https://doi.org/10.1186/s12859-024-05698-6

BMC Bioinformatics

RESEARCH

Open Access

Drug–target affinity prediction with extended graph learning-convolutional networks

Haiou Qi[1], Ting Yu[2*], Wenwen Yu[3] and Chenxi Liu[4]

*Correspondence:
3320108@zju.edu.cn

[1] Nursing Department, Sir Run Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310016, China
[2] Operating Room Department, Sir Run Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310016, China
[3] School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
[4] School of Medicine and Health Management, Tongji Medical School, Huazhong University of Science and Technology, Wuhan 430030, China

Abstract

Background: High-performance computing plays a pivotal role in computer-aided drug design, a field that holds significant promise in pharmaceutical research. The prediction of drug–target affinity (DTA) is a crucial stage in this process, potentially accelerating drug development through rapid and extensive preliminary compound screening, while also minimizing resource utilization and costs. Recently, the incorporation of deep learning into DTA prediction and the enhancement of its accuracy have emerged as key areas of interest in the research community. Drugs and targets can be characterized through various methods, including structure-based, sequence-based, and graph-based representations. Despite the progress in structure and sequence-based techniques, they tend to provide limited feature information. Conversely, graph-based approaches have risen to prominence, attracting considerable attention for their comprehensive data representation capabilities. Recent studies have focused on constructing protein and drug molecular graphs using sequences and SMILES, subsequently deriving representations through graph neural networks. However, these graph-based approaches are limited by the use of a *fixed adjacent matrix* of protein and drug molecular graphs for graph convolution. This limitation restricts the learning of comprehensive feature representations from intricate compound and protein structures, consequently impeding the full potential of graph-based feature representation in DTA prediction. This, in turn, significantly impacts the models' generalization capabilities in the complex realm of drug discovery.

Results: To tackle these challenges, we introduce GLCN-DTA, a model specifically designed for proficiency in DTA tasks. GLCN-DTA innovatively integrates a graph learning module into the existing graph architecture. This module is designed to learn a *soft adjacent matrix*, which effectively and efficiently refines the contextual structure of protein and drug molecular graphs. This advancement allows for learning richer structural information from protein and drug molecular graphs via graph convolution, specifically tailored for DTA tasks, compared to the conventional fixed adjacent matrix approach. A series of experiments have been conducted to validate the efficacy of the proposed GLCN-DTA method across diverse scenarios. The results demonstrate that GLCN-DTA possesses advantages in terms of robustness and high accuracy.

Conclusions: The proposed GLCN-DTA model enhances DTA prediction performance by introducing a novel framework that synergizes graph learning operations with graph convolution operations, thereby achieving richer representations. GLCN-DTA does not distinguish between different protein classifications, including structurally ordered and intrinsically disordered proteins, focusing instead on improving feature representation. Therefore, its applicability scope may be more effective in scenarios involving structurally ordered proteins, while potentially being limited in contexts with intrinsically disordered proteins.

Keywords: Drug–target affinity prediction, Deep learning, Drug discovery, Graph learning-convolutional networks

Background

With the U.S. Food and Drug Administration (FDA) approval taking up to 17 years and costing $2.6 billion [1–5], drug development faces challenges, particularly under strict market scrutiny [6]. Drug repurposing and repositioning have emerged as key strategies to expedite and economize drug development [7], prompting a research focus on accurate prediction of drug–target affinity (DTA) [8–12]. The ability to predict binding interactions between small molecules and targets can significantly streamline the process of identifying lead compounds, thus expediting drug research and development.

Traditional methods based on biological experiments, e.g., high-throughput screening experiments, can be effective at predicting DTA, but they are too cumbersome and demand substantial amounts of money and time [13]. Meanwhile, the vast array of drug-like compounds, estimated in the millions [14], along with numerous potential targets, poses a significant challenge to the practical application of this technology. The expansion of bioinformatics databases in recent years has resulted in a substantial collection of biological experimental data, paving the way for the advent of computational approaches [5].

Existing computational approaches are roughly categorized into two types: ligand-based and structure-based approaches, as shown in Fig. 1. Ligand-based methods [15] are based on the concept that ligands with similar chemical properties are likely to have similar biological functions and bind to similar target proteins [16–19]. This approach focuses on the information about the ligand rather than the target protein's structure. However, when the number of available ligands is low, the accuracy of predictions decreases. Structure-based methods, like docking simulations [20], use the 3D structures of drugs and target proteins to predict how well they bind. This is done using molecular docking and dynamics simulations [21–24]. While this method is accurate, it tends to be time-consuming. For proteins with available structural and site information, detailed simulations using molecular simulation and docking can yield highly precise results. However, many proteins lack such structural data, leading to molecular docking and dynamics simulations inapplicable. Sequence-based methods are an alternative method to predict their drug molecule binding affinity when there is lacking structural information on proteins. But the complexity of protein and small molecule structures makes accurately describing and extracting the features of targets and drugs a challenging aspect of affinity prediction. This challenge has become a focal point in computer-aided medicine research, particularly with the surge in deep learning applications over the last decade.

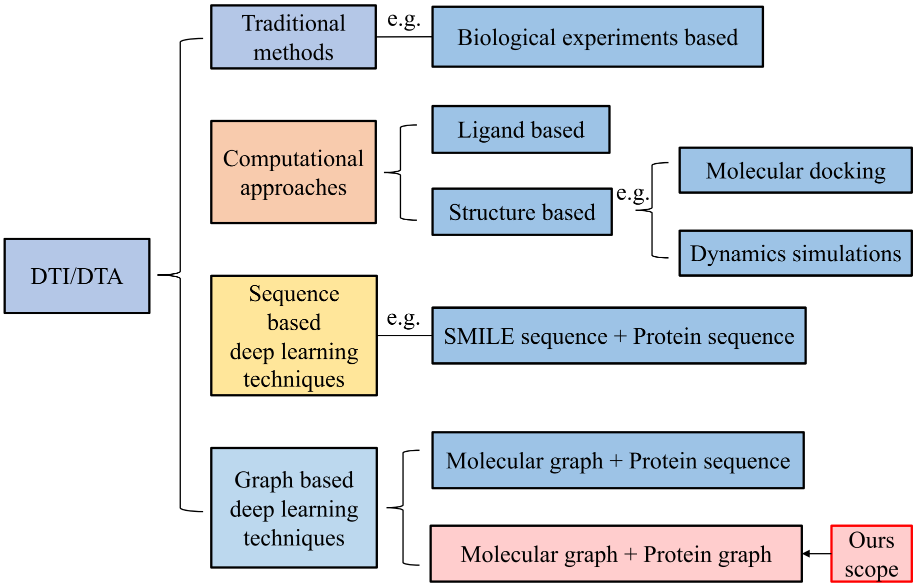

DTI/DTA
Traditional methods
e.g.
Biological experiments based
Computational approaches
Ligand based
Structure based
e.g.
Molecular docking
Dynamics simulations
Sequence based deep learning techniques
e.g.
SMILE sequence + Protein sequence
Graph based deep learning techniques
Molecular graph + Protein sequence
Molecular graph + Protein graph
Ours scope

Fig. 1 Summary of representative methods relevant to DTI and DTA prediction. Our methods apply to the scope of graph-based deep learning methods for DTA prediction

Currently, sequence-based techniques are leading the way in computational biology. The development of deep learning has inspired a wave of early studies to employ neural network architectures for the extraction of features from protein sequences and drug molecules SMILES (simplified molecular input line entry specification) [25], thereby more effectively learning about the potential characteristics of target proteins and drugs. Depending on the objectives, sequence-based methodologies are classified into two groups: drug–target interaction (DTI) prediction and DTA prediction. Specifically, within any given drug–target pair, deep learning algorithms are used to extract distinct representations of the drug and target. These are then concatenated into one comprehensive vector that is used for the final predictive analysis. DTI is a binary classification task used for determining whether the drug can bind to the target or not. For instance, TransformerCPI [26] employs a transformer-based neural network [27] to analyze sequence data for predicting drug–target interaction. NeoDTI [28] combines data from various heterogeneous networks, utilizing representations of drugs and targets that preserve their topological features for predicting interactions. Hu et al. [29] introduced a CNN-based technique for drug–target interaction prediction, incorporating both 1D and 2D structural descriptors of the drug and the protein's sequence as inputs to the network. Ezzat et al. [30] further refine the prediction accuracy by using decision tree methods and kernel ridge regression for feature dimension reduction and ensemble learning.

