)** You will notice entry 2 does not have an IC50 value. We will want to remove any molecules without inhibition data. Run the next block of code to accomplish that.

**1D)** Now use the len function again to see how many data points are left. **Record this value for your lab report**

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

**-**In Module1,we opened a large set of data and filtered out incomplete entries. Now we are going to perform analysis on our data.

**2A)** To start, we will create shortcuts for each property in this dataset. Run the first block of code. Text

Description automatically generated

**Figure S9.** Provided code showing how columns of data can be extracted from the DataFrame into variables. These variables are referred to later in the code to simplify plotting.

**2B)** Now we are going to validate GSK’s Solubility Forecast Index (Drug Discov Today. 2010 Aug;15(15-16):648-55). Create a scatter plot that shows how solubility is affected by number of aromatic rings and LogP. You will need to type your own code into the empty cell and fill in the appropriate words:

Text, letter

Description automatically generated

**Figure S10.** Empty template code to plot physicochemical properties of the dataset. At this point, students are expected to understand the parameters of the scatter function, as well as other functions provided by the Axes object generated from matplotlib.

If you get a syntax error, you likely have a missing comma or bracket somewhere – proofread your code to identify any typos. **Take a screenshot of this plot for your lab report**

**2C)** Now we are going to do some basic statistics on our data. Box plots are a common way to visualize large sets of data. Run the block of code and discuss with your group what it is doing. **Take a screenshot of the plot for your report.**

**2D)** Instead of a 5x2 grid, make a 10x1 grid of box plots.  **Take a screenshot for your lab report**

**Module 3:**

**3A)** Now we are going to be doing our principal component analysis. To do this we are going to define features and create a list of lists containing values for the features. Run the next block of code.

A picture containing graphical user interface

Description automatically generated

**Figure S11.** Template code provided to the student to execute and visualize a principal component analysis performed on the given dataset. The resultant chemical space is colored by a specific property, revealing a coorelation between the resultant axes of the PCA and drug properties.

**Take a screen shot of the plot for your lab report. Now color by another feature and screenshot. Finally color by a property from module 2 that was not included in the features. Screen shot this for your lab report as well.**

**3B)** Now we are going to filter our molecules according to the Lipinski rules. You will notice that the code that worked for filtering a json does not work for a csv. This is because data in a CSV is structured differently than in a JSON file. To figure out how to address this Google the phrase “filter pandas dataframe by multiple conditions”. Now run and plot a PCA on your Lipinski filtered set. **Take a screenshot of both your code and the plot for your lab report**

**Module 4:**

This block of code will generate an interactive graph that can be used to see values of individual data points. **Copy down the SMILES string and IC50 values of 3 molecules that clustered near each other into your lab report. It may be easier to write these by hand and type them into the word document after.**