Mathematical Biosciences and Engineering

MBE, 20 (2): 2588–2608.
DOI: 10.3934/mbe.2023121
Received: 25 August 2022
Revised: 31 October 2022
Accepted: 07 November 2022
Published: 25 November 2022

http://www.aimspress.com/journal/MBE

Research article

Multi-source transfer learning with Graph Neural Network for excellent modelling the bioactivities of ligands targeting orphan G protein-coupled receptors

Shizhen Huang[1], ShaoDong Zheng[1,2] and Ruiqi Chen[2,*]

[1] College of Physics and Information Engineering, Fuzhou University, Fuzhou 350116, China
[2] VeriMake Innovation Lab, Nanjing Renmian Integrated Circuit Co., Ltd., Nanjing 210088, China

* **Correspondence:** Email: rickychen@verimake.com.

Abstract: G protein-coupled receptors (GPCRs) have been the targets for more than 40% of the currently approved drugs. Although neural networks can effectively improve the accuracy of prediction with the biological activity, the result is undesirable in the limited orphan GPCRs (oGPCRs) datasets. To this end, we proposed Multi-source Transfer Learning with Graph Neural Network, called MSTL-GNN, to bridge this gap. Firstly, there are three ideal sources of data for transfer learning, oGPCRs, experimentally validated GPCRs, and invalidated GPCRs similar to the former one. Secondly, the SIMLEs format GPCRs convert to graphics, and they can be the input of Graph Neural Network (GNN) and ensemble learning for improving prediction accuracy. Finally, our experiments show that MSTL-GNN remarkably improves the prediction of GPCRs ligand activity value compared with previous studies. On average, the two evaluation indexes we adopted, R2 and Root-mean-square deviation (RMSE). Compared with the state-of-the-art work MSTL-GNN increased up to 67.13% and 17.22%, respectively. The effectiveness of MSTL-GNN in the field of GPCR Drug discovery with limited data also paves the way for other similar application scenarios.

Keywords: G protein-coupled receptors (GPCRs); multi-source transfer learning; Graph Neural Network; biological activity

1. Introduction

G protein-coupled receptors (GPCRs) are the most successful clinical drug targets, primarily due to their substantial involvement in human pathophysiology and their pharmacological tractability [1]. More than 40% of current drugs are GPCR-targeted, and their market value exceeds 1.3 billion dollars per drug [2]. Bioactivity value is usually used to measure the potential of drug discovery, especially the endogenous ligands [3]. There are now more than 140 kinds of GPCRs with unknown endogenous ligands not being adequately studied, called orphan GPCRs (oGPCRs) [4,5]. Strongly correlated with oGPCRs, many GPCR members are found to be widely distributed in various species. And many of them are found to have abundant experimentally validated ligand entries with bioactivity values, especially in human non-olfactory GPCRs [6,7]. This suggests that although the endogenous ligand or signaling pathway of the oGPCR is still unclear, the drug development process for the receptor can still be carried out and advanced.

Computer-aided drug design has played an important role in drug discovery in the past decades, based on oGPCR and their bioactivity values [8]. The most representative approach is structure-based drug discovery. It completes the computation by simulating the physical interaction between the target and small organic compounds, and then calculates the ligand bioactivity value [9]. However, this method has not been applied on a large scale because there are limited atomic resolution structures available for direct calculation. Molecular dynamics simulation is used to calculate the bioactive value of ligands among similar structures [10]. However, this method is hindered by low-sequence similarity in the helical regions due to the limitation that the operation basis requires high-quality models of GPCR structures. Machine learning (ML) methods and deep learning (DL) methods have been widely used in the study of drug-target prediction based on GPCR ligand bioactivity value [11]. On account of the strong predictive power of deep learning methods in recent years, various deep learning models are applied to drug repositioning, including multi-layer perceptron, deep belief network stacked auto-encoder, etc [12].

Although machine learning and deep learning are widely used in GPCR-based bio-activity value prediction. However, due to the limitation of machine learning and deep learning, which require a large amount of data, it is difficult for the model to achieve excellent results when there are insufficient samples [13], especially during the COVID-19 outbreak [14,15]. Therefore, machine learning models that achieve excellent results under common GPCR are not satisfactory under oGPCR. In order to improve the accuracy of the oGPCRs ligand bioactivity prediction with limited data, we proposed MSTL-GNN in this paper. This method utilizes multi-source transfer learning to solve the problem of insufficient data under a small-sized sample. The adoption of the graph neural network model is to generate molecular fingerprints and the generation of weighted molecular fingerprints. Finally, the weighted molecular fingerprint is used as input to improve the ability of bioactivity value prediction by ensemble learning.

The main contributions of this paper are as follows:

1. The MSTL-GNN method is proposed to effectively obtain the graph features of oGPCRs under a small data sample;

2. Based on the graph features, ensemble learning is used to predict the biological ligand activity.

3. We comprehensively compared the proposed method and previous methods based on typical 12 GPCRs data according to their family group, and the experimental results show that the proposed method effectively improves the prediction performance.

4. Experimental results show that, compared with the state-of-the-art work (WDL-RF), MSTL-GNN increased up to 67.13% and 17.22%, respectively.

The paper is structured as follows: Section 2 introduces related work. Section 3 explains our methodology and materials used. The experiment and evaluation are discussed in Section 4. Finally, section 5 concludes the research.

2. Related work & motivation

2.1. Related work

In recent years, different kinds of algorithms have been proposed to improve the bioactivate prediction accuracy. Tang et al. [16] proposed a multi-decision forest model to predict the bioactivate value for the receptor from the GPCR dataset. Lounkine et al. [17] proposed a Bayesian model. It combines molecular similarity and structure similarity to predict the bioactivates. Carpenter et al. [18] used a deep-learning-based virtual algorithm to predict the bioactivate value. Moreover, their result shows the superiority of the deep learning model. Wu et al. [19] proposed WDL-RF based on the deep learning weighted deep learning and random forest (WDL-RF) method, and input molecular fingerprints and models with variable sizes and shapes. Excellent results were obtained under R2 and RMSE Root-mean-square deviation (RMSE) evaluation parameters. Hu et al. [20] developed a deep learning method to predict biological ligand activity values through an end-to-end encode-decode model combined with CNN Convolutional Neural Networks. More recently, the Stokes team [21] has designed a directed message-passing approach for activity prediction to fit compounds.

2.2.Motivation

Previous studies have been able to perform well on the prediction of the BAV of ligand molecules binding with drug targets. However, insufficient information is available for the drug targets of oGPCR, resulting from the lack of 3D structures of some drug targets and the insufficient sample size of known active ligand molecules. Moreover, the common machine learning strategy uses existing software to compute, based on hand-crafted features with fixed lengths (e.g., molecular fingerprints, molecular descriptors), and then conducts standard machine learning methods to construct prediction models. However, the extraction of hand-crafted features requires researchers to master the relevant domain knowledge, thus limiting the popularity of the method. Also, molecular fingerprints cannot be flexibly adapted to diverse tasks.

Therefore, the remaining challenges can be concluded as (1) Effective selection of ligand molecules in the case of small samples. In order to meet this demand, the first is to extract suitable molecular characteristics of the ligand and then to meet the accuracy requirement of prediction of the BAV of the ligand molecule binding with drug target even in small sample scenarios. (2) Generating good molecular fingerprint features for various tasks to achieve end-to-end prediction.

