Constructing a Data Validation Tool for Identifying Structure-Function Matches in Protein and assay Databases

CS520 - Data Integration, Warehousing, and Provenance

Project Overview

To build reliable machine learning models, collecting the structure and function of the underlying protein interactions is essential. Current methods of relating 3D protein structures to corresponding are being completed manually and are a tedious effort for chemists. This project will explore an automated technique to ingest data from two data sources – one containing 3D protein structures and the other containing biochemical assays (laboratory experiments) – and identify the 3D structure for the proteins measured in the assay database.

The results of this project will be used to inform a machine learning research project for Dr. David Minh's lab. Initial validation of the model was completed for the μ-opioid receptor (MOR). MORs have fewer data points in both databases, making them an attractive candidate for an initial target. However, further validation is slated to be completed on the Estrogen Receptor β (ERB)—the query "estrogen receptor" results in 13,767 BioAssays and 405 protein structures. Identifying the pairs between these databases manually is time-consuming, so developing automation will prove to be the most efficient approach to conduct the mapping.

Overcoming challenges including inconsistent naming conventions for ERB between and within the databases and building a generalized query for ERB will be the focal points for the project.

Data Set Introduction

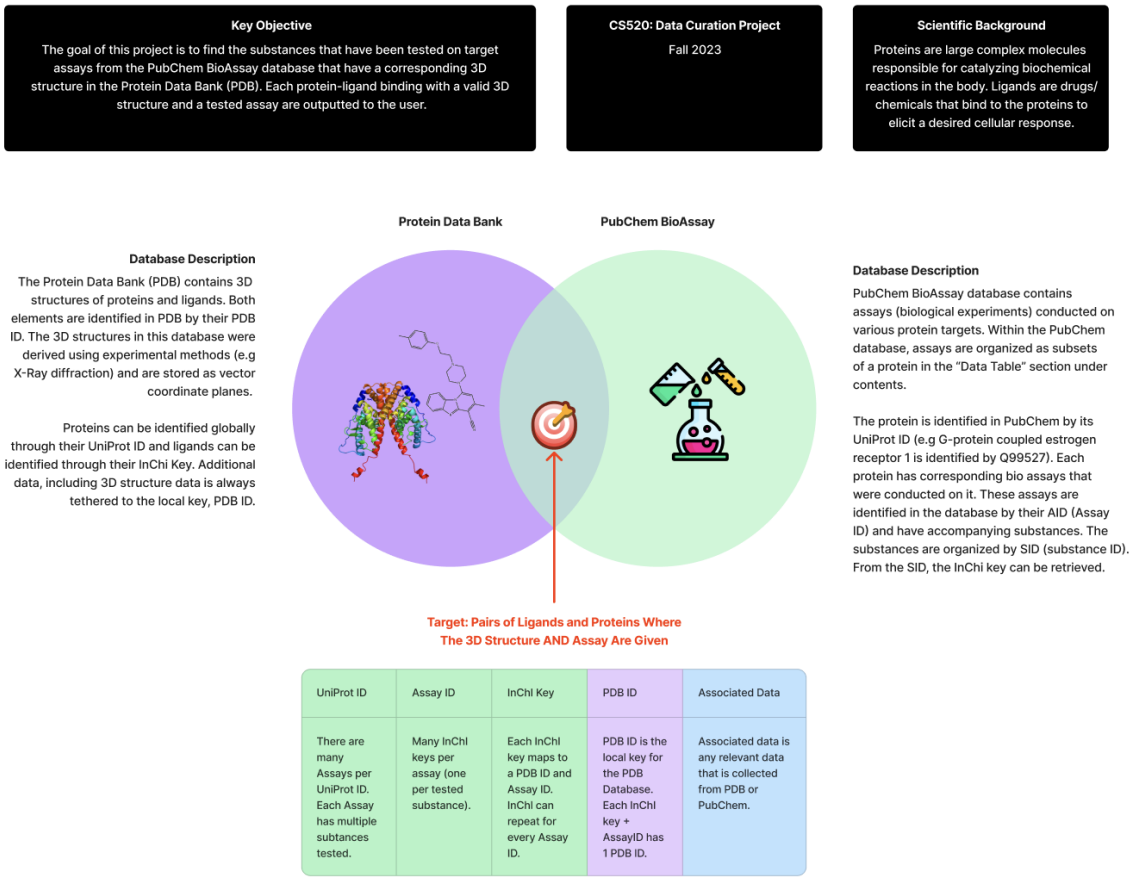
There are two open-access platforms that each contain components that are relevant to the scientific use case for mapping both databases. Experimentally derived 3D structures of proteins are stored in the Protein Data Bank (PDB); Biochemical Assays are stored in the PubChem BioAssay Database.

