Binding Affinity Prediction of Protein-Ligand complexes using Machine Learning

MSc Project

Abdus Salam Khazi

Supervisors:
Simon Bray & Alireza Khanteymoori

October 15, 2021

Table of Contents

1. Introduction: Biological Background
2. Problem Definition
3. Data Processing and Analysis
4. Testing strategy
5. Machine Learning Models & Results
6. Discussion
7. Q & A
8. References
9. Appendix

Biological Background

What are proteins and ligands?

- **Proteins:** Complex molecules that are work-horses (machines) of a living organism.
- **Ligands:** Molecules that bind to (receptor) proteins.
- Proteins and ligands bind together to form protein-ligand complexes.

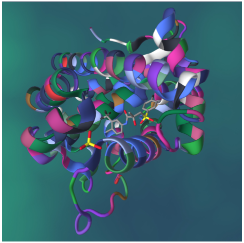


Figure: Haemoglobin transporter protein.

Biological Background

Protein-Ligand complexes

- Ligands bind to proteins at "cavity" like locations called pockets.
- The pockets and the ligands are complementary in shape.

Drugs

- Drugs are ligand molecules that bind to proteins.
- They cause a therapeutic effect after binding to the proteins.

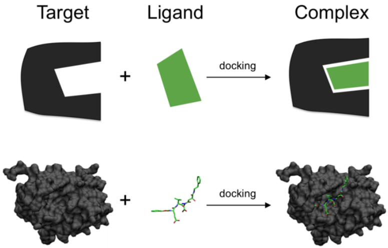


Figure: Lock and Key hypothesis in molecular docking.

Biological Background

Protein-Binding Affinity

- Assume a dynamic system in which protein P and ligand L are binding and unbinding continuously.
- Binding affinity can be quantified by using disassociation constant k_d (**at equilibrium**).

$$k_d = \frac{[P][L]}{[PL]}$$

Here $[P]$ = Protein concentration, $[L]$ = Ligand concentration and $[PL]$ = Protein-Ligand complex concentration.

Problem Definition

- Determining if a potential drug (ligand) can bind to a target protein is very expensive [3].
- The project tries to reduce the drug discovery costs by eliminating bad leads.
- **Problem definition**: Predict protein-ligand binding affinity using "In-Silico" methods.

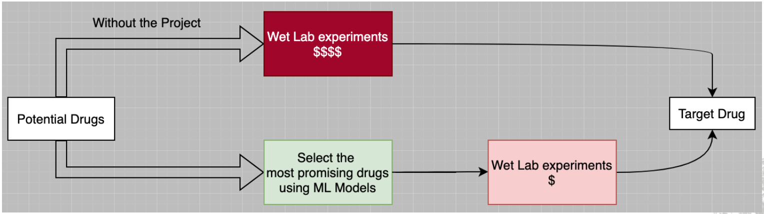


Figure: Project Overview

Problem Definition

There are various problems in the protein-ligand domain. The following figure shows the classification tree.

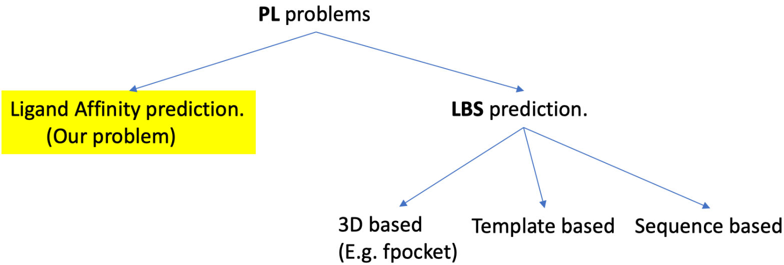


Figure: Protein-Ligand problem classifcation.

Feature Extraction

- The input data to the ML model is extracted from a database called PDB Data bank.
- *fpocket* and *RDKit* were used to extract the features of proteins and ligands.
- The input features contain information about the 3D structures of the proteins and the ligands.

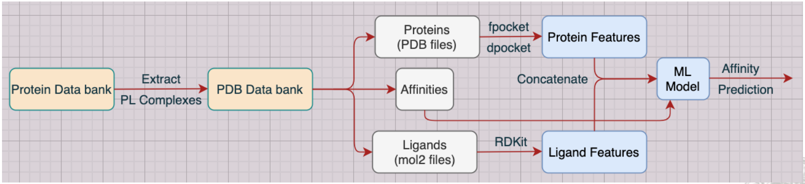


Figure: Data Input Overview.

