# Exercise: Evaluate TankBind on the KIBA dataset

**Step 1: Pre-process KIBA dataset (get SMILES strings for molecules and Uniprot and PDB ids for proteins)**

Hints:

* DeepPurpose pre-processing pipeline gives you a data frame of SMILES strings, protein sequences and affinity scores (<https://github.com/kexinhuang12345/DeepPurpose>, see “load\_process\_KIBA()”)
* DeepDTA dictionary mapping Uniprot ids to sequences from KIBA dataset (<https://github.com/hkmztrk/DeepDTA/blob/master/data/kiba/proteins.txt>)
* Map Uniprot ids to PDB ids: <https://www.uniprot.org/id-mapping>

**Step 2: Download PDB files (e.g. from PDB database where experimental structures are available, and otherwise from AlphaFold database)**

Hints:

* requests package (for downloading files in python script)
* e.g. to get protein with PDB id 2R5T from PDB: <https://files.rcsb.org/download/2R5T.pdb>
* e.g. to get protein with Uniprot id O00141 from AlphaFold DB: <https://alphafold.ebi.ac.uk/files/AF-O00141-F1-model_v2.pdb>

**Step 3: Run P2Rank to get binding pocket predictions (already part of TankBind pipeline)**

Hints:

* Installation guide: <https://github.com/rdk/p2rank>

**Step 4: Construct dataset**

Hints:

* follow virtual screening example (but adapt for multiple proteins): <https://github.com/luwei0917/TankBind/blob/main/examples/high_throughput_virtual_screening_LRRK2_WDR.ipynb>
* important: need torchdrug=0.1.2 (Python 3.8) to get the right dimension of molecular features for the model 🡪 easiest to create dataset in local conda environment and switch to Colab (per default python 3.10) for applying the model (very compute-intense, needs GPU)
* pre-calculate and save molecular features in local environment instead of calculating for each batch (similar to protein\_dict)

**Step 5: Apply TankBind model**

Hints:

* use self-dock model: (<https://github.com/luwei0917/TankBind/blob/main/saved_models/self_dock.pt>)
* to reduce runtime: take only best ranking pocket per sample or evaluate only part of the dataset (e.g. 10%)
* might have to remove very large proteins/pockets due to memory problems (e.g. if dis\_map.shape[0]>10000)

**Step 6: Evaluation:**

* metrics: concordance index, mean squared error
* maybe normalization necessary
* compare with state-of-the-art affinity prediction model