Machine learning based approaches for GPCR-Ligand binding prediction

Xuenan Mi
NetID: xmi4
Biophysics and Quantitative Biology

Xin Chen
NetID: xinc6
Biophysics and Quantitative Biology

1 Introduction

G protein-coupled receptors (GPCRs) is an important family of integral membrane proteins. Many common human diseases involve GPCR signaling, including schizophrenia, depression, and hypertension[1]. They regulate diverse physiological processes and have binding sites that are accessible at the cell surface. With binding to an external ligand, a GPCR induces the coupling of a GPCR and G-proteins, that is followed by different signal transduction. Based on the import role of GPCRs, they are widely studied as drug targets. Approximately 34 percent of all FDA-approved drugs target GPCRs[2].

Prediction of drugs / ligands which can bind to specific GPCR is an important aspect to study GPCRs as drug target. Many experimental or bioinformatics approaches to identify GPCR-ligand binding have been proposed. Most of these approaches are through calculating the binding affinity to identify the binding status of the ligands. But these approaches rely on the known structure of the GPCRs or ligands, which limited the identification of binding ligands.

In recent years, structure-based virtual screening methods have been replaced by machine learning techniques. Several features are extracted from proteins and ligands. Common molecular descriptors include molecular fingerprint, SMILE string, 3D voxelization and graph encoding [3]. Many traditional machine learning methods, such as random forest, support vector machine and logistic regression, can be applied to predict the interactions between targets and small molecules.

In this project, we propose to use various machine learning methods to predict GPCR-ligand binding. We will use two baseline machine learning method, like random forest, graph convolution model to predict the binding ligands for GPCRs. And we propose to utilize a novel deep neural network, multitask deep learning architecture, which is a powerful neural network, but not widely used in pharmaceutical and biotech industries.

Multitask network is a framework to leverage useful information contained in multiple related tasks to help improve the generalization performance of all the tasks. Among all GRCRs, we can treat each GPCR as one task, although they are not exactly same, they have similar properties, structure, etc. So the multitask network may help to train a better model than single model from each task. We would like to see whether this multitask deep learning architecture has good and robust performance in prediction of binding ligands.

2 Method

2.1 Data Sources and Preprocessing

From the GPCRdb (https://gpcrdb.org), there are 7 GPCR classes, among them class A is the largest sub-family. In GPCR class A, we downloaded data for 525 GPCRs, 132,354 ligands and 215,684 GPCR-ligand bindings.

Due to the large dataset, training model for this large data will cost a large amount time and memory. We filter the original dataset and keep GPCRs in Chemokine family, which includes 39 receptors, 8296 ligands.

The dataset downloaded from the GPCRdb are all positive samples, but in order to obtain training data for classification, we have to generate artificial negative data using algorithm as below:

For each GPCR X from positive samples,

Step1. Randomly select GPCR Y from one of the positive samples;

Step2. If X and Y have common binding ligands, go to Step1. Otherwise, randomly select one ligand Z among

ligands that bind to Y, and make a negative sample with "X-Z"

Step3. If negative samples already have "X-Z", go to Step1.

Using the above algorithm, we generate 24489 negative samples in total. Because we have 9166 positive samples in original data set, we randomly select the same size negative samples, and then generate the final filter input for the models. We label the positive samples as "1" and label negative samples as "0". And then we will train a model for these categorical data set.

Due to some GPCR has very few binding ligands, we only GPCR which has more than 10 positive samples and 10 negative samples. So finally there are 25 GPCR and 8296 ligands in the input dataset.