Feature Extraction

Protein Features:

- *fpocket* is an LBS prediction algorithm used to predict ligand binding pockets.
- There can be multiple binding pockets for a PL complex.
- Using *dpocket*, 55 descriptors were obtained for every (potentially) binding pocket as real values.

Ligand Features:

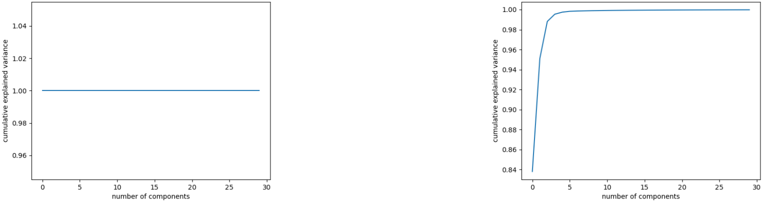
- Using *RDKit.Chem.Descriptors* module, 402 features were extracted as real values.

Concatenation:

- The (concatenated) input feature space to the model was $\mathbf{R}^{457}$.
- It was less than $\mathbf{R}^{457}$ if feature selection is done before model training.

Data Preprocessing

- Data points containing NaN (Not a number) values were removed from the data.
- PCA (Principal Component Analysis) was used to find the variance contribution of the features.
- Feature *IPC* was log scaled for numerical safety during training.



(a) With original Ligand feature IPC.

(b) With log scaled Ligand feature IPC.

Figure: Cumulative PCA of ligand features.

Feature Selection

Feature selection strategies:

- **Manual:** 121 ligand descriptors + all protein descriptors.
- **Output Correlation**: Features with the best *Pearson* or *Spearman* correlation w.r.t the affinity score (output) were selected.
- **Genetic Algorithms**: Genetic algorithms with a population score function "$\mathbf{R}2 \text{ score} * \text{Features Eliminated}$" was used to select the best features [4]:

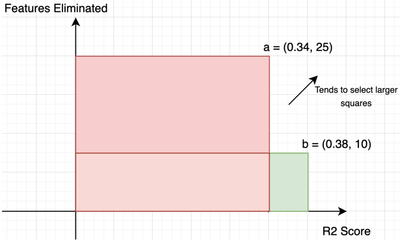


Figure: Genetic Algorithm score function representation.

Dealing with measurement resolution

- In PDB bind databank, each complex has a corresponding measurement resolution (Å units).
- The structural detail of the 3D image is inversely proportional to the measurement resolution.
- The weighting of each data point was done according to hyperbolic formulae (or) linear formulae:

$W_{\mathrm{Hyperbolic}} = \frac{\max \mathrm{R}_{1\ldots\mathrm{n}}}{R_i}$ $W_{\mathrm{Linear}} = (\max \mathrm{R}_{1\ldots\mathrm{n}} + 1) - R_i$

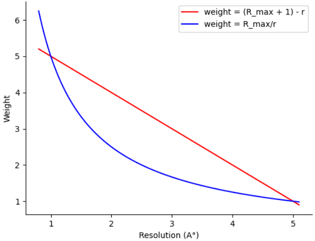


Figure: Weight calculation formulae.

Feature Family Correlations

- Features can be divided into families.
- Important ones are - AUTOCORR2d_, Chi, EState_VSA, PEOE_VSA, SMR_VSA, SlogP_VSA, VSA_EState, and fr_.
- Within Chi and AUTOCORR2d_, the features are correlated.
- ML models need to take into account this issue.

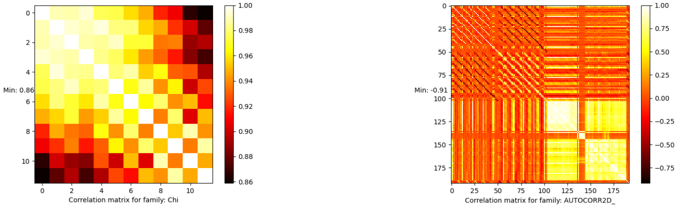


Figure: Correlation Heat Map.

