Integrating concept of pharmacophore with Graph Neural Networks for chemical property prediction and interpretation

Yue Kong

Beijing University of Chemical Technology <https://orcid.org/0000-0003-1599-7264>

Xiaoman Zhao

Beijing University of Chemical Technology <https://orcid.org/0000-0002-7049-1461>

Ruizi Liu

Beijing University of Chemical Technology <https://orcid.org/0000-0003-0634-232X>

Zhenwu Yang

Beijing University of Chemical Technology <https://orcid.org/0000-0002-4468-9490>

Hongyan Yin

Beijing University of Chemical Technology <https://orcid.org/0000-0001-9116-5908>

Bowen Zhao

Hyper-dimension insight pharmaceuticals Ltd. <https://orcid.org/0000-0003-0794-8868>

Jinling Wang

Hyper-dimension insight pharmaceuticals Ltd. <https://orcid.org/0000-0002-6805-0704>

Bingjie Qin

Hyper-dimension insight pharmaceuticals Ltd. <https://orcid.org/0000-0002-7107-5726>

Aixia Yan (✉ yanax@mail.buct.edu.cn)

Beijing University of Chemical Technology

Research Article

Keywords: graph neural networks (GNNs), pharmacophore, reduced graph (RG), hierarchical pooling

Posted Date: January 19th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1221390/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. [Read Full License](#)

Abstract

Recently, graph neural networks (GNNs) have revolutionized the field of chemical property prediction and achieved state-of-the-art results on benchmark data sets. Compared with the traditional descriptor- and fingerprint-based QSAR models, GNNs can learn task related representations, which completely gets rid of the rules defined by experts. However, due to the lack of useful prior knowledge, the prediction performance and interpretability of the GNNs may be affected. In this study, we introduced a new GNN model called PharmGNN for chemical property prediction that integrated pharmacophore information hierarchically into message-passing neural network (MPNN) architecture, specifically, in the way of pharmacophore-based reduced-graph (RG) pooling. PharmGNN absorbed not only the information of atoms and bonds from the atom-level message-passing phase, but also the information of pharmacophores from the RG-level message-passing phase. Our experimental results on eleven benchmark and ten kinase data sets showed that our model consistently matched or outperformed other existing GNN models. Furthermore, we demonstrated that applying pharmacophore-based RG pooling to MPNN architecture can generally help GNN models improve the predictive power. The cluster analysis of PharmGNN representations and the importance analysis of pharmacophore nodes will help chemists gain insights for hit discovery and lead optimization.

Introduction

With the accumulation of large-scale chemical and biological data, the improvement of computing power, and especially the major breakthroughs that deep neural networks (DNNs) having made in many fields such as image recognition [1] and natural language processing [2], there has been a surge of interest in developing DNNs for drug discovery in recent years [3–7]. This trend is especially reflected in the development of a variety of DNNs for chemical property prediction [8–10]. In the field of drug design, these models correspond to the quantitative structure-activity (property) relationship (QSAR/QSPR) models [11], belonging to the category of ligand-based drug design (LBDD) method [12, 13]. This method is not limited to the availability of the 3D structure of the target of interest but fits models or finds patterns from the collected ligand data, which is commonly used for large-scale virtual screening, chemical property evaluation and molecular structure optimization.

Before the rise of DNNs, there have been extensive QSAR models developed for drug discovery, mainly using traditional machine learning (ML) approaches, such as support vector machines (SVMs) [14, 15], Naïve Bayes (NB) [16, 17], artificial neural network (ANN) [18, 19], random forest (RF) [20] etc. Although many reports claimed that the performance of many DNN models did not meet researchers' expectation that the prediction accuracy is far beyond traditional machine learning [21, 22], it is largely because the amount of data is too small to give full play to advantages of DNNs. This advantage is bound to be brought into play as the amount of chemical and biological data gradually accumulates. Another point that motivates researchers to keep great enthusiasm for the development of DNNs is their ability of effective representation extraction. Instead of using expert features and performing feature engineering like traditional machine learning models, which is complex and time-consuming, not easy to reproduce, and limited by expert feature definition, DNNs can extract useful representations from task data for the sake of the end-to-end fashion. And this task-learned representations can be used for analogues searching in virtual screening campaigns and gaining insights of relationship between chemical structure and properties to guide molecular optimization. However, to achieve accurate predictions and extract useful representations at the same time depends on well-designed DNN architecture, which is challenging but urgent to be paid efforts to.

Given that the framework of DNN is quite flexible, there have been a variety of DNN models published for drug discovery. Most of these models can be mainly divided into two types: one is to use the RNN (recurrent neural network) or Transformer [23] frameworks to operate the molecular SMILES as strings [24, 25], the other one is to use the GNN (graph neural network) framework to operate molecules as graphs [26, 27]. Although systematic performance comparison results for these two methods are rare, GNN is more popular than RNN in chemical property prediction in recent years. This may be because the form of graph representation is closer to the intrinsic properties of the molecular structure, thus what the model learns from the graph is more able to reflect the properties of the molecule. On the other hand, the SMILES string guides the model to learn a lot of SMILES grammar rules, such as parentheses representing branched chains which are irrelevant to molecular structure. Recently, as a general GNN architecture, the message-passing neural network (MPNN) [28] has been proposed, consisting of a message-passing phase and a readout (or called pooling) phase. Researchers have developed many models based on MPNN architecture to predict chemical properties and extract task-learned representations, such as MPNN model (note it refers a specific model instead the previously mentioned MPNN architecture), D-MPNN (Directed MPNN) [29], AttentiveFP [30], R-GCN (Relational Graph Convolutional Networks) [31] and GSN (Graph Substructure Networks) [32].

However, most of the MPNN architectures only absorb node information (such as atom type, formal charge) and edge information (such as bond type, stereo type) as the original information of a molecular graph, but do not make full use of prior chemical knowledge, such as the information from pharmacophores [33], which have been widely used in drug design and discovery. Of note, there have been many successful cases that prove that the pharmacophore rules can be well combined with molecular graph, and one representative method is the pharmacophore-based graph reduction [34–36]. Based on pharmacophore rules, the reduced graphs (RGs) provide simplified representations of chemical structures by collapsing atom groups into pharmacophores while maintaining the topological properties of the original molecular structures. This drives us to think about whether embedding information of pharmacophores into MPNN architecture under graph reduction scheme will help improve the accuracy and reliability of the model and to enrich the information contained in the task-learned representation.