DTA prediction, distinct from DTI prediction, aims to accurately predict the precise binding affinity between a drug and a target. This challenge, commonly seen as a regression task, has become increasingly interesting in recent years [31–33]. At present, various approaches have yielded impressive results in the realm of affinity prediction. For example, DeepDTA [34] utilizes a pair of convolutional neural networks (CNNs) to separately extract features from the drug and protein, later combining these features for predicting affinity. Additionally, DeepDTA has gathered previous research data to develop two benchmark datasets, where drugs are denoted in SMILES format and proteins by their sequences. This approach, involving two tailored convolution networks for molecule and protein representation, has shown effective results in the benchmarks. WideDTA [35], an advancement of DeepDTA, integrated Live Max Common Substructure (LMCS) and Protein Motifs and Domains (PDM) into its structure. It uses four separate convolutional neural networks (CNNs) to encode these components into four unique representations. Huang et al. [36] introduced an innovative fingerprint feature vector for molecules, and utilized a Pseudo Substitution Matrix Representation (Pseudo-SMR) to depict protein sequences, aimed at enhancing drug–target interaction predictions. DeepPurpose [37] uses two specialized encoders for SMILES and sequence representation, combining the functionalities of CNN, RNN (recurrent neural network), and Transformer structures. For molecule representation, molecular fingerprints serve as a common technique, where the structural details of a molecule are encoded into strings or binary digits. Techniques such as extended connectivity fingerprints [38], atom environment descriptors (MOLPRINT2D) [39], and molecular access system keys (MACCS) [40] are examples of this approach. Altae-Tran et al. [41] detailed methods for learning significant small-molecule representations in scenarios where data is limited. Although sequence-based approaches have seen successes, they overlook the topological aspects of drug and protein molecules. Relying solely on CNN technology falls short in fully capturing the sequence details of drugs and proteins. Acknowledging the interactions between atoms and assigning varying weights and attention to them automatically is crucial.

Recently, graph neural networks (GNNs) [42] have gained recent prominence in bioinformatics for their ability to efficiently process non-Euclidean spatial data [5]. Utilizing the intrinsic qualities of nodes and their relational dynamics, GNNs effectively extract key feature information, essential for accurately identifying and predicting vertices or edges. Given the intricate correlation and diversity in biological data, molecular graphs aptly represent biological details, especially in terms of molecular structure and intermolecular functional relations. GNN models stand out in pinpointing molecular features by assimilating information from neighboring nodes to understand both local and overarching network details. The Graph Convolutional Network (GCN) [43] and Graph Attention Network (GAT) [44] are commonly used models within the GNN framework, and their application in computer-aided drug design is progressively growing, e.g., graph-based DTA. GraphDTA [45] integrated GNN into DTA prediction by creating molecule graphs where atoms serve as nodes and bonds as edges, effectively representing drug molecules. In this model, CNNs are employed to extract high-level representations from protein sequences, while GNN models are applied to the molecular graphs, enhancing the performance of DTA predictions. MCN-CPI [46] and PADME [47] similarly construct graphs for molecule depiction and utilize GNNs for feature extraction in DTA prediction. The success of these methods underscores the effectiveness of GNNs in accurately characterizing small molecules.

Although these methods are adept at forming graphs to represent molecules and explore their structural properties, they tend to neglect the structural information embedded within protein sequences. In recent developments, various graph-based works have focused on predicting drug–target interactions by employing both molecule graphs and protein graphs, as opposed to using proteins' primary sequences. For example, DGraphDTA [48] converts protein sequences into contact maps for graph construction. Concurrently, drug molecules are transformed from SMILES to molecular graphs using RDKit [49]. Both sets of graphs are then fed into a GNN to aid in richer feature extraction. DGraphDTA's reliance on extensive database searches, especially during sequence alignment, results in lengthy processing times and reduced prediction accuracy. To combat this, WGNN-DTA [50] integrates evolutionary scale modeling (ESM) [51] for generating protein graphs, thereby streamlining the prediction process and yielding superior results. Additionally, DGraphDTA's limitation in capturing target representation alterations due to DTIs, a result of its training on a limited protein dataset, is addressed by the newly proposed Graph Early Fusion Affinity (GEFA) [52]. GEFA transforms drug molecule graphs into protein graphs, utilizing an attention mechanism for more effective learning and prediction. To broaden the range of features derived from drug and protein graphs, STAMP-DPI [53] adopts Mol2vec [54] to analyze inter-drug relationships. Concurrently, it uses a pre-trained BERT [55] model to extract embedded representations from the large pool of unlabeled protein sequences provided by TAPE [56].

Utilizing graph-based methodologies, which focus on the topology and functionality of graphs, allows for the exploration of complex protein structure features, thereby increasing the accuracy and reliability of predictions. However, these graph-based approaches are limited by the use of a *fixed adjacent matrix* of protein and drug molecular graphs for graph convolution. This limitation restricts the learning of comprehensive feature representations from intricate compound and protein structures, consequently impeding the full potential of graph-based feature representation in DTA prediction. This, in turn, significantly impacts the models' generalization capabilities in the complex realm of drug discovery.

To tackle these challenges, we introduce GLCN-DTA, a model specifically designed for proficiency in DTA tasks. GLCN-DTA innovatively integrates a graph learning module into the existing graph architecture. This module is designed to learn a *soft adjacent matrix*, which effectively and efficiently refines the contextual structure of protein and drug molecular graphs. This advancement allows for learning richer structural information from protein and drug molecular graphs via graph convolution, specifically tailored for DTA tasks, compared to the conventional fixed adjacent matrix approach. A series of experiments have been conducted to validate the efficacy of the proposed GLCN-DTA method across diverse scenarios. The results demonstrate that GLCN-DTA possesses advantages in terms of robustness and high accuracy. The proposed GLCN-DTA model enhances DTA prediction performance by introducing a novel framework that synergizes graph learning operations with graph convolution operations, thereby achieving richer representations. Note that our method, similar to other graph-based deep learning approaches [48, 50, 52, 53], primarily focuses on better expressing and extracting the characteristics of drugs and targets, fusing these features, and then predicting affinity. It does not consider the classification of proteins, nor does it differentiate between structurally ordered proteins and intrinsically disordered proteins. The focus is more on how to better represent features. Consequently, the applicability of our method might be limited, where it may perform better in scenarios involving structurally ordered proteins but could be limited when dealing with intrinsically disordered proteins.

Methods

GLCN-DTA, a deep learning framework designed for graph-based DTA prediction, is depicted in Fig. 2 and consists of three key steps. (1) Generating drug molecule graph representation: GLCN-DTA inputs drug compounds in SMILES format and converts them into a drug graph that includes atoms and edges, on behalf of the drug's natural properties. Inspired by existing graph network literature [57, 58], GLCN-DTA integrates an extended graph learning mechanism to existing graph convolutional architectures, formatting graph learning-convolutional networks (GLCN). The GLCN is designed to learn a *soft adjacency matrix* which refines the graph's context, highlighting node relationships for DTA tasks and improving feature extraction through graph convolution. This graph convolution process efficiently uses the relationships from the graph learning module, spreading information between nodes in the graph. By default, GLCN-DTA uses a two-layer graph learning-convolutional network for extracting drug graph features. (2) Obtaining protein graph features: GLCN-DTA processes the protein sequence input and converts it into a contact map [48], using established methods such as Pconsc4. From this contact map, a protein graph is formed, applying a 0.5 threshold to create the initial adjacency matrix, as inspired by [48]. This contact map effectively represents the protein's high-dimensional structure and is key to identifying interactions among protein residues. Typically, residues are deemed to be in contact if the Euclidean distance between any two atoms is below a certain threshold [59]. To extract detailed information from these protein graphs, a two-layer graph learning-convolutional network is also employed in this step by default, taking into account their contextual relationships. (3) Performing DTA prediction: GLCN-DTA utilizes the representations from both the molecule graph and the protein graph, combining them into a single, unified vector. This vector is then processed through a two-layer feed-forward network (FFN), enabling the method to arrive at the final binding prediction.