2.2 Feature engineer

To obtain features from ligands, we use SMILE (Simplified molecular-input line-entry systems) string encoding[4]. We convert each ligand to their corresponding SMILE format, and then use this SIMILE format as input of our model. The example of SMILE format is as below:

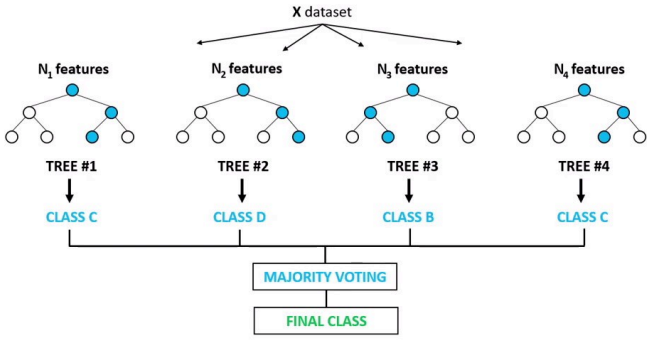
ChEMBL_ID	Smiles
CHEMBL100336	C1CCC(CC1)CN2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=C(C=C5)CC(=O)O)O
CHEMBL100596	C1CCC(CC1)C(C(=O)O)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=CC=C5)O
CHEMBL100958	COC(=O)C(C1CCCCC1)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=CC=C5)O
CHEMBL10097	C1CN(CCC1C2(CCC(=O)NC2=O)C3=CC=CC=C3)CC4=CC=C(C=C4)Cl
CHEMBL101165	CC1=CC(=NC=C1)C2=NC=CC(=C2)C
CHEMBL101333	C1CCC(CC1)C(C(=O)OCC2=CC=CC=C2)N3CC(C(C3)C4=CC=CC=C4)CN5CCC(CC5)(CCCC6=CC=CC=C6)O
CHEMBL101961	C1CN(CCC1(CCCC2=CC=CC=C2)O)CC3CN(CC3C4=CC=CC=C4)C(C5=CC=CC=C5)C(=O)O
CHEMBL102674	C1CN(CCC1(CCCC2=CC=CC=C2)O)CC3CN(CC3C4=CC=CC=C4)CC(=O)O
CHEMBL102675	CC(C(=O)O)N1CC(C(C1)C2=CC=CC=C2)CN3CCC(CC3)(CCCC4=CC=CC=C4)O
CHEMBL102826	C1CCC(CC1)C(C(=O)O)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)CCCC5=CC=CC=C5
CHEMBL10309	C1CN(CCC1C2(CCC(=O)NC2=O)C3=CC=CC=C3)CC4=CC=C(C=C4)Br
CHEMBL103268	C1CCC(CC1)CN2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=CC=C5C(=O)O)O
CHEMBL103343	C1CCC(CC1)C(CO)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=CC=C5)O
CHEMBL103782	C1CCC(CC1)C(C(=O)O)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)C5=CC=C(C=C5)F
CHEMBL103790	C[N+]1=NC(=NN1)C2=CC=C(C=C2)CCCC3(CCN(CC3)CC4CN(CC4C5=CC=CC=C5)CC6CCCCC6)O
CHEMBL103813	C1CCC(CC1)C(CO)N2CC(C(C2)C3=CC=CC=C3)CN4CCC(CC4)(CCCC5=CC=CC=C5)O
CHEMBL104	C1=CC=C(C=C1)C(C2=CC=CC=C2)(C3=CC=CC=C3Cl)N4C=CN=C4
CHEMBL104464	C=CCN(C1CCN(CC1)CCC(CN2N=C(N=N2)C3=CC=CC=C3)C4=CC=CC=C4)C(=O)OCC5=CC=C(C=C5)[N+](=O)[O-]
CHEMBL104474	C=CCN(C1CCN(CC1)CCC(CN2C=NC3=CC=CC=C32)C4=CC=CC=C4)C(=O)OCC5=CC=CC=C5
CHEMBL104628	C1CN(CCC1CCCC2=CC=CC=C2)CC3CN(CC3C4=CC=CC=C4)S(=O)(=O)C5=CC=CS5
CHEMBL104698	CC(CCN1CCC(CC1)N(CC=C)C(=O)OCC2=CC=CC=C2)(CN3C(=O)C(NC3=O)CC4=CNC5=CC=CC=C54)C6=CC=CC=C
CHEMBL104793	C=CCN(C1CCN(CC1)CCC(CN2C(=O)C(NC2=O)CC3=CNC4=CC=CC=C43)C5=CC=CC=C5)C(=O)OCC6=CC=CC=C6
CHEMBL104867	CC(CCN1CCC(CC1)N(CC=C)C(=O)OCC2=CC=CC=C2)(C3NC4=CC=CC=C4S(=O)(=O)N3)C5=CC=CC=C5
CHEMBL105109	C1CN(CCC1C2=CC=CC=C2)CCC(CN3C(=O)C(NC3=O)CC4=CC=CC=C4)C5=CC(=CC=C5)Cl
CHEMBL105460	C=CCN(C1CCN(CC1)CCC(CN2C(=NN=N2)C3=CC=CC=C3)C4=CC=CC=C4)C(=O)OCC5=CC=C(C=C5)[N+](=O)[O-]
CHEMBL105570	CC(CCN1CCC(CC1)N(CC=C)C(=O)OCC2=CC=C(C=C2)[N+](=O)[O-])(CN3C=NC4=CC=CC=C43)C5=CC=CC=C5

