Bioinformatics, 2024, **40(7)**, btae349
https://doi.org/10.1093/bioinformatics/btae349
Advance Access Publication Date: 24 June 2024
Original Paper

OXFORD

Data and text mining

HGTDR: Advancing drug repurposing with heterogeneous graph transformers

Ali Gharizadeh [1], **Karim Abbasi** [1], **Amin Ghareyazi** [1], **Mohammad R.K. Mofrad** [2,*], **Hamid R. Rabiee** [1,*]

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, P.O. Box 11155-9517, Iran
[2]Departments of Bioengineering and Mechanical Engineering, University of California, Berkeley, CA, P.O. Box 94720-1740, United States

*Corresponding authors. Departments of Bioengineering and Mechanical Engineering, University of California, Berkeley, CA, P.O. Box 94720-1740,United States. E-mail: mofrad@berkeley.edu (M.R.K.M.); Department of Computer Engineering, Sharif University of Technology, Tehran,P.O. Box 11155-9517, Iran. E-mail: rabiee@sharif.edu (H.R.R.)

Associate Editor: Zhiyong Lu

Abstract

Motivation: Drug repurposing is a viable solution for reducing the time and cost associated with drug development. However, thus far, the proposed drug repurposing approaches still need to meet expectations. Therefore, it is crucial to offer a systematic approach for drug repurposing to achieve cost savings and enhance human lives. In recent years, using biological network-based methods for drug repurposing has generated promising results. Nevertheless, these methods have limitations. Primarily, the scope of these methods is generally limited concerning the size and variety of data they can effectively handle. Another issue arises from the treatment of heterogeneous data, which needs to be addressed or converted into homogeneous data, leading to a loss of information. A significant drawback is that most of these approaches lack end-to-end functionality, necessitating manual implementation and expert knowledge in certain stages.

Results: We propose a new solution, Heterogeneous Graph Transformer for Drug Repurposing (HGTDR), to address the challenges associated with drug repurposing. HGTDR is a three-step approach for knowledge graph-based drug repurposing: (1) constructing a heterogeneous knowledge graph, (2) utilizing a heterogeneous graph transformer network, and (3) computing relationship scores using a fully connected network. By leveraging HGTDR, users gain the ability to manipulate input graphs, extract information from diverse entities, and obtain their desired output. In the evaluation step, we demonstrate that HGTDR performs comparably to previous methods. Furthermore, we review medical studies to validate our method's top 10 drug repurposing suggestions, which have exhibited promising results. We also demonstrated HGTDR's capability to predict other types of relations through numerical and experimental validation, such as drug–protein and disease–protein inter-relations.

Availability and implementation: The source code and data are available at https://github.com/bcb-sut/HGTDR and http://git.dml.ir/BCB/HGTDR

1 Introduction

The escalating costs and protracted development periods for new pharmaceuticals present significant challenges to the industry (Zhang *et al.* 2020, Pan *et al.* 2022). Drug repurposing, defined as identifying new uses for approved drugs, has emerged as a viable solution to these issues (Zhou *et al.* 2020). Employing this approach mitigates the risk and reduces the expenses and duration associated with the development of drugs (Singh *et al.* 2020, Talevi and Bellera 2020). To facilitate drug repurposing, both systematic experimental and computational techniques have been devised (Pushpakom *et al.* 2019, Talevi and Bellera 2020). Among computational strategies, network-based methods have shown encouraging outcomes, encompassing three primary stages (Zeng *et al.* 2019, Zhu *et al.* 2020, Che *et al.* 2021, Jin *et al.* 2021, Yu *et al.* 2021): (1) network construction, (2) feature extraction, and (3) link prediction. A biological network or knowledge graph is initially constructed, incorporating various biological entities and their interconnections. Subsequently, the nodes within this graph are characterized through embeddings using graph neural networks (GNNs). Finally, these embeddings are applied within algorithms, such as a matrix completion or a neural network, to predict links in the knowledge graph.

Zhu *et al.* (2020) integrated six distinct databases to develop a comprehensive knowledge graph, which comprises five types of entities and nine varieties of relationships. To characterize the graph, they employed a combination of four path-based representations, guided by predefined meta-paths, alongside three embedding-based representations. Subsequently, they harnessed three machine learning techniques—support vector machines, decision trees, and random forests—to discern new potential drug applications. The strength of this study lies in its integration of multiple data sources to account for the interplay between various types of data. Nevertheless, the reliance on manually selected meta-paths may curtail the automation of the method, necessitating specific meta-path selections for each knowledge graph constructed.

The layer attention graph convolutional network (LAGCN) method, as proposed by Yu *et al.* (2021), establishes a drug–drug similarity network by applying the Jaccard index to various drug-related datasets, including targets,

Received: 6 September 2023; **Revised:** 20 April 2024; **Editorial Decision:** 28 April 2024; **Accepted:** 23 June 2024

pathways, and substructures, all of which are sourced from the DrugBank database. Concurrently, it formulates a disease–disease similarity network, employing semantic similarity scores derived from MeSH descriptors structured as a hierarchical directed acyclic graph (DAG) based on the premise that diseases with more shared ancestral traits exhibit greater similarity. The integration of these two similarity networks, along with the drug–disease relationships, results in the construction of the final input graph. This graph is further processed as nodes are embedded using an LAGCN, followed by the employment of a bilinear decoder for link prediction. While LAGCN leverages diverse data sources, it simplifies these into uniform similarity matrices, a process that may exclude certain informative relationships. Additionally, the method needs more scalability, limiting its applicability to vast graphs.

The AttGCN-DDI model, described by Che *et al.* (2021), assembles a concise knowledge graph from six distinct databases, delineating five entities and seven relationship types. This model hypothesizes that a drug can treat a disease if it can be connected to the disease by a path of less than four steps. Focusing on COVID-19, it incorporates drugs and diseases that have the shortest path to COVID-19 of fewer than four links into the graph. After this addition, the relationships between these newly added nodes are integrated. This method of input selection necessitates expert knowledge in the domain, which, in turn, diminishes the automation aspect of the process. Topological features are extracted as node embeddings using Att-GCN, a method that treats all edges uniformly, thus not accounting for the heterogeneity of the edges. Additionally, a projection matrix is developed to reconstruct the drug–disease interaction matrix. Similar to other discussed methods, this approach also suffers from a lack of scalability.

DeepDR (Zeng *et al.* 2019) employs a network of nine databases to extract drug features. Initially, these networks are transformed into homogeneous similarity networks using techniques such as the Jaccard similarity. The method then generates probabilistic co-occurrence (PCO) matrices for each network through random surfing. To address the inherent sparsity of some networks, it constructs shifted positive pointwise mutual information (PPMI) matrices from the PCO matrices. However, these preprocessing stages may result in losing original network information. A multi-modal deep autoencoder is employed to extract features from drug nodes. DeepDR considers drug–disease associations as inputs and drug features as auxiliary information. Subsequently, it applies a collective variational autoencoder (cVAE), as introduced by Chen and de Rijke (2018), to predict drug–disease associations.