Significantly, GLCN-DTA sets itself apart from other graph-based methods such as DGraphDTA [48], WGNN-DTA [50], GEFA [52], and STAMP-DPI [53]. These methods typically require predefining specific edge types and node connectivities in the graph, or

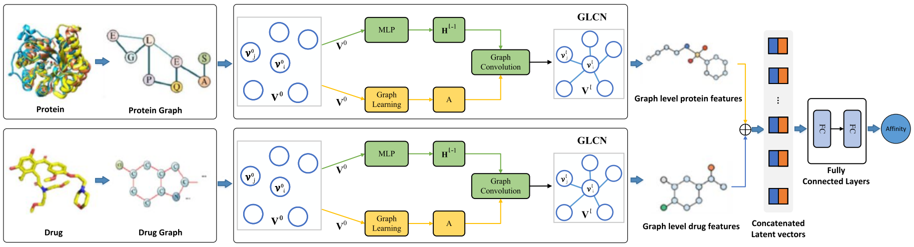


Fig. 2 The architecture of GLCN-DTA. For protein sequences, protein graphs are built upon contact maps derived from their sequences. In the case of molecules, their SMILES representations are utilized as the foundation for graph construction. Once these two graph representations are established, they are processed through two separate graph learning-convolutional networks (GLCN) to extract their respective graph-level features. These representations are then concatenated to predict the affinity via fully connected layers. $\mathbf{V}^l$ donates *node embedding* in the *l*-th graph convolution layer. $\mathbf{H}^l$ represents *hidden features* in the *l*-th graph convolution layer. **A** is *soft adjacent matrix*. ⊕ denotes concatenated operation

they implement thresholds on contact maps to establish a *fixed adjacency matrix*, which might introduce noise or inaccuracies in graph topology leading to aggregate useless and redundancy information between nodes. In contrast, GLCN-DTA leverages its graph learning module to automatically learn hidden node relationships, efficiently refining the graph's structure and filtering useless nodes and being robust to complex structures. This process uncovers a *soft adjacency matrix*, revealing latent node relations and enabling richer node representation through an extended graph learning-convolutional technique.

Molecule graph representation

It can be seen from the Fig. 3 that the molecule graph constructed for extracting the small-molecule representation is the same as that of GraphDTA [45] and DGraphDTA [48]. But for the feature extraction, we use graph learning-convolutional networks, instead of conventional GCN or GAT, to get a richer representation.

Specifically, molecules are commonly depicted using SMILES [25] string, transforming atoms and covalent bonds into sequences of ASCII characters. Conveniently, this format can be reverted to molecular structures using prevalent molecular processing software like RDKit [49]. Subsequently, a molecular graph is formed, representing atoms as nodes and chemical bonds as edges. To elaborate, in the creation of the molecular graph, each atom within the molecule is designated as a node in the graph. An edge is then introduced between two atom nodes if they share a bond. This construction method is illustrated in Fig. 3. Besides, a self-loop is incorporated into the molecular graph, connecting each atom node to itself for better aggregating information of the drug molecule.

Beyond just nodes and edges, it's crucial to define the features of each atom node to ensure they are distinguishable. This is because atoms, the smallest units in chemical reactions, exhibit varied chemical properties due to differences in size and charge. The selected molecular atom node features are the same as those in GraphDTA [45] and WGNN-DTA [50], which is illustrated in Table 1. By detailing the specific features of different atom nodes, the chemical and binding characteristics of small molecules can be more thoroughly represented. Incorporating these factors that influence molecular binding is key to enhancing prediction performance.

Following the molecular graph's construction, GLCN-DTA utilizes a two-layer graph learning-convolutional network for extracting drug graph features. Graph learning-convolutional networks will be elaborated in the subsequent section. A global pooling process also is employed for the extracted representation. Molecules with different sizes

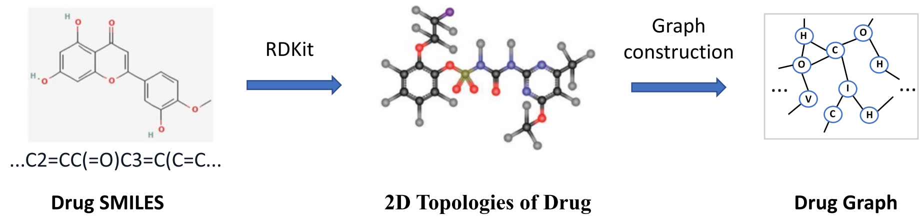


Fig. 3 The procedure of constructing a drug molecule graph. The SMILES representations are utilized for molecule graph construction. Due to the complexity of the drug graph, we have depicted only a portion of the full graph for clarity and ease of demonstration

Table 1 Atom node feature for molecule graph

Feature name	Feature description	Dimension
Atom type	One-hot encoding of the atom	44
Atom neighbors	One-hot encoding of the degree of the atom in the molecule, which is the number of directly-bonded neighbors	11
Number of hydrogens	One-hot encoding of the total number of H bound to the atom	11
Number of implicit hydrogens	One-hot encoding of the number of implicit H bound to the atom	11
Aromatic flag	Whether the atom is aromatic	1
All	All features of the atom	78

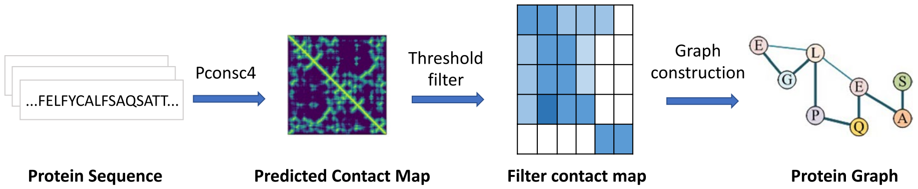


Fig. 4 The procedure of constructing a protein graph. The protein sequence are utilized for protein graph construction

are extracted into latent vectors with the same size for the following performing DTA prediction.

Protein graph representation

Similarly, for the process of the protein graph, the contact map is first predicted from the protein sequence, and a protein graph is then constructed based on it, basically the same as that of DGraphDTA [48]. But for feature extraction and learning, we are also different from existing graph-based methods in that GLCN-DTA can learn robust features with graph learning operation instead of using a fixed threshold-based adjacent matrix.

Specifically, in a manner akin to drug molecule processing, the first step in extracting protein representation is generating the protein graph. This is followed by feature extraction using a GLCN on this graph, a process illustrated in Fig. 4.

Protein structure prediction aims to analyze and assemble the three-dimensional structure of a protein based on its sequence. This structural data encompasses the angles and distances between various residue pairs. A contact map, often an output of structure prediction techniques, typically takes the form of a matrix. This contact map effectively represents the protein's high-dimensional structure and is key to identifying interactions among protein residues. Typically, residues are deemed to be in contact if the Euclidean distance between any two atoms is below a certain threshold [59]. In practice, Pconsc4 is used to predict the contact map, which is a fast, simple, open-source, and efficient method. Pconsc4 outputs the likelihood of contact between residue pairs, and applying a 0.5 threshold yields a contact map shaped (N, N), where N represents the count of nodes (residues). This result effectively serves as the protein's adjacency matrix. However, using the hard threshold to get a fixed adjacent matrix may be existing noises. So we apply the extended graph learning-convolutional network to learn a soft adjacent matrix that will be introduced in the following section.

In the protein graph that's been built, residues serve as nodes, and thus it's essential to define the attributes of these residue nodes. The features selected for the residues are listed in Table 2, consistent with the features identified in DGraphDTA [48]. The features of each residue node are defined by the distinct R functional groups they possess. These node features are represented by attributes such as hydrophobicity, polarity, charge, aromaticity, and so on. Furthermore, Position-Specific Scoring Matrix (PSSM) is a prevalent protein representation in proteomics. In PSSM, scores are assigned to each residue position based on sequence alignment outcomes, serving to characterize the residue node's features. Consequently, this paper utilizes 54-bit features to describe each residue node, with the specifics of these features presented in Table 2. The dimensions of the node features are thus (N, 54). Both the adjacency matrix and the node features are then processed via a graph learning-convolutional network to derive the vector representation of the corresponding protein.

