7.4: R Assignment

Virtual Screening

Objectives

- Perform virtual screening against PubChem using ligand-based approach
- Apply filters to prioritize virtual screening hit list.
- Learn how to use pandas' data frame.

In this notebook, we perform virtual screening against PubChem using a set of known ligands for muscle glycogen phosphorylase. Compound filters will be applied to identify drug-like compounds and unique structures in terms of canonical SMILES will be selected to remove redundant structures. For some top-ranked compounds in the list, their binding mode will be predicted using molecular docking (which will be covered in a separate assignment).

1. Read known ligands from a file.

As a starting point, let's download a set of known ligands against muscle glycogen phosphorylase. These data are obtained from the DUD-E (Directory of Useful Decoys, Enhanced) data sets (http://dude.docking.org/), which contain known actives and inactives for 102 protein targets. The DUD-E sets are widely used in benchmarking studies that compare the performance of different virtual screening approaches (https://doi.org/10.1021/jm300687e).

Go to the DUD-E target page (http://dude.docking.org/targets) and find muscle glycogen phosphorylase (Target Name: PYGM, PDB ID: 1c8k) from the target list. Clicking the target name "PYGM" directs you to the page that lists various files (http://dude.docking.org/targets/pygm). Download file "**actives_final.ism**", which contains the SMILES strings of known actives. Rename the file name as "**pygm_1c8k_actives.ism**". [Open the file in WordPad or other text viewer/editor to check the format of this file].

Now read the data from the file using the pandas library (https://pandas.pydata.org/). Please go through some tutorials available at https://pandas.pydata.org/pandas-docs/version/0.15/tutorials.html

```
colnames <- c('smiles','dat', 'id')
df_act <- read.delim("pygm_1c8k_actives.ism", header = FALSE, sep = " ")
colnames(df_act) <- colnames

head(df_act)
```

```
 print(nrow(df_act))    # Show how many structures are in the "data frame"
```

2. Similarity Search against PubChem

Now, let's perform similarity search against PubChem using each known active compound as a query. There are a few things to mention in this step:

- The isomeric SMILES string is available for each query compound. This string will be used to specify the input structure, so HTTP POST should be used. (Please review lecture02-structure-inputs.ipynb)
- During PubChem's similarity search, molecular similarity is evaluated using the **PubChem fingerprints** and **Tanimoto** coefficient. By default, similarity search will return compounds with Tanimoter scores of **0.9 or higher**. While we will use the default threshold in this practice, it is noteworthy that it is adjustable. If you use a higher threshold (e.g., 0.99), you will get a fewer hits, which are too similar to the query compounds. If you use a lower threshold (e.g., 0.88), you will get more hits, but they will include more false positives.
- PubChem's similarity search does **not** return the similarity scores between the query and hit compounds. Only the hit compound list is returned, which makes it difficult to rank the hit compounds for compound selection. To address this issue, for each hit compound, we compute **the number of query compounds that returned that compound as a hit.** [Because we are using multiple query compounds for similarity search, it is possible for different query compounds to return the same compound as a

hit. That is, the hit compound may be similar to multiple query compounds. The underlying assumption is that hit compounds returned multiple times from different queries are more likely to be active than those returned only once from a single query.]

- Add "sys.sleep()" to avoid overloading PubChem servers and getting blocked.

```
smiles_act <- list(df_act$smiles)
```

WIP WIP WIP
