Using Machine Learning Techniques to Distinguish Between Agonist and Antagonist.

Abhish Panwar(2019135) Raghav Sharma(2019189) Prashant Singh(2019188)

Introduction

Proteins are necessary for the functions that our cells carry out. A cell responds when an agonist molecule is attached to a receptor; hence, the complex is known as an agonist complex. An antagonist will prevent the binding site from being used and have the opposite outcome as an agonist. Understanding how the complexes function can aid in improving drug discovery and manufacture.

In the project so far, our motive was to classify the complex as an agonist or antagonist in nature; till now, features from the whole protein sequence have been used. Now the focus is on the ligand. The pocket surrounding the ligand, i.e., a minor part of the entire complex, is analyzed to try and do the job faster and better. Softwares such as FPocket and Cavity Ligbuilder are used to find the pocket region in the given protein-ligand complex, and then features from this pocket region are extracted. Some additional features are also calculated to better classify the protein-ligand combination as of agonist or antagonist nature. Various ML models are applied to these final extracted features to get a good accuracy for the classification as mentioned above.

Methodology

Data Collection

The pertinent protein files were acquired using the RCSB database. Protease bound with agonist" and "Protease bound with antagonist" produced all matching IDs, which were then retrieved. Files containing RNA or DNA were removed. Any files that did not have "Agonist" or "Antagonist" in their PDB title, header, or primary PubMed citation were eliminated in the following phase of automatic screening. To confirm the inclusion of the ligand and protein in the file, the files were manually screened using the criteria mentioned above and the molecules and ligands present in them. Duplicates were removed, and the final data composition was as follows:

- 551 proteins and 156 antagonists
- 395 Offenders

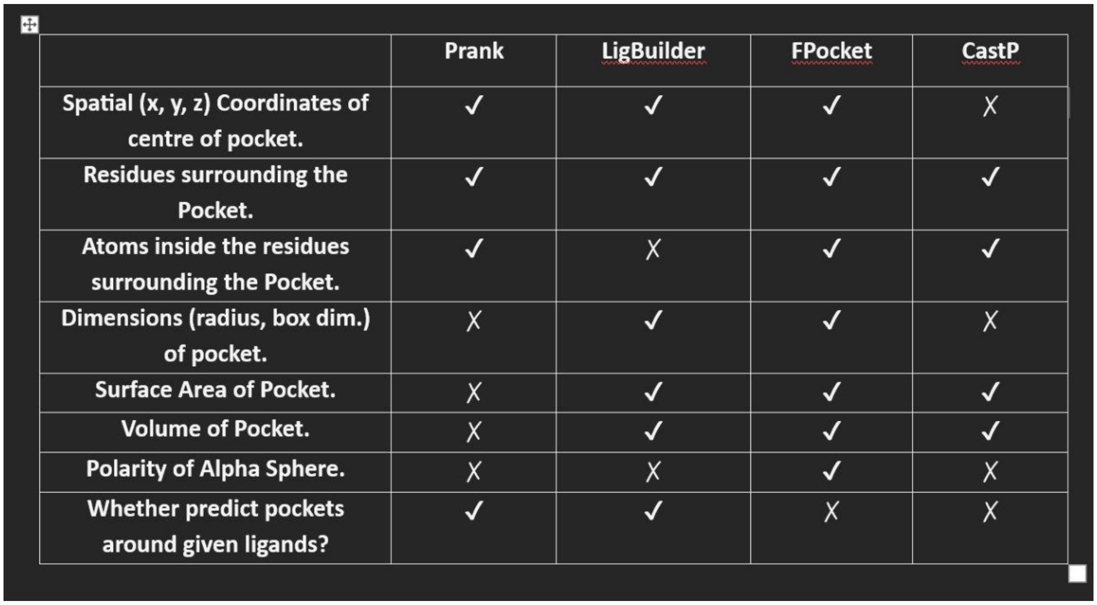
Peptide ligands were likewise eliminated when ligands were considered characteristics in the models, leaving 489 proteins, of which 131 were antagonists and 358 agonists. There are also 214 different proteins and 423 different ligands.

Filters Applied

Filter applied on Data:
The data was filtered after removing some of the ligands. Fpocket and Cavity Ligbuilder were not able to give correct data regarding these protein-ligand combinations. Combinations containing ligands like 'SO4', 'CA,' 'ZN' etc., were removed from the data file. These ligands were ions and were not playing any role in determining the agonistic/antagonistic nature of the complexes. So they played no significant role in the classification and were thus filtered out. Also, complexes containing the ligand -'NAG,' related to crystallization, were filtered out. The remaining data were used to find the pocket regions and extract the required features for applying the ML models.