As detailed by Jin *et al.* (2021), HeTDR utilizes an input network resembling the one used by DeepDR. It adopts identical procedures for network conversion to homogeneous networks, random surfing, and the creation of shifted PPMI matrices. A distinctive feature of HeTDR is its application of similarity network fusion to integrate all drug-related information into a singular network, thereby providing a comprehensive biological perspective on drugs.

The model employs a sparse autoencoder, as described in Ng (2011), to extract drug features. Additionally, it utilizes fine-tuned BioBERT (Lee *et al.* 2020) to derive disease features. Consequently, each new input requires the fine-tuning of BioBERT. Ultimately, the GATNE-I model, introduced by Cen *et al.* (2019), is applied to predict drug–disease interactions.

Several contemporary studies have made efforts to utilize heterogeneous data from multiple databases. Nevertheless, these approaches come with constraints. Some, like those proposed by Zhu *et al.* (2020), need more scalability and rely on limited datasets. While methods such as those by Zeng *et al.* (2020) employ different networks, they need to synthesize a higher-order representation that consolidates information from these varied sources. It is crucial to recognize that integrating networks can reveal insights that may need to be discernible when networks are analyzed in isolation. Furthermore, certain techniques (Zeng *et al.* 2019, Yu *et al.* 2021) transform heterogeneous networks into homogenized similarity networks, diluting the input data's richness. Most of these strategies also do not incorporate non-network data. Some approaches (Zeng *et al.* 2019, Jin *et al.* 2021) restrict the inclusion of certain relationship types in the knowledge graph, thereby preventing the use of pivotal data, such as protein–protein interactions. These methods predominantly center on drug-related data and overlook the extraction of network features pertinent to diseases, thus presenting considerable hurdles to drug repurposing endeavors. Moreover, certain methods necessitate hands-on implementation, including the designation of meta-paths as specified by Zhu *et al.* (2020), necessitating specialized domain knowledge and constraining the potential for input modification. The absence of end-to-end model structures further compromises the automation of these methods and amplifies the reliance on domain expertise.

To address the constraints outlined previously, we introduce a model endowed with the following features:

- It distinguishes between node and edge types using heterogeneous graphs.
- The model accommodates an unrestricted variety of graph data types, enabling the use of any node and edge categories in the input graph.
- Scalability is a core attribute of the method.
- The model operates autonomously, obviating the need for domain-specific knowledge during setup.
- It is an end-to-end model that integrates the task of link prediction within the feature extraction process from nodes.

Our model, Heterogeneous Graph Transformer for Drug Repurposing (HGTDR), parallels prior models by encompassing a three-step approach. In our research, PrimeKG (Chandak *et al.* 2023) serves as the foundation of our knowledge graph, complemented by the initial embeddings from BioBERT (Lee *et al.* 2020) and ChemBERTa (Chithrananda *et al.* 2020). To extract node features, we utilize the heterogeneous graph transformer (HGT) technique (Hu *et al.* 2020) as our GNN. Finally, a fully connected network predicts the relationship score within the graph between a drug and a disease.

PrimeKG significantly enhances the initial phase of our approach, broadening the scope of drugs and diseases that can be explored beyond previous capabilities. Table 1 outlines a comparative analysis of the coverage of drugs and diseases by this method and others.

Several studies have developed knowledge graphs in drug repurposing, including Hetionet (Himmelstein *et al.* 2017) and DRKG (Ioannidis *et al.* 2020). Hetionet features 755

Table 1. Comparison of the number of drugs and diseases using different methods.

Method	Drug	Disease
ATTGCN-DDI	1470	752
LAGCN	269	598
HetDR	1519	1229
DeepDDR	1519	1229
HGTDR	1801	1363

indication relations, linking 387 drugs to 77 diseases. DRKG presents 83 895 compound–disease relationships, which may not directly pertain to indications. Notably, most of these relationships, specifically 77 782, are derived from the GNBR database (Percha and Altman 2018), which utilizes text-processing techniques. However, this source may not reliably inform drug repurposing efforts due to its reliance on text extraction.

For the extraction of features from heterogeneous graphs, various methodologies have been proposed, such as HAN (Wang *et al.* 2019), HetGNN (Zhang *et al.* 2019), MHGCN (Yu *et al.* 2022), and HGT (Hu *et al.* 2020). HAN employs node-level and semantic-level attention mechanisms within heterogeneous graphs to discern the significance of nodes and their meta-path-based neighbors and the relevance of different meta-paths. HetGNN introduces a strategy for capturing the structural and content heterogeneity. It utilizes a random walk approach to sample heterogeneous neighbors that are strongly correlated, grouping them by node type. Subsequently, node embeddings are generated through a dual-module neural network that accounts for structural and content diversity. MHGCN aims to autonomously identify valuable heterogeneous meta-path interactions across varying lengths in multiplex heterogeneous networks through multi-layer convolutional aggregation, thereby producing node embeddings that merge structural and semantic information. Despite its innovations, MHGCN's scalability remains a challenge. HGT distinguishes itself by interpreting heterogeneous nodes and edges through meta-relations, utilizing node-specific and edge-specific weights. This approach enables identifying common and unique patterns across different relationships without predefined meta-paths, thus bypassing concerns about identifying critical meta-paths within the biological context.

According to the study by Hu *et al.* (2020), HGT outperforms HAN and HetGNN, further validated by its scalability advantage. Therefore, we have selected HGT for node feature extraction in our model. While the original study by Hu *et al.* highlighted HAN's effectiveness following HGT, we opted to verify HGT's superiority by substituting our feature extraction layers with HAN layers. The results in Table 4 affirm that HGT surpasses HAN in our specific task.

The rest of this article is organized as follows: Section 2 is dedicated to a comprehensive exposition of our methodology. Section 3 presents the experiments conducted and the corresponding results obtained through the application of our method. The paper concludes in Section 4, where we provide concluding remarks and outline directions for future research.

2 Materials and methods

This section begins with the presentation of the problem formulation, followed by a detailed explanation of the three steps involved in our proposed method.

2.1 Problem formulation

Let G be a heterogeneous graph whose nodes are biological entities whose edges are different relationships among the entities. We have drug and disease nodes and indication edges in the graph. We define all indications as $I = \left\{\left(Dr^i,\ Di^i\right),\ Y^i\right\}_{i=1}^{N}$ where $\left(Dr^i, Di^i\right)$ is the i$^{\text{th}}$ indication edge in the graph; Y^i is the label, which is 1 for all indications, and N is the number of all indications. During the training step, we put 20% of indications (I_{input}) in the graph G and tried to predict the label of the rest of the indications (I_{pos}) plus random negative samples, which are defined as $Ineg\ =\ \left\{\left(Dr^i,\ Di^j\ \right),\ 0\right\}_1^{N_{pos}}$ where N_{pos} is the number of samples of I_{pos}, $i \in (1 \ldots N_{Drug})$, and $j \in (1 \ldots N_{Disease})$. N_{Drug} and $N_{Disease}$ are also the number of drugs and diseases in G, respectively.

$$I_{prediction} = I_{pos} + I_{neg} \tag{1}$$

This prediction task can be viewed as a simulation of drug repurposing in which our model learns to predict indications that do not exist in the input graph. To do actual drug repurposing, we must put all indications in the graph G and try to predict novel indications.

