Chapter 24
GNN-based Biomedical Knowledge Graph Mining in Drug Development

Chang Su, Yu Hou, Fei Wang

Abstract Drug discovery and development (D^3) is an extremely expensive and time consuming process. It takes tens of years and billions of dollars to make a drug successfully on the market from scratch, which makes this process highly inefficient when facing emergencies such as COVID-19. At the same time, a huge amount of knowledge and experience has been accumulated during the D^3 process during the past decades. These knowledge are usually encoded in guidelines or biomedical literature, which provides an important resource containing insights that can be informative of the future D^3 process. Knowledge graph (KG) is an effective way of organizing the useful information in those literature so that they can be retrieved efficiently. It also bridges the heterogeneous biomedical concepts that are involved in the D^3 process. In this chapter we will review the existing biomedical KG and introduce how GNN techniques can facilitate the D^3 process on the KG. We will also introduce two case studies on Parkinson's disease and COVID-19, and point out future directions.

24.1 Introduction

Biomedicine is a discipline with lots of highly specialized knowledge accumulated from biological experiments and clinical practice. This knowledge is usually buried

Chang Su,
Department of Population Health Sciences, Weill Cornell Medicine, e-mail: chs4001@med.cornell.edu

Yu Hou,
Department of Population Health Sciences, Weill Cornell Medicine, e-mail: yuh4001@med.cornell.edu

Fei Wang,
Department of Population Health Sciences, Weill Cornell Medicine, e-mail: few2001@med.cornell.edu

in massive biomedical literature and text books. This makes effective knowledge organization and efficient knowledge retrieval a challenging task. Knowledge graph is a recently emerged concept aiming at achieving this goal. A knowledge graph (KG) stores and represents knowledge by constructing a semantic network describing entities and the relationships between them. The basic elements comprising a knowledge graph are a set of ⟨head, relation, tail⟩ tuples, where the heads and tails are concept entities and relations link these entities with semantic relationships. In biomedicine, the typical entities could be diseases, drugs, genes, etc., and the relationships could be treats, binds, interactions, etc. Large scale biomedical KG makes efficient knowledge retrieval and inference possible.

Biomedical KG can effectively complement the biomedical data analytics processes. In particular, many different types of biomedical data are heterogeneous and noisy (Wang et al, 2019f; Wang and Preininger, 2019; Zhu et al, 2019e), which makes the data-driven models developed on these data not reliable for real practice. Biomedical KGs (BKGs) effectively encode the biomedical entities and their semantic relationships, which can serve as "prior knowledge" to guide the downstream data-driven analytics procedure and improve the quality of the model. On the other hand, we can also use BKGs to generate hypotheses (such as which drug can be used to treat which disease), and get them validated in real world health data (such as electronic health records).

In this chapter, we will review existing BKGs and present examples of how BKGs can be used for generating drug repurposing hypotheses, and point out future directions.

24.2 Existing Biomedical Knowledge Graphs

This section surveys the existing BKGs that are publicly available and the ways of BKG construction and curation (Table 24.3).

A common way for constructing a BKG is to extract and integrate data from data resources, usually, which are manually curated to summarize and organize the biomedical knowledge derived from biological experiments, clinical trials, genome wide association analyses, clinical practices, etc (Santos et al, 2020; Ioannidis et al, 2020; Himmelstein et al, 2017; Rizvi et al, 2019; Yu et al, 2019b; Zhu et al, 2020b; Zeng et al, 2020b,b; Domingo-Fernández et al, 2020; Wang et al, 2020e; Percha and Altman, 2018; Li et al, 2020d,b; Goodwin and Harabagiu, 2013; Rotmensch et al, 2017; Sun et al, 2020a). In table 24.2, we summarized some public data resources that have been commonly used in the construction of BKGs. For instance, Comparative Toxicogenomics Database (CTD) (Davis et al, 2019) is an open resource providing rich, manually curated chemical–gene, chemical–disease and gene–disease relational data, for the aim of advancing understanding the impacts of environmental exposures on human health. DrugBank (Wishart et al, 2018) is a database containing information of the approved drugs and drugs under trial, as well as the pharmacogenomic data (e.g., drug-target interactions). Ontology resources like Gene Ontology

