MSc Project - Binding Affinity Prediction of Protein-Ligand Complex

Abdus Salam Khazi
abdus.khazi@students.uni-freiburg.de

Github Repository [3]

Supervisors: Simon Bray & Alireza Khanteymoori

July 22, 2021

Contents

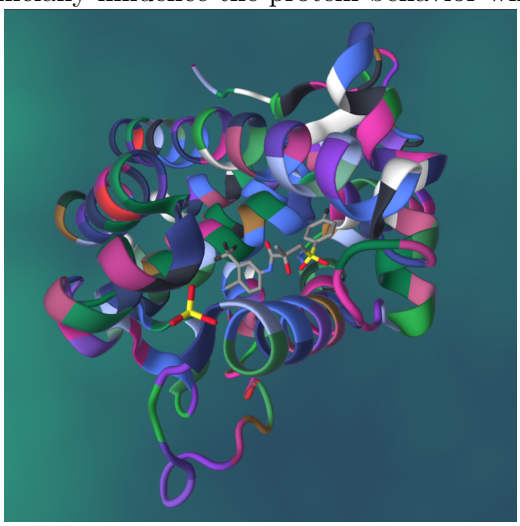
1 Introduction

.....TOBE REFINED AFTER COMPLETING THE FULL REPORT.....

1.1 Biological Background

Proteins are the workhorses of our body. Every main function is carried out by a protein or a collection of proteins. Ligands are molecules that bind to proteins to form protein-ligand complexes. They can be molecules that the protein transports e.g. Haemoglobin transporter or they can act as stimulating agents. In addition to this, they can also start/stop the protein from doing its function. The correct functioning of these protein-ligand complexes is essential for any living organism.

The study of protein-ligand complexes is an intrinsic part of the drug discovery field. This is because drugs are small molecules that act as ligands. As the drug molecules (ligands) bind to the target proteins, they can artificially influence the protein behavior which causes a therapeutic effect.

 [10]

When one proposes a target drug candidate, he has to answer questions like - How easily does the drug bind to the target protein? Does it bind to any other protein - If so is it desirable? Does it have any unforeseen effect on the protein function? etc. To answer these questions biologists and pharmacists conduct wet-lab experiments which are expensive.

One way to reduce the cost of these experiments is to make a data-driven selection of the drugs. Using experimental data collected over many years, one can build models to predict the behavior of the proposed drug

computationally. These 'In-Silico' computational methods can aid in the elimination of undesirable drugs as well guide the drug selection process.

Our project aims to answer one of the above questions - How well does a given drug bind to the target protein? We determine this computationally by building a machine learning model that is trained on the previous data. We hope that this model will be helpful in reducing the costs of drug discovery. But from where do we get the data to build our model?

1.2 PDBBind Dataset

Over the last few decades, researchers have been successful in building a single data archive for proteins. This archive, called **Protein Data bank** [5] , holds 3-D structural data of the proteins determined by experiments like X-ray crystallographic, Nuclear magnetic resonance (NMR), and cryoelectron microscopy (cryoEM). A subset of this data also contains information about how well a given protein and ligand bind together. It is called binding affinity between a protein and ligands. (It also contains data about protein-protein complexes which isn't dealt with in our project) [4]

As we are studying the protein-ligand binding affinity, we would like to filter out this data from the protein data bank. It is what is done by the maintainers of the **PDBBind Data bank**. [8] Using the curated protein-ligand affinity data present in the PDBBind Data bank, we build a machine learning model that learns to predict the affinity. But how is the binding affinity quantified?

1.3 Understanding Binding Affinity

A binding affinity between a protein and a ligand is quantified by the K_d, K_i and IC_{50} measures in the PDBBind Data bank. Here K_d refers to dis-association constant, K_i refers to the inhibition constant and IC_{50} refers to inhibitory concentration 50%. The reason for having different measurements is because it is not possible to use the same measurement techniques for all biological complexes/processes.