2.2 Network construction

In this step, we should construct a network. Previous methods built their networks using databases such as DrugBank (Law *et al.* 2014). There is a problem with this approach. Domain knowledge is necessary to select the appropriate databases. Moreover, not all relations within a database are used in those works, raising the question of how the relations are chosen. We address this problem by utilizing a previously constructed knowledge graph, PrimeKG (Chandak *et al.* 2023), and allowing our model to determine which entities and relationships are relevant to the task. To achieve this goal, HGT is suitable because it can control the overall contribution of a particular node or edge type to embedding creation by using node and edge-specific weights. PrimeKG integrates 20 high-quality resources to generate a graph with 10 node types and 30 edge types. The existing graph can also be expanded with new data without node and edge type restrictions. The graph is defined as $G = (V,\ E,\ A,\ R)$ in which every node $v \in V$ and every edge $e \in E$ are associated with their type-mapping functions $\tau(v):\ \ V \rightarrow A$ and $\phi(e):$ $E \rightarrow R$ respectively. Also, for every edge $e = (s,\ t)$, where s and t are the source and target nodes of the edge, its meta-relation is denoted as $\langle \tau(s),\ \ \phi(e),\ \ \tau(t) \rangle$.

Since previous works only contain drugs and diseases that contribute to indication relations, we remove drug and disease nodes that do not contribute to at least one indication edge. These nodes can increase accuracy unrealistically, in comparison with previous works, due to the ease of predicting no indications related to the drugs and diseases. Consequently, some nodes are removed from the graph. Tables 2 and 3 demonstrate graph statistics before and after the removal of the nodes.

2.2.1 Initial embeddings

We add BioBERT and ChemBERTa embeddings to our knowledge graph to integrate different types of information into the graph. BioBERT is a domain-specific language representation model pre-trained on large-scale biomedical

Table 2. PrimeKG node counts.

Entity	Count before removal	Count after removal	Removal percent
Biological process	28 642	28 642	0
Protein	27 671	27 573	0.35
Disease	17 080	1363	92.01
Phenotype	15 311	15 082	1.49
Anatomy	14 035	14 035	0
Molecular function	11 169	11 169	0
Drug	7957	1801	77.36
Cellular component	4176	4176	0
Pathway	2516	2516	0
Exposure	818	780	4.64
Total	**129 375**	**107 137**	**17.17**

Table 3. PrimeKG directed edge counts.

Entity	Count before removal	Count after removal	Removal percent
Anatomy–protein (present)	3 036 406	3 036 406	0
Drug–drug	2 672 628	743 328	72.18
Protein–protein	642 150	642 150	0
Disease–phenotype (positive)	300 634	24 490	91.85
Biological process–protein	289 610	289 610	0
Cellular component–protein	166 804	166 804	0
Disease–protein	160 822	99 232	38.29
Molecular function–protein	139 060	139 060	0
Drug–phenotype	129 568	102 736	20.70
Biological process–biological process	105 772	105 772	0
Pathway–protein	85 292	85 292	0
Disease–disease	64 388	1772	97.24
Drug–disease (contraindication)	61 350	32 194	47.52
Drug–protein	51 306	24 848	51.56
Anatomy–protein (absent)	39 774	39 774	0
Phenotype–phenotype	37 472	37 472	0
Anatomy–anatomy	28 064	28 064	0
Molecular function–molecular function	27 148	27 148	0
Drug–disease (indication)	18 776	18 776	0
Cellular component–cellular component	9690	9690	0
Phenotype–protein	6660	6660	0
Drug–disease (off-label use)	5136	3836	25.31
Pathway–pathway	5070	5070	0
Exposure–disease	4608	2924	36.54
Exposure–exposure	4140	4140	0
Exposure–biological process	3250	3250	0
Exposure–protein	2424	2424	0
Disease–phenotype (negative)	2386	140	94.13
Exposure–molecular function	90	90	0
Exposure–cellular component	20	20	0
Total	**8 100 498**	**5 683 172**	**29.84**

Table 4. Comparison of HGT and HAN layers as feature extractors.

Feature extraction layer	AUROC	AUPR
HAN	0.925	0.918
HGT	0.944	0.946

corpora. Therefore, its embeddings add information extracted from biomedical literature to our knowledge graph. ChemBERTa is a similar model pre-trained on SMILES (Weininger 1988) representation of molecules that can extract their structural information. The initial embeddings are added to the graph in the following manner. The names of all entities, except drugs, are first obtained from PrimeKG. BioBERT embeddings are then extracted and added to our graph nodes. Moreover, SMILES (Weininger 1988) representations of the drugs are obtained from DrugBank. Afterward, they are embedded using ChemBERTa and added to drug nodes. Interestingly, we can add any embedding to any of our node types, and we do not require different node types to have similar embeddings. This lets us add any auxiliary information to our graph if it can be represented as an embedding. We denote the initial embeddings of the graph as H^{init}.

After constructing the graph, we divide indication edges into masked and unmasked groups, with 80% of indication edges being masked. Unmasked indications are used like other edges as part of the input graph. However, masked indications are used as positive samples when computing loss function. This technique helps the model predict indications not present in the input graph, which can act like a simulation of drug repurposing.

2.3 Feature extraction

For embedding graph nodes, attention-based GNNs usually follow the following formula for source node s and target node t (Hu *et al.* 2020):

$$H^{l}[t] \leftarrow \underset{\forall s \in N(t), \forall e \in E(s,t)}{Aggregate} \left(Attention(s,t).Message(s) \right) \qquad (2)$$

There are three basic operators: Attention, which estimates the importance of each source node; Message, which extracts the message using only the source nodes; and Aggregate, which aggregates the neighborhood message by the attention weight.

To understand different distributions of different node and edge types, HGT introduces a heterogeneous mutual attention mechanism to calculate a target node and all its neighbors ($s \in N(t)$) mutual attention grounded by their meta-relation. The attention for each edge $e = (s,t)$ is defined as follows:

$$Attention_{HGT}(s,e,t) = \underset{\forall s \in N(t)}{Softmax} \left(\underset{i \in [1,\ h]}{\Vert} ATT-head^{i}(s,e,t) \right)$$

$$ATT-head^{i}(s,e,t) = \left(K^{i}(s) W^{ATT}_{\Phi(e)} Q^{i}(t)^{T} \right) . \frac{\mu_{<\tau(s),\ \Phi(e), \tau(t)>}}{\sqrt{d}}$$

$$K^{i}(s) = K-Linear^{i}_{\tau(s)} \left(H^{(l-1)}[s] \right)$$
$$Q^{i}(t) = Q-Linear^{i}_{\tau(t)} \left(H^{(l-1)}[t] \right) \qquad (3)$$

For the i-th attention head $ATT-head^{i}(s,\ e,\ t)$, source nodes that have type $\tau(s)$ is projected into the i-th key vector $K^{i}(s)$, with a node-specific linear projection to consider the distribution differences of node types. Similarly, the target node t is projected to the i-th query vector. Furthermore, unlike the vanilla Transformer that directly computes the dot product between query and key vectors for calculating the similarity between key and query, HGT keeps a distinct edge-specific matrix $W^{ATT}_{\Phi(e)}$ for each edge type $\Phi(e)$. Thus, the model can capture different semantic relations even between the same node-type pairs. Additionally, a prior tensor μ is used to denote the general significance of each meta-relation triplet, serving as an adaptive scaling to the attention. Finally, h attention heads are concatenated to get each node pair's attention vector. Then, for each target node t, all attention vectors of neighbors $N(t)$ are gathered, and softmax is conducted, making it fulfill $\Sigma_{\forall s \in N(t)} Attention_{HGT}(s, e, t) = 1_{h \times 1}$.