Testing strategy

For every execution, we report the **random seed** used in it. This random seed can be used as a script argument (Execution ID) to reproduce the exact results. To determine the result quality we use:

- **R^2 score** (*Coefficient of determination*): $R^2 \in (-\infty, 1.0]$ where 1.0 is the best score.
- **Visualization:** The model's prediction is visualized as a 2D scatter plot. The best plot is $y = x$ line which corresponds to the best R^2 score of 1.0.

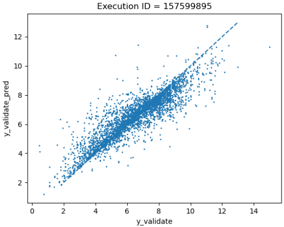


Figure: (Sample) Visualizing accuracy. $R^2 \approx 0.805$.

Machine Learning models

The ML model should approximate the following function:

$$\text{Binding affinity prediction} : \mathbf{R}^n \mapsto \mathbf{R} \quad \text{where} \quad n \in \mathbf{I}^+$$

The following ML models were studied

- Simple Linear Regression
- Random Forest Regression
- Support Vector Regression
- Rotation Forest Regression

Also Note:

- DNNs cold not be used due to lack of data.
- The project only had ≈ 16000 data points to train.
- A simple DNN a model of size $[457, 20, 10, 1]$ has 9350 parameters.
- The DNN would overfit drastically.

Simple Linear Regression

- This model approximates the binding affinity using a linear hyperplane. (By minimizing the square of errors).
- It is the cheapest computational model.
- Assumes strong linear relationship between input features and binding affinity.
- Genetic algorithms sucessfully used for feature selection.
- Alternate weighting strategy: Data duplication.

No. features	Feature selection	Weighting	Training	Validation	Testing
457	-	-	0.461	0.415	0.320
457	-	Hyperbolic	0.454	0.427	0.337
457	-	Hyperbolic duplication	0.465	0.416	0.326
457	-	Linear	0.458	0.419	0.328
457	-	Linear Duplication	0.460	0.428	0.327
49	**Genetic**	**Hyperbolic**	≈**0.377**	≈**0.374**	≈**0.364**
40	Pearson Correlation	Hyperbolic	0.287	0.278	0.285
40	Spearman Correlation	Hyperbolic	0.289	0.294	0.290
176	Manual	Hyperbolic	0.362	0.346	0.331

Table: R^2 scores of the Linear Regression Model.

Random Forest Regression

Introduction

- It is a non-linear ensemble model of regression trees. (Sampling with replacement aka Bagging is used)
- For each tree, the data is split recursively till a stopping criterion. Each data subset has lesser entropy than the superset.
- There is no need for any assumption w.r.t data.
- Additional feature selection strategy: Genetic Elitism.

Dealing with correlated features

- The RF was forced to randomly use only 20% of the features to determine the best (feature, value) combination for each split step.
- The RF model does not depend on some features exclusively.
- The correlated features were used as an advantage.

Random Forest Regression

Feature Importances

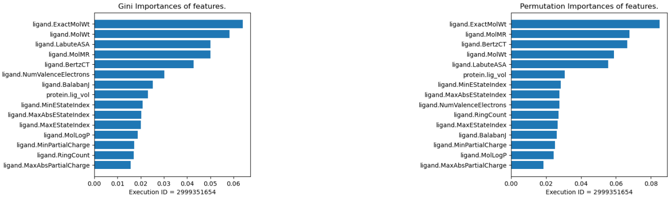
- **Gini Importance:** (Provided by the RF model)
  - The most important features are the ones that contribute the most in the reduction in entropy.
- **Permutation Importance:** (Model agnostic)
  - The most important features are the ones that are most relevant for prediction accuracy.

$$\mathrm{Importance}(f) = \mathrm{Accuracy}_{\mathrm{pred}}(X) - \mathrm{Accuracy}_{\mathrm{pred}}(X_{f\_shuffled})$$

Genetic feature selection

- Fitting an RF model is expensive. However evaluation of feature importance is cheap.
- Hence the following scoring function is used, $s = \mathrm{Importance}(f_1, f_2 ...) * \mathrm{features\_eliminated}$

Random Forest Regression



(a) Gini importance.

(b) Permutation importance.

Figure: Feature Importance calculation of Random Forest Regressor (With manual feature selection).