Another limitation of current MPNN architectures for chemical property prediction is that they ignore any hierarchical structure and information that might exist in the molecular graph, which will hinder the models from effectively extracting the information in the graph. On the other hand, the global pooling such as the maximizing pooling, average pooling, and pooling with attention mechanisms [30], has been adopted as the standard readout phase for a model with the MPNN architecture, leading to a “flat” nature. More recently, hierarchical pooling has attracted research attention, for instance, Diffpool [37] uses a learned distribution matrix to collapse atom groups. However, these current hierarchical structures still do not make full use of prior chemical knowledge. To date, there has been no hierarchical pooling method leverages knowledge of pharmacophore to design GNN models, specifically, the ones with the MPNN architecture.

In this work, we proposed a new GNN model, PharmGNN, for chemical property prediction. The core idea of PharmGNN was to integrate pharmacophore information hierarchically into MPNN architecture, specifically, in the way of pharmacophore-based RG pooling. As illustrated in Figure 1, the PharmGNN absorbed not only the information of atoms and bonds from the atom-level message-passing phase, but also the information of pharmacophores from the RG-level message-passing phase. Our models achieved state-of-the-art prediction performance and these results were also transferred to data sets of ten popular kinases. Furthermore, the cluster analysis of the task-learned representation of PharmGNN showed that the representation can be used to identify molecules with similar activities but different scaffolds in the context of virtual screening and lead optimization.

Methods

Data sets

Benchmark datasets. To compare the performance of the PharmGNN with those of other GNN models, we tested our models on eleven benchmark datasets from MoleculeNet [38]. Among MoleculeNet, the physical chemistry, bioactivity and physiology data sets except for the PDBbind data sets were tested in this work. Three of the eleven datasets were used for regression tasks and eight for classification tasks.

Kinase datasets. The core idea of our model PharmGNN is to integrate information of pharmacophores which are regarded as abstract features of molecules for molecular recognition of a ligand by a biological target. Therefore, in theory, our model is more suitable for the task of predicting molecular bioactivity towards targets of interest. To systematically test the prediction performance of various algorithms on the bioactivity datasets, we collected inhibitors of ten kinase targets (see Table 1). The principle of kinase target selection was to cover each kinase family as much as possible and select the targets of great prospects for drug development. All these datasets were derived from ChEMBL [39]. After a series of operations such as data deduplication, salt removal, and electrical neutrality, ten kinase data sets were ready for classification task, with the numbers of molecules ranging from 807 to 8800 and the ratios of positive and negative samples ranging from 0.19 to 0.82.

Table 1
Basic information of kinase datasets used in this work

kinase family	kinase full name ^a	short name	total ^b	active	inactive	ratio ^c
TK	Epidermal growth factor receptor erbB1	EGFR	8800(8718)	4514	4286	0.51
TKL	Serine/threonine-protein kinase B-raf	BRAF	4761(4669)	3629	1132	0.76
CAMK	Serine/threonine-protein kinase PIM1	PIM1	4507(4437)	3322	1185	0.74
Atypical	Serine/threonine-protein kinase mTOR	mTOR	4135(3924)	3390	745	0.82
AGC	Serine/threonine-protein kinase AKT	AKT1	3967(3904)	2175	1792	0.55
Other	Serine/threonine-protein kinase Aurora-A	AURKA	3882(3854)	2286	1596	0.59
TK	Tyrosine-protein kinase BTK	BTK	2640(2560)	1894	746	0.72
CMGC	Cyclin-dependent kinase 2	CDK2	2603(2531)	1302	1301	0.50
STE	Mitogen-activated protein kinase kinase kinase kinase 2	MAP4K2	897(891)	280	617	0.31
CK1	Casein kinase I alpha	CK1	807(801)	154	653	0.19
a: preferred name listed on ChEMBL website; b: number of all molecules (number of molecules that are suitable for reduced-graph processing); c: active/inactive ratio.						

Molecular Graph

In graph neural networks (GNNs), a molecule is regarded as a graph $G = (V, E)$, where atom is regarded as node V and chemical bond is regarded as edge E . The nodes and edges are encoded according to the rules shown in the Table S1 and Table S2. For instance, node features include atom type, formal charge, etc., and edge features include bond type, stereo type, etc. These encoded features are the initial features of molecular graphs which are used as raw inputs to train GNN models. After training, we can get the final task prediction value, together with the task-learned graph representations that also can be called molecular fingerprints.

Reduced Graphs (RGs)

RGs provide simplified representations of chemical structures by collapsing atom groups into super nodes while maintaining the topological properties of the original molecular structures. RGs have been mainly implemented to the varied applications of similarity searching, scaffold hopping, de novo design and structure-activity relationships extracting [34, 36, 40, 41].

By altering the rules used for collapsing atom groups, RGs provide flexible ways of generalizing super node features. There is a research trend to collapse the atom groups into RGs through the pharmacophore rules and the resulting RGs can be regarded as topological pharmacophore [36, 40]. It is worth emphasizing that the pharmacophore rules need to be improved before applied to graph reduction. This is because each atom in RGs needs to be mapped to one or more super nodes, while atoms that do not belong to any pharmacophore are not labeled according to classical pharmacophore rules.

In this work, we adopted the graph reduction scheme developed by Harper [34], which defines 18 types of super nodes as shown in Figure 2 (a): three types about defining rings (aromatic ring, aliphatic ring, or acyclic) intersected with six types about defining features (positively ionizable, negatively ionizable, joint

H-bond donor and acceptor, donor, acceptor, or no feature), and it should be noted that the items within the three ring types and the six feature types are listed in order of priority from high to low. Figure 2 (b) lists some comparative examples of molecules and their RGs. Readers can find more graph reduction schemes in literature [34, 35, 42].

Message-Passing Neural Network (MPNN)

MPNN is a general framework for supervised learning on graphs. Within its forward pass, there are two phases: a message-passing phase and a readout phase. Here we take an undirected graph G as an example, within which the node (atom) features are represented as x_v and the edge (bond) features as e_{vw} . In terms of the message-passing phase, the message function is defined as M_t and the vertex update function is defined as U_t , where t is the running time step. During message-passing process, the hidden state of each node h_v^{t+1} can be updated based on message m_v^{t+1} according to:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw})$$

1

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1})$$

2

where $N(v)$ is the set of neighbors of the node v in G . In addition, h_v^0 is derived from the initial node features x_v through some function.