Finding Pockets Regions

Many different software and websites were used to find a protein-ligand combination's pocket region. Some of the examples of these Software are FPocket, Cavity Ligbuilder, P-Rank, and CastP. Among these software, two were finalized because they provided the most favorable features according to our project. Various characteristics of the outputs from the different software are compared below in a tabular form. These two pieces of software provided some extra information about the pocket region of the complex, which was later featured for the ML part. For example, Dimensions of the pocket (radius, volume) were obtained in Fpocket and Ligbuilder-cavity but not in P-Rank or Castp. Both software supported pdb files as input and were easily automated for large datasets.

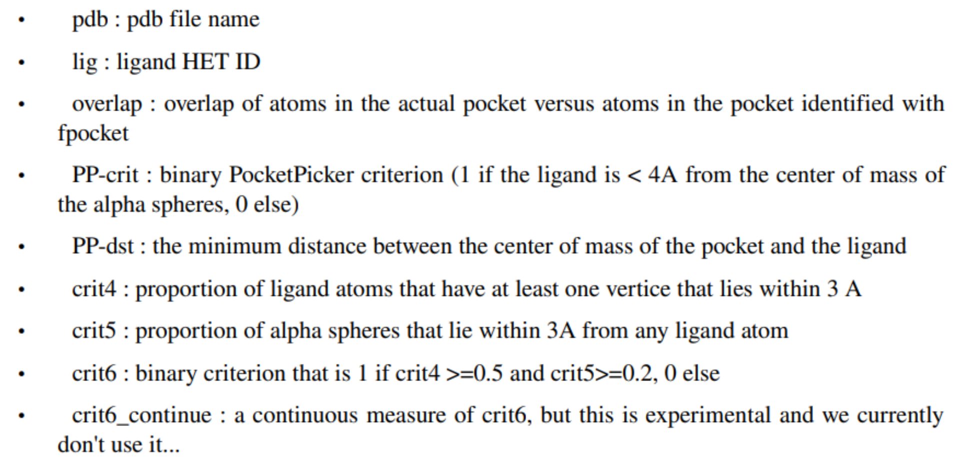


	Prank	LigBuilder	FPocket	CastP
Spatial (x, y, z) Coordinates of centre of pocket.	✓	✓	✓	✗
Residues surrounding the Pocket.	✓	✓	✓	✓
Atoms inside the residues surrounding the Pocket.	✓	✗	✓	✓
Dimensions (radius, box dim.) of pocket.	✗	✓	✓	✗
Surface Area of Pocket.	✗	✓	✓	✓
Volume of Pocket.	✗	✓	✓	✓
Polarity of Alpha Sphere.	✗	✗	✓	✗
Whether predict pockets around given ligands?	✓	✓	✗	✗

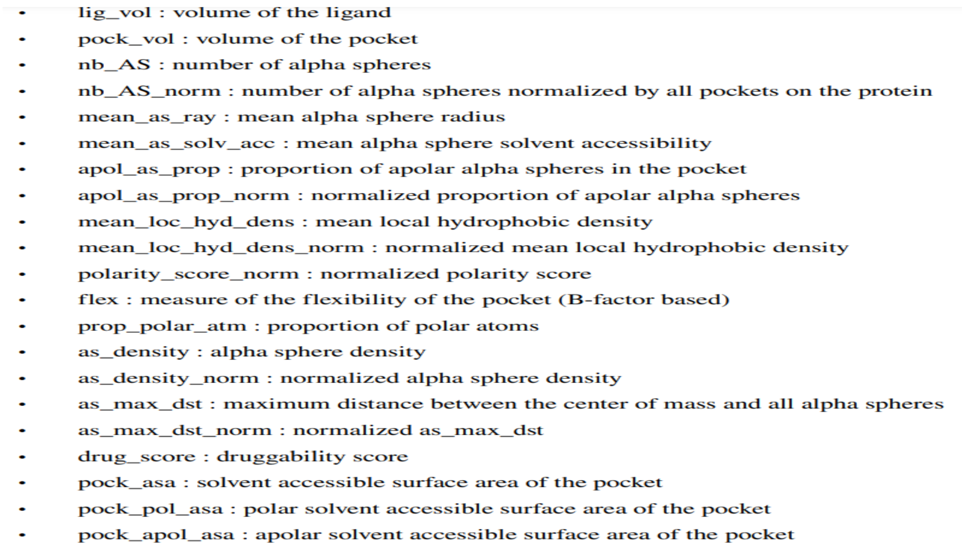
Output

Apart from giving the pocket region in the protein-ligand complex, various pocket characteristics were obtained. Output characteristics taken from Cavity-Ligbuilder are Total Surface area, Maximal pKd, Average pKd, Drug Score, and Druggability.