3. Method

3.1. Overall framework

The SMILES [22] string serves as input to the entire algorithm and the output is the biological activity value of the ligand molecule bound to the receptor. Biological activity is usually determined by ED50 (Half of the effective quantity), IC50 (Half inhibition concentration), Ka (Binding constant, affinity constant), Ki (Inhibition constant), to measure [23], the overall MSTL-GNN architecture is shown in Figure 1.

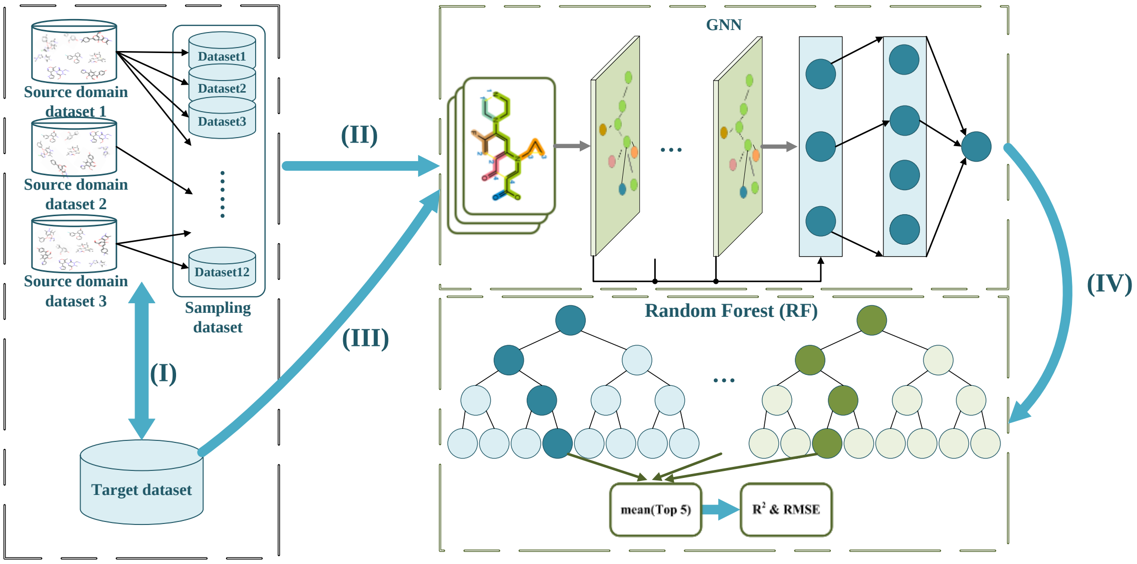


Figure 1. Schematic diagram of the architecture.

The specific experimental steps are as follows:

(I) Generated based on datasets with drop-back sampling. New drug targets can often be used to discover homologous or similar target proteins and even multiple ones that act more easily with similar compounds and tend to interact in more similar ways and mechanisms. Therefore, we utilized the information on the abundant compound samples of these target proteins to help establish virtual screening models for the uninformative drug targets of these samples. After determining the drug target, its dataset was obtained, and the homologous drug target was found to obtain its dataset. First, the GPCRs dataset was obtained from the Uniprot database. The dataset contains the ligand SMILES and the biological activity values of the action. Meanwhile, the relevant homologous GPCRs were found and the homologous GPCRs dataset was obtained according to the family where GPCRs were located. Finally, the homologous GPCRS dataset was put back to obtain the sampled homologous GPCRs dataset.

(II) Construction of a virtual screening model based on a graph neural network. SMILES represents the molecular structure as arbitrary manual feature graphs, which is essentially a two-dimensional picture describing the molecular structure. SMILES serves as an input to the graph neural network, and the output is the bioactivity value for the ligand upon binding to GPCRs. Specifically, the graph neural network virtual screening model can be divided into three stages: generation of molecular fingerprint, generation of weighted molecular fingerprint, and prediction of biological activity value. The generation stage of molecular fingerprints does the operation as follows: First, each atom in SMILES is added and calculated to obtain all the atomic information. Another convolution operation of the atom to obtain information on the edges between the atom, and then another convolution operation to eliminate the effect of the extreme. Finally, all the information was summed to obtain the molecular fingerprints of the entire molecule. The generation stage of the weighted molecular fingerprint is as follows: the molecular fingerprint generated in the previous step obtains the weighted molecular fingerprint through a layer of weighted weight. The biological activity value prediction stage is done as follows: pass the weighted molecular fingerprints through two full-connected layers to obtain their predicted biological activity values.

(III) Construction of a multi-source transfer learning model based on parameter migration. The weight matrix was first obtained by entering the source domain data into the graph neural network model. The data of the target domain is then input into the graph neural network, and the weight matrix obtained from the source domain data trained in the model is migrated to the target domain as the initial value of the target domain feature matrix, and then retrained to obtain the new weighted molecular fingerprint of the target domain data.

(IV) Prediction of ligand biological activity values based on integrated learning. First, the different source domains were obtained by training for the new weighted molecular fingerprints. Weighted molecular fingerprints serve as input to the random forest. The result of the random forest output is the predicted biological activity values. Then RMSE and R2 values were used to evaluate the predicted performance and select the 5 best groups from the biological activity values. These 5 sets of activity values were then averaged as the final predicted biological activity values. Finally, activity values were validated in test set T and used with R2 evaluation of the performance was predictive with RMSE.

3.2. MSTL-GNN algorithm

The multi-Source Transfer-Graph Neural Network algorithm is as follows: first, generate a homologous data set by putting back to random sampling. The parameters were then trained in the graph neural network using the homology dataset to obtain weight parameters. The parameters were then migrated to the trained model of the target domain dataset and the model was trained to obtain a weighted molecular fingerprint. Finally, the weighted molecular fingerprints were used as an input to the random forest. The output of the top 5 was averaged as the final predicted value.

The specific steps are as follows:

(I) Generate homologous datasets. According to the family of GPCRs, the relevant homologous GPCRs were found and obtained dataset. The dataset contains the name of GPCRs, SMILES of ligand molecules, and biological activity values for ligand molecules and GPCRs binding. Each dataset had four homologous datasets that sampled the four homologous datasets, set the sampling ratio to 0.5, and randomly repeated sampling three times. These yielded 12 homologous datasets that serve as those required for homologous migration operations. The purpose of this step is to prepare the second step for the training model.

(II) Training of the graph neural network. The trained graph neural networks are used to generate weighted molecular fingerprints while acquiring weight parameters in preparation for parameter migration. The SMILES in the dataset obtained in the first step cannot be used directly as an input to the graph neural network, which requires the SMILES string to be converted into a two-dimensional graph via the software RDKit so that the ligand molecule SMILES to a 2D graph can be converted as an input to the graph neural network. For example, the SMILES of dopamine is 'C1=CC (=C (C =C1CCN) O) O', whose corresponding 2D graph is shown in Figure 3.2. Among them, the SMILES string 'C1=CC (=C (C=C1CCN) O) O' represents a graph, '-' represents a single key, '=' represents a double key, '#' represents three keys, ':' represents an aromatic key, atoms or ions separated with '.', represents disconnected molecules structures. Each point in Figure 2 (a) represents an atom or ion in the SMILES string, such as a carbon atom (C), or an oxygen atom (O), and each edge represents the chemical bond formed by atomic bonding between atoms or ions and whether the molecule is a closed-loop structure. The software RDKit was used to generate the graph beginning to train the graph neural network model. The algorithm can be divided into three stages: molecular fingerprint generation weighted molecular fingerprint generation, and predicted output prediction. To visually show the graph neural network training process in the second.