No. Features	Feature Selection	Weighting	Training	Validation	OOB score	Testing
457	-	-	0.961	0.790	0.791	0.447
457	-	Hyperbolic	0.961	0.773	0.794	0.448
457	-	Linear	0.960	0.785	0.794	0.443
40	Spearman Correlation	Hyperbolic	0.930	0.664	0.671	0.878
40	Pearson Correlation	Hyperbolic	0.925	0.642	0.645	0.868
229	Genetic Elitism	Hyperbolic	0.958	0.779	0.793	0.444
394	Genetic Normal	Hyperbolic	0.961	0.776	0.795	0.448
176	**manual**	**Hyperbolic**	**0.949**	**0.734**	**0.736**	**0.463**

Table: Random Forest Regression R^2 Score table.

Support Vector Regression

- It is a non linear model.
- It fits a ϵ radius pipe to the data.
- The model was not suitable for running genetic algorithms (for both normal and permutation genetics). Training took $\approx 99.07s$ and validation took $\approx 27.01s$.
- Features were selected using either output correlation or manual feature selection.
- Contrary to other models, linear weighting of data gave the best results.

No. Features	Feature Selection	Weighting	Training	Validation	Testing
457	-	-	0.311	0.290	0.314
457	-	Hyperbolic	0.330	0.323	0.327
457	-	**Linear**	**0.344**	**0.319**	**0.335**
40	Spearman Correlation	Linear	0.260	0.268	0.262
40	Pearson Correlation	Linear	0.198	0.196	0.197
40	Spearman Correlation	Hyperbolic	0.254	0.265	0.256
40	Pearson Correlation	Hyperbolic	0.168	0.171	0.168
176	Manual	Linear	0.311	0.283	0.310

Table: Table showing R^2 Scores for SVR.

Rotation Forest Regression

- Random forest trees divide data using axis aligned hyperplanes.
- The complexity of the trees can be reduced by linearly transforming (rotating) the data. The eigen vectors of the training data become the basis vector. For example:
    - Representing $y = kx$ would take a very deep tree.
    - If $y = kx$ is linearly transformed to $y = 0$, only 1 node is enough.
- Computationally very expensive - It took $\approx$ 25 min 29 seconds to train.

No. Features	Selected	Training	Validation	Testing
457	-	0.967	0.767	0.449
40	Spearman Correlation	0.952	0.650	0.890
40	Pearson Correlation	0.951	0.629	0.883
176	**manual**	**0.960**	**0.716**	**0.471**

Table: Rotation Forest R^2 Score overview.

Discussion

Notable points:

- Best models: Linear Regression and Random Forest Regression.
- Random Forest uses correlated features to make itself more robust.
- Random Forest can deal with both discrete and real valued features.
- Rotation Forest did not improvement accuracy as they are good only for Real valued features.
- Testing results were sometimes better than validation results. It is because test data $<$ validation data. But the difference is minimal.

Limitations:

- Linear regression assumes data linearity.
- Random Forest has heavy reliance on ligand features.
- Both models were black box models.
- Genetic feature selection was not helpful for Random Forests.

Discussion

Further work:

- A new weighting strategy: Weighting a pocket descriptor based on the overlap between the pocket and the ligand. For example,

$$W_{\text{Total}} = W_{\text{Hyperbolic}} * W_{\text{Overlap}}$$

- Improvement of feature selection: Build 1 model per family of features. Use the best feature as a family surrogate.
- A more explainable model can be built.