In terms of the readout phase, it uses a readout function R to make a task prediction for the whole graph according to:

$$\hat{y} = R(\{h_v^T | v \in G\})$$

3

where the output $\hat{y}$ can be a scalar or a vector, depending on whether it is used for single task prediction or multi-task predictions.

During training process, taking the molecular graphs as inputs, the model predicts the properties of each molecule. The loss is computed based on the predicted properties and the true ones, then of which the gradient is backpropagated through the readout phase and the message-passing phase.

Reduced-Graph MPNN architecture (RG-MPNN)

The reduced-graph message-passing neural network architecture (shorted as RG-MPNN architecture) was proposed in this work, which consists of four phases: a message-passing phase at atom level, a graph reducing phase, a message-passing phase at RG level and a molecule readout phase. In short, compared with common MPNN architecture, the RG-MPNN architecture has one more graph-reducing phase and one message-passing phase at RG level. The RG-MPNN architecture works as follows.

Atom-level Message-Passing. During the atom-level message-passing phase, the operation of RG-MPNN architecture is very similar to the message-passing phase of typical MPNNs, with one difference that e is not directly considered in the message function M_k since h_{atom}' is derived from $cat(x_{atom}', e_{atom-atom'})$ by linear transformation. This phase runs for K time steps. The hidden state of each atom h_{atom}^{k+1} can be updated based on message m_{atom}^{k+1} according to:

$$m_{atom}^{k+1} = \sum_{atom' \in N(atom)} M_k(h_{atom}^k, h_{atom'}^k)$$

4

$$h_{atom}^{k+1} = U_k(h_{atom}^k, m_{atom}^{k+1})$$

5

Graph Reducing. During this phase, the whole graph G is operated by the function *Reduce* which maps each atom to one or more super nodes with the rules we have mentioned in the method part of reduced graphs, resulting in a reduced graph RG . Then, we define $RG = (V', E')$, where V' represents the super node, which is one of the 18 predefined super nodes, and E' represents the edge between super nodes, which is equal to one plus to the number of chemical bonds shared between two adjacent super nodes. The hidden state of initial super node h_{rg}^0 according to:

$$h_{rg}^0 = Reduce(\{h_{atom}^K | atom \in rg\})$$

6

RG-level Message-Passing. This phase runs for T time steps and the hidden state of each super node h_{rg}^{t+1} can be updated based on message m_{rg}^{t+1} according to:

$$m_{rg}^{t+1} = \sum_{rg' \in N(rg)} M_t(h_{rg}^t, h_{rg'}^t)$$

$$h_{rg}^{t+1} = U_t(h_{rg}^t, m_{rg}^{t+1})$$

Molecule Readout. During this phase, the molecule embedding h_{mol} , also as the task-learned representation of molecular graph, is achieved by a readout function R based on the hidden states h_{rg}^T within RG :

$$h_{mol} = R(\{h_{rg}^T | rg \in RG\})$$

then the prediction of molecular property is achieved through MLP layers:

$$\hat{y} = MLP(h_{mol})$$

where the output $\hat{y}$ can be a scalar or a vector same as that in the MPNN process, depending on whether it is used for single task prediction or multi-task predictions.

Theoretically, the RG-MPNN architecture proposed in this paper can be applied to any model under the MPNN architecture, that is, before readout of the whole molecule, the graph reducing and message-passing at RG level can added, and the latter operation is optional.

PharmGNN

Applying the RG-MPNN architecture, we proposed a model called PharmGNN (short for pharmacophore-based graph neural network), which was designed by adding RG-MPNN architecture based on the residual graph neural network (shorted as RGNN). As shown in Figure 1 (b), the PharmGNN follows the pattern of RG-MPNN architecture, going through the four processes mentioned above in turn: a message-passing phase at atom level, a graph reducing phase, a message-passing phase at RG level and a molecule readout phase.

Within the message-passing phase at atom level, when gathering messages from neighbor atoms, our model adopts the attention mechanism, which was proposed by Velickovic and Bengio et al. in constructing GAT [43] model. The core idea of this mechanism is to receive messages from neighbors according to a certain weight that is calculated based on the feature vectors of the center atom and its neighbor atom. This mechanism is in line with our basic chemical understanding, that is, each atom is influenced by its neighbor atom with different degrees, which may lie in factors such as the strength of the electrostatic attraction, the shift of the electron cloud, etc. Moreover, in the update process, the new hidden state of atom h_{atom}^k is obtained by adding attention message and residuals. It is worth emphasizing that there are two residual items based on skip-connection mechanism, the linearly transformed values of the previous two hidden states h_{atom}^{k-1} and h_{atom}^{k-2} (when $k \geq 2$), since the skip-connection residual can effectively avoid the problem of gradient disappearance during process of the network training.

The graph reducing process can be regarded as a pharmacophore-based graph reduction along with a one-step message-passing. Firstly, the graph G is reduced into RG (reduced graph) according to the previously defined pharmacophore-based RG pooling rules, and the sum of vectors of child nodes inside one pharmacophore is regarded as the initial state S_{rg} . Then it comes to the message-gathering step in MPNN architecture. Each pharmacophore node receives the messages from their child nodes through their attention weights. This is consistent with the chemical intuition that each atom contributes differently to its pharmacophore. Finally, the pharmacophore nodes are updated through a GRU (gated recurrent unit) [44], with the expectation that the network can weigh the initial state S_{rg} and the messages passed over.

During the message-passing phase at RG level, the operation is similar to the MPNN step in the graph reducing process, that is, the attention mechanism is applied to gather messages, and then the GRU is used to update the nodes.

The implementation of the molecule readout is very similar to that of graph reducing process since the readout operation can be regarded as a special case of graph reducing, that it, all child nodes belong to one super node.

Model Evaluation

In this work, we used two methods to split each dataset into a training set, a validation set and a test set. The first was to split randomly according to the ratio of 8: 1: 1. Noted that in each round of comparing the performance of algorithms, the random seed was kept the same to eliminate the impact of different dataset divisions. The second is scaffold splitting. The core idea of scaffold splitting is to put molecules with different scaffolds into different sets to evaluate the prediction ability on new scaffolds that not encountered during training. Additionally, each model was repeated for five rounds.