Since the development of the protein graph and its feature set is reliant on sequence alignment outcomes, preparatory procedures including sequence alignment and screening are vital. These preliminary steps are the same as used in DGraphDTA [48]. Specifically, in the operation of Pconsc4 and the computation of PSSM, the protein sequence alignment results serve as the initial input. Therefore, in the pre-processing phase, it's essential to first align all protein sequences in the benchmark datasets. To enhance computation speed, HHblits [60] is employed for the protein sequence alignment. Following this, the alignment results are processed using HHfilter [60] and CCMPred [61] scripts to obtain alignments formatted in the PSICOV [62] style.

Graph learning-convolutional network

We incorporate extended graph *learning-convolutional network* (GLCN) inspired by [57, 58] into existing graph architecture to learn a *soft adjacent matrix* $\mathbf{A}$ to model the graph context for DTA tasks, The detailed GLCN is illustrated in Fig. 2.

Table 2 Residue node feature for protein graph

Number	Feature description	Dimension
1	One-hot encoding of the residue symbol	21
2	Position-specific scoring matrix (PSSM)	21
3	Whether the residue is aliphatic	1
4	Whether the residue is aromatic	1
5	Whether the residue is polar neutral	1
6	Whether the residue is acidic charged	1
7	Whether the residue is basic charged	1
8	Residue weight	1
9	The negative of the logarithm of the dissociation constant for the -COOH group [64]	1
10	The negative of the logarithm of the dissociation constant for the $-NH_3$ group [64]	1
11	The negative of the logarithm of the dissociation constant for any other group in the molecule [64]	1
12	The pH at the isoelectric point [64]	1
13	Hydrophobicity of residue (pH $= 2$) [65]	1
14	Hydrophobicity of residue (pH $= 7$) [66]	1
15	All features of the residue	54

Mathematically, a graph $\mathcal{D}$ is defined as $G = (V, E)$, where $V = \{v_1, \ldots, v_N\}$ represents a set of N nodes. The edge set $E \subset V \times V$ comprises edges $e_{ij} = (v_i, v_j) \in E$, each indicating the existence of relation $\alpha_{ij} \in R$ from node v_i to node v_j.

Graph learning

Given a set of graph nodes input $\mathbf{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_N]^T \in \mathbb{R}^{N \times d_{model}}$, where N represents the total number of nodes, and each $\boldsymbol{v}_i \in \mathbb{R}^{d_{model}}$ denotes the feature of the i-th node in the graph. The initial state of $\mathbf{V}^0$ corresponds to either the atom node features in the molecule graph or the residue node features in the protein graph. GLCN-DTA initially creates a *soft adjacency matrix* $\mathbf{A}$ via a graph learning operation, which illustrates the weight of pairwise relationships between nodes. It then employs a multi-layer perceptron (MLP) network, akin to the method in [57], to extract features $\mathbf{H}$ for each node v_i from the input $\mathbf{V}$. Following this, a graph convolution operation is applied to the features $\mathbf{H}$, enabling the propagation and aggregation of information between nodes, ultimately leading to a new feature representation $\mathbf{V}'$.

Mathematically, a *soft adjacency matrix* $\mathbf{A}$ is learned using a single-layer neural network, which can be represented as follows:

$$\begin{cases} \mathbf{A}_i = \mathrm{softmax}(\mathbf{e}_\mathrm{i}), \quad \mathrm{i} = 1, \ldots, \mathrm{N}, \quad \mathrm{j} = 1, \ldots, \mathrm{N}, \\ \mathrm{e}_\mathrm{ij} = \mathrm{LeakyRelu}(\mathbf{w}_\mathrm{i}^\mathrm{T} |\boldsymbol{v}_\mathrm{i} - \boldsymbol{v}_\mathrm{j}|), \end{cases} \tag{1}$$

where $\mathbf{w}_i \in \mathbb{R}^{d_{model}}$ represents a learnable weight vector. To address the issue of vanishing gradients during the training phase, the LeakyRelu activation function is utilized instead of the traditional Relu function. The softmax$(\cdot)$ operation is applied to each row of $\mathbf{A}$, ensuring that the learned *soft adjacency matrix* $\mathbf{A}$ adheres the following property:

$$\sum\nolimits_{j=1}^{N} A_{ij} = 1, A_{ij} \geq 0. \tag{2}$$

The optimization of the learnable weight vector $\mathbf{w}_i$ is conducted using a modified loss function, adapted from [58], which can be expressed as follows:

$$\mathcal{L}_{\mathrm{GL}} = \frac{1}{N^2} \sum\nolimits_{i,j=1}^{N} \exp(A_{ij} + \eta \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2) + \gamma \|\mathbf{A}\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Intuitively, the first term implies that when nodes $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ are spatially distant in higher-dimensional space, it encourages a smaller weight value A_{ij}, with the exponential function amplifying this effect. Conversely, nodes closer in the higher-dimensional space are likely to have stronger connection weights. This approach helps in minimizing the influence of noise nodes during graph convolution. η serves as a balancing parameter, determining the significance of nodes within the graph. Additionally, averaging the loss is crucial due to the variable number of nodes across different graphs. The second term focuses on maintaining the sparsity of the *soft adjacency matrix* $\mathbf{A}$, with γ as the trade-off parameter. A larger γ results in a sparser *soft adjacency matrix* $\mathbf{A}$. The loss function $\mathcal{L}_{\mathrm{GL}}$ is incorporated as a regularization term in the final loss equation [Eq. (6)], as suggested in [58], to avoid trivial solutions like $\mathbf{w}_i = \mathbf{0}$.

Graph convolution

Graph convolutional network (GCN) is utilized to capture global information and the features of nodes from the graph. We perform graph convolution on the node features v_i.

Firstly, with $\mathbf{V}^0 \in \mathbb{R}^{N\times d_{model}}$ serving as the initial input layer of the graph, *hidden features* $\mathbf{h}_{ij}^{l}$ between nodes v_i and v_j are extracted from the graph. This extraction process utilizes the *node-node* pairs (v_i, v_j) data in the l-th convolution layer, and the computation is performed as follows:

$$\boldsymbol{h}_{ij}^{l} = \sigma(\mathbf{W}_{v_i h}^{l}\mathbf{v}_i^{l} + \mathbf{W}_{v_j h}^{l}\mathbf{v}_j^{l} + \mathbf{b}^{l}), \tag{4}$$

where $\mathbf{W}_{v_i h}^{l}$ and $\mathbf{W}_{v_j h}^{l} \in \mathbb{R}^{d_{model}\times d_{model}}$ are the learnable weight matrices specific to the l-th convolution layer, while $\mathbf{b}^{l} \in \mathbb{R}^{d_{model}}$ serves as a bias parameter. The terms $\mathbf{v}_i^{l}$ and $\mathbf{v}_j^{l}$ represent the features of the i-th and j-th nodes in the l-th convolution layer, respectively. The function $\sigma(\cdot) = \max(0, \cdot)$ is a non-linear activation function. The *hidden features* $\boldsymbol{h}_{ij}^{l} \in \mathbb{R}^{d_{model}}$ represent the combination of graph features and the relational embedding between nodes v_i and v_j, which is vital for compiling a more comprehensive representation for the DTA prediction task.

Next, the *node embedding* $\mathbf{v}_i^{l+1}$ aggregates information from the *hidden features* $\boldsymbol{h}_{ij}^{l}$ through graph convolution, thereby updating the node representation. Since the graph learning layer is capable of generating an optimal adaptive graph *soft adjacency matrix* $\mathbf{A}$, the graph convolution layers can acquire task-specific *node embeddings* by applying a layer-wise propagation rule. For node v_i, the process can be described as follows:

$$\mathbf{v}_i^{(l+1)} = \sigma(\mathbf{A}_i \mathbf{h}_i^{l} \mathbf{W}^{l}), \tag{5}$$

where $\mathbf{W}^{l} \in \mathbb{R}^{d_{model}\times d_{model}}$ represents a layer-specific learnable weight matrix in the l-th convolution layer. The term $\mathbf{v}_i^{(l+1)} \in \mathbb{R}^{d_{model}}$ signifies the node embedding for node v_i in the $(l+1)$-th convolution layer. After progressing through L layers, the contextual information $\mathbf{v}_i^{L}$ is obtained, encompassing global information for every node v_i. This information, $\mathbf{v}_i^{L}$, is then carried forward to subsequent steps for the execution of the DTA prediction task.