In Figure 2 (b), the training procedure is as follows: First, the generation of molecular fingerprints contains L different individual units. A SMILES molecule represents the graph after one accumulation, convolution, seeks the mean accumulation as the input value, and then inputs to the next same computation module. Such a computation module has the number of L in total. Each unit consists of convolutional and involvement layers, operated as follows for each unit.

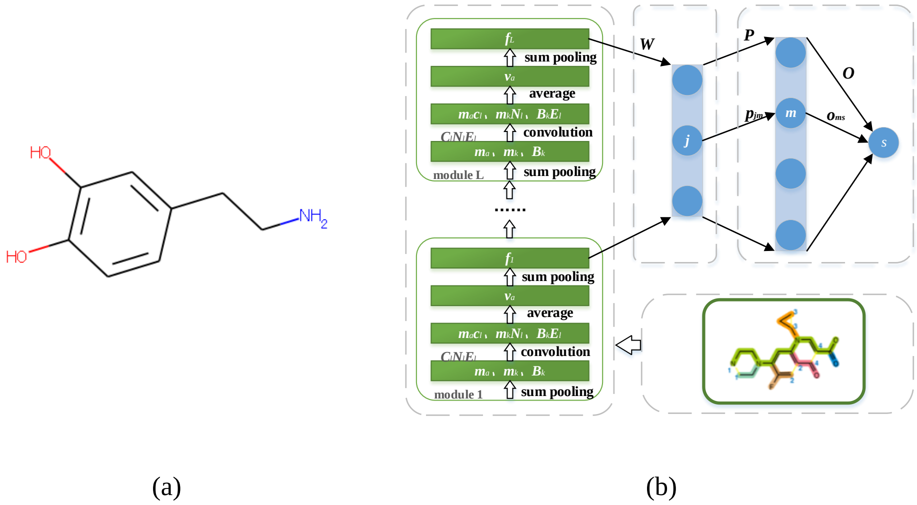


Figure 2. Schematic diagram of the MSTL-GNN architecture. (a) Dopamine 2D vision. (b) Schematic diagram of the GNN.

Enter the SMILES string of the *i*th ligand molecule, set to contain an atom, after RDKit treatment to obtain the attribute vector of each atom$m_j \in R^A$, where$j = 1,\dots,N_a$, A is the dimension of the attribute vector of the atom. In the *l*th unit, m_a represents the property vector of atom a. The atoms and their surrounding neighbor atom *t* are expressed as:

$$m_a = Rdkit(a) \tag{1}$$

$$m_t = Rdkit(t) \tag{2}$$

In the *l*th unit, the jth atom is output into $z_j \in R^{1*B}$via the first convolutional layer:

$$z_j = \sigma\left(m_j W_l^C + \frac{1}{|N_j|}\sum_{t\in N_j} m_t W_l^N + \frac{1}{|N_j|}\sum_{j\in N_j} A_{ij} W_l^E + b\right) \tag{3}$$

The dimension of the attribute vector in each module is A, fingerprint length is B, $m_j \in \mathbb{R}^{1*A}$is the attribute vector of the jth atom; N_j is the nearest neighbor set of the atoms; $A_{ij} \in \mathbb{R}^{1*6}$is related to the keys connecting the atom. Key information contains single key, double key, triple key or aromatic key, whether the key is conjugated, whether the key is part of the ring;$W_l^C \in R^{A*B}$, $W_l^N \in R^{A*B}$, $W_l^E \in R^{6*B}$ are the weight matrix; $b \in R^B$ is the bias vector. $\sigma(x) = \begin{cases} 0, & x < 0 \\ 1, x \geq 0 \end{cases}$

Then, manipulate the molecule with sum pooling to obtain the molecular fingerprint of each module: f

$$f = f + z_j \tag{4}$$

Among them, $f \in R^{1*B}$.

The molecular fingerprint f (recorded asf_l) obtained from each unit passes through the weight layer$W \in R^{B*B}$ to generate a weighted molecular fingerprint $F \in R^B$as:

$$F = \sigma(\sum_{l=1}^{L} W \cdot f_l) \tag{5}$$

Among these $f_l$is the molecular fingerprint of the l_{th} unit.

After obtaining the weighted molecular fingerprint F, the predicted activity value $\hat{y}_i$ of ligand molecules x_i was obtained by two fully ligated layers as:

$$z_m = \sigma(\sum_j p_{jm} F_j) \tag{6}$$

$$\hat{y}_i = \sigma(\sum_m o_{ms}\, z_m) \tag{7}$$

In the middle layer, there are neurons in the number of M, where p_{jm} is the weight of neuron $m$connected to the neuron j, o_{ms} is the weight of neuron m connected to the neuron s the value of the first dimension; F_j is the result of F in the j_{th} dimension; $\hat{y}_i$ is the i_{th} of the predicted ligand to the drug target.

Then, based on the predicted activity $\hat{y}_i$ of the i_{th} ligand binding to the drug target, the minimum quantization error function was introduced, continuously iterating parameters θ with the objective function of:

$$\min_{\theta} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \frac{\lambda}{2n}\sum_{\theta}\theta^2 \tag{8}$$

Among them, $n$is for the number of ligand molecules, $y_i$is for the real biological activity value of the i_{th} ligand molecule, $\hat{y}_i$ is for the i_{th} ligand molecule to predict biological activity value, θ is for the neural network model parameter, λ is for the regularization term coefficient, in order to balance the regularization term and loss function. When λ is large, the target function focus more on the regularization term, making the loss value cannot be minimized, resulting in underfitting. When λ is too small, the objective function focuses more on the loss function term, resulting in overfitting. In order to best match the loss function term to the regularization term, the regularization term coefficient λ is needed to help reach balance.

For function (8), the graph neural network model parameters θ were updated using the Adam algorithm. Adam is a stochastic optimization of a ladder degree-based method to calculate the adaptive learning rate for each parameter. The function (8) is abbreviated as$f(\theta)$, $g_t = \Delta_\theta f_t(\theta)$ is the gradient of the θparameters of the target function$f(\theta)$ at round tth iteration. The Ad am algorithm calculates the first-order matrix estimate of the gradient by using the decay coefficient$\beta_1, \beta_2 \in [0,1)$, with the second-order$v_t$ matrixm_t being:

$$m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \tag{9}$$

$$v_t = \beta_1 * v_{t-1} + (1 - \beta_2) * g_t^2 \tag{10}$$

in which, g_t^2 is the second derivative$g_t \odot g_t$.

Since the first estimate $m_t$and second moment estimatev_t are initialized to a zero vector, the impulse estimates are also biased towards zero (especially at the initial step length or decay coefficient comparison hours). The problem of momentum estimation bias towards zero is solved by calculating bias correction $\hat{m}_t$ and $\hat{v}_t$.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{11}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{12}$$

in which the sum of β_1^t andβ_2^t is the t power of the sum ofβ_1 and β_2.

