Bioinformatics, 2024, **40(1)**, btad774
https://doi.org/10.1093/bioinformatics/btad774
Advance Access Publication Date: 22 December 2023
Original Paper

Data and text mining

GeNNius: an ultrafast drug–target interaction inference method based on graph neural networks

Uxía Veleiro[1], Jesús de la Fuente[2,3], Guillermo Serrano[1,2], Marija Pizurica[4,5], Mikel Casals[2], Antonio Pineda-Lucena[1], Silve Vicent[1], Idoia Ochoa[2,6], Olivier Gevaert[4,*], Mikel Hernaez[1,6,*]

[1]CIMA University of Navarra, IdiSNA, 31008 Pamplona, Spain
[2]TECNUN, University of Navarra, 20016 San Sebastian, Spain
[3]Center for Data Science, New York University, New York, NY 10012, United States
[4]Stanford Center for Biomedical Informatics Research, Department of Medicine and Department Biomedical Data Science, Stanford University, Stanford, CA 94305, United States
[5]Internet Technology and Data Science LAB (IDLab), Ghent University, Gent 9052, Belgium
[6]Instituto de Ciencia de los Datos e Inteligencia Artificial (DATAI), University of Navarra, 31008 Pamplona, Spain

*Corresponding authors. CIMA University of Navarra, IdiSNA, 31008 Pamplona, Spain. E-mail: mhernaez@unav.es (M.H.); Stanford Center for Biomedical Informatics Research, Department of Medicine and Department Biomedical Data Science, Stanford University, Stanford, CA 94305, United States. E-mail: ogevaert@stanford.edu (O.G.)

Associate Editor: Jonathan Wren

Abstract

Motivation: Drug–target interaction (DTI) prediction is a relevant but challenging task in the drug repurposing field. In-silico approaches have drawn particular attention as they can reduce associated costs and time commitment of traditional methodologies. Yet, current state-of-the-art methods present several limitations: existing DTI prediction approaches are computationally expensive, thereby hindering the ability to use large networks and exploit available datasets and, the generalization to unseen datasets of DTI prediction methods remains unexplored, which could potentially improve the development processes of DTI inferring approaches in terms of accuracy and robustness.

Results: In this work, we introduce GeNNius (Graph Embedding Neural Network Interaction Uncovering System), a Graph Neural Network (GNN)-based method that outperforms state-of-the-art models in terms of both accuracy and time efficiency across a variety of datasets. We also demonstrated its prediction power to uncover new interactions by evaluating not previously known DTIs for each dataset. We further assessed the generalization capability of GeNNius by training and testing it on different datasets, showing that this framework can potentially improve the DTI prediction task by training on large datasets and testing on smaller ones. Finally, we investigated qualitatively the embeddings generated by GeNNius, revealing that the GNN encoder maintains biological information after the graph convolutions while diffusing this information through nodes, eventually distinguishing protein families in the node embedding space.

Availability and implementation: GeNNius code is available at https://github.com/ubioinformat/GeNNius.

1 Introduction

The process of identifying new drugs to treat a specific disease can be simplified by seeking a chemical compound that modulates a pharmacological target implicated in that disease, with the goal of altering its biological activity. Even though different biological entities can be chosen as targets, such as RNA or proteins, the latter are the most common pharmacological targets (Santos *et al.* 2017). Targeting proteins allows the modulation of many biological processes implicated in maintaining health and potentially preventing or treating diseases. For example, drugs targeting metabolic enzymes can alter how cells process nutrients and energy (Duggan *et al.* 2023).

Although high-throughput wet-lab techniques were developed to accelerate drug discovery pipelines, both in vitro and in vivo approaches are time-consuming and can be costly (DiMasi *et al.* 2016). To address these limitations, computational methods have arisen as promising tools to reduce the time and resources required to bring new treatments to market. The field of drug repurposing involves predicting novel drug–target interactions (DTIs) that will ultimately enable the discovery of new uses for already approved drugs (Pushpakom *et al.* 2019). In recent years, the availability of large amounts of data has made it possible to design machine learning models that can assist in these drug development tasks, through, e.g. the identification of complex molecular patterns that were not previously uncovered. These models typically leverage multiple types of data, including amino acid sequences (Zhao *et al.* 2021) and the 3-D protein structures (Verma *et al.* 2021), as recent advances in protein structure prediction such as AlphaFold (Jumper *et al.* 2021, Varadi *et al.* 2021) have significantly increased the amount of structural information available.

Specific to DTI prediction, several different machine learning architectures have been proposed in recent years (see Section 2.1). From these, several are computationally expensive, thereby hindering the ability to use large networks that better describe the available corpus of drug–target interaction,

Received: 19 June 2023; **Revised:** 20 November 2023; **Editorial Decision:** 1 December 2023; **Accepted:** 21 December 2023

but further running times are often not reported. In terms of architecture, while some models utilize simple linear regression techniques, others contain more complex mechanisms such as transformers (Huang *et al.* 2020). However, most of these technologies do not consider the global view of how proteins and drugs are connected, which could be informative towards the discovery of novel relationships. To allow for modeling the network topology, recent works have been proposed to represent DTI data as a graph (Luo *et al.* 2017, Peng *et al.* 2021). Specifically, DTIs can be modeled as a heterogeneous graph connecting drugs and proteins (both represented as nodes) based on recorded interactions in wet-lab experiments (edges). This representation can be augmented by adding additional node types (e.g. diseases), or edge types (e.g. protein similarity). The DTI prediction model is then trained to predict whether a drug has the potential to interact with a protein.

Advances in machine learning for graphs have highlighted Graph Neural Networks (GNNs) as a powerful tool to model these complex networks for a wide range of applications across diverse fields including economics (Xu *et al.* 2022), particle physics (Que and Luk 2022), and especially biomedicine (Yi *et al.* 2021). The defining characteristic of a GNN is that it uses a form of neural message passing, where at each iteration the hidden embeddings of the nodes are updated (Hamilton 2020). Further, contrary to most previous node embedding techniques such as Node2Vec (Grover and Leskovec 2016), several GNN architectures are able to generalize from a set of training examples to unseen data points. This capability is of utmost importance to guarantee the generalization capabilities of the developed models when facing unseen interactions.