To understand K_d, consider a protein and a ligand binding and unbinding continuously in a kinetic system. In this system let $[P]$, $[L]$ and $[PL]$ represent the concentrations of the Protein, Ligands and the protein-ligand complex respectively. This system can be represented by the equation.

$$[P] + [L] \rightleftharpoons [PL]$$

We can quantify the binding affinity K_d by using the concentrations in the above system at equilibrium.

$$K_d = \frac{[P][L]}{[PL]} = \frac{k_{-1}}{k_1}$$

where k_{-1} is the disassociation rate constant and k_1 is the association rate constant. Similarly, K_i and IC_{50} are defined using concentration albeit non-trivially. [7]

Our problem, hence, boils down to this - Given $K_d/K_i/IC_{50}$ for various complexes in the PDBBind Data bank, can we predict this affinity measure for new protein-ligand complexes?

The next questions that need to be answered before we build our agent are - how exactly are the proteins and ligands represented in the PDBBind Data bank? How do we extract the properties of proteins and ligands for predicting this affinity?

2 Problem Set-up and Formulation

The binding of proteins and ligands is heavily influenced by their respective 3D structures. The **PDBBind Data bank** extracts information about these complexes from the **Protein Data bank** and creates the following files for every complex

- **PDB Format** - For the Protein.
- **Mol2** - For the ligand.
- **SDF** - For the ligand.

All of the above formats contain the 3D information that is essential in the prediction of the binding affinity. The 3D representation of the proteins and ligands is encoded like in the **XYZ format**.

2.1 XYZ format

XYZ format is a chemical file format that is used to represent the geometry of a molecule. It specifies the number of atoms and their Cartesian X, Y, Z coordinates hence the name XYZ format. These coordinates given are relative to each other hence, translation and rotation do not change the molecule's representation. The following text shows the XYZ format representation and an example. [11]

```
<number of atoms>
comment line
<element> <X> <Y> <Z>
...
```

The units of distance used is Angstrom (Å). $1\,\text{Å} = 10^{-10}$ m. For example, the pyridine molecule is represented in the following format. [11]

```
11

C       -0.180226841      0.360945118     -1.120304970
C       -0.180226841      1.559292118     -0.407860970
C       -0.180226841      1.503191118      0.986935030
N       -0.180226841      0.360945118      1.29018350
C       -0.180226841     -0.781300882      0.986935030
C       -0.180226841     -0.837401882     -0.407860970
H       -0.180226841      0.360945118     -2.206546970
H       -0.180226841      2.517950118     -0.917077970
H       -0.180226841      2.421289118      1.572099030
H       -0.180226841     -1.699398882      1.572099030
H       -0.180226841     -1.796059882     -0.917077970
```

2.2 PDB, SDF and Mol2 formats

PDB format is a human readable file format used to represent the protein molecules (macro molecules). In addition to the 3D information of atoms, it contains information regarding protein's primary, secondary, tertiary and the quaternary structures. Because of the presence of 3D positional data, molecular visualization is possible using specialized software. [9] [2]

SDF format file is a type of molecular descriptive file which contains x,y,z format similar to pdb format. It also contains bond information. This data can be used to create a graph using which SMILES string can be created. SDF format

Mol2 format file is also a file similar to the sdf format used to represent the ligands. We chose to use this format because we could extract the features of more molecules using this format.

2.3 Problem Formulation

Protein-Ligand complex problems can be broadly classified into 2 types -

- LBS - Ligand binding site prediction.
- Ligand affinity prediction.

LBS (Ligand binding side) prediction can be further classified into 3 types -

- 3D structure based
- Template based
- Sequence based

The deal with binding affinity prediction in our project. In our problem we take care of the following 2 things

- We do not take the features of the whole protein but take the features of the location of the protein which binds to the ligand. This is called a pocket.
- We try to keep the features of proteins and ligands distinct till the input so that we can have a plug and play sort of input for our model. The binding affinity of any protein or ligand can be found out because of this property of our modelling.

However for this we make use of existing 3D structure based LBS tool chain called fpocket. The reason this is helpful is because 3D structural features of proteins are very crucial for the binding between the proteins and ligands as proteins and ligand interactions can be considered as machines of 2 parts interacting with each other.