For the benchmark data sets, we used RMSE (root mean square error) to evaluate regression tasks, and AUC (area under curve) to evaluate classification tasks, to be consistent with other models on benchmark evaluation. For kinase data sets, two indicators were used to evaluate the model – AUC and MCC

(Matthews correlation coefficient), as the two are not sensitive to data imbalance [45]. In different scenarios, the best model can be selected according to different indicators. The AUC indicator is suitable to select models in the scenarios where the correct sorting is counted such as shortlisting compounds for bioactivity testing in virtual screening, since it measures the ability of model to rank positive samples before negative ones. While the MCC indicator is suitable for the models used in the scenarios where the correct classification is counted such as evaluating whether the molecule is active or toxic.

Model training

Pytorch [46], a deep-learning framework, was used for developing all parts of the PharmGNN, RDKit (v.2018.09.2) [47] for processing molecules and Pytorch Geometric [48] for transforming a molecule into a graph. MSELoss and CrossEntropyLoss were used as loss functions for regression and classification tasks, respectively, whereas Adam [49] was used for gradient descent optimization.

Results And Discussion

Model performance on benchmark data sets

To compare the performance of PharmGNN with other existing GNNs, we used the benchmark data sets to test these models. Considering the reported performance and code availability of the models, we selected MPNN [28] and AttentiveFP [30] for comparison. The former is the classic MPNN model as the baseline, and the latter is the model with superior performance reported in the recent period. When comparing performance, we not only listed the performance reported on the original literature, but also reproduced these models locally, and used the same test set (randomly split) to test these models in each round, aiming to eliminate the impact of different data sets.

Table 2 and Figure S1 summarize the performance on the benchmark data sets covering a variety of molecular bioactivities, toxicities, and physical chemistry properties. On the three regression tasks (ESOL, FreeSolv and Lipophilicity), although our model (PharmGNN) did not perform better than the reported AttentiveFP, it performed best in the comparison of locally reproduced models (see numbers in bold in Table 2). This may be due to different data set divisions and slight differences in training strategies.

Table 2
Model performance on benchmark datasets

Category	Dataset	# Compounds ^a	Task Type	# Tasks	Metrics	AttentiveFP ^b	MPNN ^c	AttentiveFP (reproduced)	MPNN (reproduced)	PharmGNN (our model)
Physical chemistry	ESOL	1128 (1030)	Regression	1	RMSE	0.503 ± 0.076	0.580 ± 0.030	0.650 ± 0.123	0.853 ± 0.057	0.605 ± 0.037
	FreeSolv	642 (566)	Regression	1	RMSE	0.736 ± 0.037	1.150 ± 0.120	1.162 ± 0.180	1.255 ± 0.229	0.939 ± 0.067
	Lipophilicity	4200 (4085)	Regression	1	RMSE	0.578 ± 0.018	0.719 ± 0.031	0.627 ± 0.055	0.662 ± 0.019	0.579 ± 0.020
bioactivity	MUV	93087 (91470)	Classification	17	ROC-AUC	0.843±0.012	-	0.772 ± 0.031	0.740 ± 0.012	0.819 ± 0.011
	HIV	41127 (38686)	Classification	1	ROC-AUC	0.832 ± 0.021	-	0.815 ± 0.022	0.803 ± 0.015	0.824 ± 0.019
	BACE	1513 (1419)	Classification	1	ROC-AUC	0.850 ± 0.012	-	0.868 ± 0.024	0.846 ± 0.026	0.889 ± 0.018
Physiology or toxicity	BBBP	2039 (1928)	Classification	1	ROC-AUC	0.920 ± 0.015	-	0.888 ± 0.025	0.824 ± 0.038	0.879 ± 0.035
	Tox21	7831 (7372)	Classification	12	ROC-AUC	0.858 ± 0.014	-	0.852 ± 0.025	0.836 ± 0.018	0.873 ± 0.008
	ToxCast	8575 (8058)	Classification	617	ROC-AUC	0.805±0.022	-	0.860 ± 0.012	0.848 ± 0.008	0.866 ± 0.009
	SIDER	1427 (1270)	Classification	27	ROC-AUC	0.637 ± 0.017	-	0.827 ± 0.008	0.812 ± 0.012	0.825 ± 0.014
	ClinTox	1478 (1437)	Classification	2	ROC-AUC	0.940 ± 0.018	-	0.940 ± 0.029	0.941 ± 0.026	0.965 ± 0.011

a: number of compounds used in MPNN and AttentiveFP models / number of compounds used in PharmGNN model.

b: model performance taken from reference³⁰.

c: model performance taken from reference²⁸.

For classification tasks, our model performed best on half of the eight ones (BACE, Tox21, ToxCast and ClinTox), compared with all the models either reported or locally reproduced (see numbers both in bold and underlined in Table 2), which indicating that our model performed well for bioactivity and tox prediction

tasks. It is worth emphasizing that toxicity data sets are all multi-task classification tasks, so this indicates the potential of our models in multi-task prediction, and more extensive experiments are worthy to test this hypothesis.

Overall, from the comparison of the models, our model PharmGNN performed slightly better than AttentiveFP, and to a large extent better than the MPNN model, suggesting our model is a promising method to solve problems in drug discovery especially for bioactivity and tox prediction problems.

Predicting kinase bioactivities

PharmGNN integrates the concept of pharmacophore from drug discovery field into the graph neural network, aiming to improve the predicting ability for molecular bioactivity towards targets of interest. Under these circumstances, we constructed a series of kinase molecular activity data sets, aiming to test the model's ability to predict molecular bioactivity on a larger scale. In addition to the data sets split randomly, we also trained and evaluated models on data sets split on scaffolds. This is because there has been research shown that the model trained on the data sets split based on scaffolds has better generalization ability in industry, given that this split can simulate the scene of the data set split by time periods in industry [38].

Table 3 lists the model performance under two indicators – MCC and AUC. Here, we only compare the models in terms of AUC. On randomly divided data sets, our model performed slightly better than AttentiveFP, achieving the best performances on seven out of ten, while AttentiveFP performed best on the other three data sets. In addition, both models mentioned above outperformed the MPNN model. The result is in line with our expectations because the pooling method is very important for task prediction. In theory, the hierarchical pooling of PharmGNN and the attention pooling of AttentiveFP can extract representation or fingerprints more effectively than the average pooling in typical models with MPNN architecture.

Table 3
Model performance on kinase datasets.