Recently, entire libraries have been developed to work with GNNs. Special mention should be made to PyTorch Geometric (PyG), a geometric deep learning library built on top of PyTorch (Fey and Lenssen 2019). Among other functions and layers, PyG implements the SAGEConv layer, which corresponds to the GraphSAGE operator that was originally designed to allow the training of GNNs in large networks (Hamilton *et al.* 2017). SAGEConv simultaneously learns the topological structure of the neighborhood of each node, as well as the distribution of the features of the nodes in the neighborhood.

In this work, we present a novel DTI prediction method, termed Graph Embedding Neural Network Interaction Uncovering System (GeNNius), built upon SAGEConv layers followed by a neural network (NN)-based classifier. GeNNius outperforms state-of-the-art DTI prediction methods across several datasets, not only in AUROC and AUPRC, but also in execution time. Since ensuring the capabilities of in silico drug repurposing approaches to find new interactions is of utmost importance, we also evaluated the ability of GeNNius to predict true interactions reported as negative in a given dataset, yielding promising results. We further assessed the generalization capability of our model by training in one dataset and testing in a different one. This procedure mimics more realistically how the model would behave in a real-world setting.

Finally, while drug repurposing approaches based on complex machine learning models have eased the discovery of new targets, they often lack explainability. We analyzed qualitatively how drug features (such as commonly used molecular descriptors) and protein features (such as the amino acid ratio of protein sequences) are combined in a nonlinear manner while solving the DTI prediction task. This analysis revealed that the GNN encoder maintains biological information while diffusing this information through nodes, eventually distinguishing protein families in the node embeddings.

Overall, the results of our evaluation provide strong support for the effectiveness of GeNNius, and introduce relevant guidelines to build GNN-based drug repurposing approaches.

2 Materials and methods

2.1 Methods

2.1.1 Model architecture

GeNNius architecture is composed of a Graph Neural Network (GNN) encoder that generates node embeddings and a Neural Network (NN)-based classifier that aims to learn the existence of an edge (i.e. an interaction) given the concatenation of a drug and protein node embeddings (Fig. 1).

In GNNs, nodes in the graph exchange messages with their neighbors to update their feature representation, which is formulated with two fundamental functions: the message and the update functions.(Wu *et al.* 2022):

$$\mathbf{m}_v^k = \sum_{u \in N(v)} M_k(\mathbf{h}_v^{k-1}, \mathbf{h}_u^{k-1}, e_{vu}). \tag{1}$$

$$\mathbf{h}_v^k = U_k(\mathbf{h}_v^{k-1}, \mathbf{m}_v^k), \tag{2}$$

where $k \in \{1, \ldots, K\}$ represents the layer, $\mathbf{m}_v$ the aggregated message vector for node v, $N(v)$ the neighbor nodes of v, and $\mathbf{h}_v \in \mathcal{R}^d$ the node v embedding, of dimension d. $M(\mathbf{h}_v, \mathbf{h}_u, e_{vu})$ defines the message between node v and its neighbor node u, which depends on the edge information e_{vu}. Finally, U_k is the node update function, which combines aggregated messages from the node's neighbors with the node's own representation.

GeNNius's encoder is composed of four SAGEConv layers, which are responsible for generating network-preserving node embeddings $\mathbf{h} \in \mathcal{R}^d$ ($d = 17$ in our case) by aggregating information from the embeddings of each node's local neighborhood. Thus, in GeNNius, the embedding of node v at SAGEConv layer k is given by:

$$\mathbf{h}_v^k = f(\mathbf{W}_0^k \mathbf{h}_v^{k-1} + \mathrm{AGG}(\mathbf{W}_1^k \{\mathbf{h}_u^{k-1}, u \in \mathcal{N}(v)\})), \tag{3}$$

where f is the activation function (Tanh in our case) and AGG represents the aggregation function (SUM in our case). $\mathbf{W}_0^k$ and $\mathbf{W}_1^k$ are the learnable weight matrices; since we are working with heterogeneous graphs, where a drug is only connected to proteins and vice versa, if $\mathbf{W}_0^1 \in \mathcal{R}^{d \times d_P}$ then $\mathbf{W}_1^1 \in \mathcal{R}^{d \times d_N}$, or the other way around, being d_P (d_N) the initial dimension of proteins (drugs) node features. For $k > 1$, both matrices have dimension $d \times d$.

The NN-based classifier is composed of two dense layers, both using ReLu as the activation function, followed by the output layer, which is composed of a single neuron with a sigmoid activation function. The input to the classifier is a vector of dimension $2d$ (corresponding to the concatenation of a drug and protein embeddings), and the output is the estimated probability of having an interaction (positive edge).

GeNNius architecture (depicted in Fig. 1) and hyperparameters were chosen through a grid search with ten independent

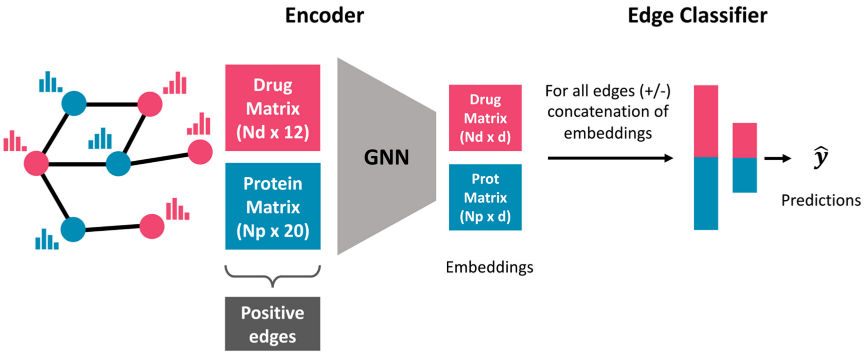


Figure 1. Schematic of GeNNius architecture. GeNNius inputs a graph containing drug (red) and protein (blue) nodes, where N_d and N_p represent the number of drugs and proteins, respectively. First, a GNN generates node representations with an embedding of dimension $d = 17$. Second, a NN-based classifier aims at learning the existence of an edge given a set of concatenations of drug and protein embeddings. Note that at this stage a negative set of edges is generated.

runs, using different types and number (K) of GNN layers, different embedding dimension d, activation functions, aggregation functions, and different number of heads for layers with attention. This approach helped us to fine-tune the model (see Supplementary Material S1 for a detailed description of the process and hyperparameters).