In practice, the representations of both the molecule graph and the protein graph are input into the GLCN to separately generate drug features and protein features. Following this, a global pooling process is applied to these extracted features. This step is crucial as it standardizes the size of the resultant latent vectors, preparing them for use in subsequent DTA prediction tasks.

Performing DTA prediction

Following the feature extraction process by GLCN, latent vectors for both the protein and molecular graphs are derived. These two latent vectors are then concatenated and further processed through two fully connected layers to achieve the final binding prediction. One LeakyRelu activation function is after the first fully connected layer. We also employ dropout to avoid overfitting. This process is visually detailed in the right part of Fig. 2.

Model optimization

The model parameters for the entire network are trained jointly by minimizing the loss function expressed as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{dta}} + \lambda \mathcal{L}_{\text{GL}}, \tag{6}$$

where $\mathcal{L}_{\text{GL}}$ is as defined in Eq. 3, with λ serving as a balancing parameter. In practical terms, $\mathcal{L}_{\text{dta}}$ is computed using the mean squared error (MSE) loss to minimize the discrepancy between the predicted affinity values and the actual (ground truth) values.

Results and discussion

Datasets

We conducted experiments using four benchmark datasets, Davis [63], KIBA [64], Metz [65], and ToxCast [66], for training our model and evaluating its performance. Davis [63] and KIBA [64] datasets are widely recognized benchmarks for DTA prediction and have been previously used in DeepDTA [34]. They are also publicly available. To more thoroughly assess our model's performance, we have incorporated the Metz [65] and ToxCast [66] datasets into this study, which include a variety of different types of targets, encompassing both kinases and non-kinases. Distinct from the previously mentioned benchmark datasets, the Metz and ToxCast datasets measure the binding affinities of drug–target pairs using inhibition constant (K_i) and concentration for 50% of maximal effect (EC_{50}), respectively. Detailed information about these four datasets can be found in Table 3.

The Davis [63] dataset was compiled by choosing specific kinase proteins and their inhibitors, where the binding affinity is indicated by the dissociation constant K_d. This dataset encompasses 442 proteins, 68 drugs, and a total of 30,056 drug–target interactions. The average length of the drug SMILES strings is 64, while the average length of the protein sequences is 788. The processing of affinity values in this dataset follows the same methodology as used in DeepDTA [34], employing the following equation:

$$pK_d = -\log_{10} \frac{K_d}{10^9} \tag{7}$$

The KIBA [64] dataset compiles kinase inhibitor biological activities from multiple sources, including the inhibition constant (K_i), dissociation constant (K_d), and the half-maximal inhibitory concentration (IC_{50}). It utilizes the KIBA score to predict biological activity. This dataset features 229 proteins, 2111 drugs, and a total of 118,254 drug–target interactions. The average length of the drug SMILES strings in this dataset is 58, and the average protein sequence length is 728.

Table 3 Summary of the benchmark datasets

Dataset	Proteins	Drugs	Binding entries	Train	Test
Davis	442	68	30,056	25,046	5010
KIBA	229	2111	118,254	98,545	19,709
Metz	170	1423	35,259	28,207	7052
ToxCast	37	3098	114,626	80,251	34,375

In the benchmark setup, each dataset is partitioned into various parts, with one part designated for testing and the remaining parts used for cross-training and validation. By conducting tests on these two datasets, the predictive capability of the method can be thoroughly assessed.

Evaluation metrics

DTA prediction is treated as a regression problem, and our model's performance was evaluated using three metrics: mean squared error (MSE), concordance index (CI), and regression toward the mean (denoted as r_m^2 index).

MSE measures the deviation between predicted and actual values using a squared loss function, which can be expressed as:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)^2, \tag{8}$$

where $\hat{y}_i$ represents the predicted value, y_i is the actual (true) value, and n denotes the total number of drug–target pairs in the dataset.

CI is employed to assess whether the predicted DTA values for two randomly selected drug–target pairs maintain the same rank order as their actual values. This measure is important for evaluating the model's ability to accurately rank the interactions in terms of their binding affinities, which can be expressed as:

$$CI = \frac{1}{Z}\sum_{d_x > d_y} h\left(b_x - b_y\right), \tag{9}$$

$$h(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0.5, & \text{if } x = 0 \\ 0, & \text{if } x < 0 \end{cases}, \tag{10}$$

where b_x represents the predicted value corresponding to the larger affinity d_x, and b_y denotes the predicted value for the smaller affinity d_y. The function $h(x)$ is a step function used in the calculation. The term Z serves as a normalization constant, signifying the total number of drug–target pairs involved in the analysis.

r_m^2 metric is utilized to assess the external predictive capability of the model. This evaluation is conducted as follows:

$$r_m^2 = r^2 \times \left(1 - \sqrt{r^2 - r_0^2}\right), \tag{11}$$

where r^2 represents the squared correlation coefficient between the true and predicted values with intercepts. Conversely, r_0^2 denotes the squared correlation coefficient for the true and predicted values without intercepts.

The setting of the hyperparameters

Our methods were executed using PyTorch, with the model trained on mini-batches of size 128. We employed an initial learning rate of 0.0005, complemented by a warm setting. Additionally, we utilized learning rate decay, reducing it by 20% every 40 epochs.

The Adam optimizer was the choice for optimization. The training spanned over 1000 epochs. Detailed information on the training hyperparameters for our model is listed in Table 4.

Comparison with existing methods

To thoroughly assess our method's efficacy in DTA prediction, we benchmarked it against existing methods such as DeepDTA [34], GraphDTA [45], DGraphDTA [48], and others. This comparison utilized the same benchmark datasets, Davis [63], KIBA [64], Metz [65], and ToxCast [66], and implemented identical training and test sets. Additionally, uniform performance measures were employed for evaluation. The results from other methods were sourced from their respective publications. Performance comparisons are presented in Tables 5 and 6, where lower MSE, or higher CI and r_m^2 scores, indicate superior model performance.

Specifically, Table 5 displays the performance metrics (MSE, CI, and r_m^2) of various models on the Davis dataset. Our method markedly outperformed others on this dataset, achieving an MSE of 0.215, a CI of 0.903, and an r_m^2 of 0.720. These results represent improvements of + 0.3%, + 0.3%, and + 0.1%, respectively, over the previously best-performing method.

Additionally, we assessed our model on the KIBA dataset. As indicated in Table 6, GLCN-DTA excelled among existing methods, achieving an MSE of 0.127, which is + 0.2% higher than the prior best method. The CI performance matched that of the leading WGNN-DTA method. In terms of r_m^2, GLCN-DTA was very close to the top-performing SUbMDTA, differing by only 0.001. These outcomes suggest that our proposed method is effective for DTA prediction.

We also report the performance of GLCN-DTA on the Metz and ToxCast test sets to verify the effectiveness of GLCN-DTA on a vast majority of different types of targets, with MSE, CI, and r_m^2 values of 0.236, 0.848, and 0.713 for Metz, and 0.152, 0.917, and 0.492 for ToxCast, respectively, as detailed in Table 7. In terms of prediction accuracy, GLCN-DTA demonstrates a notable advantage over GraphDTA [45], MGraphDTA [74], and GPCNDTA [11] on both the Metz and ToxCast datasets, which reveals the strong effectiveness of GLCN-DTA.