2.4 fpocket/dpocket descriptors

We extract the features of a pocket using dpocket tool (a submodule in the fpocket toolchain). fpocket uses voronoi tessalation (3D) to find out pockets in our protein structure. To get the descriptors/features of the pockets at which ligands bind we use the dpocket (aka descriptors pocket). dpocket creates 3 types of descriptors for the pockets in the protein -

- fpocketp. This lists all the possible pockets (with descriptors) that could bind to ligand according to a criteria. Multiple pockets can bind with the same ligand. Here the descriptor called overlap maybe 100% or less.

- fpocketnp. This lists all pockets (with descriptors) that are not binding according to the criteria.

- explicitp. This lists all the explicitly binding pocket (with descriptors). Here the overlap feature is always 100%.

We use both fpocketp and explicitp descriptors to train our model. There are in total 55 descriptors in total obtained using dpocket descriptors.

3 Feature selection

3.1 Dichotomous problem

Since the protein and the ligand are equally responsible for the affinity between them, our problem (also the binding site prediction problems) can be classified as a dichotomous problem in which we need data from both the protein and the ligand. The data if it is complementary is better for solving the problem. The accuracy measure of the protein LBS prediction is the same as the dichotomous problems in math.

As we are dealing with a dichotomous problem we have to select features of the protein and the ligand separately. This helps us make our model plug and play w.r.t the proteins and ligands. The following feature selection mechanisms were used to select features -

- Each feature of both protein and ligand were correlated with the output i.e the Dissociation constant. We used both pearson and spearmann correlation to calculate this. The assumption to be made when taking these correlations is that of monotonic increase of the feature with respect to the output. The features with the highest correlation were taken from both proteins and ligands as inputs.

- By using genetic algorithms[1]. Here each feature is represented by a binary number. 1 indicating inclusion and 0 indicating exclusion from the input of our model. A binary string of 456 binary numbers (401 for ligands + 55 for proteins) is called a chromosome. See pseudocode below

- Features selected by expert. Given by Simon Bray. (Yet to do)

Algorithm 1 Selection of features in our model using genetic algorithm [1]

procedure GENETIC_ALGORITHM_BASED_SELECTOR
    model_type ← Linear Regression
    $population = \{C_1, C_2, C_3...C_n\} \in \mathbb{B}^{456}$ (initial chromosomes).
    best ← C_1
    $i \leftarrow 0$
    gen ← number of generations to run.
    for $i < gen$ **do**
        Fit model_type for each feature selection (chromosome).
        Let $\{S_1, S_2, S_3...S_n\} \in \mathbb{R}$ be scores for each chromosome.
        best ← C_i with the highest score S_i
        for $j < len(population)$ **do**
            Set ← 3 random chromosomes from *population*
            $c \leftarrow best(\text{Set})$
            $genetically\_better\_population.add(c)$
        end for
        $population \leftarrow genetically\_better\_population$
        for $j < len(population)$ `with step 2` **do**
            $P_1, P_2 \leftarrow population[j], population[j+1]$
            $c_1, c_2 \leftarrow crossover(P_1, P_2)$
            $c_1 \leftarrow mutation(c_1)$
            $c_2 \leftarrow mutation(c_2)$
            $children.add(c_1, c_2)$
        end for
        $new\_population \leftarrow population$
    end for
    return best
end procedure