Both the PDB and BioAssay Databases are accessible by RESTful APIs.
Protein Data Bank API
PubChem BioAssay API

Protein Data Bank (PDB) – The PDB is a federated collection of experimentally derived 3D protein structures. The data in this repository was developed in laboratory settings using methods like X-ray crystallography and spectroscopy.

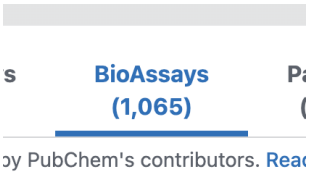
PubChem BioAssay Database – The NIH compiled 943 data sources of relevant biochemical data, including protein targets, bioassays, and proteins. PubChem BioAssays contains more than 1.6 million experiments (assays) conducted by PubChem contributors. Specifically, small-molecule and RNAi screening data with accompanying annotations are stored in the database.

Challenges to Data Integrity



Data Integrity Challenges for the PubChem Database

When querying assays in the PubChem database, the GUI allows users to enter whatever data they find to be most relevant in identifying the desired result. Additionally, there is a RESTful API that can be used to access this data. In the example of the Estrogen Receptor Beta, if you query Q92731, its accompanying UniProt ID, 1,065 BioAssays will be returned. These experiments are all experiments that involve this receptor as a target. However, many of these results aren't good fits for integration into a machine learning model. If they do not have a corresponding match to a 3D structure in the Protein Data Bank, that result can be eliminated. Moreover, assays with fewer than 10 tested substances and non-confirmatory (experimental) assay types are not good fits for training a machine learning model. Unfortunately, the PubChem API can't query for this directly because



the number of test results for an assay isn't a queryable field. These challenges will have to be overcome to make this data useful.

Data Integrity Challenges for the PDB Database

The Protein Data Bank carries a host of challenges, particularly since many results did not appear when queried because they are of a similar sequence (chemically similar) but not directly related. For example, querying UniProt ID Q92731 results in 37 identified structures. These structures are of the Estrogen Receptor Beta protein and its binding behavior with a ligand (target drug). The limitations exist because this UniProt ID is specific to the Estrogen Receptor Beta in humans. This results in acceptable targets being thrown out of the returned dataset.

Challenges Related to Integrating PubChem and PDB Databases

Both the data integrity challenges for querying the underlying databases need to be initially addressed. However, once they are, the process of mapping the PubChem results to their corresponding PDB results is the final challenge. There is no global key that can be used to identify a match, so instead each UniProt ID and InChI key need to be queried. If an assay and structure both involve both elements, it is a match. However, there is considerable preparation involved to complete the validation.

Steps to Improve Data Integrity and Integrate Data

1. **User Input is Requested –** upon running the notebook, the user will be prompted to enter a valid UniProt ID. This ID will identify the relevant structure-assay matches for this protein. For the Vizier notebook specifically, this ID is hardcoded but can be edited by the user to get the desired output.
2. **UniProt ID is Validated and FASTA is Extracted –** Directly querying the user's untrusted input could result in unintended responses. This is why the official UniProt database is queried to validate that the input is correct. Once the input is validated, the tool queries the FASTA sequence from the UniProt database (a third database and source of truth) to be used later. The FASTA sequence is a representation of the amino acid chains for a protein, which will help identify sequence-similar proteins in the PDB.
3. **PDB Results Are Gathered Using FASTA Sequence –** Once the FASTA sequence is extracted, it is used to query the Protein Data Bank API. The fetch_similarpid takes an argument of fasta_sequene and pid from the UniProt API and returns an array of identifiers (PDB ID) that share sequence parity or similarity to the user-inputted protein. To query this, the FASTA sequence was used with an evalue_cutoff of 0.0001 and an identify_cutoff of 0.6.
4. **Fetch InChI Keys from PDB –** Now that a list of eligible proteins is identified, the program iterates through the list of PDB IDs returned by the previous function and calls the fetch_inchi_key function, which takes a PDB ID as an argument. This function

queries the PDB and searches for structures that include the queried protein and a "non-polymer entity" (which is a small molecule like a drug/ligand and is called an NME). The response is returned to the Main function as a JSON packet. These packets are processed in sum and using another function called process_pdb_results, they are returned as a tuple containing a set of proteins and their corresponding NMEs. This data is more robust than what would have been achieved using a regular query by API/GUI through the use of the FASTA sequence from the UniProt API.

5. **Check 3D Protein Structure Relation to Assays** – This set is sent to a function that further cleans the data, find_matching_aids, and then to pubchem_match. Each of these is combined to query the PubChem BioAssays database. We query both the inchi_key and cid(compound ID was extracted using the UniProt ID) from the PDB results to return a list of assays (AIDs) that correspond to each. Then, by comparing these two lists we find assays that use the target protein and an NME that is paired with that protein in PDB. Thus, at this point, we have a list of protein-chemical pairings with the structure and function that can be returned to the user.
6. **Return Results to the User –** The data is returned to the user in an easily presented fashion.