Finally, the model parameters θ are updated with the following iterative steps:

$$\theta_t = \theta_{t-1} - \frac{\lambda}{\sqrt{\hat{v}_t} + \varepsilon} m_t \tag{13}$$

Among these, λ is step length (step size), and parameters $\beta_1, \beta_2, \varepsilon$ are used in Equations (9) to (13). Set the default parameters in Adam: λ =0.01, β_1=0.9, ε =10-8.

Based on the Ad am algorithm, mini-batch serves as the optimization strategy. That is, 100 samples were randomly selected in each round of update iteration, with a maximum number of iterations of 250.

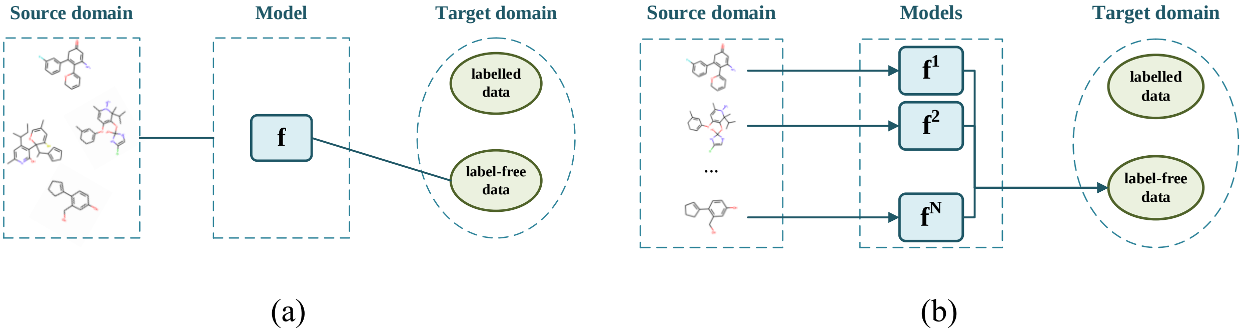


Figure 3. Schematic diagrams of the different multi-source transfer learning method. (a) Multi-source transfer learning model I. (b) Multi-source transfer learning model II.

Single-source domain transfer learning aids the construction of the target domain model with a single source domain, and the effect of the results largely depends on the selection of single-source domains. Multi-source transfer learning is proposed to reduce the impact of individual source domains and enhance generalization. Often in practice, multiple different source domains can be found, but migration using a single source domain only leads to a waste of resources. The most common multi-source transfer learning method is to add all the source domains as one source (Figure 3(a)). However, the drawback of this approach is that the differences between the different source domains are ignored. Another approach is to train individual source domain classification regression models and then combine these models to select different weight parameters for each model (Figure 3(b)). In MSTL-GNN, the weight for each model is set as 1 because the source domain improves similarly to the target domain.

The 12 source domain datasets were input into the GNN model trained by formula (1)–(13), setting the number of training rounds to 100 rounds empirically and then adjusting the mini-batch and step lengths to achieve the optimal parameters, the currently trained source domain model and parameter weights were extracted.

(III) Before training the target domain, assign the parameter weights obtained from the previous source domain model to the corresponding target domain, modify the initialization operation of the previous step, and replace the initialization, further continue to train in the target domain model to obtain the corresponding weighted molecular fingerprints. Training 12 source domain models will get 12 weighted molecular fingerprints.

(IV) Random Forest, an integrated learning method that can be used on classification, regression, and other issues, is based on supervised learning algorithms consisting of multiple decision trees. During the training phase, a large number of decision trees are constructed that are often irrelevant and stochastic. In the output stage, the different decision trees can be voted to determine the final prediction results. Each decision tree has its results, and multiple decision trees help build the overall model. Thus, random forests often tend to perform better than decision trees.

The weighted molecular fingerprints obtained from the previous step were used as input to the random forest, and the corresponding predicted values were output. The random forest was constructed as follows: The molecular fingerprints obtained in the third step were first used as the training set, where the i_{th} ligand molecular weighted molecular fingerprint is the true activity value of the i_{th} ligand and GPCR binding, with samples of in all. Then operate as follows: Select from the random and put back sampling samples.

A decision tree was then constructed with the samples, best segmented at each node in a randomly selected feature subset, allowing the tree to grow continuously.

Repeat the above steps to guide the growth of individual trees to completion.

Finally, the constructed random forest to make the prediction. Each ligand molecule can gain M predictions, which were averaged as the final result.

(V) Four source domains were selected in this experiment, without knowing which source domain of the four source domains, which source domains showed a poor migration effect. In order to minimize the influence of source domain selection, using the integrated learning method, integrated learning can also combine each learner to form powerful learners, improving the performance of prediction, predicting the activity value of ligand molecules and GPCR binding accurately, improving the efficiency of lead compound selection, and improving the efficiency of drug development. In this experiment, the dataset of the source domain was sampled. In the final prediction phase, according to R2And R MSE, the five best-predicted values are selected and averaged as the final prediction.

3.3. A flow framework for ligand biological activity prediction based on multi-source migration maps

The following procedure shows the ligand bioactivity prediction algorithm based on multisource graph neural networks:

Algorithm: Multi-source transfer learning method based on graph neural network

```
Enter:
    Source domain homologous GPCRs sample S and target domain sample T
Training stage:
1   Source domain datasets were randomly sampled
2   Initialization:$C_l, N_l, E_l, b_l (l \in [1, L]), W, P, O$; $f \leftarrow 0, F \leftarrow 0$
3   Repeat (the 4-14 operation until the convergence occurs)
4       A subset of S was randomly selected from the dataset
5           for$(x_i, y_i) \in S$
6               for$a \in x_i$
```

```
            $z_j = \sigma\left(m_j W_l^C + \frac{1}{|N_j|}\sum_{t\in N_j} m_t W_l^N + \frac{1}{|N_j|}\sum_{j\in N_j} A_{ij} W_l^E + b\right)$
            $f = f + z_j$
        end for
        $F = \sigma(\sum_{l=1}^{L} W \cdot f_l)$
        $\hat{y}_i = \sigma(\sum_m o_{ms} z_m)$
        $\min_{\theta} \frac{1}{2n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \frac{\lambda}{2n}\sum_{\theta}\theta^2$
        Update the model parameters$\theta$ using Adam
    end for
  get the weight matrix $W$and set the initial value
  Enter the target domain dataset
  A random forest regression predicting
  Top 5 from domain dataset
Output stage:
  Calculating RMSE and $R^2$
```

4. Experimental and analysis

The experiment was divided into two parts. The first part explores explore the influence of different training samples on this method. We specifically explore the effect of fallback sampling and no-fallback sampling, target domain samples, and training sample size on performance. Another part is the performance comparison of this method. Results with past relevant methods were compared under the same datasets, and the relevant results were analyzed.