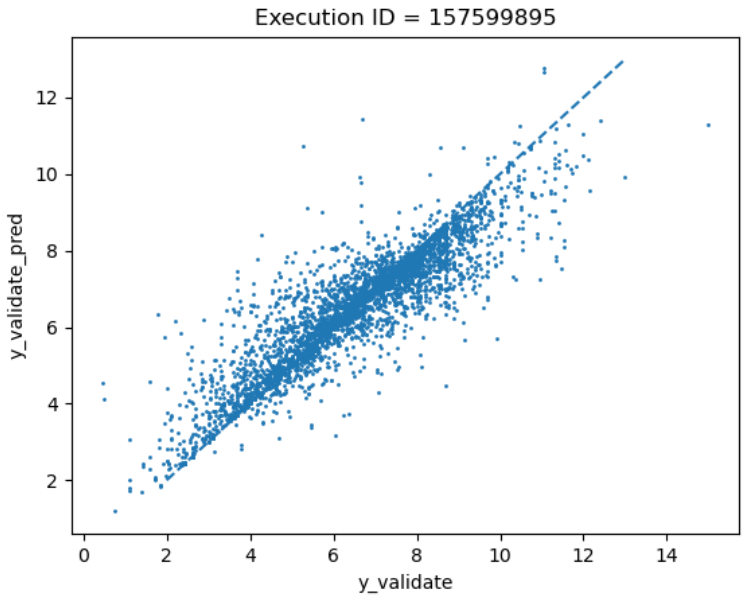
4 Testing strategy

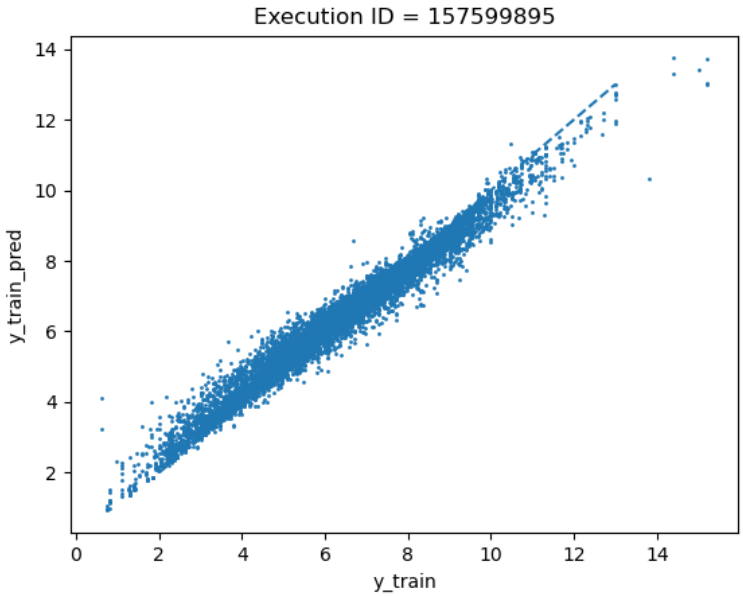
Our problem is a regression problem. The input space of our model is multi-dimensional. Hence we cannot fully visualize our model as a function of the input space. To get around this we use R^2 score, coefficient of determination, of to test the quality of the models. $R^2 \in (-\infty, 1.0]$ where 1.0 is the best score. [6]

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

Where $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ and $\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} \epsilon_i^2$

For visualizing the results given by our regression model, we plot a graph of $y_{predicted}$ vs y_{actual}. This is plotted for both the training and the validation data.





The perfect model would have all the points in the graph on the $y = x$ line.

5 Machine Learning Models

Various machine learning models were trained using the extracted data. We tried out Linear Regression, Support Vector Regression, A small Neural Network, as well as a Random Forest Regression. But far the most impressive performance was given by Random Forest Regression.

5.1 Simple linear regression

This is the most computationally cheap model that we used to fit our model. The R^2 score for this model we got ≈ 0.455 when we used all the features of the ligands and the proteins. Since the fitting of the model was very cheap, we tried to use the validation score of the model as a fitness score for our genetic algorithm.

5.2 Random Forest Regression

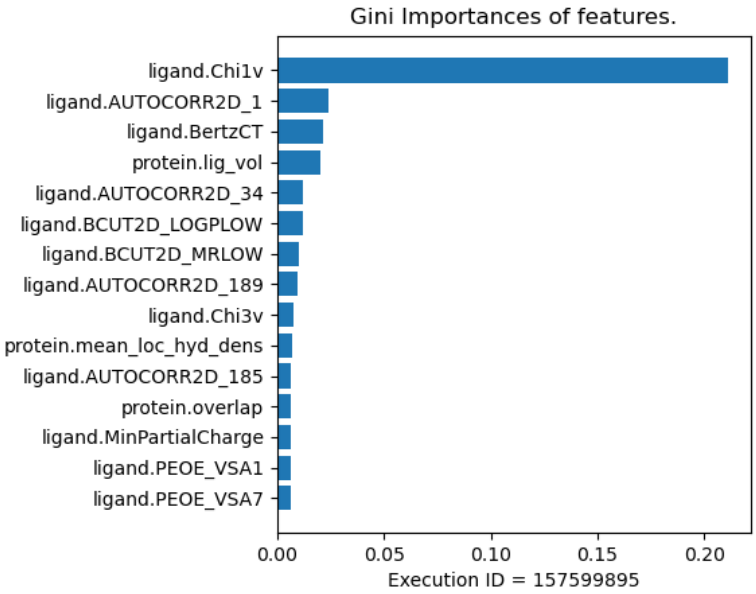
A random forest regression is an ensemble model which uses an ensemble of trees to calculate the output variable. Each node in the tree reduces tries