3 Baseline method

3.1 Random Forest

In order to convert each ligand to a vector, we use extended connectivity fingerprints (ECFP4) generated from RDkit [5] to featurize each ligand. The ligand can be decomposed into a set of fragments, each fragment extends radially along bonds to neighboring atom and has a unique identifier. We collect the all identifiers for a ligand and hash the identifiers into a fixed length vector to construct the molecular fingerprint. Typically, we convert each ligand into 1024 dimensional vectors.

A random forest classifier consists a large number of individual decision tree classifier on various sub-sample of the dataset. In this method, the random forest contains 100 individual decision trees and build each tree on the whole dataset. And due to the dataset is very sparse, we use "balanced" mode to adjust the weight of each class.



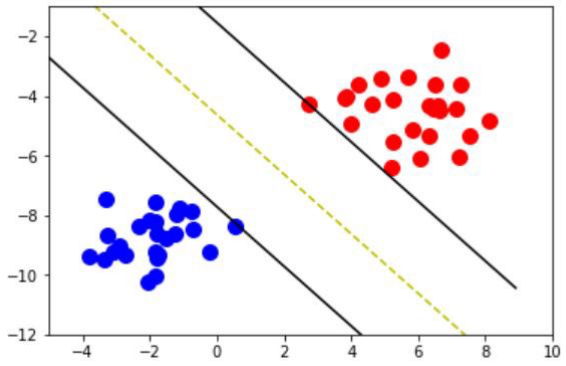
For each GPCR, we split the processed data into training set and testing set based on the proportion 4:1. We train a single model for each GPCR and use the area under the ROC curve (AUC) to evaluate the classification performance. The evaluation results on test dataset are shown in Table1.

We use the random forest classifier from sklearn package.

3.2 Support Vector Machine

The second baseline method we use is support vector machine. The same featurization method, ECFP4, is applied to converting ligand into a 1024-dimensional vector.

The basic principle of SVM is to find a hyperplane in an N-dimensional space which can classifies the data points. There are many possible hyperplanes that can be chosen. Our objective is to find a hyperplane which could maximize the distance between data points of different classes.



Similarly, for each GPCR, we train a SVM model and evaluate the results using AUC score. The results are shown in Table 1.

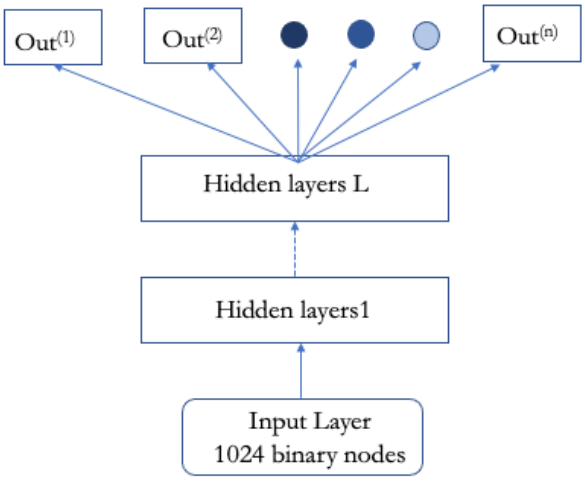
We use the support vector machine classifier from sklearn package