Dataset	Splitting	MPNN		AttentiveFP		PharmGNN	
		MCC	AUC	MCC	AUC	MCC	AUC
EGFR	random	0.699 ± 0.022	0.923 ± 0.009	0.729 ± 0.017	0.933 ± 0.010	0.738 ± 0.029	0.942 ± 0.008
	scaffold	0.648 ± 0.03	0.901 ± 0.010	0.684 ± 0.029	0.908 ± 0.006	0.706 ± 0.020	0.925 ± 0.007
BRAF	random	0.703 ± 0.052	0.916 ± 0.020	0.753 ± 0.035	0.938 ± 0.006	0.774 ± 0.020	0.915 ± 0.012
	scaffold	0.614 ± 0.040	0.883 ± 0.006	0.685 ± 0.018	0.905 ± 0.008	0.677 ± 0.044	0.915 ± 0.006
PIM1	random	0.691 ± 0.069	0.933 ± 0.029	0.741 ± 0.026	0.957 ± 0.009	0.758 ± 0.029	0.951 ± 0.013
	scaffold	0.612 ± 0.050	0.881 ± 0.024	0.681 ± 0.021	0.921 ± 0.006	0.710 ± 0.042	0.928 ± 0.007
mTOR	random	0.591 ± 0.037	0.888 ± 0.036	0.641 ± 0.033	0.927 ± 0.010	0.674 ± 0.038	0.921 ± 0.008
	scaffold	0.408 ± 0.060	0.792 ± 0.033	0.588 ± 0.022	0.876 ± 0.011	0.574 ± 0.015	0.886 ± 0.009
AKT1	random	0.669 ± 0.068	0.905 ± 0.035	0.751 ± 0.038	0.933 ± 0.011	0.771 ± 0.031	0.941 ± 0.014
	scaffold	0.605 ± 0.041	0.883 ± 0.016	0.657 ± 0.029	0.914 ± 0.010	0.649 ± 0.023	0.910 ± 0.008
AURKA	random	0.634 ± 0.061	0.892 ± 0.020	0.665 ± 0.021	0.909 ± 0.010	0.690 ± 0.028	0.917 ± 0.007
	scaffold	0.471 ± 0.031	0.793 ± 0.012	0.475 ± 0.023	0.807 ± 0.008	0.522 ± 0.037	0.836 ± 0.006
BTK	random	0.670 ± 0.065	0.915 ± 0.022	0.748 ± 0.083	0.947 ± 0.017	0.759 ± 0.042	0.954 ± 0.017
	scaffold	0.545 ± 0.044	0.849 ± 0.021	0.626 ± 0.044	0.902 ± 0.014	0.682 ± 0.031	0.893 ± 0.015
CDK2	random	0.567 ± 0.041	0.865 ± 0.019	0.624 ± 0.074	0.886 ± 0.021	0.652 ± 0.050	0.902 ± 0.013
	scaffold	0.376 ± 0.052	0.752 ± 0.024	0.412 ± 0.026	0.773 ± 0.022	0.495 ± 0.015	0.820 ± 0.008
MAP4K2	random	0.457 ± 0.131	0.792 ± 0.059	0.484 ± 0.123	0.813 ± 0.041	0.540 ± 0.046	0.863 ± 0.025
	scaffold	0.174 ± 0.039	0.578 ± 0.037	0.277 ± 0.03	0.652 ± 0.033	0.306 ± 0.045	0.706 ± 0.034
CK1	random	0.156 ± 0.162	0.673 ± 0.111	0.313 ± 0.128	0.751 ± 0.047	0.433 ± 0.090	0.800 ± 0.065
	scaffold	-0.020 ± 0.051	0.576 ± 0.029	0.159 ± 0.084	0.653 ± 0.011	0.333 ± 0.107	0.687 ± 0.033

It can be seen from Table 3 and Figure 3 that the model tended to perform better on data sets with a relatively large number of molecules. On the other hand, it showed the limitation of GNNs for task prediction on small data sets, as the model did not perform well on the two small data sets of CK1 and MAP4K2. This limitation is mainly due to the relatively larger number of parameters of GNN that need to be trained and the parameters having not been fully trained would lead to underfitted models if the data set is too small. In addition, the data imbalance may be one another reason for the bad prediction performance. Nevertheless, PharmGNN performed significantly better than AttentiveFP on these two tasks, and equally to or slightly better than AttentiveFP on most of other tasks.

On the models that based on scaffold split methods, we can see almost the same trend as that based on random split, but the scaffold split can be more challenging: the model performance is generally lower than the case of random split.

Applying RG-MPNN architecture

From the tests on different data sets (benchmark data sets and kinase data sets), it has been proved that the PharmGNN model is effective, which means that the RG-MPNN architecture is effective when applied to a basic RGNN (see the previous methods section). In this part, we aim to explore the effect of applying this architecture to other models with MPNN architecture, to see whether the RG-MPNN architecture can help MPNNs improve their predictive ability.

We compared three sets of models in pairs (RGNN vs PharmGNN, AttentiveFP vs PharmAttentiveFP, MPNN vs PharmMPNN), each consisting of two models before and after being applied RG-MPNN architecture. Totally, we built 600 models across ten kinase datasets, with two splitting methods (random and scaffold splitting) and repeated by five times. The detailed performance of each model is shown in Table S3 and S4. The effects on prediction performance that RG-MPNN architecture brings to basic models are shown in Figure 4 (detailed difference can be seen in Table S5 and S6), where the following points can be concluded: a) the RG-MPNN architecture can improve most basic models, including all the RGNNs, 80% of AttentiveFP models, and 55% of MPNNs, with the improvements of AUC ranging from -0.010 to 0.046; b) the RG-MPNN architecture helps RGNN gain more improvement on scaffold splits than on random splits, while the gains of RG-MPNN architecture for the other two models (AttentiveFP and MPNN), could not see a consistent trend between these two splitting methods. This indicated that the basic model RGNN is more compatible with the RG-MPNN architecture.

Overall, from the experimental results, RG-MPNN architecture can improve the models with MPNN architecture to varying degrees, increasing our prospects for applying this architecture in industry.

