**PSPG Cheminformatics Workshop -- 2021**

In this workshop, you will use machine learning tools from scikit-learn (aka sklearn, http://scikit-learn.org) along with chemical fingerprinting and visualization tools from RDKit (http://rdkit.org) to predict the protein targets of cancer-related compounds. This workshop uses two Python scripts that are mostly complete. Your job is to fill in the missing sections and use these scripts to explore the protein (target) profiles of these cancer-related compounds. Your goal is to decide which compound is most applicable to your tumor for treatment.

Work together in groups of 2 or 3 to complete the scripts and write up a short report. You should be able to do this in 2-3 PowerPoint slides.

Explain common substructures or functional groups (“warheads”) that are present in the compound you choose and the most similar ligands for the predicted off-targets of this compound. Do these off-targets explain the compounds efficacy, it’s side effects, or perhaps a repurposing opportunity? Is there an experiment you would recommend to better understand your compound? Explain why or why not. If an off-target, for what indication might it be used? Is this worth pursuing? Why or why not?

As a sanity measure along the way, you can always double-check your predictions against the SEA web tool, at http://sea.bkslab.org/search, which will make similar (but not identical) target predictions for each drug you provide. Simply input the drug SMILES from the **cancer\_compounds.csv** file and the drug name or ID, followed by a space.

**Script 1: "Predict" notebook**

**2017-01-24\_\_PSPG\_Workshop\_Predict\_Script\_Fill\_Jupy-Env.ipynb**

This notebook takes in a file containing compounds (SMILES), along with two reference files containing data on molecules and targets from ChEMBL. It loads the data, converts the SMILES into in-memory representations of molecules using RDKIT, calculates chemical fingerprints from them, and trains a machine-learning method called a Naïve Bayesian Classifier to predict the drug targets of each compound.

http://scikit-learn.org/stable/modules/naive\_bayes.html#multinomial-naive-bayes

Some parts of the script are incomplete, and these are marked with question marks (?). It is your job to fill them in.

**Script 2: "Compare" notebook**

**2017-01-24\_\_PSPG\_Workshop\_Compare\_Script\_Fill\_Jupy-Env.ipynb**

This notebook is used to explore why a prediction from the first script was made. It takes a file of drugs along with a target ID (e.g., “CHEMBL...”), and the two ChEMBL reference files as mentioned before. It uses RDKit to calculate the molecular similarity of all the ligands for the target you specified as compared to the drugs you gave it, and shows the most similar ligands.

http://www.rdkit.org/docs/GettingStartedInPython.html

Do these ligands share common patterns, functional groups, “warheads”, etc with your compound? Which ones?

**Data: cancer\_compounds.csv**

This file can be opened in Excel, or a simple text editor, and contains the SMILES and names or IDs of various cancer-related compounds.

**Data: sample\_cpds\_to\_compare.csv**

This file can be opened in Excel, or a simple text editor, and likewise contains SMILES and IDs of several cancer compounds from the first data file. It demonstrates the file format for to be used for the query drugs as input to the "Compare" notebook.

**Setting up our Working Environment**

Let's create a directory for us to work from. This directory is where we will store all the files we plan to use, and where we will install Miniconda from as well.

First create the directory:

cd ~/Desktop/

mkdir -p 19-02-21\_cheminfo\_workshop/predictions/

cd 19-02-21\_cheminfo\_workshop/

**Download the files from Google Drive.**

We have set up a google drive folder that contains all the files you will need, here:

<https://goo.gl/tYUVDf>

Download the files to the folder you just created.

**Install Miniconda and Additional Packages**

Unzip the file you downloaded. If you do not have Conda installed, find the Miniconda bash file that matches your operating system (Linux and OSX file provided) and follow the site's instructions to install Miniconda.

bash Miniconda2-latest-MacOSX-x86\_64.sh # For Mac

bash Miniconda2-latest-Linux-x86\_64.sh # For Linux

Once Miniconda has been installed, close your terminal window, and reopen another one. Check to see conda was installed correctly. Type the following command to see the list of packages that were installed.

conda list

**Create and Activate a new Conda Environment**

Now that we have the infrastructure in place, we need to create a new conda environment, and install the additional packages, such as rdkit and jupyter, we will need to generate our predictions.

Create a new environment using the following code:

conda create --name workshop\_env python=2.7 jupyter seaborn pandas scikit-learn

source activate workshop\_env

conda install -c rdkit rdkit

**Start a Jupyter Notebook Session**

To test if everything is working, let's open up a jupyter notebook session. We'll import the modules we need to run our scripts predict.py and compare\_cpd\_to\_ligands.py correctly.

jupyter notebook --no-browser --port=8886

Open Chrome or Firefox and type the following into the url address:

localhost:8886

This should bring up a jupyter notebook session that is being hosted from your local machine. Open the scripts provided and in the first cell try importing the modules we'll need to ensure all our packages are installed correctly.