4 Proposed method

The GPCR dataset is a challenging benchmark in ligand prediction which consists of 25 different GPCR where there are only a few binding ligands per target. There are 8,296 ligands in total, some GPCRs only have 10 active ligands. Training a model with such a small number of positive ligands is very challenging.

We would like to apply multitask deep learning architecture to predict the ligand of GPCRs. Multitask deep learning is a powerful network, but, it has not been widely used in the drug discovery. We would like to train a single model which predicts all the different targets at once. When a feature useful for predicting one target, it's often important for predicting other targets. In the DeepChem open-source platform, a multitasks classifier can be used for our project**[6]**.

A standard multitask architecture is as below:



From the input layer, we perform the linear and nonlinear transformation repeatedly. The formula of transformation is

$$x_{i+1} = \sigma\,(W_i x_i + b_i)$$

where x_i is the input to the i-th layer of the network, W_i is the weight matrix and b_i is bias for the i-th layer, σ is nonlinearity. After L transformations, the final layer of the network x_L is fed into a simple nonlinear classifier, softmax, which predict the probability that input x_0 has class j:

$$P\,(y = j \mid x_0) = \frac{e^{(w^j)^T x_L}}{\sum_{m=1}^{M} e^{(w^m)^T x_L}}$$

Where M is the number of classes, in our method, M =2. $w^1, \ldots, w^m$ are weight vectors.

The multitask network have n softmax classifiers, one for each task (GPCR).

5 Results

5.1 Compare multitasks network with baseline methods

In the multitask network, we use one hidden layer with 1000 nodes, and set the dropout probability of layer 0.5. Based on the AUC score for each GPCR (task) from three methods, multitasks network have best prediction results for most

GPCRs (19/25). The evaluation results for three methods is shown in Table 1.

Table 1. AUC score of each GPCR using Random forest, support vector machine and multitask network

GPCR	Random Forest	SVM	Multitasks Network
5ht1d_rat	0.7143	0.8571	**0.9973**
ackr3_human	0.8176	0.9539	**0.9943**
ccr1_human	0.9688	0.9859	**0.9897**
ccr1_mouse	0.6872	0.9353	**0.9815**
ccr2_human	0.9718	**0.9860**	0.9785
ccr2_mouse	0.8434	0.9350	**0.9474**
ccr3_human	0.9859	**0.9926**	0.9923
ccr3_mouse	0.8997	0.8990	**0.9995**
ccr3_rat	0.9996	0.9993	**0.9997**
ccr4_human	0.9244	0.9607	**0.9940**
ccr5_human	0.9759	**0.9807**	0.9765
ccr5_mouse	0.7077	0.9148	**0.9790**
ccr6_human	0.6047	0.8934	**0.9608**
ccr8_human	0.9359	1.0	**1.0**
ccr9_human	0.9773	1.0	**1.0**
ccr12_human	1.0	**1.0**	0.9999
cx3c1_human	0.9115	0.9996	**0.9997**
cxcr1_human	0.8904	0.9425	**0.9858**
cxcr2_human	0.9841	0.9837	**0.9971**
cxcr3_human	0.9722	0.9888	**0.9964**
cxcr3_mouse	0.8738	**0.9945**	0.9940
cxcr3_rat	0.4997	0.7490	**0.9963**
cxcr4_human	0.9743	0.9902	**0.9986**
cxcr5_human	0.9583	**0.9996**	0.9992
q9jly8_rat	0.7137	0.9972	**0.9978**

5.2 Ablation study for layer size

An important parameter in the multitask network is size of each layer. We use the single hidden layer multitask network in Table 1, and that layer contains 1000 nodes, which is default parameter in the multitask network. In addition to 1-hidden layer multitask network, we would like to see how the pyramidal multitask network performs. The pyramidal network means there are more than one hidden layer in the network, and the size of first layer is larger than the size of next layer.