4.1. Evaluation indicators

This experiment is a regression problem of activity value, using the common correlation coefficient R^2 And the mean-squared error is RMSE [24–26]. The specific formula is as follows:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{14}$$

y_i represents the actual value; $\hat{y}_i$ represents the predicted value; $n$represents the number of samples. The smaller the value of RMSE represents, the smaller error there is in the prediction.

$$R^2 = \frac{(\sum_{i=1}^{n}(y_i - \bar{y})\,(\hat{y}_i - \bar{\hat{y}}))^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2(\hat{y}_i - \bar{\hat{y}})^2} \tag{15}$$

y_i: activity value of the ith ligand and target binding; $\hat{y}_i$:predicted activity value of the ith ligand and target binding; $\bar{y}$: mean activity value of ligand and target binding; $\bar{\hat{y}}$: mean value of predicted activity value of ligand and target binding; $n$representing the number of samples. The larger the R2 is, the more stable the representative model, and so a more reliable the model.

4.2. Introduction of the experimental dataset

The experimental dataset in this paper comes from the Internet open-source database. One from the Uniprot [27] database contains a total of 3,052 GPCRs data. The 825 GPCRs, related to humans were downloaded from (http://zhanglab.ccmb.med.umich.edu/GLASS/) [28]. Then, download the GLASS file, which includes 519,051 ligand molecular data interacting with GPCRs, from which 12 groups of fewer ligand molecules were selected and 4 homologous GPCRs. for each group of GPCRs based on the family where GPCRs belong.

As shown in Appendix Table a, the GPCRs dataset was classified into 12 groups by families, with a larger number of numbered A–L ligand molecules as the source domain dataset (Source Database, SD), the remaining group being the target domain dataset (Target Database, TD), the dataset included group name, type, UniPort ID name, number of ligand molecules, and the GPCR subfamily. Each GPCR dataset includes the GPCR name, the SMILES of the ligand molecules, and the biological activity values of the GPCR versus ligand action, as shown in Table 1.

Table 1. The A group target domain dataset.

GPCR	SMILES	Activity values
Q9Y2T5	C1C(OC2=CC=CC(=C21)C3=CC(=CC=C3)C(=O)NCCO)CC4=CC(=CC=C4)C(F)(F)F	-2.81
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(C=CC(=C3S2)C4=CC(=CC=C4)C(=O)NCC(=O)N)	-1.92
Q9Y2T5	C1=CC(=CC(=C1)C(=O)NCCO)C2=NC(=CC=C2)OCCC3=CC(=CC(=C3)Cl)F	-2.54
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(O2)C(=CC=C3)C4=CC(=CC=C4)C(=O)NCCO	-1.542
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CCOC2=CC=CC(=N2)C3=CC(=CC=C3)C(=O)NCCO	-2.81
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(S2)C(=CC=C3)C4=CC(=CC=C4)C(=O)NCCO	-1.569
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(S2)C(=CC=C3)C4=CC(=CC=C4)C(=O)NCC(=O)N	-1.74
Q9Y2T5	C1C(OC2=C(C=CC=C21)C3=CC(=CC=C3)C(=O)NCCO)CC4=CC(=CC=C4)C(F)(F)F	-2.32
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CN2C=C3C=CC=C(C3=N2)C4=CC(=CC=C4)C(=O)NCCO	-1.811
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(C=CC=C3S2)C4=CC(=CC=C4)C(=O)NCCO	-1.58
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CN2C=C3C(=N2)C=CC=C3C4=CC(=CC=C4)C(=O)NCCO	-1.591
Q9Y2T5	CC1=C(OC2=C1C=CC=C2C3=CC(=CC=C3)C(=O)NCCO)CC4=CC(=CC=C4)C(F)(F)F	-1.909
Q9Y2T5	COCCNC(=O)C1=CC=CC(=C1)C2=CC=CC3=C2SC(=C3)CC4=CC(=CC(=C4)Cl)F	-1.474
Q9Y2T5	C1=CC(=CC(=C1)C(F)(F)F)CC2=CC3=C(C=CC=C3O2)C4=CC(=CC=C4)C(=O)NCCO	-1.45
Q9Y2T5	COCCNC(=O)C1=CC=CC(=C1)C2=CC=CC3=C2SC(=C3)CC4=CC(=CC=C4)C(F)(F)F	-1.632

4.3. Effects of the different sources of training samples

To deal with source domain dataset, there are two solutions, one with drop-back sampling and one with no drop-back sampling. The one with drop-back sampling is as follows. For each source domain dataset, the method is put back sampling as follows, sample each source domain dataset and set a sampling ratio of 0.5. The source domain dataset in such a group was changed from the original 4 groups to 12 groups. The one with no drop-back sampling is as follows. For each source domain dataset, each dataset is picked 3 times, and the ratio is set to 0.5 so that the source domain dataset in one set changes from the original 4 to 12 groups. This section mainly verifies whether differences in obtaining source domain datasets affect the predicted performance. Table 4.7 shows the experimental results on 12 datasets based on MSTL-GNN with put-back sampling and MSTL-GNN based on no drop-back sampling methods.

Table 2. Comparison between R2 and RMSE in MSTL-GNN with and without drop-back sampling mode. F-MSTL-GNN represents the MSTL-GNN, two figures in bold are those without drop-back sampling for better data in both ways.

Group	R^2 (↑)		RMSE (↓)	
	MSTL-GNN	F-MSW-WDL-RF	MSTL-GNN	F-MSTL-GNN
A	**0.938**	0.923	**0.117**	0.132
B	0.615	0.646	0.778	0.793
C	0.745	0.746	0.64	0.601
D	0.69	0.702	0.247	0.23
E	**0.493**	0.492	**0.633**	0.634
F	0.393	0.396	0.612	0.611
G	0.397	0.41	0.724	0.715
H	**0.587**	0.583	**0.582**	0.585
I	**0.504**	0.491	**0.734**	0.743
J	**0.599**	0.598	0.598	0.598
K	0.436	0.448	0.585	0.578
L	0.562	0.569	0.719	0.713

It can be observed from Table 2 that A group, E group, H group, I group experiment, the put-back sampling method RMSE is slightly lower than the no-back sampling method. R2 in A group, H group, I group, and J group with the put-back sampling method is slightly higher than the no-back sampling method. The lower the RMSE represents, the smaller error in the prediction. The higher R2 represents, the more stable the prediction model is. To compare whether there were significant differences in drop-back and no-dropback sampling methods, we calculated with the Wilcoxon signed-rank test. Results for both sampling methods are shown in Table 2. The calculation was performed in two ways. The p-value is 0.656, and the p-value is much greater than 0.05. The p-value between RMSE was 0.504, also much greater than 0.05. This represents no significant difference between no put-back sampled dataset generation and no put-back sampling data generation methods. Not put-back sampling, with completely different source domain samples, has similar results to put-back sampling. This shows that multi-source transfer learning is multiple-source domains together and can learn features from a different domain, rather than only learning from a single domain. Also indirectly verified that the multiple-source transfer learning algorithm is superior to the single-source transfer learning algorithm.