2.1.2 Model configuration

The model was trained with the Adam optimizer (Kingma and Ba 2017) and a learning rate of 0.01. We use a loss that combines the sigmoid of the output layer and the binary cross entropy in a single function. This combination takes advantage of the log-sum-exp trick for numerical stability (Paszke *et al.* 2019). Given a dataset divided into batches of size N, the loss l_n for sample n in the batch is computed as follows:

$$l_n = -[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log\Big(1 - \sigma(x_n)\Big)], \qquad (4)$$

where $y_n \in \{0, 1\}$ is the associated label for sample n, $\hat{y}_n = \sigma(x_n)$ the estimated probability of the sample belonging to the positive class (i.e. existence of an interaction), and x_n the output of the last linear layer (before the activation function). The final batch loss L is then computed as the average of $(l_1, \ldots, l_N)$. Finally, a dropout of 0.2 is used at the encoder stage (the GNN) to address potential collinearities of node features (Wager *et al.* 2013) (dropout rate chosen through hyperparameter-tuning, see Supplementary Section S1).

The model is implemented with early stopping, calculated on validation data, with a minimum of 40 training epochs. The latter is especially useful for small datasets where an early stop may occur during the first epochs, eventually causing underfitting. The model was built with the latest version of PyTorch Geometric (2.3.0), with PyTorch 2.0.0-cuda11.7, and the following packages: pyg-lib (0.2.0+pt20cu117), torch-scatter (2.1.1+pt20cu117) and torch-sparse (0.6.17+pt20cu117). A Dockerfile for running the model is available at https://github.com/ubioinformat/GeNNius.

2.1.3 Model training and evaluation

In the standard setting in which a single dataset (graph in our case) is used to evaluate model performance, the input graph is randomly split into a 70:10:20 ratio for train, validation, and test, respectively, via the random link split function of PyG. This function also randomly selects the negative edges needed for training and testing the NN-based classifier for a 1:1 positive/negative ratio. The training set requires further shuffling of positive and negative edges. Only 70% of train edges is used for training the encoder, while the rest are kept apart for the edge prediction part (i.e. the edge classifier).

To assess the performance of the models in the edge classification task on test data, we use the area under the Receiver Operating Characteristic curve (AUROC), as well as the area under the precision-recall curve (AUPRC), both widely used for evaluating DTI prediction models. We refer to Supplementary Section S2 for a more extended description of these metrics.

2.1.4 Node features

Due to the different nature of drugs and proteins, we choose a vastly different set and dimension of features for drug and protein nodes. The protein node features are encoded as a 20D vector, accounting for the 20 different amino acids, where each feature indicates the proportion of the corresponding amino acid in the protein sequence associated to the node. Drug node features are chosen to be well-known molecular descriptors, calculated with RDKit (Landrum 2022), from their SMILES (Simplified Molecular-Input Line-Entry System). Specifically, the 12 selected features for drug nodes are: LogP value, molecular weight, number of hydrogen bond acceptors, number of hydrogen bond donors, number of heteroatoms (i.e. any atom that is not carbon or hydrogen), number of rotatable bonds, topological polar surface area (surface sum over all polar atoms, primarily O and N, also including their attached H atoms), number of rings and aromatic rings, number of NHs and OHs, number of nitrogen and oxygen atoms, number of heavy atoms a molecule (atomic number > 1), and number of valence electrons. While some of the above-mentioned features are related, model learning and performance is not expected to deteriorate as a dropout layer was introduced to reduce the potential effect of features' collinearity (the correlation matrices of drug/protein features are provided in Supplementary Section S3). While other node features could be considered, such as protein pre-computed embeddings, training the model with those features showed almost no increase in performance (Supplementary Section S4). In addition, the pre-computed protein embeddings were not available for all proteins, which led to a decrease in nodes, hindering the training process and making it impossible for the model to generalize when trained in small networks (Supplementary Section S4).

2.1.5 Related work

In order to benchmark our proposed method GeNNius, we focus on the latest DTI prediction models that have been shown to outperform previously developed models in their respective publications.

- **DTINet** (Luo *et al.* 2017). It considers a heterogeneous graph with four node types (drugs, proteins, side effects and diseases) and six edge types (DTIs, protein-protein interaction, drug-drug interaction, drug-disease association, protein-disease association, drug-side-effect association, plus similarity edges between drugs and proteins). After compact feature learning on each network drugs/proteins, it calculates the best projection of one space onto another using a matrix completion method, and then infers interactions according to the proximity criterion. We note that the large quantity of data required to run the method hampered its reproducibility, as the code for generating all these matrices was not available.
- **EEG-DTI** (Peng *et al.* 2021). EEG-DTI, which stands for end-to-end heterogeneous graph representation learning-based method, also considers a heterogeneous network, where nodes and edges are the same as in DTINet (see above). The model first generates a low-dimensional embedding for drugs and proteins with three Graph Convolutional Networks (GCN) layers, and then concatenates these layers for drugs and proteins separately. Finally, it calculates the inner product to get the protein-drug score.
- **HyperAttentionDTI** (Zhao *et al.* 2021). This method only requires the SMILES string for drugs and the amino acid sequence for proteins. Then, it embeds each character of the different sequences into vectors. The model is based on the attention mechanism and Convolutional Neural Networks (CNNs), in order to make DTI predictions.
- **MolTrans** (Huang *et al.* 2020). Its name stands for Molecular Interaction Transformer for predicting DTIs. As HyperAttentionDTI above, it needs the SMILES for drugs and amino acid sequences for proteins. Then, it makes use of unlabeled data to decompose both drug and nodes into high-quality substructures, to later create an augmented embedding using transformers. Due to this architecture, the model is able to identify which substructures are contributing more to the overall interaction between a protein and a drug.
- **MCL-DTI** (Qian *et al.* 2023). This model inputs drug SMILES to generate two multimodal information of drugs (molecular image and chemical text) and the amino acid sequence for target representation. After, the model uses two decoders with the generated embeddings, which include a multi-head self-attention block and a bi-directional multi-head cross-attention block for cross-information learning. Both drug and protein decoders are used separately to obtain the interaction feature maps. Finally, these feature maps generated by decoders are fed into a fusion block for feature extraction and output the DTI prediction results.

We selected these models following two main criteria: reproducibility and architectural diversity. Firstly, all these models have available working codes. Secondly, our choice of models encompassed both diverse architectural approaches and feature selection, allowing for a comprehensive exploration of the problem space. Notably, we decided to include DTINet, even if it is an older model compared to the rest because it is an example of a widely cited model and used as a comparison several times in the field. Alternatively, some other recent methods were excluded from this review due to reproducibility issues (Li *et al.* 2022, 2021, Binatlı and Gönen 2023, Yao *et al.* 2023).