(Ashburner et al, 2000) and Disease Ontology (Schriml et al, 2019) stored functional and semantic context of genes and diseases, respectively. By integrating data from these rich resources, a number of BKGs have been constructed (Santos et al, 2020; Ioannidis et al, 2020; Himmelstein et al, 2017; Rizvi et al, 2019; Yu et al, 2019b; Zhu et al, 2020b; Zeng et al, 2020b,b; Domingo-Fernández et al, 2020; Wang et al, 2020e). For example, Hetionet (Himmelstein et al, 2017), released in 2017, is a well-curated BKG that integrates 29 publicly available biomedical databases. It contains 11 types of 47031 biomedical entities and 24 types of over 2 million relations among thoses entities. Similar to Hetionet, Drug Repurposing Knowledge Graph (DRKG) (Ioannidis et al, 2020) was built by integrating data from six different existing biomedical databases, containing 13 types of about 100K entities and 107 types of over 5 million relationships. Zhu et al (2020b) constructed a drug-centric BKG by systematically integrating multiple drug databases such as DrugBank (Wishart et al, 2018) and PharmGKB (Whirl-Carrillo et al, 2012). Hetionet, DRKG, and BKGs have been used in accelerating computational drug repurposing. PreMedKB (Yu et al, 2019b) includes the information of disease, genes, variants, and drugs by integrating relational data among them from existing resources. By integrating multiple dietary related databases, Rizvi et al (2019) built a BKG, named Dietary Supplements Knowledge Base (iDISK), which covers knowledge of dietary supplements, including vitamins, herbs, minerals, etc. The Clinical Knowledge Graph (CKG)(Santos et al, 2020) was constructed by integrating relevant existing biomedical databases such as DrugBank (Wishart et al, 2018), Disease Ontology (Schriml et al, 2019), SIDER (Kuhn et al, 2016), etc. and knowledge extracted from scientific literature. It contains over 16 million nodes and over 220 million relationships. Compared to other BKGs, CKG has a finer granularity of knowledge as it involves more entity types such as metabolite, modified protein, molecule function, transcript, genetic variant, food, clinical variable, etc.

As the rapid development of biomedical research, a continuously increasing volume of biomedical articles have been published every day. Manually extracting knowledge from literature for BKG cuuration is no longer sufficient to meet current needs. To this end, efforts have been made in using text mining methods to extract biomedical knowledge from scientific literature to construct BKGs (Domingo-Fernández et al, 2020; Wang et al, 2020e; Percha and Altman, 2018; Li et al, 2020d). For example, Sun et al (2020a) constructed a knowledge graph by extracting biomedical entities and relationships from drug descriptions, medical dictionaries, and literature to identify suspected cases of Fraud, Waste, and Abuse from claim files. COVID-KG (Wang et al, 2020e) and COVID-19 Knowledge Graph (Domingo-Fernández et al, 2020) were built by extracting COVID-19 specific knowledge from biomedical literature. The resulting COVID-19 specific BKGs contain entities such as diseases, chemicals, genes, and pathways, along with their relationships. KGHC (Li et al, 2020d) is a BKG with the specific focus on hepatocellular carcinoma. It was built by extracting knowedge from literature and contents on the internet, as well as structured triples from SemMedDB (Kilicoglu et al, 2012). In addition, some studies (Goodwin and Harabagiu, 2013; Li et al, 2020b; Rotmensch et al, 2017; Sun et al, 2020a) tried to build BKGs from clinical data such as electronic

health records (EHRs) and electronic medical records (EMRs). For instance, Rotmensch et al (2017) constructed a BKG by extracting disease-symptom associations from EHR data using the data-driven approach. Li et al (2020b) proposed a systematic pipeline for extracting BKG from large scale EMR data. Compared to other BKGs based on triplet structure, the resulting KG is based a quadruplet structure, i.e., $\langle head, relation, tail, property \rangle$. Here the property includes information such as co-occurrence number, co-occurrence probablity, specificity, and reliability of the corresponding $\langle head, relation, tail \rangle$ triplet.