Similarly, meta-relations of edges are also used in message passing process to alleviate the distribution differences of nodes and edges of different types as follows:

$$Message_{HGT}(s,e,t) = \Vert_{i \in [1,\ h]} MSG - head^i(s,e,t)$$

$$MSG - head^i(s,e,t) = M - Linear^i_{\tau(s)}\left(H^{(l-1)}[s]\right) W^{MSG}_{\Phi(e)} \quad (4)$$

Source node's ID with a linear projection $M - Linear^i_{\tau(s)}$. It is then followed by a matrix $W^{MSG}_{\Phi(e)}$ for incorporating the edge dependency. Then, all h message heads are concatenated for each node pair.

In the aggregation step, the attention vector is used as the weight to average the corresponding messages from the source nodes to get the updated target vector $\hat{H}^{(l)}[t]$ as:

$$\tilde{H}^{(l)}[t] = \bigoplus_{\forall s \in N(t)} \left(Attention_{HGT}(s,e,t).Message_{HGT}(s,e,t)\right) \quad (5)$$

Finally, the target node's vector is mapped back to its type-specific distribution with a linear projection $A - Linear_{\tau(t)}$ followed by a non-linear activation and residual connection as:

$$H^{(l)}[t] = \sigma\left(A - Linear_{\tau(t)} \tilde{H}^{(l)}[t]\right) + H^{(l-1)}[t] \quad (6)$$

As described, the method relies on using meta-relation to parameterize the weight matrices separately, enabling the model to distinguish the operators for different relations and thus be more capable of handling the distribution differences in heterogeneous graphs.

2.3.1 HGSampling

Using the entire input graph in the training process can make any method unscalable. Therefore, a sampling method is needed to make the method scalable. Consequently, we use HGSampling (Hu *et al.* 2020) to get mini-batches for training. We denote a subgraph generated by the sampler as G_{batch}. The method keeps a similar number of nodes and edges for each type and keeps the subgraph dense to minimize information loss and reduce sample variance. HGSampling has been used in large-scale graphs and applied to a graph with 178 million nodes and 2.2 billion edges (Hu *et al.* 2020). As a result, there will not be any problem regarding the size of the graph since the current large biological graphs have fewer than 10 million edges (Himmelstein *et al.* 2017, Zhang *et al.* 2021, Chandak *et al.* 2023).

2.3.2 Model architecture

Our model has three layers of HGT ($L = 3$) whose input and output feature vector sizes are 64. The initial embeddings are 768 in size, which are the outputs of the BioBERT and ChemBERTa models. Therefore, a linear layer is first defined, which decreases the input size to 64.

$$H^0_{batch} = Dropout\left(ReLU\left(Linear_\tau\left(H^{init}_{batch}\right)\right)\right) \quad (7)$$

where H^{init}_{batch} is the input feature vector containing the initial embedding of batch nodes, H^0_{batch} is the input feature vector of the first HGT layer, $Linear_\tau$ is a node-specific linear layer, and ReLU is the rectified linear unit activation function. Also, dropout is applied to the output with probability = 0.5.

HGT layers are defined as follows:

$$H^l_{batch} = HGT\left(H^{(l-1)}_{batch}, \quad G_{batch}\right) \quad (8)$$

where G_{batch} is the sampled input graph, and $H^{(l-1)}_{batch}$ and H^l_{batch} are the input and output feature vectors of layer l. All layers have eight HGT attention heads like the original work on HGT (Hu *et al.* 2020). The outputs of all three layers are concatenated and fed to the following linear layer.

$$Feat = ReLU\left(Linear\left(i \in [i,\ L] || H^i_{batch}\right)\right) \quad (9)$$

where *Feat* is the output feature vector for subgraph nodes.

2.4 Link prediction

Link prediction tasks are carried out with a simple two-layer fully connected network, which receives input from a concatenated collection of drug and disease feature vectors derived from the HGT network. Suppose we want to check the score of the drug i and disease j in our input subgraph. We first concatenate HGT embeddings of the drug and the disease:

$$H^0_{FC} = Feat_{Drug_i} \ || \ Feat_{Disease_j} \quad (10)$$

where $Feat_{Drug_i}$ and $Feat_{Disease_j}$ are extracted features for drug i and disease j, respectively. The first layer's input dimension size is 128 (concatenation of two 64-dimensional embeddings), and the output dimension size is 64. Batch normalization is used in this layer, ReLU is the activation function, and the dropout rate is 0.2.

$$H^1_{FC} = Dropout\left(ReLU\left(BatchNorm\left(Linear_{128 \to 64}\left(H^0_{FC}\right)\right)\right)\right) \quad (11)$$

The second layer is as follows:

$$Score = \sigma\left(Linear_{64 \to 1}\left(H^1_{FC}\right)\right) \quad (12)$$

where σ is the sigmoid function, and *Score* is the output of a fully connected network, which is the indication score between drug i and disease j. Figure 1 demonstrates each step of the method in detail.

2.5 Training

The graph is generated, as explained in Section 2.2. During training, subgraphs are generated by the HGSampling algorithm using drugs as initial sample nodes. The batch size of