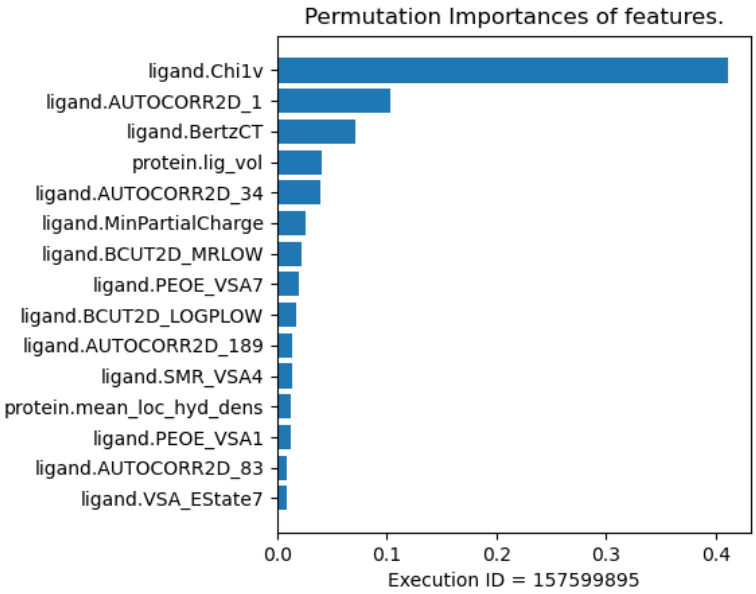
to reduce the entropy of the data at hand. The R^2 score we got was > 0.75 for an ensemble of 100 trees.

5.2.1 Feature Importance calculation

The Random Forest regression model fitting is very expensive as compared to the simple linear regression. Hence the same strategy cannot be used for feature selection as used in the linear models. We used 2 main methods to determine the importance of features in the model

- **Gini Importance** - This is given by default by the model. The model calculates the amount of entropy reduction of that each feature does when it divides the data using the OOB score. (Out of bag Score)
- **Permutation Importance** - This is a model agnostic method to determine the importance of features in a regression. It tries to calculate the reduction in the accuracy we shuffle each column.





The issue with the above methods is that if there is any hidden correlation between input features then it will reduce the importance of the correlated features.

5.3 Permutation Importance and genetic algorithm

To overcome the issue with the above feature importance selection, we plan to use a combination of the concepts of permutation importance and genetic algorithm to find out the importance of the selected features.

Results - YET TO IMPLEMENT

6 Discussion

7 Conclusion

References

[1] Jason Brownlee. Simple Genetic Algorithm From scratch. Link. [Online; accessed 28-June-2021].

[2] TMP Chem. Computational Chemistry 1.2 - PDB File Format. Link. [Online; accessed 22-July-2021].

[3] Abdus Salam Khazi. Code for the whole project. Link. [Online; accessed 22-July-2021].

[4] PDBank. PDBank History. Link. [Online; accessed 22-July-2021].

[5] PDBank. PDBank Homepage. Link. [Online; accessed 22-July-2021].

[6] scikit learn. R2 score, the coefficient of determination. Link. [Online; accessed 22-July-2021].

[7] The Science Snail. Difference between Ki, Kd, IC50 and EC50 values. Link. [Online; accessed 22-July-2021].

[8] Wikipedia. PDBbind database. Link. [Online; accessed 22-July-2021].

[9] Wikipedia. Protein Data Bank (file format). Link. [Online; accessed 22-July-2021].

[10] Wikipedia. Protein–ligand complex. Link. [Online; accessed 24-June-2021].

[11] Wikipedia. XYZ Format. Link. [Online; accessed 22-July-2021].