Table 4 Hyperparameters used in our experiments

Hyperparameter	Setting
Learning rate	0.0005
Epoch	1000
Batch size	128
Optimizer	Adam
GLCN layers	2
Fully connected layers after concatenation	2
dropout rate	0.2
d_{model}	128
λ	1
η	1
γ	1

Table 5 Prediction performance on the Davis dataset

Model	MSE	CI	r_m^2
KronRLS [67]	0.379	0.871	0.407
SimBoost [68]	0.282	0.872	0.644
DeepDTA [34]	0.261	0.878	0.630
WideDTA [35]	0.262	0.886	0.633
MATT_DTI [69]	0.229	0.890	0.682
DeepGS [70]	0.252	0.882	0.686
AttentionDTA [71]	0.245	0.887	0.657
GraphDTA [45]	0.229	0.893	0.649
DeepGLSTM [72]	0.232	0.895	0.680
SubMDTA [73]	0.218	0.894	0.719
DGraphDTA[a] [48]	0.240	0.890	0.659
GEFA] [52]	0.228	0.893	–
STAMP-DPI [53]	0.474	–	–
WGNN-DTA[a] [50]	0.215	0.900	0.711
GLCN-DTA (ours)	0.215	0.903	0.720

MSE mean squared error, *CI* concordance index. The r_m^2 index was used in DeepDTA [34], which can be used to evaluate the external predictive performance of quantitative structure-activity relationship models

[a] The experimental results are acquired from [5]

Table 6 Prediction performance on the KIBA dataset

Model	MSE	CI	r_m^2
KronRLS [67]	0.411	0.782	0.342
SimBoost [68]	0.222	0.836	0.629
DeepDTA [34]	0.194	0.863	0.673
WideDTA [35]	0.179	0.875	0.675
MATT_DTI [69]	0.150	0.889	0.756
DeepGS [70]	0.193	0.860	0.684
AttentionDTA [71]	0.162	0.882	0.735
GraphDTA [45]	0.147	0.889	0.674
DeepGLSTM [72]	0.133	0.897	0.792
SubMDTA [73]	0.129	0.898	0.793
DGraphDTA[a] [48]	0.147	0.891	0.767
WGNN-DTA[a] [50]	0.140	0.899	0.749
GLCN-DTA (ours)	0.127	0.899	0.792

MSE mean squared error, *CI* concordance index. The r_m^2 index was used in DeepDTA [34], which can be used to evaluate the external predictive performance of quantitative structure-activity relationship models

[a] The experimental results are acquired from [5]

Table 7 Prediction performance on the Metz and ToxCast datasets

Model	Metz			ToxCast		
	MSE	CI	r_m^2	MSE	CI	r_m^2
GraphDTA [45]	0.282	0.816	0.681	0.215	0.843	0.330
MGraphDTA [74]	0.265	0.822	0.701	0.176	0.902	0.430
GPCNDTA [11]	0.248	0.834	0.686	0.165	0.904	0.474
GLCN-DTA (ours)	0.236	0.848	0.713	0.152	0.917	0.492

Table 8 Ablation study for the number of GLCN layers on the Davis dataset with GLCN-DTA methods

The number of GLCN layers	MSE	CI	r_m^2
1	0.232	0.885	0.700
2	0.215	0.903	0.720
3	0.226	0.893	0.713
4	0.248	0.876	0.692

Table 9 Ablation study for the graph learning module with GLCN-DTA methods

Model	Davis			KIBA		
	MSE	CI	r_m^2	MSE	CI	r_m^2
GLCN-DTA	0.215	0.903	0.720	0.127	0.899	0.792
w/o graph learning	↓0.03	↓0.02	↓0.05	↓0.05	↓0.04	↓0.08

The superiority of our model can be attributed to our used extended graph learning-convolutional network. To derive more distinctive molecular and protein representations, we employed the soft adjacency matrix, which aggregates complex information from molecules and protein graphs via graph learning and graph convolution operations. Consequently, our model integrates the intrinsic information of complex compounds and protein graphs into a more comprehensive representation, enhancing both the accuracy and robustness of the model.

Ablation studies

We conducted a range of ablation studies to examine the effects of various hyper-parameters on the model's DTA prediction performance. For these studies, all models were trained from scratch using the default standard settings.

Ablation study for the number of GLCN layers

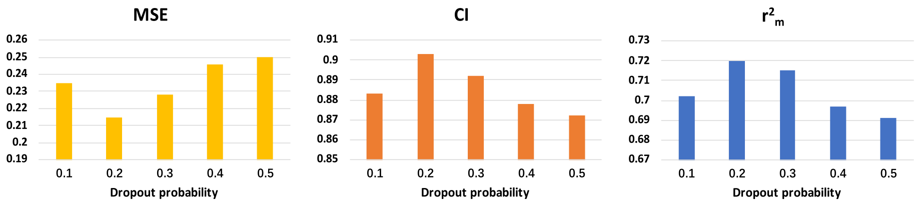
We conducted an ablation study on the Davis dataset to evaluate how the number of GLCN layers affects DTA prediction performance. As indicated in Table 8, the optimal results were consistently achieved with a 2-layer model, rather than with models having 1, 3, or 4 layers. This outcome highlights a known aspect of GCN [43] models: the deeper the model (i.e., the greater the number of layers), the higher the likelihood of overfitting. Therefore, in practical applications, it's advisable to set a specific number of graph layers to optimize performance.

Ablation study for the graph learning module

As indicated in Table 9, the removal of the graph learning component from GLCN-DTA results in a significant decrease in performance metrics on both Davis and KIBA datasets, particularly on the more complex, longer sequences of the KIBA dataset. This suggests that the graph learning element plays a crucial role in handling complex structures in datasets, enhancing generalization, and learning superior representations for improved DTA prediction tasks.

Table 10 Ablation study for the weight of graph learning loss on the Davis dataset with GLCN-DTA methods

The weight of the graph learning λ	MSE	CI	r_m^2
0	0.245	0.883	0.670
0.5	0.215	0.903	0.720
1	0.227	0.894	0.712
2	0.238	0.890	0.685
4	0.241	0.885	0.679



(a) MSE scores (b) CI scores (c) r_m^2 scores

Fig. 5 Ablation study for various dropout probabilities on the Davis datasets with GLCN-DTA methods

Ablation study for the weight of graph learning loss

We conducted an ablation study focused on a key hyper-parameter: the trade-off ratio λ, which is the weight of the graph learning loss in Eq. 6. Specifically, we carried out a series of experiments varying the λ value to assess its impact on DTA prediction performance. The results, reported in Table 10, explore λ values within the range [0, 0.5, 1, 2, 4]. It was observed that our method performed optimally at $\lambda = 1$, surpassing other configurations. Therefore, in practical applications, setting a task-specific value for λ is crucial to ensure the best possible results.

Ablation study for various dropout probabilities

In the stage of performing DTA prediction, the latent vectors representing protein and drug features are concatenated and then passed through two fully connected layers to determine the final binding prediction. To mitigate overfitting, dropout is applied after each fully connected layer. During forward propagation, dropout randomly deactivates neurons with a probability p, enhancing the model's generalization and effectively addressing overfitting. To assess the impact of varying dropout probabilities, we conducted tests on the Davis dataset with different values of p. The results, depicted in Fig. 5, indicate that the optimal dropout probability is 0.2, as evidenced by the lowest MSE value and the best CI and r_m^2 performances at this setting. Excessively high dropout probabilities can cause underfitting and ineffective extraction of protein features, while overly low probabilities might not sufficiently prevent overfitting. Therefore, an appropriately balanced dropout probability is essential for achieving the best predictive performance.

Conclusions

To enhance DTA prediction capabilities, this paper introduces the GLCN-DTA method, effectively characterizing small molecules and protein sequences through the construction of molecular graphs and protein graphs based on contact maps. Utilizing the expanded capabilities of GLCN for advanced feature extraction, the resulting latent vectors provide a more comprehensive representation of proteins and molecules. Extensive experimental results show that GLCN-DTA is not only applicable to DTA prediction but also excels in processing complex and longer datasets. This makes it a robust tool for virtual screening of target proteins and aiding in the discovery of lead compounds.