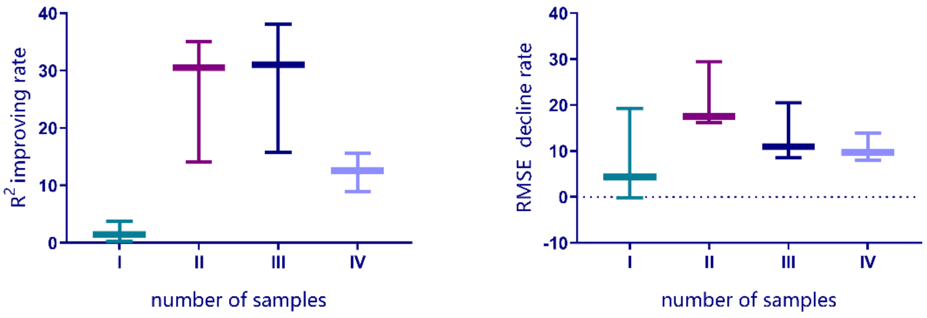
We went on to compare the effect of the different sample numbers on the MSTL-GNN. We combine group A, group B and group C with sample amount fewer than 30 into Group I; we combine group D, group E and Group F with sample amount larger than 30 into Group II; we combine group G, group H and Group I with sample amount larger than 300 into Group III. Group J, K and L with sample amount larger than 300 were combined into Group IV. Then separately compare MSTL-GNN with WDL-RF in these four groups. The average improvement rate of R2, and the average reduction rate of RMSE. Specifically, Table 3 shows the results in these four groups.

Table 3. Rate of RMSE reduction between MSTL-GNN and WDL-RF versus increased rate of R2 for the different number of samples. I represents sample size less than 30, II represents sample size less than 30 and greater than 100, III represents sample size greater than 100 and less than 300, and IV represents sample size greater than 300.

Number of samples	R^2 (↑)	RMSE (↓)
I	2.45%	3.88%
II	67.13%	19.22%
III	39.20%	14.00%
IV	17.95%	10.37%

As is known from Table 3, the average improvement rate of R2 in I is 2.45%, with RMSE of 3.88%, MSTL-GNN only slightly improved over WDL-RF with sample size less than 30, and high R2 of both algorithms can also be seen in Table 4.8. The lowest R2 in WDL-RF is at 0.59, in which case, promotion is difficult. R2 in III sees the average improvement rate of 67.13%, RMSE of 19.22%, and MSTL-GNN performance over WDL-RF at sample size less than 100 and greater than 30, which was poor at this number of samples with the highest R2 only 0.334. MSTL-GNN largely boosted R2 in the case of this small sample, and reducing the RMSE, reduces the error of the prediction model, while improving the reliability. Judging from Table 3, the same situation also occurs in III, both with insufficient samples, while the WDL-RF situation performs better than WDL-RF. In the large sample of III, the R2 of MSTL-GNN with an average improvement rate of 17.95% and RMSE of 10.37%, MSTL-GNN experiences a significant boost compared to WDL-RF.

The Figure 4 shows that RMSE has the rate of a 10% reduction either in the case of large or small samples, R2 sees an improvement rate of nearly 20%, meaning that the MSTL-GNN algorithm not only has good performance in small samples, but also improves the prediction performance with large sample size, with the best improvement at the sample size of 30-300. In the large sample case, the graph neural network learns good features and predicts activity values, and adding multi-source transfer learning is equivalent to adding the training sample, so it can better fit the model. In a very small sample, although adding transfer learning enhances the training effect, the model can easily underfit due to the small sample size, reducing the performance of the algorithm. The addition of multi-source transfer learning has solved the multi-fitting problem well and thus had the best performance in the three cases.



(a) (b)

Figure 4. Comparison of MSTL-GNN performance with WDL-RF for different numbers of samples. (a) R2 improving rate. (b) RMSE decline rate. I represents sample size less than 30, II represents sample size less than 30 and greater than 100, III represents sample size greater than 100 and less than 300, and IV represents sample size greater than 300.

To verify the effect of training samples on MSTL-GNN, 300,100,30 and 15 samples were randomly taken from the K (with 709 samples) and L (with 2776 samples), respectively, and the remaining samples served as test samples, greatly improving the test sample size to see the effect of training sample size on MSTL-GNN.

Table 4. MSTL-GNN under different training samples.

Number of samples	R^2 (↑)	RMSE (↓)
0–30	2.45%	3.88%
31–100	67.13%	19.22%
101–300	39.20%	14.00%
>300	17.95%	10.37%

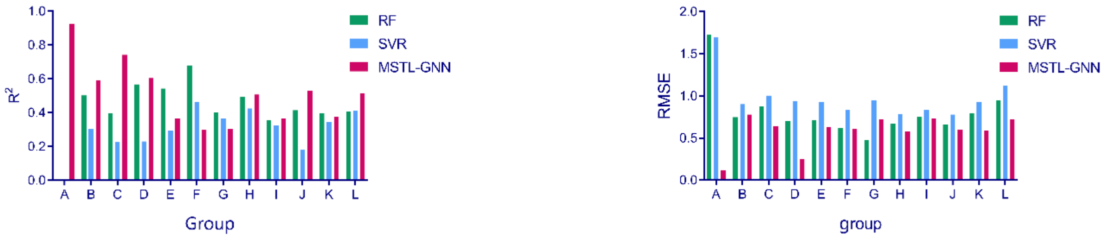
From Table 4, at sample sizes of 15 and 30, R2 is close to 0, RMSE is close to 1, showing that MSTL-GNN is as poor as the other two algorithms, which is also because although the multi-source transfer learning is added, multi-source transfer learning only plays an auxiliary role if the sample is so small that the graph neural network cannot build a strong generalization model, and multi-source transfer learning is unable to establish a good model. However, when the sample size is greater than 100, the algorithm improves significantly, which verifies the strong generalization ability of the algorithm in smaller samples and its effectiveness of the algorithm. When the sample size is greater than 300, the algorithm has better performance, and the model of the graph neural network has stronger generalization.

4.4. Comparison

MSTL-GNN is used by graph neural networks to generate new molecular fingerprints and then builds models to predict the biological activity values of ligand molecules when binding to GPCRs. Random Forest (RF) and Support Vector Regression (SVR) is performed using software to calculate the molecular fingerprints of ligand molecules that are fixed and cannot be adjusted with the task. Here we specifically compare the predicted performance of MSTL-GNN, RF, SVR, and WDL-RF on the 12 datasets. The results are shown in table 5 and figure 5. Specifically, nine of the 12 data sets performed better than the other three algorithms, with R2 of RF in E, G, and H slightly higher than MSTL-GNN. However, in these three groups, G group, and H group, MSTL-GNN had lower RMSE than RF. In many cases, the R2 was not completely positively associated with RMSE. Although RF improved R2, RMSE is high, which demonstrates that there is a contradictory relationship between the R2 and RMSE. To better fit the data and reduce the error, the reliability of the model is reduced. In addition, it can be seen that the MSTL-GNN is significantly better than the SVR and RF when the sample size is relatively small. At the same time, when the sample size is very large, the MSTL-GNN algorithm is also significantly better than the SVR algorithm, which proves that the graph neural network generates a new weighted molecular fingerprint, which is superior to the traditional software calculation, and the multi-source migration graph neural network algorithm also shows its superiority under small samples.

Table 5. SVR, RF, MSTL-GNN2 Comparison with RMSE.