Generally, the first hidden layer is very wide (2000 nodes), which allows for complex, expressive features to be learned. While the last layer is narrow (100 nodes), which will help to limit the parameter specific to each task.

Here, we would like to investigate the sensitivity of our results to the sizes of each layer by running multitask networks with all combination of hidden layer sizes: (1000, 2000, 3000) and (50, 100, 150). The evaluation results for multitask networks with different layer parameter is shown in Table 2.

Table 2. AUC score of each GPCR using multitask network with different hidden layer size

	(1000,50)	(1000,100)	(1000,150)	(2000,50)	(2000,100)	(2000,150)	(3000,50)	(3000,100)	(3000,150)
0									
5ht1d_rat	0.8719	0.9488	0.8976	0.9588	0.95	0.9652	0.6936	0.9259	0.9812
ackr3_human	0.933	0.9843	0.9194	0.9454	0.9788	0.9768	0.8401	0.8915	0.9769
ccr1_human	0.9819	0.9841	0.9887	0.9673	0.9877	0.9858	0.8392	0.9679	0.943
ccr1_mouse	0.9848	0.987	0.9896	0.9896	0.9961	0.9782	0.9331	0.9651	0.9941
ccr2_human	0.9633	0.9392	0.9555	0.9572	0.95	0.9647	0.9372	0.9729	0.9595
ccr2_mouse	0.9367	0.9707	0.9668	0.9569	0.9714	0.9752	0.905	0.9675	0.9611
ccr3_human	0.9855	0.9811	0.9893	0.9915	0.9879	0.9909	0.9725	0.9839	0.9903
ccr3_mouse	0.9977	0.9994	0.9961	0.9886	0.9954	0.9981	0.9836	0.9917	0.993
ccr3_rat	0.9995	0.9997	0.9997	1.0	0.9997	0.9998	0.9819	0.9997	0.9903
ccr4_human	0.9882	0.9879	0.9878	0.9467	0.9583	0.9895	0.9475	0.9873	0.9804
ccr5_human	0.948	0.956	0.9582	0.9537	0.9615	0.9729	0.8855	0.957	0.9426
ccr5_mouse	0.9825	0.9639	0.9765	0.9313	0.9739	0.9652	0.9272	0.9645	0.956
ccr6_human	0.9505	0.9543	0.9314	0.9184	0.9198	0.9487	0.9656	0.9443	0.8879
ccr8_human	0.9999	1.0	1.0	0.9997	1.0	0.9999	0.978	1.0	0.9993
ccr9_human	1.0	1.0	1.0	0.9999	1.0	1.0	0.9194	0.9952	0.9995
ccrl2_human	0.9998	0.9999	1.0	0.9997	0.9999	0.9999	0.9808	0.9998	1.0
cx3c1_human	0.9997	0.9995	0.9996	0.9997	0.9997	0.9997	0.999	0.9993	0.9995
cxcr1_human	0.9789	0.9759	0.9782	0.9426	0.9631	0.976	0.9742	0.9847	0.9646
cxcr2_human	0.9949	0.9954	0.9932	0.9896	0.9877	0.9964	0.9838	0.9912	0.9949
cxcr3_human	0.9959	0.9974	0.9959	0.9875	0.9966	0.9978	0.9827	0.9967	0.9968
cxcr3_mouse	0.9744	0.9745	0.9838	0.9828	0.9858	0.9849	0.8943	0.9798	0.9621
cxcr3_rat	0.9967	0.9946	0.9949	0.9683	0.994	0.9952	0.8676	0.9922	0.9644
cxcr4_human	0.9988	0.9984	0.9988	0.995	0.9962	0.9984	0.9918	0.9979	0.9976
cxcr5_human	0.9964	0.9918	0.9918	0.9808	0.9702	0.9935	0.9951	0.9755	0.9711
q9jly8_rat	0.9971	0.9971	0.9973	0.9903	0.998	0.9974	0.8967	0.9966	0.9957

In order to investigate the sensitivity of our results to the sizes of each layer, we calculate the means and medians of each architecture's AUC score and results are shown in Table 3. Based on the results, we notice means and medians are shifted by ~0.01 across all architectures except the network with layer size (3000, 50).