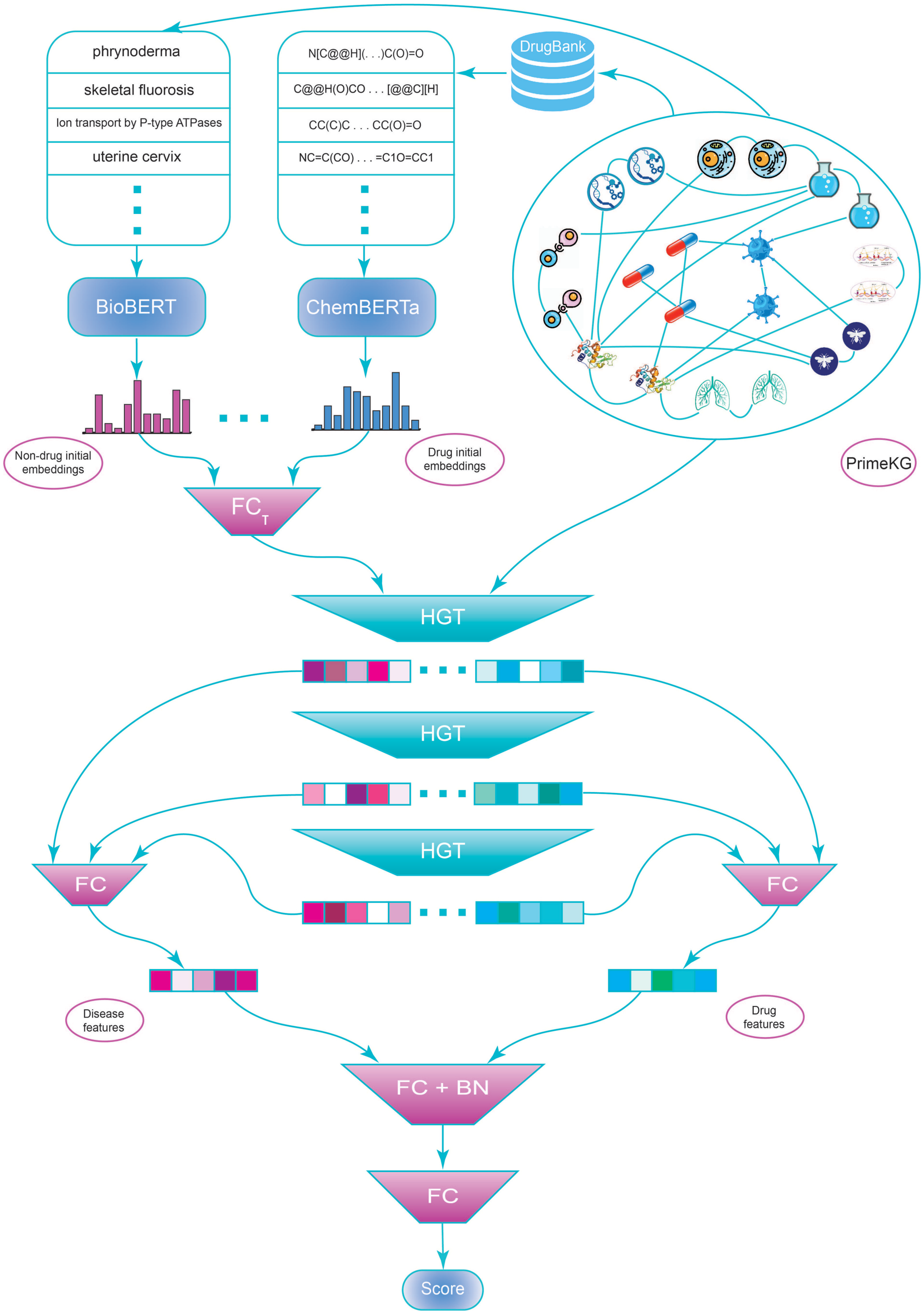


Figure 1. (1) PrimeKG is obtained as the knowledge graph, (2) drug SMILES representations are extracted from DrugBank, and entity names are extracted from PrimeKG, (3) initial embeddings are extracted using BioBERT and ChemBERTa, (4) node embeddings are computed using HGT layers, (5) final embeddings are obtained with applying a fully connected layer on a concatenation of node embeddings of different layers, and (6) drug-disease relation score is computed using a 2-layer FC network.

sampling is 164 because we have 1801 drugs in our graph, and we want our subgraphs to be similar in size. And other batch sizes can let one batch have very few drugs. HGSampling produces a sampled sub-graph of L depth from the initial nodes when the sampling depth is L. As we employ three HGT layers in the model, embeddings are constructed from neighboring nodes within a distance of three nodes. Therefore, we chose to have a sampling depth of three. As demonstrated in Table 6, similar results are obtained when different sampling depths (L = 2, 3, or 4) are used; however, sampling depths of 2 and 3 perform slightly better than depths of 4. Also, the number of nodes sampled in each iteration for each node type is 512. The HGT network and link-predicting network are trained as an end-to-end model.

We use the binary cross-entropy function as the model's loss function on $I_{prediction_{batch}}$ which contains the positive and negative samples for G_{batch}.

$$Loss = -\frac{1}{n}\sum_{i=1}^{n} Y_i \ . \ \log(\hat{Y}_i) + (1 - Y_i) \ . \ \log(1 - \hat{Y}_i) \tag{13}$$

where n is the number of all samples, Y_i is the label of the ith sample, and $\hat{Y}_i$ is the output of the model for the ith sample. Positive samples are masked indications of the subgraph, and negative samples are randomly selected from all possible edges between the subgraph's drugs and diseases, which are not actually in the subgraph's masked or unmasked indications.

The model has been optimized via the AdamW (Loshchilov and Hutter 2019) optimizer with a cosine annealing learning rate scheduler (Loshchilov and Hutter 2017). The model is trained for 300 epochs.

3 Experiments

Our experiments and their results are explained in this section. We compared HGTDR with LAGCN (Yu *et al.* 2021), DeepDR (Zeng *et al.* 2019), and HeTDR (Jin *et al.* 2021) using 5-fold cross-validation to evaluate the proposed method. The results for all methods were regenerated for this work. AUROC and AUPR are used as evaluation metrics for comparison.

Hyperparameter optimization is also done on the percent of masked indications (20%, 50%, 80%), number of HGT layers (3, 4), and layer feature space dimension (32, 64). The model is run using an NVIDIA GeForce GTX 1080 Ti GPU with a memory size of 11GB.

As described in the introduction, we compared the effectiveness of HGT layers with HAN layers to show their effectiveness. Furthermore, to show the result's insensitivity to the sampling method, we tested different sampling depths and repeated validation with different samplings.

In Section 2.2, we claim that drug and disease nodes should be removed to make our data comparable to previous works, and using those removed nodes can increase our method's performance unrealistically. In an effort to validate this fact, we carried out an experiment where the nodes were not removed.

We conducted another experiment to compare HeTDR and HGTDR's performance for new diseases, where literature and indication information are limited. In this experiment, we used 90% of diseases for training and 10% for testing. We removed BioBERT embeddings for diseases of each method, and only one indication from every validation disease was used in training data. Since one of HeTDR's limitations is that it does not support validating diseases that do not indicate training data.

Furthermore, we ran another experiment to check the robustness of our model to different input types. In this experiment, we ran our model five times, and each time, we removed one type of relation in the input graph (five edge types with most instances are used).

In addition, to investigate the model's ability to extract the necessary information for its task, we use it without any modifications to predict other relations instead of indications. Input data and model are both the same in this experiment.

3.1 Results

For cross-validation, we divided indications into five subsets. Four subsets were used for training the model, and one subset was used for validation in each fold. For each metric, the average of 5-fold is reported. Figure 2 shows the comparison results of the four methods. Only HeTDR performs slightly better than our method.

We showed the effectiveness of HGT layers by swapping them with HAN layers. Table 4 demonstrates the higher performance of HGT layers in comparison with HAN layers.

Additionally, we demonstrated that different samplings using the HGSampling algorithm do not lead to much difference in prediction results. Table 5 shows validation results for 10 repeats on one cross-validation fold with different samplings.

Moreover, Table 6 shows that the method is not sensitive to sampling depth and performs well when using 2 or 3 sampling depths.