2.2 Materials

2.2.1 Datasets

In this work we selected various datasets that have been widely used for DTI prediction tasks:

- **DrugBank** (Wishart *et al.* 2006). Drug–Target interactions collected from DrugBank Database Release 5.1.9. Its first release was in 2006, although it has had significant upgrades during the following years.
- **BioSNAP** (Zitnik *et al.* 2018) (http://snap.stanford.edu/biodata). Dataset created by Stanford Biomedical Network Dataset Collection. It contains proteins targeted by drugs on the U.S. market from DrugBank release 5.0.0 using MINER (Stanford-SNAP-Group 2017).
- **BindingDB** (Liu *et al.* 2007). Database that consists of measured binding affinities, focusing on protein interactions with small molecules. The binarization of the dataset was done by considering interactions positive if their K_d was lower than 30. Data downloaded from Therapeutics Data Commons (TDC) (Huang *et al.* 2021).
- **Davis** (Davis *et al.* 2011). Dataset of kinase inhibitors with kinases covering >80% of the human catalytic protein kinome. The acquisition and preprocessing of the dataset was similar to BindingDB.
- **Yamanishi *et al.*** (Yamanishi *et al.* 2008). It is composed of four subsets of different protein families: Enzymes (E), Ion-Channels (IC), G-protein-coupled receptors (GPCR), and nuclear receptors (NR). Yamanishi dataset has been considered the golden standard dataset for DTI prediction and has been used in several published models (Zong *et al.* 2019, Peng *et al.* 2021, Zhang and Xie 2022). DTIs in this dataset come from KEGG BRITE (Kanehisa *et al.* 2006), BRENDA (Schomburg *et al.* 2004), SuperTarget (Günther *et al.* 2008), and DrugBank.

For all considered datasets, we dropped those drugs and proteins from which SMILES or amino acid sequences could not be generated, yielding slightly smaller networks (see Supplementary Section S5).

Note that the above-mentioned datasets, with the exception of BindingDB and Davis, contain only positive samples, i.e. positive links in the network. Nevertheless, when choosing negative samples, we performed random subsampling to have a balanced dataset prior to training the model.

Datasets statistics are summarized in Table 1. These datasets were released in different years, and thus some drug–target interactions can be shared across datasets (See Supplementary Section S6).

2.2.2 Dataset configuration for inferring unknown positives

DTI datasets contain information from diverse sources, have been released in different years, and may be curated in various

Table 1. Dataset statistics.

	DrugBank	BIOSNAP	BindingDB	Davis	Yamanishi			
					E	GPCR	IC	NR
Number of drugs	6823	4499	3084	59	444	222	210	53
Number of proteins	4652	2113	718	218	660	94	203	25
Total number of nodes	11475	6612	3802	277	1104	316	413	78
Total number of edges	23708	13838	5937	673	2920	634	1471	86
Sparsity (%)	0.07	0.15	0.27	5.52	1.01	3.13	3.57	6.94
No. of connected components	412	174	231	1	44	18	3	10
Avg degree drug nodes	3.47	3.08	1.93	11.41	6.58	2.86	7.00	1.62
Avg degree protein nodes	5.10	6.55	8.27	3.09	4.42	6.74	7.25	3.44

ways. As a result, negatively labeled edges in one dataset may be reported as positive in other datasets. We evaluate these unknown positive edges for each dataset to assess if GeNNius can predict them (see Supplementary Section S6 for details on the number of these edges for each dataset). Importantly, we ensured that testing edges do not appear as negatives during training to assess how well GeNNius predicts these specific interactions; we repeated the process ten independent times, enabling us to investigate the variability of the prediction depending on training edges, which is often not reported in DTI prediction models.

2.2.3 Data leakage prevention during evaluation on unseen datasets

Contrary to previously proposed models, we assess the generalization capability of GeNNius by training it on one dataset and testing it on another. For a fair assessment, it is necessary to ensure that there is no data leakage of DTIs between training and testing.

Let us consider two nodes that are present both in the training and test datasets. There are four possible scenarios for an edge connecting these nodes. A positive edge in both datasets is a clear example of data leakage from the train to the test set, as we already informed the model about that positive edge during training. Hence those repeated DTIs are removed during training. On the other hand, edges that appear in one dataset but not on the other one are kept. Keeping the negative edges in the training data makes sense from a usability perspective since a nonreported DTI in a given dataset does not necessarily mean that that pair does not interact, and we would like to test the capabilities of the model under this general scenario. Further, a negative edge may be shared in both datasets; however, since negative edges are randomly selected when generating the training and testing sets, the probability of picking the same edge in both datasets is very low. As an illustrative example, when using DrugBank for training and NR for testing, the probability of selecting the same negative edge is approximately $3e^{-6}$.

We performed five independent training runs on each dataset, i.e. randomly selecting each time a different set of edges for training the model. Next, for each trained model, we performed five independent testing runs. We report the average and standard deviation of the AUROC and AUPRC metrics, of the test set, across the total 25 runs per training-testing dataset pair.

2.2.4 Protein and drug annotation

Protein family and enzyme annotation was retrieved from the ChEMBL database (release 31), as its family hierarchy is manually curated and according to commonly used nomenclature (Gaulton *et al.* 2016). Drug chemical annotation was generated using ClassyFire, an automated structural classifier of chemical entities (Djoumbou Feunang *et al.* 2016). Annotation was used for coloring t-SNEs.

2.2.5 Hardware

All simulations were performed on a server with 64 intel xeon gold 6130 2.1 Ghz cores with 754 GB of RAM and a NVIDIA GeForce RTX 3080, driver version 515.43.04, with cuda 11.7. version.

3 Results

3.1 GeNNius outperforms state-of-the-art methods

The proposed model was run on the eight selected datasets with five independent runs. The resulting AUROC and AUPRC metrics on the test sets across all datasets, as well as running times (corresponding to train, validation and test), are presented in Fig. 2 (see also Supplementary Section S7). GeNNius returned AUROC and AUPCR performance close to 1 (>0.9) for large datasets, and while smaller datasets reported worse results, they are still compelling (>0.8 in almost all runs). NR, being the smallest one, achieved the worst results (>0.7). In addition, the large datasets showed stable results, with a low standard deviation, across the five independent runs. Further, the model execution time was ultrafast for all datasets (less than a minute for the largest dataset). Note that the time variance in the large datasets is due to early stopping. Furthermore, we also performed experiments with different subsampling ratios of positive: negative pairs (1:1, 1:2, 1:10), showing that the ratio 1:1 is not oversimplifying the problem of predicting DTIs (see Supplementary Section S8).