Visualization and analysis of task-learned fingerprints

We expect a good QSAR model not only to accurately predict the potential activity of each molecule, but also to help pharmacologists visually observe why some molecule is active (such as what substructure or property it has), and to measure the effect the structural differences on activity between two molecules. With this expectation, we extracted the hidden state of molecules in PharmGNN as the task-learned representations and trained the representations into spatial arrangement via a self-organizing map (SOM).^{50,51} Figure 5 (a) shows the molecular representation distribution on a two-dimensional map where being projected to adjacent neurons means that two molecules are similar at the level of task-learned representation. It can be seen from the figure that active molecules and inactive molecules are mapped to different zones, with the diagonal line as the dividing line, the upper left corner mostly lies the active samples, and the lower right corner lies inactive ones. The conflicting neurons are concentrated near the diagonal, which is the junction of the two types of molecules. Molecules in this area are the most challenging ones to distinguish since these molecules have similar representations but different labels. Notably, prediction credibility of model is also implied in the map: the closer to the upper left and lower right corner, the higher the credibility to be active molecules and inactive ones, respectively.

To explore whether the task-learned representation obtained by the PharmGNN is superior to the fixed (or expert-defined) fingerprint, we took ECFP_4 [52], a representative and widely used fingerprint type, for comparison. The SOM obtained by ECFP_4 fingerprints is shown in Figure 5 (b), and the activity labels of the molecules are marked on the figure. A trend can be seen that negative samples tend to gather in the center and positive ones on the edge, but it is obvious that its differentiating effect is not as good as the task-learned fingerprints that shown in Figure 5 (a).

Furthermore, to visually see the difference between similar molecules under different representations, we took two typical AURKA inhibitors (VX-680 and pha-739358) as examples and dive to look at their analogs under the task-learned representation and ECFP_4 system, respectively (see Figure S2 and S3). It can be concluded that in the two representation systems, the molecular structures in the same neuron are very similar, but different analogs are extracted. In terms of the consistency of molecular labels in the same neuron, the task-learned representation is better than ECFP, which is consistent with the previously observed phenomenon that the task-learned system has fewer conflicting neurons than the ECFP system. Strikingly, the task-learned system shows the possibility of completing scaffold hopping while ensuring activity.

Activity interpretation for AURKA inhibitors

The task-learned representations are often criticized for being difficult to interpret, and it is difficult to gain intuitive knowledge from them, which is not conducive to the understanding of pharmacologists when applied to the practice of drug discovery. Therefore, we extracted the attention weights to learn the importance of each pharmacophore nodes, aiming at dig and provide some intuitive information to help drug development.

Take the aforementioned pha-739358 (an AURKA inhibitor, shown in Figure 6 (a)) as an example, we have annotated the degrees of the effect of the pharmacophore node in the molecule on the activity, as shown in Figure 6 (b). We can see that Y (aliphatic and positively ionizable), Co (acyclic and donor), Ti (aromatic and donor), Ni (acyclic and acceptor) and Sc (aromatic and no feature) play important roles in molecular activity. Combining the ligand-receptor interaction diagram in the crystal complex (PDB ID: 2J50) for comparative analysis (Figure 6 (c)), we found that the two findings have a certain consistency. For example, Co and Ti, which form hydrogen bonds to the backbone, are labeled, which is consistent with the interactions appeared in the crystal complex. However, the findings are not completely overlapped, which is not surprising though, since one interpretation is derived from the interaction with receptor and the other from the knowledge from ligands. The interpretations of the two can be used for reference in practical applications. After all, in drug development practice, the more information from more perspectives, the more novel ideas can be provided, and the chance of discovering new drugs will increase.

Conclusions

With the goal of integrating more prior chemical knowledge to establish predictive GNN models for chemical properties, we introduced a pharmacophore-based RG pooling method for MPNNs that can extract pharmacophore information hierarchically from molecular graphs. Therefore, in this work, we proposed the PharmGNN model and compared it with the state-of-the-art GNN algorithms on the MoleculeNet benchmarks. The results showed that our models outperformed other models on ten out of twelve tasks. These results were also transferred to ten kinase data sets which were selected because they are

representative kinase targets from each kinase family with great potential to be drug targets. Models built on these kinase data sets can be used in drug screening for inhibitors of these kinases. Furthermore, three groups of comparative experiments on the kinase data sets by applying the RG-MPNN architecture were conducted, suggesting that this architecture can generally improve the predictive power of many MPNNs. It showed that this architecture had the potential to be extended to more MPNNs. Moreover, the SOM analysis of PharmGNN showed that the active molecules had a clustering effect, and it is possible to identify molecules with similar activities but different scaffolds. Additionally, the importance of pharmacophore nodes gained from our models will help chemists gain insights for hit discovery and lead optimization.

Abbreviations

DNNs: deep neural networks; QSAR/QSPR: quantitative structure-activity (property) relationship; LBDD: ligand-based drug design; ML: machine learning; SVMs: support vector machines; NB: Naïve Bayes; ANN: artificial neural network; RF: random forest; RNN: recurrent neural network; GNN: graph neural network; MPNN: message-passing neural network; D-MPNN: directed MPNN; R-GCN: relational graph convolutional networks; GSN: graph substructure networks; RGs: reduced graphs; RG-MPNN architecture: reduced-graph message-passing neural network architecture.

Declarations

Availability of data and materials

The eleven benchmark data sets and ten kinase data sets are available at Github (<https://github.com/ChloeKong/PharmGNN>).

The code is available at GitHub (<https://github.com/ChloeKong/PharmGNN>).

Competing interests

The authors declare no competing financial interest.

Funding

Not applicable.

Authors' contributions

YK implemented the method and evaluated the models, XMZ, RZL, ZWY, HYY, BWZ, JLW performed the analysis. BJQ and AXY provided the main idea of this work. All authors read and approved the final manuscript.

Acknowledgements

Not applicable.

Associated content

Supplementary tables (Tables S1- S6) and supplementary figures (Figures S1- S3) are shown in the SI.docx.

Funding

There is no funding for this work.