Q & A

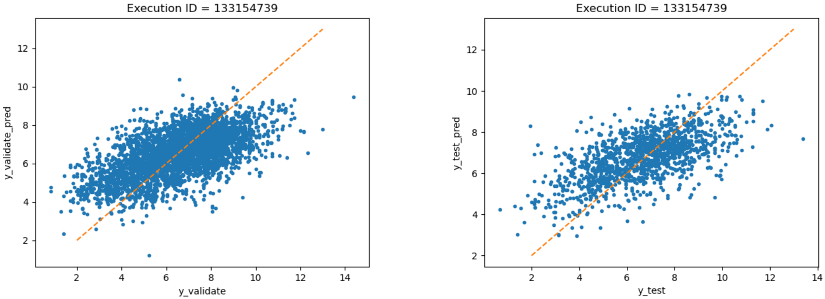
References

- Du, Li, Xia, Ai, Liang, Sang, Ji and Liu; Insights into Protein–Ligand Interactions: Mechanisms, Models, and Methods (2016)
- Le Guilloux, Schmidtke, and Tuffery; Fpocket: An open source platform for ligand pocket detection(2009)
- DiMasi, Grabowski and Hansen; nnovation in the pharmaceutical industry: New estimates of R & D costs (2016)
- John H. Holland. Genetic Algorithms. (1960)
- Is rotation forest the best classifier for problems with continuous features? A. Bagnall, M. Flynn, J. Large, J. Line, A. Bostrom, and G. Cawley (2020)

Appendix - Definitions

- Binding affinity between a protein and a ligand is quantified by the K_d, K_i and IC_{50}. Here K_d refers to the dissociation constant, K_i to inhibition constant, and IC_{50} to inhibitory concentration 50%.

Appendix - Visualizing Linear Regression results

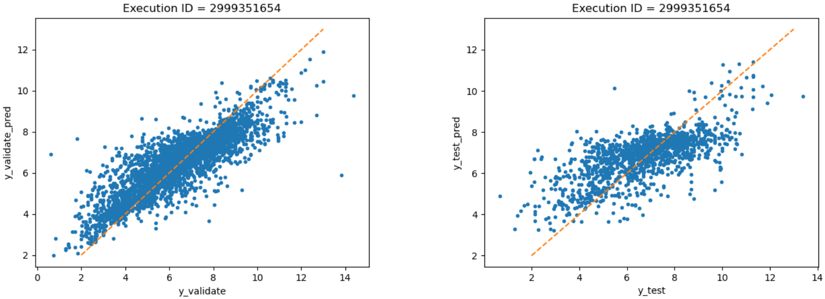


(a) Validation Accuracy.

(b) Test Accuracy.

Figure: Linear Model using Hyperbolic weighting and all 49 features selected by genetic algorithm.

Appendix - Visualizing Random Forest results

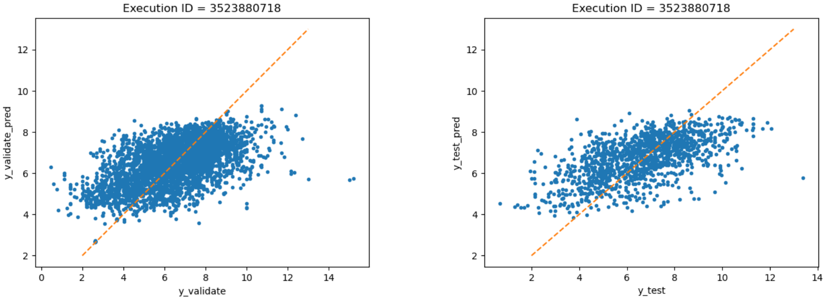


(a) Validation Accuracy.

(b) Testing Accuracy.

Figure: Random Forest Regressor with 176 manually selected features and hyperbolic weighting.

Appendix - Visualizing SVR Results


Execution ID = 3523880718

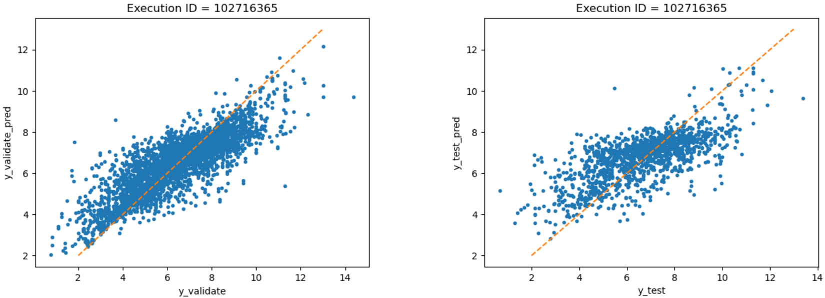
Execution ID = 3523880718

(a) Validation accuracy.

(b) Testing accuracy.

Figure: SVR accuracy visualization for all features 457 and Linear weighting.

Appendix - Visualizing Rotation Forest Results



(a) Validation accuracy.

(b) Testing accuracy.

Figure: Rotation Forest Accuracy visualization for manually selected features (176). The rotation forest implementation does not support data weighting.