Fpocket provided various characteristics as output. Their names and meanings are mentioned below:



- pdb : pdb file name
- lig : ligand HET ID
- overlap : overlap of atoms in the actual pocket versus atoms in the pocket identified with fpocket
- PP-crit : binary PocketPicker criterion (1 if the ligand is < 4A from the center of mass of the alpha spheres, 0 else)
- PP-dst : the minimum distance between the center of mass of the pocket and the ligand
- crit4 : proportion of ligand atoms that have at least one vertice that lies within 3 A
- crit5 : proportion of alpha spheres that lie within 3A from any ligand atom
- crit6 : binary criterion that is 1 if crit4 >=0.5 and crit5>=0.2, 0 else
- crit6_continue : a continuous measure of crit6, but this is experimental and we currently don't use it...



- lig_vol : volume of the ligand
- pock_vol : volume of the pocket
- nb_AS : number of alpha spheres
- nb_AS_norm : number of alpha spheres normalized by all pockets on the protein
- mean_as_ray : mean alpha sphere radius
- mean_as_solv_acc : mean alpha sphere solvent accessibility
- apol_as_prop : proportion of apolar alpha spheres in the pocket
- apol_as_prop_norm : normalized proportion of apolar alpha spheres
- mean_loc_hyd_dens : mean local hydrophobic density
- mean_loc_hyd_dens_norm : normalized mean local hydrophobic density
- polarity_score_norm : normalized polarity score
- flex : measure of the flexibility of the pocket (B-factor based)
- prop_polar_atm : proportion of polar atoms
- as_density : alpha sphere density
- as_density_norm : normalized alpha sphere density
- as_max_dst : maximum distance between the center of mass and all alpha spheres
- as_max_dst_norm : normalized as_max_dst
- drug_score : druggability score
- pock_asa : solvent accessible surface area of the pocket
- pock_pol_asa : polar solvent accessible surface area of the pocket
- pock_apol_asa : apolar solvent accessible surface area of the pocket

Additional Features

From the output characteristics of Fpocket, we found the Amino acid composition of all the obtained pockets. The properties of these Amino acids were used to define some additional features for the entire pocket. These other features were calculated to better classify the protein-ligand combination as agonists/antagonists. Hydrophobicity, Polar, Aromatic, Small, and Aliphatic were additional features. The percent composition of the amino acids in the whole pocket gave results to the value of these features, which were normalized between 0 and 1.

The final features were 53 in number. After converting them all in numerical format (e.g., Druggable was converted to 1, undruggable to -1). All final features were normalized for all the input data. These features were added in a .csv file which was analyzed further for more feature reduction after some filtering. The final CSV file, with 0/1 labels, was used to train the ML model for the given dataset.

Result

After filtering out the samples, we have a total of 444 samples left. Out of these, 324 were agonists, while 120 were antagonists To handle this class imbalance, we oversampled the data using SMOTE.

On combining features from fpocket and ligbuilder, we have a total of 53 features. Since the data across each feature has a different range, we have normalized the data using standard scalar. We have reduced the feature from 53 to 40 using truncated SVD. Reducing the features to 40 gives us the best test accuracy.

We have then applied various machine learning models – Random Forest Classifier, Support Vector Classifier, Gradient Boosting, KNN, and Gaussian Naïve Bias.

SVC Classifier gives out the best test accuracy (85%), as against Random Forest Classifier (82%) and GB Classifier (76%). Five-fold cross-validation accuracy from SVC is coming out to be around 82%. We have finally used the SVC classifier to predict any unknown sample.

Results:

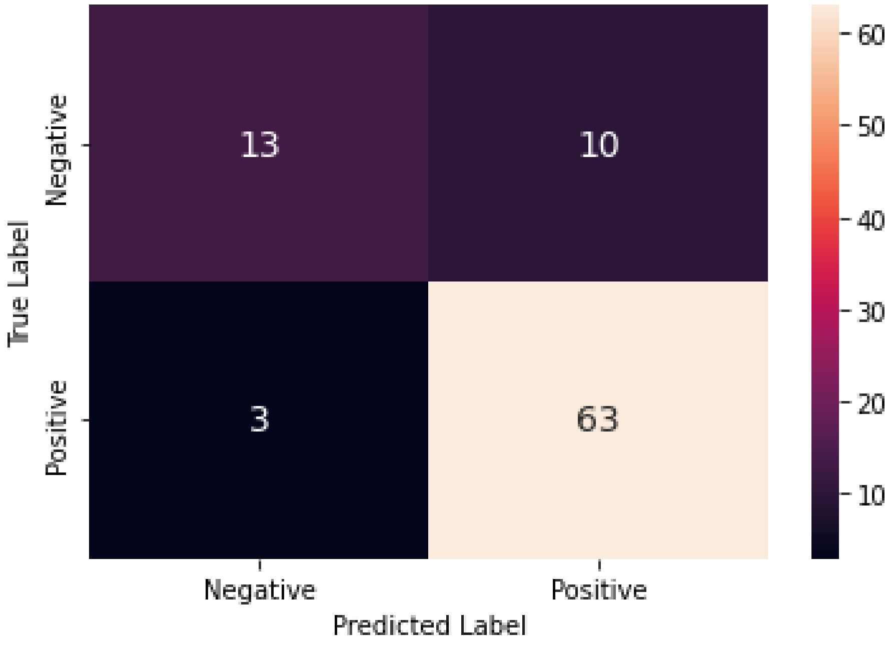
Out of 53 features, only 40 features were passed through ML models after normalizing the data through standard scalar. The whole dataset was split into an 80% training set and a 20% testing set. ML models were trained on a training set. The best test accuracy is achieved on the SCV classifier (85%).