Next, we compared the performance of GeNNius with previously proposed methods. Table 2 shows the performance results of GeNNius and the state-of-the-art methods for both DrugBank and BIOSNAP, the largest standard DTI datasets. We focus on these datasets as they better characterize the current size of testable available drugs. GeNNius outperformed all benchmarked methods in terms of AUROC and AUPRC. Importantly, the execution time is significantly reduced, even when executed without GPU (see Supplementary Section S9). Previous methods' running time was in the order of tens of minutes (except DTINet, which took 4.23 min), while GeNNius took <0.6 min to perform the training, validation, and testing. The closest performance in AUROC and AUPRC to GeNNius was achieved by EEG-DTI. However, EGG-DTI took four orders of magnitude more time to run (917.39 versus 0.58 min in DrugBank). Finally, we also compared GeNNius to off-the-shelf machine learning baselines Logistic Regression (LR) and Random Forest (RF), to assess the actual improvement in accuracy using the same features (see Supplementary Section S10 for further details on those baselines). Comparing our model with LR and RF, we observed an increase in AUROC of 31.75% and 16.73%, respectively, indicating that GeNNius is superior due to its architecture: it not only uses node features but also incorporates network topological information.

3.2 GeNNius prediction capabilities for inferring previously unreported drug–target interactions

To analyze the capability of GeNNius to detect unknown interactions, we first identified those target-protein pairs lacking an edge in one dataset (negative label) but connected in the other datasets (positive label). Then, we assessed whether GeNNius was able to annotate these edges as positive. We trained the model ensuring that the edges for testing were not seen during the training process and repeated the process ten times. Further details of experiment set-up in Section 2.1 (Section 2.2.2).

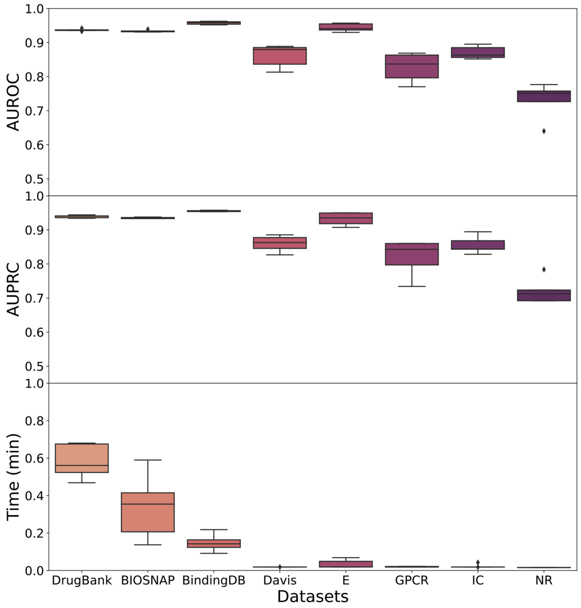


Figure 2. Boxplots of AUROC and AUPRC metrics on test data for five independent runs of GeNNius for the eight selected datasets. (Upper) AUROC results. (Middle) AUPRC results. (Lower) Time results in minutes.

The ratio of correctly predicted edges for each dataset is presented in Fig. 3. When trained with large datasets, GeNNius returned good prediction capabilities, detecting more than 80% of edges in almost all cases. It is worth noting that with DrugBank GeNNius successfully predicted more than 90% of these edges across all runs. Further, when using Yamanishi datasets (E, GPCR, IC, and NR), GeNNius returned satisfactory results, predicting 70% of DTIs on average across different runs, although with higher variability than when using large datasets. This suggests that training on a small dataset hinders the inference of new interactions, as the random choice of edges for training has larger impact on the predictive power in these cases. We note that the observed outliers could be due to a noninformative random selection of training edges. Finally, the Davis dataset yielded significantly worst results than the other datasets. At first sight, this behavior could be due to the origin of the Davis dataset, as it is generated from affinity experiments. However, BindingDB, which is also generated from affinity data, does not yield such low performance. Hence, this may indicate that the problem comes from the significant difference in the topology of Davis versus all the other datasets. Davis is the only dataset formed as a uniquely connected network, while other datasets have more than one connected component. It also presents significantly different average degree values in drug nodes (see Table 1).

3.3 GeNNius generalization capabilities

We evaluated GeNNius performance when training and testing on different datasets. In order to ensure that there is no data leakage that might oversimplify the prediction task, DTIs that were common to train and test datasets were discarded prior to applying the model (see Section 2.1, Section 2.2.3, for a more detailed description of the set-up).

AUROC results are presented in Fig. 4 (obtained AUPRC results are similar, see Supplementary Section S11), where each entry of the heatmap shows the performance of GeNNius on the row dataset when trained on the column dataset. The reported values correspond to 25 runs, where statistical deviation in AUROC and AUPRC arise from the random selection of edges.

In general terms, GeNNius returned compelling results in its generalization capabilities; however, there was a strong dependence on the training dataset. GeNNius reported the best generalization capabilities when trained on larger datasets,

Table 2. Benchmarking results of GeNNius against four state-of-the-art DTI methods and two off-the-self machine learning baselines, for BIOSNAP and DrugBank datasets.[a]

Method	BIOSNAP			DrugBank		
	AUROC	AUPRC	Time (min)	AUROC	AUPR	Time (min)
DTINet	0.8557 ± 0.0011	0.8856 ± 0.0009	4.23	0.8154 ± 0.0004	0.8569 ± 0.0005	7.99
HyperAttentionDTI	0.8616 ± 0.0026	0.7716 ± 0.0627	332.83	0.8624 ± 0.0034	0.7756 ± 0.0456	610.45
MolTrans	0.7921 ± 0.0084	0.6452 ± 0.0037	43.35	0.7982 ± 0.0079	0.6622 ± 0.0053	79.86
EEG-DTI	0.9021 ± 0.0094	0.9046 ± 0.0098	41.39	0.8886 ± 0.0049	0.8795 ± 0.0066	917.39
MCL-DTI	0.8702 ± 0.0045	0.8664 ± 0.0062	281.02	0.8706 ± 0.0032	0.8698 ± 0.0052	473.74
GeNNius	**0.9340** ± 0.0032	**0.9349** ± 0.0021	**0.34**	**0.9371** ± 0.0033	**0.9392** ± 0.0041	**0.58**
Logistic regression	0.6173 ± 0.0026	0.5731 ± 0.0020	0.02	0.6196 ± 0.0048	0.5747 ± 0.0035	0.06
Random forest	0.7910 ± 0.0050	0.7519 ± 0.0056	0.05	0.7698 ± 0.0032	0.7212 ± 0.0031	0.09