References

1. Pak M, Kim S (2017) A review of deep learning in image recognition. In: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT). pp 1–3
2. Young T, Hazarika D, Poria S, Cambria E (2018) Recent Trends in Deep Learning Based Natural Language Processing. IEEE Comput Intell Mag 13:55–75. <https://doi.org/10.1109/MCI.2018.2840738>
3. Zhavoronkov A, Ivanenkov YA, Aliper A et al (2019) Deep learning enables rapid identification of potent DDR1 kinase inhibitors. Nat Biotechnol 37:1038–1040. <https://doi.org/10.1038/s41587-019-0224-x>
4. Yang X, Wang Y, Byrne R et al (2019) Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. Chem Rev 119:10520–10594. <https://doi.org/10.1021/acs.chemrev.8b00728>
5. Mater AC, Coote ML (2019) Deep Learning in Chemistry. J Chem Inf Model 59:2545–2559. <https://doi.org/10.1021/acs.jcim.9b00266>

6. Chen H, Engkvist O, Wang Y et al (2018) The rise of deep learning in drug discovery. *Drug Discov Today* 23:1241–1250. <https://doi.org/10.1016/j.drudis.2018.01.039>
7. Elton DC, Boukouvalas Z, Fuge MD, Chung PW (2019) Deep learning for molecular design - a review of the state of the art. *Mol Syst Des Eng* 4:828–849. <https://doi.org/10.1039/C9ME00039A>
8. Wieder O, Kohlbacher S, Kuenemann M et al (2020) A compact review of molecular property prediction with graph neural networks. *Drug Discov Today: Technologies* S 1740674920300305. <https://doi.org/10.1016/j.ddtec.2020.11.009>
9. Goh GB, Hodas NO, Vishnu A (2017) Deep learning for computational chemistry. *J Comput Chem* 38:1291–1307. <https://doi.org/10.1002/jcc.24764>
10. Shen J, Nicolaou CA (2019) Molecular property prediction: recent trends in the era of artificial intelligence. *Drug Discov Today: Technologies* 32–33:29–36. <https://doi.org/10.1016/j.ddtec.2020.05.001>
11. Cherkasov A, Muratov EN, Fourches D et al (2014) QSAR Modeling: Where Have You Been? Where Are You Going To? *J Med Chem* 57:4977–5010. <https://doi.org/10.1021/jm4004285>
12. Sharma V, Wakode S, Kumar H (2021) Chapter 2 - Structure- and ligand-based drug design: concepts, approaches, and challenges. In: Sharma N, Ojha H, Raghav PK, Goyal R k. (eds) *Chemoinformatics and Bioinformatics in the Pharmaceutical Sciences*. Academic Press, pp 27–53
13. Baskin II (2020) The power of deep learning to ligand-based novel drug discovery. *Expert Opin Drug Dis* 15:755–764. <https://doi.org/10.1080/17460441.2020.1745183>
14. Noble WS (2006) What is a support vector machine? *Nat Biotechnol* 24:1565–1567. <https://doi.org/10.1038/nbt1206-1565>
15. Zernov VV, Balakin KV, Ivaschenko AA et al (2003) Drug Discovery Using Support Vector Machines. The Case Studies of Drug-likeness, Agrochemical-likeness, and Enzyme Inhibition Predictions. *J Chem Inf Comput Sci* 43:2048–2056. <https://doi.org/10.1021/ci0340916>
16. Ballabio D, Grisoni F, Consonni V, Todeschini R (2019) Integrated QSAR Models to Predict Acute Oral Systemic Toxicity. *Mol Inf* 38:1800124. <https://doi.org/10.1002/minf.201800124>
17. Zhang H, Cao Z-X, Li M et al (2016) Novel naïve Bayes classification models for predicting the carcinogenicity of chemicals. *Food Chem Toxicol* 97:141–149. <https://doi.org/10.1016/j.fct.2016.09.005>
18. Mandlik V, Bejugam PR, Singh S (2016) Chapter 6 - Application of Artificial Neural Networks in Modern Drug Discovery.. In: Puri M, Pathak Y, Sutariya VK et al (eds) *Artificial Neural Network for Drug Design, Delivery and Disposition*. Academic Press, Boston, pp 123–139
19. Zou J, Han Y, So S-S (2008) Overview of artificial neural networks. *Methods Mol Biol* 458:15–23. https://doi.org/10.1007/978-1-60327-101-1_2
20. Breiman L (2001) Random Forests. *Mach Learn* 45:5–32. <https://doi.org/10.1023/A:1010933404324>
21. Jiang D, Wu Z, Hsieh C-Y et al (2021) Could graph neural networks learn better molecular representation for drug discovery? A comparison study of descriptor-based and graph-based models. *J Cheminform* 13:12. <https://doi.org/10.1186/s13321-020-00479-8>
22. Lane TR, Foil DH, Minerali E et al (2021) Bioactivity Comparison across Multiple Machine Learning Algorithms Using over 5000 Datasets for Drug Discovery. *Mol Pharmaceutics* 18:403–415. <https://doi.org/10.1021/acs.molpharmaceut.0c01013>
23. Vaswani A, Shazeer N, Parmar N et al (2017) Attention is All you Need. *Advances in Neural Information Processing Systems*. Curran Associates, Inc
24. Karpov P, Godin G, Tetko IV (2020) Transformer-CNN: Swiss knife for QSAR modeling and interpretation. *J Cheminform* 12:17. <https://doi.org/10.1186/s13321-020-00423-w>
25. Zhang X, Zhao J, LeCun Y (2015) Character-level Convolutional Networks for Text Classification. *Advances in Neural Information Processing Systems*. Curran Associates, Inc
26. Wu Z, Pan S, Chen F et al (2021) A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32:4–24. <https://doi.org/10.1109/TNNLS.2020.2978386>
27. Sun M, Zhao S, Gilvary C et al (2020) Graph convolutional networks for computational drug development and discovery. *Briefings Bioinform* 21:919–935. <https://doi.org/10.1093/bib/bbz042>
28. Gilmer J, Schoenholz SS, Riley PF et al (2017) Neural Message Passing for Quantum Chemistry. In: *Proceedings of the 34th International Conference on Machine Learning*. PMLR, pp 1263–1272
29. Yang K, Swanson K, Jin W et al (2019) Analyzing Learned Molecular Representations for Property Prediction. *J Chem Inf Model* 59:3370–3388. <https://doi.org/10.1021/acs.jcim.9b00237>
30. Xiong Z, Wang D, Liu X et al (2020) Pushing the Boundaries of Molecular Representation for Drug Discovery with the Graph Attention Mechanism. *J Med Chem* 63:8749–8760. <https://doi.org/10.1021/acs.jmedchem.9b00959>
31. Schlichtkrull M, Kipf TN, Bloem P et al (2017) Modeling Relational Data with Graph Convolutional Networks. *arXiv:170306103*
32. Bouritsas G, Frasca F, Zafeiriou S, Bronstein MM (2021) Improving Graph Neural Network Expressivity via Subgraph Isomorphism Counting. *arXiv:200609252*
33. Yang S-Y (2010) Pharmacophore modeling and applications in drug discovery: challenges and recent advances. *Drug Discov Today* 15:444–450. <https://doi.org/10.1016/j.drudis.2010.03.013>
34. Harper G, Bravi GS, Pickett SD et al (2004) The reduced graph descriptor in virtual screening and data-driven clustering of high-throughput screening data. *J Chem Inf Comput Sci* 44:2145–2156. <https://doi.org/10.1021/ci049860f>
35. Birchall K, Gillet VJ, Willett P et al (2009) Use of reduced graphs to encode bioisosterism for similarity-based virtual screening. *J Chem Inf Model* 49:1330–1346. <https://doi.org/10.1021/ci900078h>