Table 7 investigates the effect of drug and disease nodes' removal, as discussed in Section 2.2. We can see that not removing those nodes increases the method's performance. However, since other methods do not use such drug and disease nodes where there is no indication related, we decided not to use those nodes in our main method.

In a second experiment designed for new diseases, we compared HGTDR and HeTDR. Table 8 demonstrates the results of the two methods with the data limitations of new diseases.

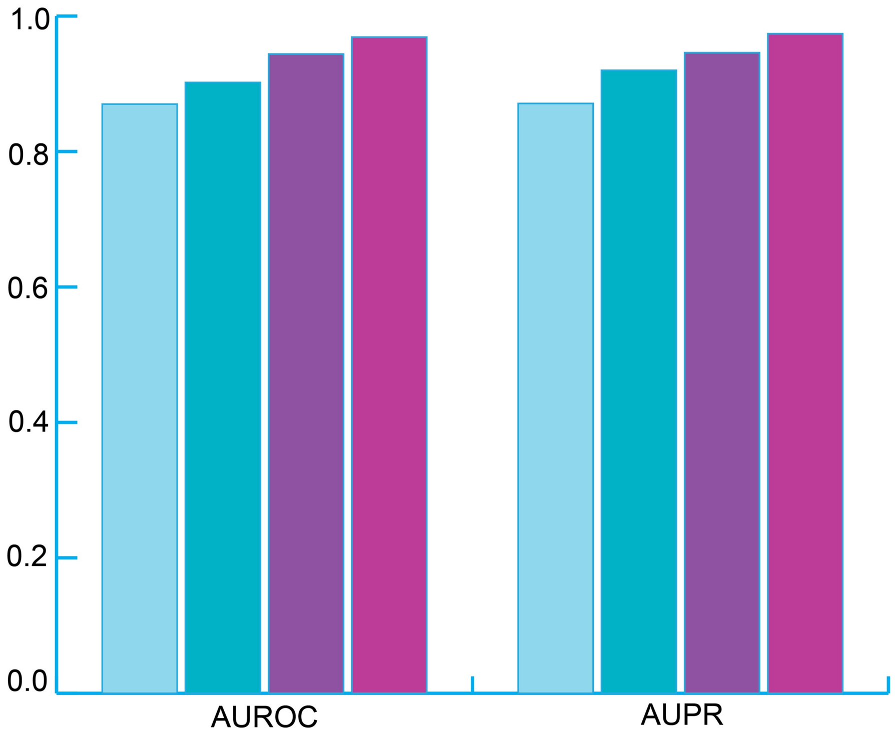
As explained in Section 3, cross-validation was repeated for each of the five robustness testing experiments. Table 9 demonstrates the results of robustness testing of HGTDR to input variation. There is no significant difference in the results of the six types of inputs, which shows that HGTDR is robust to relation removals. Notably, more than half of the edges of our original graph are in anatomy-protein type, and removing the type does not have much effect on the results. Any relationship contains valuable information for drug repurposing or can be considered noise in the input graph. Using this method, we do not need to know which relation type is proper and which is noise for our task.

3.1.1 Novel relation prediction

Although common metrics like AUROC and AUPR show the general correctness of these methods, they are not entirely reliable since the ultimate goal of drug repurposing is to find false positive examples. Hence, a method with an accuracy of 100% would not be helpful because it cannot suggest novel drug repurposing candidates. To predict novel indications, we trained the model with all indications and made 110 batches (10 sets of batches with batch size 164) from the input graph with positive and negative samples. False positive predictions can be inferred as drug repurposing candidates. We further investigated medical literature to find evidence for novel relations predicted by HGTDR. Table 10 demonstrates discovered evidence from the literature for 10 highly scored false positive predictions by HGTDR.

3.1.2 Other applications

Even though our primary goal in developing this method was to predict drug–disease links, its generality enables us to use it to predict any relationship in the graph. Thus, we used it to predict other relation types. Table 11 shows evaluation results for those relations.



	Method	AUROC	AUPR
●	LAGCN	0.87	0.871
●	DeepDR	0.902	0.92
●	HGTDR	0.944	0.946
●	HeTDR	0.969	0.974

Figure 2. Comparing AUROC and AUPR metrics for LAGCN, DeepDR, HeTDR, and HGTDR.

Table 5. Validation results for 10 repeats on 1-fold of cross-validation.

AUROC	AUPR
0.951	0.954
0.950	0.952
0.949	0.950
0.951	0.953
0.950	0.951
0.950	0.953
0.951	0.954
0.945	0.947
0.950	0.951
0.954	0.958

Table 6. Comparing HGSampling with different sampling depths.

Sampling depth	AUROC	AUPR
2	0.946	0.945
3	0.944	0.946
4	0.943	0.944

We also investigated other research to evaluate these predictions experimentally. Tables 12 and 13 show the experimental validation for disease–protein and drug–protein relations. Medical literature supports five out of five disease–protein relation predictions and three out of five drug–protein relation predictions. Experimental and numerical evidence suggests that the model can extract the information it needs to complete its task without changing the inputs or providing additional information.

To determine which information from the graph is most relevant to each downstream task, we used meta-relation attention to identify the most important relations for each task. Supplementary Tables S1–S7, in supplementary materials, present the top five meta-relations for all three layers in our model for different tasks. For indication prediction, we can see that other relations between drugs and diseases, like contraindication and off-label use, are among the important meta-relations, which is logical. Furthermore, we can see that

Table 7. HGTDR performance compared with when no node is removed from the input graph.

Method	AUROC	AUPR
HGTDR	0.944	0.946
HGTDR without node removal	0.979	0.970

Table 8. HGTDR and HeTDR results without literature and indication information.

Method	AUPR	AUROC
HGTDR	0.859	0.871
HeTDR	0.779	0.782

Table 9. The method results when a relation type is removed from the input graph.

Removed relation	AUPR	AUROC
Nothing removed	0.946	0.944
Anatomy–protein(present)	0.949	0.949
Drug–drug	0.945	0.944
Protein–protein	0.944	0.943
Biological process–protein	0.940	0.942
Cellular component–protein	0.939	0.939

Table 10. Experimental evidence of novel predicted relations.

Drug	Disease	Evidence
Betamethasone	Lichen planus	(Cawson 1968)
Dactinomycin	Acute myeloid leukemia	(Falini *et al.* 2015)
Vincristine	Ewing sarcoma	(Wagner *et al.* 2017)
Paclitaxel	Classic Hodgkin lymphoma	(Sinha *et al.* 2013)
Prednisolone	Trichinellosis	(Shimoni *et al.* 2007)
Dexamethasone	Blastomycosis	No evidence
Vinblastine	Malignant Sertoli-Leydig Cell tumor of the ovary	(Finan *et al.* 1992)
Escitalopram	Social phobia	(Pelissolo 2008)
Dactinomycin	Plasmablastic lymphoma	No evidence
Cetirizine	Urticaria	(Kalivas *et al.* 1990)

Table 11. Evaluation results when predicting relations other than indication.