Table 3. Means and medians of all architectures.

	(1000,50)	(1000,100)	(1000,150)	(2000,50)	(2000,100)
Mean	0.9782	0.9832	0.9796	0.9737	0.9809
Median	0.9882	0.9879	0.9896	0.9828	0.9877
	(2000,150)	(3000,50)	(3000,100)	(3000,150)	
Mean	0.9860	0.9310	0.9771	0.9761	
Median	0.9909	0.9475	0.9847	0.9812	

6 Discussion

In this work, we investigate the performance of multitasks network in prediction of GPCR-ligand interaction. Compared to two based machine learning methods, random forest and support vector machine, in most GPCRs (19/25), multitask network has better prediction result.

The multitasks framework is inspired by human learning activities where people always apply the knowledge learned from previous tasks to help learn a new task. Several reasons about input dataset may also result in good performance of multitask network in prediction of GPCR-ligands binding. First, each task in GPCR dataset has limited samples, which is insufficient to obtain a good model. Multitasks network could overcome this issue, it learns multitasks jointly, which can lead to much performance improvement compared with learning them individually. Second, the tasks in GPCR dataset are not totally independent, they share similar physical properties, structure and function. It's reasonable we could obtain some generalization information across all tasks.

In the ablation study, we investigate the sensitivity of multitask framework on size of each hidden layer. We compare the prediction performance among 9 different combination of layer sizes, first layer size can be 1000 /2000 /3000, second layer size can be 50/ 100/ 150. The first hidden layer size is often large, which allows for complex features of input. The second hidden layer is often narrow, which helps to limit the parameters specific to each task. Based on the results of all combination of layer size, we find most architectures have robust prediction performance. Therefore, we think this multitask frame is not very sensitive on size of each hidden layer.

For future work, we can focus on two directions. One is small molecule featurization. Regardless of single network or multitask network, they all heavily depend on the design of the feature engineering pipeline. In our project, we use extended connectivity fingerprints (ECFP4) method to describe each ligand, but there exist many others, like graph convolutional network. Graph-structural information will be passed, which is benefit for feature extraction. Another possible future direction, we may consider few shot learning and meta learning framework, which is also reasonable solution for problems with limited data of each task.

7 Contributions

In this project, Xuenan Mi works on cleaning up the original dataset, and random forest method. Xin Chen works on feature engineering and support vector machine method. Xuenan Mi and Xin Chen works together on the proposed method: multitask network and ablation study.

REFERENCES

[1] Garland, S. L. Are GPCRs still a source of new targets? *J Biomol Screen*, 18, 9 (Oct 2013), 947-966.

[2] Hauser, A. S., Attwood, M. M., Rask-Andersen, M., Schioth, H. B. and Gloriam, D. E. Trends in GPCR drug discovery: new agents, targets and indications. *Nat Rev Drug Discov*, 16, 12 (Dec 2017), 829-842.

[3] Raschka, S. and Kaufman, B. Machine learning and AI-based approaches for bioactive ligand discovery and GPCR-ligand recognition. *Methods*, 180 (Aug 1 2020), 89-110.

[4] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information & Computer Sciences*, 28, 1 (1988), 31-35.

[5] Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *J Chem Inf Model*, 50, 5 (May 24 2010), 742-754.

[6] Ramsundar, B., Liu, B., Wu, Z., Verras, A., Tudor, M., Sheridan, R. P. and Pande, V. Is Multitask Deep Learning Practical for Pharma? *J Chem Inf Model*, 57, 8 (Aug 28 2017), 2068-2076.