36. Pogány P, Arad N, Genway S, Pickett SD (2019) De Novo Molecule Design by Translating from Reduced Graphs to SMILES. *J Chem Inf Model* 59:1136–1146. <https://doi.org/10.1021/acs.jcim.8b00626>
37. Ying Z, You J, Morris C et al (2018) Hierarchical Graph Representation Learning with Differentiable Pooling. *Advances in Neural Information Processing Systems*. Curran Associates, Inc
38. Wu Z, Ramsundar B, Feinberg EN et al (2018) MoleculeNet: a benchmark for molecular machine learning. *Chem Sci* 9:513–530. <https://doi.org/10.1039/C7SC02664A>
39. Mendez D, Gaulton A, Bento AP et al (2019) ChEMBL: towards direct deposition of bioassay data. *Nucleic Acids Res* 47:D930–D940. <https://doi.org/10.1093/nar/gky1075>
40. Stiefl N, Watson IA, Baumann K, Zaliani A (2006) ErG: 2D pharmacophore descriptions for scaffold hopping. *J Chem Inf Model* 46:208–220. <https://doi.org/10.1021/ci050457y>
41. Wollenhaupt S, Baumann K (2014) inSARa: intuitive and interactive SAR interpretation by reduced graphs and hierarchical MCS-based network navigation. *J Chem Inf Model* 54:1578–1595. <https://doi.org/10.1021/ci4007547>
42. Barker EJ, Buttar D, Cosgrove DA et al (2006) Scaffold hopping using clique detection applied to reduced graphs. *J Chem Inf Model* 46:503–511. <https://doi.org/10.1021/ci050347r>
43. Veličković P, Cucurull G, Casanova A et al (2018) Graph Attention Networks
44. Cho K, van Merriënboer B, Bahdanau D, Bengio Y (2014) On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *arXiv:1409.1259*
45. Powers D, Ailab (2011) Evaluation: From precision, recall and F-measure to ROC, informedness, markedness & correlation. *J Mach Learn Technol* 2:2229–3981. <https://doi.org/10.9735/2229-3981>
46. Pytorch (accessed Oct 10, 2021) <https://github.com/pytorch/pytorch>
47. Landrum GI (2006) : RDKit: Open-Source Cheminformatics; <https://rdkit.org/docs/index.html> (accessed Oct 10 2021)
48. Fey M, Lenssen JE (2019) Fast Graph Representation Learning with PyTorch Geometric. *arXiv:1903.02428*
49. Kingma DP, Ba J (2017) Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*
50. Kong Y, Qu D, Chen X et al (2016) Self-Organizing Map (SOM) and Support Vector Machine (SVM) Models for the Prediction of Human Epidermal Growth Factor Receptor (EGFR/ ErbB-1) Inhibitors. *Comb Chem High T Scr* 19:400–411
51. Kong Y, Yan A (2017) QSAR models for predicting the bioactivity of Polo-like Kinase 1 inhibitors. *Chemometr Intell Lab* 167:214–225. <https://doi.org/10.1016/j.chemolab.2017.06.011>
52. Rogers D, Hahn M (2010) Extended-Connectivity Fingerprints. *J Chem Inf Model* 50:742–754. <https://doi.org/10.1021/ci100050t>

Figures

Figure 1

Illustration of our proposed PharmGNN. (a) General architecture of PharmGNN. Taking Acetaminophen as an example, there are four phases that a molecule goes through from the molecular graph to the task prediction. (b) Specific architectural details of each phase of PharmGNN.

Figure 2

Scheme and examples of graph reduction. (a) The graph reduction scheme adopted in this work. The three ways of defining rings and six ways of defining features combine to eighteen types of reduced graphs. And prioritize at the ring and feature level. (b) Comparative examples of molecules and their reduced graphs.

Figure 3

Model performance on kinase datasets in terms of AUC. (a) Model performance based on random splitting datasets. Our model PharmGNN performs slightly better or comparable than AttentiveFP, better than MPNN model. (b) Model performance based on scaffold splitting datasets. Same trend is seen as (a), but performance is generally lower than that of (a).

Figure 4

Effects on predictive performance that RG-MPNN architecture brings to basis models. (a) AUC gains based on random splitting datasets. (b) AUC gains based on scaffold splitting datasets. Most of the AUC gains in both subfigures are positive, which means that the RG-MPNN architecture is helpful to improve predictive performance of models.

Figure 5

Comparison of representation distribution of the task-learned representation and the ECFP₄ fingerprint. (a) The SOM of the representation learned by PharmGNN model on the AURKA bioactivity prediction task. Active molecules and inactive molecules are mapped to different zones, which means the

representation learned by PharmGNN has a good differentiating effect. (b) The SOM of the ECFP_4 fingerprints. The negative samples tend to gather in the center and positive ones on the edge, but ECFP_4's differentiating effect is not as good as the task-learned representation.

Figure 6

Explanation of the importance of pharmacophore or substructure. (a) The chemical structure of the pha-739358, an AURKA inhibitor. (b) The importance of pharmacophore learned by PharmGNN. We can see that Y (aliphatic and positively ionizable), Co (acyclic and donor), Ti (aromatic and donor), Ni (acyclic and acceptor) and Sc (aromatic and no feature) play important roles in AURKA bioactivity. (c) Ligand-receptor interactions in the crystal complex (PDB ID: 2J50). Of note, Nitrogen atoms within Co and Ti form two important hydrogen bonds with receptor, which is consistent with the analysis of pharmacophore importance.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- [GraphicalAbstract.png](#)
- [SI20211221.docx](#)