Relation	AUPR	AUROC
Disease–protein	0.912	0.926
Drug–protein	0.951	0.948
Pathway–protein	0.951	0.953
Drug–phenotype	0.885	0.911
Biological process–protein	0.889	0.888
Protein–protein	0.881	0.887

most of the important meta-relations in all tables have a common node type with the meta-path that is being predicted. For example, in protein–protein prediction, all important meta-paths have node types of protein in them.

4 Discussion

Drug repurposing is a promising strategy to overcome current drug development limitations such as high failure risk and long development time. Therefore, many works have tried to develop systematic experimental and computational drug repurposing methods. Some previous works suggested methods that limit input data in different ways to increase numerical evaluation performance or fit the data to their models. However, numerical evaluation cannot be fully trusted in this task since a method's lower accuracy is probably due to suggesting more actual drug repurposing candidates. Hence, manual data manipulation may unnoticeably hinder a method's drug-repurposing ability.

Table 12. Experimental evidence of novel disease–protein relations.

Protein	Disease	Evidence
VEGFA	Lynch syndrome	(Tjalma *et al.* 2016)
CCND1	Acute lymphoblastic leukemia	(Hsu *et al.* 2021)
MTHFR	Carcinoma of esophagus	(Wang *et al.*, 2008)
IL6	Scleroderma	(Feghali *et al.* 1992)
IL1B	Hepatitis	(Hirankarn *et al.* 2006)

Table 13. Experimental evidence of novel drug–protein relations.

Protein	Drug	Evidence
OPRM1	Clozapine	(Solismaa *et al.* 2018)
SHBG	Fostamatinib	No evidence
UGT1A8	Acetaminophen	(Yoshizawa *et al.* 2020)
DRD5	Pentobarbital	No evidence
CYP1A1	Amitriptyline	(Mayer *et al.* 2007)

This work presents an end-to-end, automatic method that removes previous works' data limitations while maintaining comparable performance. This method can use any graph data type, even with massive scales, plus embedded side information to predict novel indications. We do not force the method to utilize some parts of the data with a greater emphasis. Nevertheless, the model learns which input part is relevant to the task. As a result of this feature, the method also performs well on other tasks, such as predicting drug–protein and disease–protein relationships.

A three-step method for drug repurposing is proposed using a heterogeneous input graph that overcomes some of the data-related limitations of previous studies. It is important to note that data limitations can make a method incapable of helping in the case of new disease outbreaks such as COVID, in which only limited information is available regarding the disease. In the first step, network construction, we attempt to use an existing graph rather than create a new one instead of previous works. As a result, we are preventing the pipeline from being biased by our domain knowledge. Additionally, we avoid any further preprocessing, such as constructing similarity matrices, which is another source of bias, and allow our method to determine how information should be extracted from the input data. We only remove drug and disease nodes that do not contribute to any indication edge to make the results comparable to previous works. It is noteworthy that removing the nodes adversely impacts our performance, which aligns with expectations since it is easy for the model to predict no indication relating to those nodes. Our method could add any embedded information to graph entities. Therefore, we add BioBERT and ChemBERTa embeddings to the graph, providing literature and drug structure information. In the second step, heterogeneous graph transformer layers extract node features. The architecture of HGT can extract implicit meta-paths of the graph and capture the common and specific patterns of different relationships. Due to these properties, we can use any entity and relations in the input with side information without manipulating them to increase performance. As high-throughput technologies make vast amounts of data available in various areas, scalable drug repurposing has become increasingly important. Thus, we use HGSampling to make the method scalable as well. Lastly, we use a fully connected neural network end-to-end with the previous steps to customize the extracted features for the downstream task. This enables the method to be employed for other similar tasks, too.

A 5-fold cross-validation procedure was utilized for evaluation, and the AUPR and AUROC metrics were compared with three other state-of-the-art methods. The method named HeTDR performed better than HGTDR by about 2% in both measures with their default conditions. However, HGTDR outperformed HeTDR by about 8% when new disease conditions are assumed in both methods. These metrics, however, cannot demonstrate the same performance since we cannot evaluate all the false positive samples that may be unknown true positives. We examined the experimental evidence for this case's top 10 false positives. Eight out of 10 predictions had evidence from literature suggesting an indication relationship. We also found that our method was robust to input variation when some relations from the input graph were removed. To show the applicability of HGTDR in other tasks, we changed the downstream task of indication prediction to six other tasks without modifying the model. Both numerical and experimental evaluations suggest that the model can extract informative features when the task is shifted.

Our method facilitates adding information to a knowledge graph and predicting hitherto unidentified information. Nonetheless, the extension of medical data encounters constraints, notably due to the lack of unique identifiers for entities, which hampers the addition of fresh information. While there is a potential to map and convert identifiers, the process is not exhaustive, with some data remaining unmapped and hence unconverted. Another limitation inherent in our method is its focus on predicting relationships numerically without the capacity to substantiate these predictions with empirical evidence. For instance, although our method can predict whether a drug is suitable for treating a disease, it cannot elucidate the drug's mechanism of action. Additionally, this method does not allow us to add side information to edges like we do for nodes. It is important to note that some information does not relate to a node but rather to an edge. As an example, the expression of a gene in a disease is information related to an edge that this approach cannot use.

In future research, there is the opportunity to enrich the knowledge graph by expanding the existing graph structure and incorporating varied types of ancillary information from emergent sources. Given the multifaceted nature of drug development, a holistic assimilation of information could substantially enhance performance. Additionally, the methodology has potential applications in combination therapies, which could benefit from the integrated embeddings of multiple drugs. Furthermore, efforts can be directed toward augmenting the interpretability of the end-to-end framework, thus increasing the utility and understanding of the predictive outcomes.

Supplementary data

Supplementary data are available at *Bioinformatics* online.

Conflict of interest

None declared.

Funding

K.A. was supported by Iran National Science Foundation (INSF) under Grant No. 99025741, and H.R.R. was supported by Iran National Science Foundation (INSF) under Grant No. 96006077.

Data availability

Our code and data are available at: https://github.com/bcb-sut/HGTDR; http://git.dml.ir/BCB/HGTDR.

References

Cawson R. Treatment of oral lichen planus with betamethasone. *Br Med J* 1968;**1**:86–9.

Cen Y, Zou X, Zhang J *et al.* Representation learning for attributed multiplex heterogeneous network. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, New York, NY, USA: ACM, 2019, 1358–68.

Chandak P, Huang K, Zitnik M. Building a knowledge graph to enable precision medicine. *Sci Data* 2023;**10**:67.

Che M, Yao K, Che C *et al.* Knowledge-graph-based drug repositioning against COVID-19 by graph convolutional network with attention mechanism. *Future Internet* 2021;**13**:13.

Chen Y, de Rijke M. A collective variational autoencoder for top-n recommendation with side information. In: *Proceedings of the 3rd workshop on deep learning for recommender systems*, New York, NY, USA: ACM, 2018, 3–9.

Chithrananda S, Grand G, Ramsundar B. Chemberta: large-scale self-supervised pretraining for molecular property prediction. arXiv, arXiv:2010.09885, 2020, preprint: not peer reviewed.