[a] Best values are highlighted in bold, excluding baseline results. All AUROC and AUPRC reported results correspond to test set, execution time correspond to train/validation/test. SOTA models were run in their default configuration, i.e. Moltrans and MCL-DTI correspond to 5 independent runs (execution time corresponds to an unique iteration), while DTINet and EEG-DTI correspond to a 10-Fold Cross Validation, and HyperAttentionDTI to 10-times repeated 5-fold Cross-Validation. All the simulations were run in the hardware specified in Section 2.1.

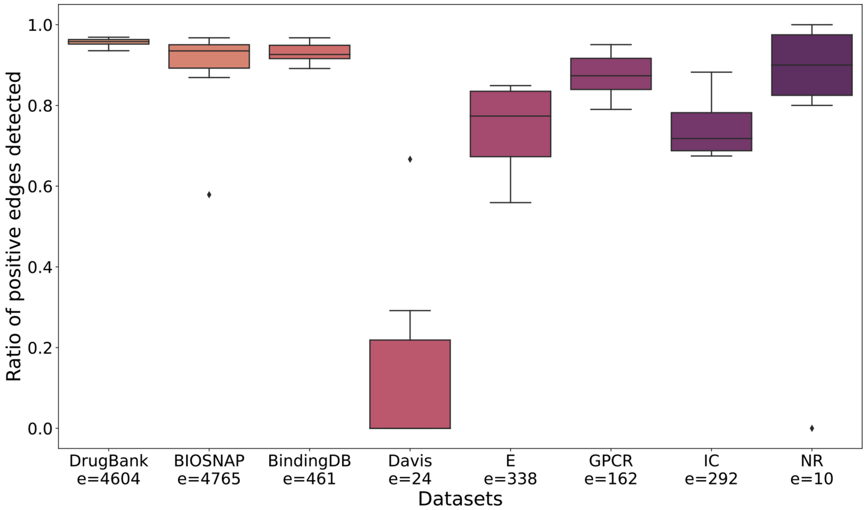
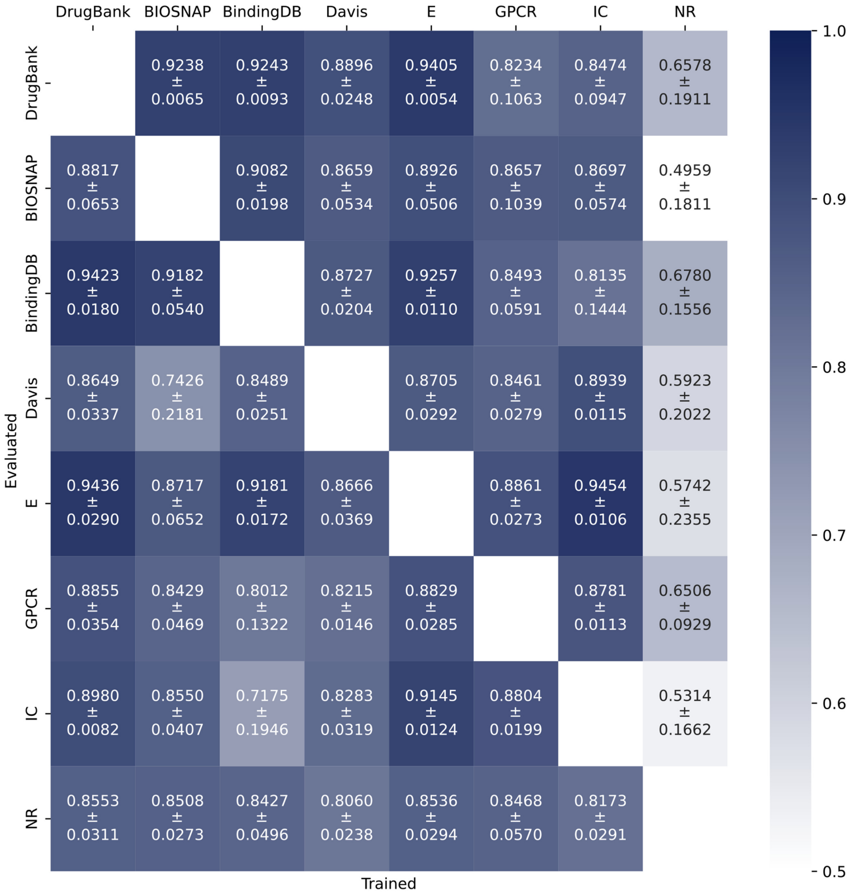


Figure 3. Boxplot of the ratio of correctly identified positive edges in 10 independent runs trained and tested on the same dataset. Note that *e* is the number of edges to be evaluated.



	DrugBank	BIOSNAP	BindingDB	Davis	E	GPCR	IC	NR
DrugBank		0.9238 ± 0.0065	0.9243 ± 0.0093	0.8896 ± 0.0248	0.9405 ± 0.0054	0.8234 ± 0.1063	0.8474 ± 0.0947	0.6578 ± 0.1911
BIOSNAP	0.8817 ± 0.0653		0.9082 ± 0.0198	0.8659 ± 0.0534	0.8926 ± 0.0506	0.8657 ± 0.1039	0.8697 ± 0.0574	0.4959 ± 0.1811
BindingDB	0.9423 ± 0.0180	0.9182 ± 0.0540		0.8727 ± 0.0204	0.9257 ± 0.0110	0.8493 ± 0.0591	0.8135 ± 0.1444	0.6780 ± 0.1556
Davis	0.8649 ± 0.0337	0.7426 ± 0.2181	0.8489 ± 0.0251		0.8705 ± 0.0292	0.8461 ± 0.0279	0.8939 ± 0.0115	0.5923 ± 0.2022
E	0.9436 ± 0.0290	0.8717 ± 0.0652	0.9181 ± 0.0172	0.8666 ± 0.0369		0.8861 ± 0.0273	0.9454 ± 0.0106	0.5742 ± 0.2355
GPCR	0.8855 ± 0.0354	0.8429 ± 0.0469	0.8012 ± 0.1322	0.8215 ± 0.0146	0.8829 ± 0.0285		0.8781 ± 0.0113	0.6506 ± 0.0929
IC	0.8980 ± 0.0082	0.8550 ± 0.0407	0.7175 ± 0.1946	0.8283 ± 0.0319	0.9145 ± 0.0124	0.8804 ± 0.0199		0.5314 ± 0.1662
NR	0.8553 ± 0.0311	0.8508 ± 0.0273	0.8427 ± 0.0496	0.8060 ± 0.0238	0.8536 ± 0.0294	0.8468 ± 0.0570	0.8173 ± 0.0291	