Group	R^2 (↑)				RMSE (↓)			
	RF	SVM	WDL-RF	MSTL-GNN	RF	SVR	WDL-RF	MSTL-GNN
A	0.001	0.002	0.915	**0.925**	1.724	1.698	0.145	**0.117**
B	0.502	0.304	0.59	**0.593**	0.748	0.902	0.813	**0.778**
C	0.398	0.228	0.738	**0.743**	0.879	1.004	**0.639**	0.64
D	0.568	0.23	0.399	**0.605**	0.703	0.936	0.35	**0.247**
E	**0.543**	0.295	0.334	0.365	0.716	0.927	0.755	**0.633**
F	**0.68**	0.462	0.21	0.301	0.62	0.836	0.742	**0.612**
G	**0.403**	0.364	0.294	0.303	**0.478**	0.948	0.813	0.724
H	0.495	0.425	0.516	**0.507**	0.67	0.786	0.636	**0.582**
I	0.355	0.324	0.259	**0.365**	0.749	0.836	0.923	**0.734**
J	0.415	0.18	0.528	**0.532**	0.663	0.778	0.662	**0.598**
K	0.397	0.347	0.333	**0.377**	0.796	0.925	0.679	**0.585**
L	0.406	0.413	0.493	**0.516**	0.945	1.125	0.781	**0.719**



(a)

(b)

Figure 5. Comparison of MSTL-GNN performance with RF and SVR. (a) R2 comparison. (b) RMSE comparison.

5. Conclusions

This paper mainly studies the application of multi-source transfer learning and graph neural networks in predicting molecular biological activity, proposing the MSTL-GNN algorithm. We study the establishment of multi-source transfer learning while using integrated learning to improve the ability of activity value prediction. MSTL-GNN still has good performance under small samples. Compared to single-source domains, MSTL-GNN selected multiple-source domains to improve model performance through transfer learning while reducing the impact of negative migration from single-source migration. Experimental results show that MSTL-GNN performs better in both large and small samples than traditional molecular fingerprint-based methods. Specifically, an average improvement of 29.67% in R2 is seen compared with the previous algorithm MSTL-GNN, and an average RMSE improvement of 11.23%. Moreover, compared with the state-of-the-art work (WDL-RF), MSTL-GNN increased up to 67.13% and 17.22%, respectively.

MSTL-GNN is a multi-source transfer learning method based on parameter migration, which has no source domain to target domain adaptation process and is suitable only for similar source domains and target domains. In normal life, homologous drug target and ligand binding activity values are not necessarily the same, so the domain adaptation process is required to make the source domain and target domain as similar as possible. In addition, MSTL-GNN average results of multiple transfer learning in the integrated learning section and does not weigh the results of migration from different source domains. In the future, weights can be used to increase the proportion of source fields similar to the target domain and reduce the proportion of dissimilar source domains.

Acknowledgments

We would like to thank you for following the instructions above very closely in advance. It will definitely save us lot of time and expedite the process of your paper's publication.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. A. S. Hauser, M. M. Attwood, M. Rask-Andersen, H. B. Schiöth, D. E. Gloriam, Trends in GPCR drug discovery: new agents, targets and indications, *Nat. Rev. Drug. Discov.*, **16** (2017), 829–842. https://doi.org/10.1038/nrd.2017.178
2. L. M. Slosky, M. G. Caron, L. S. Barak, Biased allosteric modulators: New frontiers in GPCR drug discovery, *Trends Pharmacol. Sci.*, **42** (2021), 283–299. https://doi.org/10.1016/j.tips.2020.12.005
3. F. Zhang, V. Lemaur, W. Choi, P. Kafle, S. Seki, J. Cornil, et al., Repurposing DNA-binding agents as H-bonded organic semiconductors, *Nat. Commun.*, **10** (2019), 4217. https://doi.org/10.1038/s41467-019-12248-9

4. S. Chung, T. Funakoshi, O. Civelli, Orphan GPCR research, *British J. Pharmacol.*, **153** (2008), S339–S346. https://doi.org/10.1038/sj.bjp.0707606
5. W. K. Kroeze, M. F. Sassano, X.-P. Huang, K. Lansu, J. D. McCorvy, P. M. Giguère, et al., PRESTO-Tango as an open-source resource for interrogation of the druggable human GPCRome, *Nat. Struct. Mol. Biol.*, **22** (2015), 362–369. https://doi.org/10.1038/nsmb.3014
6. A. T. Ehrlich, G. Maroteaux, A. Robe, L. Venteo, M. T. Nasseef, L. C. van Kempen, et al., Expression map of 78 brain-expressed mouse orphan GPCRs provides a translational resource for neuropsychiatric research, *Commun. Biol.*, **1** (2018), 1–14. https://doi.org/10.1038/s42003-018-0106-7
7. M. Zhao, Z. Wang, M. Yang, Y. Ding, M. Zhao, H. Wu, et al., The Roles of Orphan G Protein-Coupled Receptors in Autoimmune Diseases, *Clinic. Rev. Allerg. Immunol.*, **60** (2021), 220–243. https://doi.org/10.1007/s12016-020-08829-y
8. J. Colette, E. Avé, B. Grenier-Boley, A.-S. Coquel, K. Lesellier, K. Puget, Bioinformatics-based discovery and identification of new biologically active peptides for GPCR deorphanization, *J. Peptide Sci.*, **13** (2007), 568–574. https://doi.org/10.1002/psc.898
9. A. Jabeen, S. Ranganathan, Applications of machine learning in GPCR bioactive ligand discovery, *Current Opin. Structural Biol.*, **55** (2019), 66–76. https://doi.org/10.1016/j.sbi.2019.03.022
10. H. A. L. Filipe, L. M. S. Loura, Molecular dynamics simulations: Advances and applications, *Molecules*, **27** (2022), 2105. https://doi.org/10.3390/molecules27072105
11. A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, and G. Pujadas, Molecular fingerprint similarity search in virtual screening, *Methods*, **71** (2015), 58–63. https://doi.org/10.1016/j.ymeth.2014.08.005
12. R. Wang, S. Li, L. Cheng, M. H. Wong, K. S. Leung, Predicting associations among drugs, targets and diseases by tensor decomposition for drug repositioning, *BMC Bioinform.*, **20** (2019), 628. https://doi.org/10.1186/s12859-019-3283-6
13. B. Jan, H. Farman, M. Khan, M. Imran, I. U. Islam, A. Ahmad, et al., Deep learning in big data Analytics: A comparative study, *Comput. Electr. Eng.*, **75** (2019), 275–287. https://doi.org/10.1016/j.compeleceng.2017.12.009
14. P. Singh, S. S. Bose, Ambiguous D-means fusion clustering algorithm based on ambiguous set theory: Special application in clustering of CT scan images of COVID-19, *Knowledge-Based Systems*, **231** (2021), 107432. https://doi.org/10.1016/j.knosys.2021.107432
15. O. Cabral-Marques, G. Halpert, L. F. Schimke, Y. Ostrinski, A. Vojdani, G. C. Baiocchi, et al., Autoantibodies targeting GPCRs and RAS-related molecules associate with COVID-19 severity, *Nat. Commun.*, **13** (2022), 1220. https://doi.org/10.1038/s41467-022-28905-5
16. W. Tong, H. Hong, H. Fang, Q. Xie, R. Perkins, Decision Forest: Combining the Predictions of Multiple Independent Decision Tree Models, *J. Chem. Inf. Comput. Sci.*, **43** (2003), 525–531. https://doi.org/10.1021/ci020058s
17. E. Lounkine, F. Nigsch, J. L. Jenkins, M. Glick, Activity-Aware Clustering of High Throughput Screening Data and Elucidation of Orthogonal Structure–Activity Relationships, *J. Chem. Inf. Model.*, **51** (2011), 3158–3168. https://doi.org/10.1021/ci2004994
18. K. A. Carpenter, D. S. Cohen, J. T. Jarrell, X. Huang, Deep learning and virtual drug screening,