Falini B, Brunetti L, Martelli MP. Dactinomycin in NPM1-mutated acute myeloid leukemia. *N Engl J Med* 2015;**373**:1180–2.

Feghali C *et al.* Mechanisms of pathogenesis in scleroderma. I. Overproduction of interleukin 6 by fibroblasts cultured from affected skin sites of patients with scleroderma. *J Rheumatol* 1992; **19**:1207–11.

Finan M, Roberts W, Kavanagh J. Ovarian Sertoli-Leydig cell tumor: success with salvage therapy. *Int J Gynecol Cancer* 1992;**3**:189–91.

Himmelstein DS, Lizee A, Hessler C *et al.* Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *Elife* 2017; **6**:e26726.

Hirankarn N, Kimkong I, Kummee P *et al.* Interleukin-1β gene polymorphism associated with hepatocellular carcinoma in hepatitis B virus infection. *World J Gastroenterol* 2006;**12**:776–9.

Hsu P-C, Pei J-S, Chen C-C *et al.* Significant association of CCND1 genotypes with susceptibility to childhood acute lymphoblastic leukemia. *Anticancer Res* 2021;**41**:4801–6.

Hu Z, Dong Y, Wang K *et al.* Heterogeneous graph transformer. In: *Proceedings of the web conference 2020*, New York, NY, USA: ACM, 2020, 2704–10.

Ioannidis VN, Song X, Manchanda S *et al.* Drkg-drug repurposing knowledge graph for COVID-19. arXiv, arXiv:2010.09600, 2020, preprint: not peer reviewed.

Jin S, Niu Z, Jiang C *et al.* HeTDR: drug repositioning based on heterogeneous networks and text mining. *Patterns* 2021;**2**:100307.

Kalivas J, Breneman D, Tharp M *et al.* Urticaria: clinical efficacy of cetirizine in comparison with hydroxyzine and placebo. *J Allergy Clin Immunol* 1990;**86**:1014–8.

Law V, Knox C, Djoumbou Y *et al.* DrugBank 4.0: shedding new light on drug metabolism. *Nucleic Acids Res* 2014;**42**:D1091–7.

Lee J, Yoon W, Kim S *et al.* BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 2020;**36**:1234–40.

Loshchilov I, Hutter F. SGDR: Stochastic gradient descent with warm restarts. In: *International Conference on Learning Representations (Poster), Toulon, France*, 2017.

Loshchilov I, Hutter F. Decoupled weight decay regularization. In: *7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, May* 2019, 6–9. OpenReview.net.

Mayer RT, Dolence EK, Mayer GE. A real-time fluorescence assay for measuring N-dealkylation. *Drug Metab Dispos* 2007;**35**:103–9.

Ng A. Sparse autoencoder. *CS294A Lecture Notes* 2011;**72**:1–19.

Pan X *et al.* Deep learning for drug repurposing: methods, databases, and applications. *Wiley Interdiscip Rev Comput Mol Sci* 2022; **12**:e1597.

Pelissolo A. Efficacy and tolerability of escitalopram in anxiety disorders: a review. *Encephale* 2008;**34**:400–8.

Percha B, Altman RB. A global network of biomedical relationships derived from text. *Bioinformatics*2018;**34**:2614–24.

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

Shimoni Z *et al.* The use of prednisone in the treatment of trichinellosis. *Age (Yrs)* 2007;**32**:22.

Singh TU, Parida S, Lingaraju MC *et al.* Drug repurposing approach to fight COVID-19. *Pharmacol Rep* 2020;**72**:1479–508.

Sinha R, Shenoy PJ, King N *et al.* Vinorelbine, paclitaxel, etoposide, cisplatin, and cytarabine (VTEPA) is an effective second salvage therapy for relapsed/refractory Hodgkin lymphoma. *Clin Lymphoma Myeloma Leuk* 2013;**13**:657–63.

Solismaa A, Kampman O, Lyytikäinen L-P *et al.* Genetic polymorphisms associated with constipation and anticholinergic symptoms in patients receiving clozapine. *J Clin Psychopharmacol*2018;**38**:193–9.

Talevi A, Bellera CL. Challenges and opportunities with drug repurposing: finding strategies to find alternative uses of therapeutics. *Expert Opin Drug Discov* 2020;**15**:397–401.

Tjalma JJ, Garcia-Allende PB, Hartmans E *et al.* Molecular fluorescence endoscopy targeting vascular endothelial growth factor a for improved colorectal polyp detection. *J Nucl Med* 2016;**57**:480–5.

Wagner MJ, Gopalakrishnan V, Ravi V *et al.* Vincristine, ifosfamide, and doxorubicin for initial treatment of Ewing sarcoma in adults. *Oncologist* 2017;**22**:1271–7.

Wang J, Sasco AJ, Fu C *et al.* Aberrant DNA methylation of P16, MGMT, and hMLH1 genes in combination with MTHFR C677T genetic polymorphism in esophageal squamous cell carcinoma. *Cancer Epidemiol Biomarkers Prev* 2008;**17**:118–25.

Wang X, Ji H, Shi C *et al.* Heterogeneous graph attention network. In: *The World Wide Web Conference*, New York, NY, USA: ACM, 2019, 2022–32.

Weininger D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J Chem Inf Comput Sci* 1988;**28**:31–6.

Yoshizawa K, Arai N, Suzuki Y *et al.* Synergistic antinociceptive activity of tramadol/acetaminophen combination mediated by μ-opioid receptors. *Biol Pharm Bull* 2020;**43**:1128–34.

Yu P, Fu C, Yu Y *et al.* Multiplex heterogeneous graph convolutional network. In: *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM 2022, 2377–87.

Yu Z, Huang F, Zhao X *et al.* Predicting drug-disease associations through layer attention graph convolutional network. *Brief Bioinform* 2021;**22**:bbaa243.

Zeng X, Zhu S, Lu W *et al.* Target identification among known drugs by deep learning from heterogeneous networks. *Chem Sci* 2020; **11**:1775–97.

Zeng X, Zhu S, Liu X *et al.* deepDR: a network-based deep learning approach to in silico drug repositioning. *Bioinformatics* 2019; **35**:5191–8.

Zhang C, Song D, Huang C *et al.* Heterogeneous graph neural network. In: *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, New York, NY, USA: ACM*, 2019, 793–803.

Zhang R, Hristovski D, Schutte D *et al.* Drug repurposing for COVID-19 via knowledge graph completion. *J Biomed Inform* 2021; **115**:103696.

Zhang Z, Zhou L, Xie N *et al.* Overcoming cancer therapeutic bottleneck by drug repurposing. *Signal Transduct Target Ther* 2020; **5**:113.

Zhou Y, Wang F, Tang J *et al.* Artificial intelligence in COVID-19 drug repurposing. *Lancet Digital Health* 2020;**2**:e667–76.

Zhu Y, Che C, Jin B *et al.* Knowledge-driven drug repurposing using a comprehensive drug knowledge graph. *Health Informatics J* 2020; **26**:2737–50.

Bioinformatics, 2024, 40, 1–11
https://doi.org/10.1093/bioinformatics/btae349
Original Paper