Following **evaluation matrices** were used to evaluate the performance of this model.

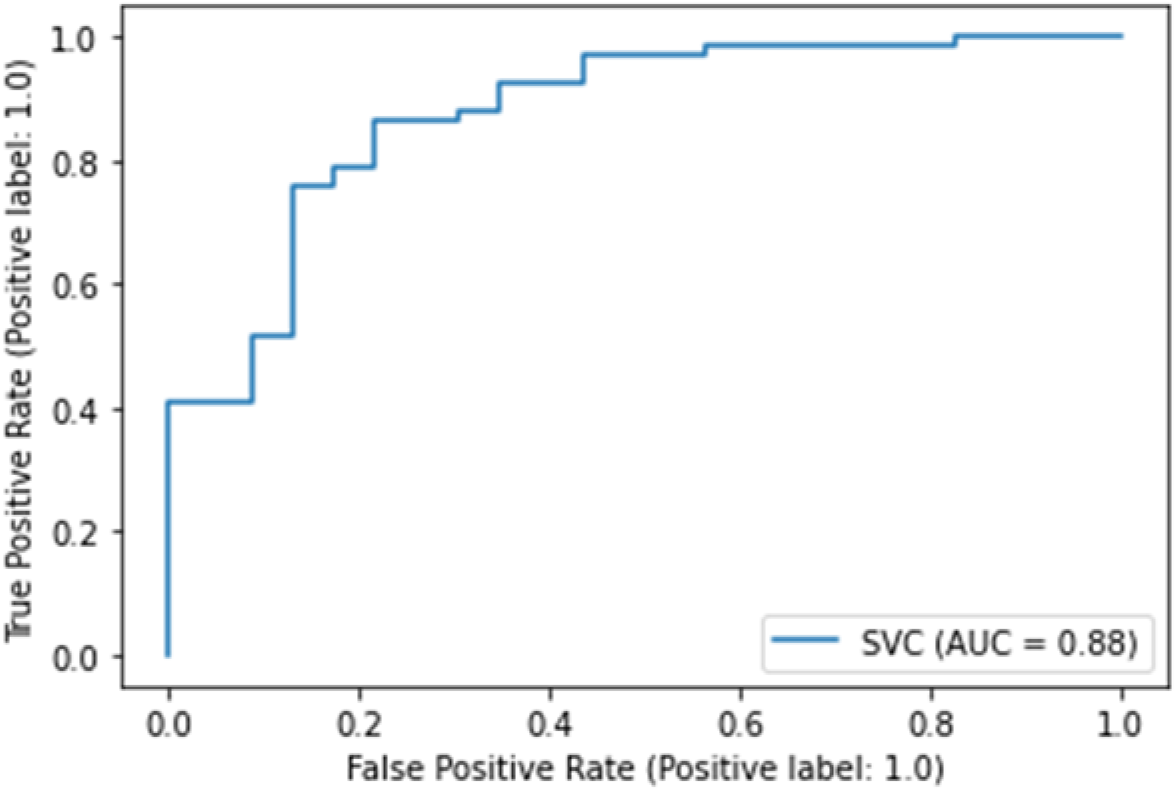
Fivefold cross-validation: Cross-validation is a statistical method that uses different portions of the data to test and train a model on different iterations. On performing five-fold cross-validation on the whole dataset, we achieved 82% accuracy, 0.72 f1 scores, 82% precision, and 0.83 AUC from SVC Classifier.

Confusion Matrix: A confusion matrix visualizes and summarizes the performance of a classification algorithm. It states the following four variables: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN).

Following confusion matrix is obtained on predicting test values from SVC Classifier.



AUC: The area Under the false positive rate and true positive rate tells how much the model can distinguish between classes. The plot of the false positive rate vs. true positive rate for the SVC Classifier is shown below.


True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)
SVC (AUC = 0.88)