However, our graph-based DTA prediction approach does not currently account for protein classification or distinguish between structurally ordered and intrinsically disordered proteins. Similar to other graph-based methods [48, 50, 52, 53], it just concentrates on enhancing the extraction of drug and target representations and feature fusion, and subsequent affinity prediction. This specificity might restrict its applicability scope in contexts involving intrinsically disordered proteins and be more suitable for scenarios with structurally ordered proteins. Moreover, our model solely provides predictions for binding affinity without considering the specific binding cavities, lacking the intuitive interpretability of the model. In our future work, we aim to incorporate domain knowledge such as the classification of proteins into structurally ordered protein and intrinsically disordered protein categories into the model, to improve DTA prediction. We also plan to incorporate binding sites as prior knowledge during training to enhance the interpretability of the model. Efforts are underway to organize relevant datasets for these enhancements. Additionally, we intend to integrate additional drug characteristics, such as textual descriptions of proteins and drugs, into the model. This integration aims to further enhance the performance of drug–target binding prediction models by incorporating diverse aspects of drug information.

Acknowledgements
Not applicable.

Author contributions
HQ proposed the idea. TY wrote the draft and provided supervision. WY and CL conceived the experiments. HQ and TY analyzed the results. All authors read and approved the final manuscript.

Funding
This work was supported by the Zhejiang Province Medical and Health Science and Technology Plan Project—General Project for Clinical Observation Research (No. 2024KY1128) and General Project for Nursing Clinical Research at the Sir Run Run Shaw Hospital affiliated with the Zhejiang University School of Medicine (No. 2024HLKY09).

Availability of data and materials
The Davis, KIBA, Metz, and ToxCast datasets used in the current study can be downloaded from https://github.com/LiZhang30/GPCNDTA.

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 15 January 2024 Accepted: 12 February 2024
Published online: 16 February 2024

References

1. Mullard A. New drugs cost us\$2.6 billion to develop. Nat Rev Drug Discov. 2014;13:877–877.
2. DiMasi JA, Grabowski HG, Hansen RW. Innovation in the pharmaceutical industry: new estimates of r &d costs. J Health Econ. 2016;47:20–33.
3. Roses AD. Pharmacogenetics in drug discovery and development: a translational perspective. Nat Rev Drug Discov. 2008;7:807–17.
4. Ashburn TT, Thor KB. Drug repositioning: identifying and developing new uses for existing drugs. Nat Rev Drug Discov. 2004;3:673–83.
5. Zhang Y, Hu Y, Han N, Yang A, Liu X, Cai H. A survey of drug–target interaction and affinity prediction methods via graph neural networks. Comput Biol Med. 2023;666:107136.
6. Pereira JC, Caffarena ER, Santos CN. Boosting docking-based virtual screening with deep learning. J Chem Inf Model. 2016;56(12):2495–506.
7. Xia L, Shen D, Wang H, Ren L, Chen Y, Li G. Identification of small-molecule regulators of testicular receptor 4 via a drug repurposing screening. ACS Omega. 2020;5:30625–32.
8. Strittmatter SM. Overcoming drug development bottlenecks with repurposing: old drugs learn new tricks. Nat Med. 2014;20:590–1.
9. Chen X, Yan CC, Zhang X, Zhang X, Dai F, Yin J, Zhang Y. Drug–target interaction prediction: databases, web servers and computational models. Brief Bioinform. 2016;17(4):696–712.
10. Chen X, Liu M, Yan G. Drug–target interaction prediction by random walk on the heterogeneous network. Mol BioSyst. 2012;8(7):1970–8.
11. Zhang L, Wang C-C, Zhang Y, Chen X. Gpcndta: prediction of drug–target binding affinity through cross-attention networks augmented with graph features and pharmacophores. Comput Biol Med. 2023;166: 107512.
12. Zhang L, Wang C-C, Chen X. Predicting drug–target binding affinity through molecule representation block based on multi-head attention and skip connection. Brief Bioinform. 2022;23:6.
13. Pathak S, Cai X (2017) Ensemble learning algorithm for drug–target interaction prediction. In: International conference on computational advances in bio and medical sciences. https://api.semanticscholar.org/CorpusID:27917753
14. Deshpande M, Kuramochi M, Karypis G. Frequent substructure-based approaches for classifying chemical compounds. IEEE Trans Knowl Data Eng. 2003;17:1036–50.
15. Keiser MJ, Roth BL, Armbruster BN, Ernsberger PR, Irwin JJ, Shoichet BK. Relating protein pharmacology by ligand chemistry. Nat Biotechnol. 2007;25:197–206.
16. Ahneman DT, Estrada JG, Lin S, Dreher SD, Doyle AG. Predicting reaction performance in c–n cross-coupling using machine learning. Science. 2018;360:186–90.
17. Balakin KV, Tkachenko SE, Lang SA, Okun I, Ivashchenko AA, Savchuk NP. Property-based design of gpcr-targeted library. J Chem Inf Comput Sci. 2002;42(6):1332–42.
18. Napolitano F, Zhao Y, Moreira VM, Tagliaferri R, Kere J, D'Amato M, Greco D. Drug repositioning: a machine-learning approach through data integration. J Cheminform. 2013;5:30–30.
19. Liu Z, Fang H, Reagan KA, Xu X, Mendrick D, Slikker W, Tong W. In silico drug repositioning: what we need to know. Drug Discov Today. 2013;18(3–4):110–5.
20. Cheng AC, Coleman RG, Smyth KT, Cao Q, Soulard P, Caffrey DR, Salzberg AC, Huang ES. Structure-based maximal affinity model predicts small-molecule druggability. Nat Biotechnol. 2007;25:71–5.
21. Levin NMB, Pintro VO, Ávila MB, Mattos BB, Azevedo WF. Understanding the structural basis for inhibition of cyclin-dependent kinases. New pieces in the molecular puzzle. Curr Drug Targets. 2017;18(9):1104–11.
22. Azevedo WF, Santos GD, Santos DMG, Olivieri JR, Canduri F, Silva RG, Basso LA, Renard G, Fonseca IO, Mendes MA, Palma MS, Santos DS. Docking and small angle X-ray scattering studies of purine nucleoside phosphorylase. Biochem Biophys Res Commun. 2003;309(4):923–8.
23. Morris GM, Huey R, Lindstrom W, Sanner MF, Belew RK, Goodsell DS, Olson AJ. Autodock4 and autodocktools4: automated docking with selective receptor flexibility. J Comput Chem. 2009;30:66.
24. Lang PT, Brozell SR, Mukherjee S, Pettersen EF, Meng EC, Thomas VL, Rizzo RC, Case DA, James TL, Kuntz ID. Dock 6: combining techniques to model rna-small molecule complexes. RNA. 2009;15(6):1219–30.
25. Weininger D. Smiles, a chemical language and information system. 1. Introduction to methodology and encoding rules. J Chem Inf Comput Sci. 1988;28(1):31–6.
26. Chen L, Tan X, Wang D, Zhong F, Liu X, Yang T, Luo X, Chen K, Jiang H, Zheng M, Elofsson A. Transformercpi: improving compound-protein interaction prediction by sequence-based deep learning with self-attention mechanism and label reversal experiments. Bioinformatics. 2020;6:66.
27. Vaswani A, Shazeer NM, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I. Attention is all you need. In: Neural information processing systems; 2017. https://api.semanticscholar.org/CorpusID:13756489.
28. Wan F, Hong L, Xiao A, Jiang T, Zeng J. Neodti: neural integration of neighbor information from a heterogeneous network for discovering new drug–target interactions. Bioinformatics. 2019;35(1):104–11.
29. Hu S, Zhang C, Chen P, Gu P, Zhang J, Wang B. Predicting drug–target interactions from drug structure and protein sequence using novel convolutional neural networks. BMC Bioinform. 2019;20:66.
30. Ezzat A, Wu M, Li X, Kwoh C. Drug–target interaction prediction using ensemble learning and dimensionality reduction. Methods. 2017;129:81–8.
31. Zhai H, Hou H, Luo J, Liu X, Wu Z, Wang J. Dgdta: dynamic graph attention network for predicting drug–target binding affinity. BMC Bioinform. 2023;24:66.