Future Med Chem, **10** (2018), 2557–2567. https://doi.org/10.4155/fmc-2018-0314

19. J. Wu, Q. Zhang, W. Wu, T. Pang, H. Hu, W. K. B. Chan, et al., WDL-RF: predicting bioactivities of ligand molecules acting with G protein-coupled receptors by combining weighted deep learning and random forest, *Bioinformatics*, **34** (2018), 2271–2282. https://doi.org/10.1093/bioinformatics/bty070
20. S. Hu, P. Chen, P. Gu, and B. Wang, A Deep Learning-Based Chemical System for QSAR Prediction, *IEEE Journal of Biomedical and Health Informatics*, **24** (2020), 3020–3028. https://doi.org/10.1109/JBHI.2020.2977009
21. J. M. Stokes, K. Yang, K. Swanson, W. Jin, A. Cubillos-Ruiz, N. M. Donghia, et al., A Deep Learning Approach to Antibiotic Discovery, *Cell*, **180** (2020), 688–702. e13. https://doi.org/10.1016/j.cell.2020.01.021
22. A. P. Bento, A. Hersey, E. Félix, G. Landrum, A. Gaulton, F. Atkinson, et al., An open source chemical structure curation pipeline using RDKit, *J. Cheminform.*, **12** (2020), 51. https://doi.org/10.1186/s13321-020-00456-1
23. S. Vijayakumar, V. Kant, and P. Das, LeishInDB: A web-accessible resource for small molecule inhibitors against Leishmania sp, *Acta Trop.*, **190** (2019), 375–379. https://doi.org/10.1016/j.actatropica.2018.12.022
24. K. P. Singh, S. Gupta, Nano-QSAR modeling for predicting biological activity of diverse nanomaterials, *RSC Adv.*, **4** (2014), 13215–13230. https://doi.org/10.1039/C4RA01274G
25. K. Lech, A. Figiel, A. Wojdyło, M. Korzeniowska, M. Serowik, M. Szarycz, Drying Kinetics and Bioactivity of Beetroot Slices Pretreated in Concentrated Chokeberry Juice and Dried with Vacuum Microwaves, *Dry. Technol.*, **33** (2015), 1644–1653. https://doi.org/10.1080/07373937.2015.1075209
26. J. Wu, C. Lan, X. Ye, J. Deng, W. Huang, X. Yang, et al., Disclosing incoherent sparse and low-rank patterns inside homologous GPCR tasks for better modelling of ligand bioactivities, *Front Comput. Sci.*, **16** (2021), 164322. https://doi.org/10.1007/s11704-021-0478-6
27. The UniProt Consortium, UniProt: A hub for protein information, *Nucleic Acids Res.*, **43** (2015), D204–D212. https://doi.org/10.1093/nar/gku989
28. W. K. B. Chan, H. Zhang, J. Yang, J. R. Brender, J. Hur, A. Özgür, et al., GLASS: A comprehensive database for experimentally validated GPCR-ligand associations, *Bioinformatics*, **31** (2015), 3035–3042. https://doi.org/10.1093/bioinformatics/btv302

Supplementary

Table a. The GPCRs dataset.

Group	Type	ID	Number of Ligand	Family	Species
A	Target	Q9Y2T5	15	Orphan receptors	Human
	Source	P07550	3155	Adrenergic receptors	Human
		P50406	3378	Serotonin receptors	Human
		P35368	1394	Adrenergic receptors	Human
		P13945	1655	Adrenergic receptors	Human

B	Target	O43194	27	Orphan receptors	Human
	Source	Q92847	1769	Releasing hormones receptors	Human
		P24530	1111	Endothelin receptors	Human
		P41144	2229	Opioid peptides receptors	Guinea pig
		O43614	3882	Orexins receptors	Human
C	Target	Q6DWJ6	30	Orphan receptors	Human
	Source	P41146	1380	Opioid peptides receptors	Human
		P31391	657	Somatostatin and urotensin receptors	Human
		P32246	693	Chemokines and chemotactic factors receptors	Human
		P61073	623	Chemokines and chemotactic factors receptors	Human
D	Target	P46093	38	Orphan receptors	Human
	Source	P25106	562	Chemokines and chemotactic factors receptors	Human
		P21556	1231	Platelet-activating factor receptors	Guinea pig
		P47900	718	Adenosine and adenine nucleotide receptors	Human
		P32246	693	Chemokines and chemotactic factors receptors	Human
E	Target	Q96LB2	74	Orphan receptors	Human
	Source	Q9Y5Y4	2776	Orphan receptors	Human
		P25090	568	Chemokines and chemotactic factors receptors	Human
		P30874	827	Somatostatin and urotensin receptors	Human
		P30872	635	Somatostatin and urotensin receptors	Human
F	Target	P32249	97	Orphan receptors	Human
	Source	Q2NNR5	538	Cysteinyl leukotriene receptors	Guinea pig
		P32246	693	Chemokines and chemotactic factors receptors	Human
		P47900	718	Adenosine and adenine nucleotide receptors	Human
		P34976	1108	Angiotensin receptors	Rabbit
G	Target	Q7Z601	107	Orphan receptors	Human
	Source	P47901	952	Vasopressin / oxytocin receptors	Human
		P32246	693	Chemokines and chemotactic factors receptors	Human
		P08912	898	Acetylcholine (muscarinic) receptors	Human
		P08482	1790	Acetylcholine (muscarinic) receptors	Rat
H	Target	Q5NUL3	195	Orphan receptors	Human
	Source	O43614	3882	Orexins receptors	Human
		P49146	671	Neuropeptide Y receptors	Human
		P35346	947	Somatostatin and urotensin receptors	Human
		Q99705	3669	Melanin-concentrating hormone receptors	Human
I	Target	Q9GZN0	204	Orphan receptors	Human
	Source	P30872	635	Somatostatin and urotensin receptors	Human
		P41146	1380	Opioid peptides receptors	Human
		P35462	4268	Dopamine receptors	Human
		P43140	777	Adrenergic receptors	Rat
J	Target	Q9HC97	331	Orphan receptors	Human
	Source	Q9Y2T6	709	Orphan receptors	Human
		Q8TDS4	500	Nicotinic acid receptors	Human
		P21556	1231	Platelet-activating factor receptors	Guinea pig

		P30411	717	Bradykinin receptors	Human
K	Target	Q9Y2T6	709	Orphan receptors	Human
	Source	P32246	693	Chemokines and chemotactic factors receptors	Human
		P21556	1231	Platelet-activating factor receptors	Guinea pig
		P35351	593	Angiotensin receptors	Rat
		P41144	2229	Opioid peptides receptors	Guinea pig
L	Target	Q9Y5Y4	2776	Orphan receptors	Human
	Source	P25090	568	Chemokines and chemotactic factors receptors	Human
		P35351	593	Angiotensin receptors	Rat
		P30556	1049	Angiotensin receptors	Human
		P32300	1552	Opioid peptides receptors	Mouse