Figure 4. Performance of GeNNius in terms of AUROC when training in one dataset (column) and testing in other (row). Train datasets do not contain positive edges that appear in the test dataset.

such as DrugBank, BIOSNAP, and E. On the other hand, when the model is trained on the smallest dataset, NR, it cannot generalize, resulting in lower AUROC/AUPRC values compared to others (whiter colors in the NR column). In addition, despite the Davis dataset being similar in size to other Yamanishi datasets, it returned the second-to-worst results for both training and testing. As mentioned previously, Davis' topology is different from the rest of the networks. In addition, Davis and BindingDB, unlike other datasets, come from affinity experiments. However, the latter seems to perform similarly, albeit slightly worse, than DrugBank when used for training.

We also found that, for smaller networks, our method obtains better results when trained on large datasets and tested on smaller ones compared to when trained and tested on the same small dataset (see results Section 3.1). For instance, GeNNius obtained an AUROC of 0.86 when trained on DrugBank and tested on NR (lower left corner of heatmap), while it achieved an AUROC of 0.73, using NR for training and testing. This suggests that training on large networks helps the model learn and generalize to unseen and smaller datasets.

In addition, to assess how much these results depend on the node features, we compared them with a random forest model that has no information on network topology. RF showed incapability to generalize, contrasting with the results obtained when training and testing on the same dataset (Table 2). The presented results indicate that GeNNius is capable of generalizing by employing both features and the network's topology without overfitting to the training network (see Supplementary Section S11).

Finally, we confirmed that GeNNius's generalization capability is not dependent on the embedding dimension by experimenting with different values (see Supplementary Section S12).

3.4 GeNNius encoder preserves biological information in edges and diffuses it in nodes

To qualitatively interpret the generated embeddings by GeNNius's GNN encoder, we computed the t-SNE of both the input features and the computed embeddings for all nodes and DTI edges. We focused on the DrugBank dataset, since as shown in previous sections, it reports one of the best AUROC/AUPRC results (Sections 3.1 and 3.3) and yields one of the lowest variabilities during DTI evaluation (Section 3.2). We aimed at shedding some light on whether the embeddings generated by GeNNius carry meaningful biological information beyond the ability to uncover new DTIs.

Firstly, we observe that the edge space with the input features contains information about drug chemical categories and protein families (see Supplementary Section S13, Supplementary Fig. S9a–c). Using the generated embeddings instead, we observe that despite the new shapes in the t-SNE, the biological information is conserved after graph convolutions, i.e. we can still distinguish groups by drug chemical classification but especially by protein families (see Supplementary Section S13, Supplementary Fig. S9d–f).

Secondly, we analyzed the nodes, and found that node input features contain almost no information about protein families, i.e. nodes do not form groups by protein families or by sub-classification of enzymes, conversely to drug nodes grouped by chemical categories (see Supplementary Section S13, Supplementary Fig. S10a–c). The next emerging question we wanted to answer is whether the network diffuses edge biological information during encoding such that the embedding of protein nodes reflects it. We found that the grouping of drug nodes concerning their chemical classification spread after applying the encoder; this is an awaited result, as we desire drugs in a DTI prediction model to be promiscuous (Supplementary Section S13, Supplementary Fig. S10d). However, protein node embeddings displayed better identifiable groups than before (Supplementary Section S13, Supplementary Fig. S10e). Protein families, such as membrane receptors (orange) and ion channels (violet), revealed some grouping at the top of the figure, despite not forming evident groups. Moreover, enzymes now gather in separate groups across the embedding space and, further, upon its annotation, we found a more clear grouping, e.g. kinases (fuchsia) formed a small group on the right of the t-SNE (Supplementary Section S13, Supplementary Fig. S10f).

Ultimately, the encoder maintains biological information in edge space while spreading biological information through nodes, such as protein family classification in protein nodes and sub-classification of enzymes.

In addition, we visualized the embedding space of negative and positive edges, adding those DTI used for validation in Section 3.2. The edges embedding space differentiates areas of positive and negative edges, vastly facilitating the task to the NN-classifier (Supplementary Section S13, Supplementary Fig. S11).

4 Conclusion

We introduced a novel Drug–Target Interaction (DTI) model, termed GeNNius, composed of a GNN encoder followed by an NN-edge classifier. GeNNius outperformed state-of-the-art models in terms of AUROC and AUPRC while being several orders of magnitude faster. Further, we showed that the generalization capabilities of GeNNius and demonstrated its ability to infer previously unreported drug–target interactions. In addition, we showed that GeNNius GNN encoder exploits both node features and graph topology to maintain biological information in edge space while spreading biological information through nodes. Ultimately, GeNNius's ability to generalize and predict novel DTIs reveals its suitability for drug repurposing. In addition, its remarkable speed is key in its usability as it enables fast validation of multiple drug–target pairs.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

This work was supported by the following grants: DoD of the US—CDMR Programs [W81XWH-20-1-0262], Ramon y Cajal contracts [MCIN/AEI RYC2021-033127-I] [RYC2019-028578-I], DeepCTC [MCIN/AEI TED2021-131300B-I00], Gipuzkoa Fellows [2022-FELL-000003–01], the Spanish MCIN [PID2021-126718OA-I00], Fulbright Predoctoral Research Program [PS00342367], and FEDER/MCIN—AEI [PID2020-116344-RB-100/MCIN/AEI/10.13039/501100011033].

References

Binatlı OC, Gönen M. Mokpe: drug–target interaction prediction via manifold optimization based kernel preserving embedding. *BMC Bioinformatics* 2023;**24**:276.

Davis MI, Hunt JP, Herrgard S *et al.* Comprehensive analysis of kinase inhibitor selectivity. *Nat Biotechnol* 2011;**29**:1046–51.