32. Wei J, Lu L, Shen T. Predicting drug–protein interactions by preserving the graph information of multi source data. BMC Bioinform. 2024;25:66.
33. Lin S, Shi C, Chen J. Generalizeddta: combining pre-training and multi-task learning to predict drug–target binding affinity for unknown drug discovery. BMC Bioinform. 2022;23:66.
34. Öztürk H, Olmez EO, Özgür A. Deepdta: deep drug–target binding affinity prediction. Bioinformatics. 2018;34:821–9.
35. Öztürk H, Olmez EO, Özgür A. Widedta: prediction of drug–target binding affinity; 2019. ArXiv arXiv:1902.04166
36. Huang Y-A, You Z, Chen X. A systematic prediction of drug–target interactions using molecular fingerprints and protein sequences. Curr Protein Peptide Sci. 2018;19(5):468–78.
37. Huang K, Fu T, Glass L, Zitnik M, Xiao C, Sun J. Deeppurpose: a deep learning library for drug–target interaction prediction. Bioinformatics. 2020;36:5545–7.
38. Rogers D, Hahn M. Extended-connectivity fingerprints. J Chem Inf Model. 2010;50(5):742–54.
39. Bender A, Mussa HY, Glen RC, Reiling S. Similarity searching of chemical databases using atom environment descriptors (molprint 2d): evaluation of performance. J Chem Inf Comput Sci. 2004;44(5):1708–18.
40. Durant JL, Leland BA, Henry DR, Nourse JG. Reoptimization of mdl keys for use in drug discovery. J Chem Inf Comput Sci. 2002;42(6):1273–80.
41. Altae-Tran H, Ramsundar B, Pappu AS, Pande VS. Low data drug discovery with one-shot learning. ACS Cent Sci. 2016;3:283–93.
42. Zhang Y, Hu Y, Han N, Yang A, Liu X, Cai H. A survey of drug–target interaction and affinity prediction methods via graph neural networks. Comput Biol Med. 2023;163: 107136.
43. Kipf T, Welling M. Semi-supervised classification with graph convolutional networks; 2016. ArXiv arXiv:1609.02907.
44. Velickovic P, Cucurull G, Casanova A, Romero A, Lio' P, Bengio Y. Graph attention networks; 2017. arXiv: 1710.10903.
45. Nguyen T, Le H, Quinn TP, Nguyen T, Le TD, Venkatesh S. Graphdta: predicting drug–target binding affinity with graph neural networks. Bioinformatics. 2021;37(8):1140–7.
46. Wang S, Jiang M, Zhang S, Wang X, Yuan Q, Wei Z, Li Z. Mcn-cpi: multiscale convolutional network for compound–protein interaction prediction. Biomolecules. 2021;11(8):1119.
47. Feng Q, Dueva EV, Cherkasov A, Ester M. Padme: a deep learning-based framework for drug–target interaction prediction; 2018. arXiv:1807.09741.
48. Jiang M, Li Z, Zhang S, Wang S, Wang X, Yuan Q, Wei Z. Drug–target affinity prediction using graph neural network and contact maps. RSC Adv. 2020;10:20701–12.
49. Greg L, et al. Rdkit: a software suite for cheminformatics. Computational chemistry, and predictive modeling [Google Scholar]; 2013.
50. Jiang M, Wang S, Zhang S, Zhou W, Zhang Y, Li Z. Sequence-based drug–target affinity prediction using weighted graph neural networks. BMC Genomics. 2022;23:66.
51. Rao R, Meier J, Sercu T, Ovchinnikov S, Rives A. Transformer protein language models are unsupervised structure learners. bioRxiv. 2020;6:66.
52. Nguyen TM, Nguyen T, Le TM, Tran T. Gefa: early fusion approach in drug–target affinity prediction. IEEE/ACM Trans Comput Biol Bioinf. 2020;19:718–28.
53. Wang P, Zheng S, Jiang Y, Li C, Liu J, Wen C, Patronov A, Qian D, Chen H, Yang Y. Structure-aware multimodal deep learning for drug–protein interaction prediction. J Chem Inf Model. 2022;62(5):1308–17.
54. Jaeger S, Fulle S, Turk S. Mol2vec: unsupervised machine learning approach with chemical intuition. J Chem Inf Model. 2018;58(1):27–35.
55. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. In: North American Chapter of the Association for computational linguistics. https://api.semanticscholar.org/CorpusID:52967399; 2019.
56. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen P, Canny J, Abbeel P, Song Y. Evaluating protein transfer learning with tape. Adv Neural Inf Process Syst. 2019;32:66.
57. Yu W, Lu N, Qi X, Gong P, Xiao R. Pick: processing key information extraction from documents using improved graph learning-convolutional networks. In: 2020 25th International conference on pattern recognition (ICPR). IEEE; 2021. p. 4363–70.
58. Jiang B, Zhang Z, Lin D, Tang J, Luo B. Semi-supervised learning with graph learning-convolutional networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019, pp. 11313–20.
59. Wu Q, Peng Z, Anishchenko IV, Cong Q, Baker D, Yang J. Protein contact prediction using metagenome sequence data and residual neural networks. Bioinformatics. 2020;6:66.
60. Steinegger M, Meier M, Mirdita M, Vöhringer H, Haunsberger SJ, Söding J. Hh-suite3 for fast remote homology detection and deep protein annotation. BMC Bioinform. 2019;20(1):1–15.
61. Seemayer S, Gruber M, Söding J. Ccmpred-fast and precise prediction of protein residue–residue contacts from correlated mutations. Bioinformatics. 2014;30(21):3128–30.
62. Jones DT, Buchan DW, Cozzetto D, Pontil M. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. Bioinformatics. 2012;28(2):184–90.
63. Davis MI, Hunt JP, Herrgård S, Ciceri P, Wodicka LM, Pallares G, Hocker M, Treiber DK, Zarrinkar PP. Comprehensive analysis of kinase inhibitor selectivity. Nat Biotechnol. 2011;29:1046–51.
64. Tang J, Szwajda A, Shakyawar SK, Xu T, Hintsanen P, Wennerberg K, Aittokallio T. Making sense of large-scale kinase inhibitor bioactivity data sets: a comparative and integrative analysis. J Chem Inf Model. 2014;54(3):735–43.
65. Metz JT, Johnson EF, Soni NB, Merta PJ, Kifle L, Hajduk PJ. Navigating the kinome. Nat Chem Biol. 2011;7(4):200–2.
66. Feng Q, Dueva E, Cherkasov A, Ester M. Padme: a deep learning-based framework for drug–target interaction prediction; 2018. arXiv preprint arXiv:1807.09741.
67. Pahikkala T, Airola A, Pietilä S, Shakyawar SK, Szwajda A, Tang J, Aittokallio T. Toward more realistic drug–target interaction predictions. Brief Bioinform. 2014;16:325–37.
68. He T, Heidemeyer M, Ban F, Cherkasov A, Ester M. Simboost: a read-across approach for predicting drug–target binding affinities using gradient boosting machines. J Cheminform. 2017;9:66.

69. Rong Y, Bian Y, Xu T, Xie W-y, Wei Y, Huang W-b, Huang J. Self-supervised graph transformer on large-scale molecular data. In: Advances in neural information processing systems; 2020. p. 12559–71.
70. Lin X, Zhao K, Xiao T, Quan Z, Wang Z-J, Yu PS. Deepgs: deep representation learning of graphs and sequences for drug–target binding affinity prediction. ArXiv arXiv:2003.13902; 2020.
71. Zhao Q, Xiao F, Yang M, Li Y, Wang J. Attentiondta: prediction of drug–target binding affinity using attention model. In: 2019 IEEE international conference on bioinformatics and biomedicine (BIBM); 2019. p. 64–9.
72. Mukherjee S, Ghosh M, Basuchowdhuri P. Deep graph convolutional network and lstm based approach for predicting drug–target binding affinity. In: Proceedings of the 2022 SIAM international conference on data mining (SDM); 2022. p. 729–37.
73. Pan S, Xia L, Xu L, Li Z. Submdta: drug target affinity prediction based on substructure extraction and multi-scale features. BMC Bioinform. 2023;24:66.
74. Yang Z, Zhong W, Zhao L, Chen CY-C. Mgraphdta: deep multiscale graph neural network for explainable drug–target binding affinity prediction. Chem Sci. 2022;13:816–33.

Publisher's Note