DiMasi JA, Grabowski HG, Hansen RW *et al.* Innovation in the pharmaceutical industry: new estimates of R&D costs. *J Health Econ* 2016;**47**:20–33.

Djoumbou Feunang Y, Eisner R, Knox C *et al.* Classyfire: automated chemical classification with a comprehensive, computable taxonomy. *J Cheminform* 2016;**8**:61.

Duggan BM, Marko DM, Muzaffar R *et al.* Kinase inhibitors for cancer alter metabolism, blood glucose, and insulin. *J Endocrinol* 2023; **256**:e220212.

Fey M, Lenssen JE. Fast graph representation learning with PyTorch geometric. In: *ICLR Workshop on Representation Learning on Graphs and Manifolds*, *New Orleans, USA*, 2019.

Gaulton A, Hersey A, Nowotka M *et al.* The ChEMBL database in 2017. *Nucleic Acids Res* 2016;**45**:D945–54.

Grover A, Leskovec J. *node2vec: Scalable Feature Learning for Networks*. arXiv:1607.00653, 2016.

Günther S, Kuhn M, Dunkel M *et al.* Supertarget and matador: resources for exploring drug–target relationships. *Nucleic Acids Res* 2008; **36**:D919–22.

Hamilton WL. Graph representation learning. *Synth Lect Artif Intell Mach Learn* 2020;**14**:1–159.

Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. CoRR, arXiv:abs/1706.02216, 2017, preprint: not peer reviewed.

Huang K, Xiao C, Glass LM *et al.* MolTrans: molecular interaction transformer for drug–target interaction prediction. *Bioinformatics* 2020;**37**:830–6.

Huang K, Fu T, Gao W *et al.* Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development, In: *Proceedings of Neural Information Processing Systems*, NeurIPS Datasets and Benchmarks, 2021.

Jumper J, Evans R, Pritzel A *et al.* Highly accurate protein structure prediction with alphafold. *Nature* 2021;**596**:583–9.

Kanehisa M, Goto S, Hattori M *et al.* From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Res* 2006;**34**: D354–7.

Kingma DP, Ba J. *Adam: A Method for Stochastic Optimization*, arXiv:1412.6980, 2017.

Landrum G. *Rdkit: Open-Source Cheminformatics. Version 2022.09.1*, 2022.

Li Y, Qiao G, Wang K *et al.* Drug–target interaction predication via multichannel graph neural networks. *Brief Bioinform* 2021;**23**:bbab346.

Li Y, Qiao G, Gao X *et al.* Supervised graph co-contrastive learning for drug–target interaction prediction. *Bioinformatics* 2022;**38**: 2847–54.

Liu T, Lin Y, Wen X *et al.* Bindingdb: a web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res* 2007;**35**:D198–201.

Luo Y, Zhao X, Zhou J *et al.* A network integration approach for drug–target interaction prediction and computational drug repositioning from heterogeneous information. *Nat Commun* 2017;**8**:573.

Paszke A, Gross S, Massa F *et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library*. Red Hook, NY, USA: Curran Associates Inc., 2019.

Peng J, Wang Y, Guan J *et al.* An end-to-end heterogeneous graph representation learning-based framework for drug–target interaction prediction. *Brief Bioinform* 2021;**22**:bbaa430.

Pushpakom S, Iorio F, Eyers PA *et al.* Drug repurposing: progress, challenges and recommendations. *Nat Rev Drug Discov* 2019;**18**:41–58.

Qian Y, Li X, Wu J *et al.* MCL-DTI: using drug multimodal information and bi-directional cross-attention learning method for predicting drug–target interaction. *BMC Bioinformatics* 2023;**24**:323.

Que Z, Luk W. Reconfigurable acceleration of graph neural networks for jet identification in particle physics. In: *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, *Incheon, Korea*, 2022, 202–5.

Santos R, Ursu O, Gaulton A *et al.* A comprehensive map of molecular drug targets. *Nat Rev Drug Discov* 2017;**16**:19–34.

Schomburg I, Chang A, Ebeling C *et al.* Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Res* 2004;**32**: D431–3.

Stanford-SNAP-Group. 2017. Miner: Gigascale multimodal biological network. *GitHub Repository*.

Varadi M, Anyango S, Deshpande M *et al.* AlphaFold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res* 2021; **50**:D439–44.

Verma N, Qu X, Trozzi F *et al.* Ssnet: a deep learning approach for protein-ligand interaction prediction. *Int J Mol Sci* 2021;**22**:1392.

Wager S, Wang S, Liang P *et al. Dropout Training as Adaptive Regularization*, arXiv:1307.1493, 2013.

Wishart DS, Knox C, Guo AC *et al.* Drugbank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 2006;**34**:D668–72.

Wu L, Chu P, Pei J *et al. Graph Neural Networks: Foundations, Frontiers, and Applications*. Singapore: Springer Singapore, 2022.

Xu C, Huang H, Ying X *et al.* Hgnn: hierarchical graph neural network for predicting the classification of price-limit-hitting stocks. *Inform Sci* 2022;**607**:783–98.

Yamanishi Y, Araki M, Gutteridge A *et al.* Prediction of drug–target interaction networks from the integration of chemical and genomic spaces. *Bioinformatics* 2008;**24**:i232–40.

Yao K, Wang X, Li W *et al.* Semi-supervised heterogeneous graph contrastive learning for drug–target interaction prediction. *Comput Biol Med* 2023;**163**:107199.

Yi H-C, You Z-H, Huang D-S *et al.* Graph representation learning in bioinformatics: trends, methods and applications. *Brief Bioinform* 2021;**23**:bbab340.

Zhang J, Xie M. Graph regularized non-negative matrix factorization with prior knowledge consistency constraint for drug–target interactions prediction. *BMC Bioinformatics* 2022;**23**:564.

Zhao Q, Zhao H, Zheng K *et al.* HyperAttentionDTI: improving drug–protein interaction prediction by sequence-based deep learning with attention mechanism. *Bioinformatics* 2021;**38**:655–62.

Zitnik M, Rok Sosič SM, Leskovec J. *BioSNAP Datasets: Stanford Biomedical Network Dataset Collection*. 2018.

Zong N, Wong RSN, Yu Y *et al.* Drug–target prediction utilizing heterogeneous bio-linked network embeddings. *Brief Bioinform* 2019;**22**: 568–80.