Table 24.1: Summary of existing BKGs.

BKGs	Entities	Relations	Focus	Construction method	URL
Clinical Knowledge Graph (Santos et al. 2020)	16 million entities from 33 entity types	220 million relations from 51 relation types	General	Resources Integration	https://github.com/MannLabs/CKG
Drug Repurposing Knowledge Graph (Ioannidis et al. 2020)	97,238 entities from 13 entity types	5,874,261 relations from 107 relation types	General	Resources Integration	https://github.com/gnn4dr/DRKG
Hetionet (Himmelstein et al. 2017)	47,031 entities from 11 entity types	2,250,197 relations from 24 relation types	General	Resources Integration	https://het.io/
iDISK (Rizvi et al. 2019)	144,059 entities from 6 entity types	708,164 relations from 6 relation types	Dietary Supplements	Resources Integration	https://conservancy.umn.edu/handle/11299/204783
PreMedKB (Yu et al. 2019b)	404,904 entities from 4 entity types	496,689 relations from 52 relation types	General	Resources Integration	http://www.fudan-pgx.org/premedkb/index.html#/home
Zhu et al. (2020b)	5 entity types	9 relation types	General	Resources Integration	-
Zeng et al. (2020b)	145,179 entities from 4 entity types	15,018,067 relations from 39 relation types	General	Resources Integration	-
COVID-19 Knowledge Graph (Domingo-Fernández et al. 2020)	3,954 entities from 10 entity types	9,484 relations	COVID-19	Literature Mining	https://github.com/covid19kg/covid19kg
COVID-KG (Wang et al. 2020e)	67,217 entities from 3 entity types	85,126,762 relations from 3 relation types	COVID-19	Literature Mining	http://blender.cs.illinois.edu/covid19/
Global Network of Biomedical Relationships (Percha and Altman 2018)	Three entity types (Chemical, Disease, Gene)	2,236,307 relations from 36 relation types	General	Literature Mining	https://zenodo.org/record/1035500
KGHC (Li et al. 2020d)	5,028 entities from 9 entity types	13,296 relations	Hepatocellular Carcinoma	Literature Mining	http://202.118.75.18:18895/browser/
Li et al. (2020b)	22,508 entities from 9 entity types	579,094 relations	General	EHR Mining	-
QMKG (Goodwin and Harabagiu 2013)	634,000 entities	1,390,000,000 relations	General	EHR Mining	-
Rotmensch et al. (2017)	647 entities from 2 entity types	Disease-Symptom	General	EHR Mining	-
Sun et al. (2020a)	1,616,549 entities from 62 entity types	5,963,444 relations from 202 relation types	General	EHR Mining	https://web.archive.org/web/20191231152615if_/http://121.12.85.245:1347/kg_test/#/

Table 24.2: Publicly available resources for BKG construction

Database	Entities	Relations	Short Description	URL
Bgee (Bastian et al. 2021)	60,072 Anatomy and Gene entities	11,731,369 relations in terms of presence/absence of expression	A database for Anatomy-Gene Expression	https://bgee.org/
Comparative Toxicogenomics Database (Davis et al. 2019)	73,922 Disease, Gene, Chemical, Pathway entities	38,344,568 Chemical-Gene, Chemical-Disease, Chemical-Pathway, Gene-Disease, Gene-Pathway, and Disease-Pathway relations	A database that is manually curated includes chemical-disease-gene-pathway relations	http://ctdbase.org/
Drug–Gene Interaction Database (Cotto et al. 2018)	160,054 Drug and Gene entities	96,924 Drug-Gene Interaction relations	A database for drug-gene interactions	https://www.dgidb.org/
DISEASES (Pletscher-Frankild et al. 2015)	22,216 Disease and Gene entities	543,405 relations	A database for Disease-Gene Association	https://diseases.jensenlab.org/
DisGeNET (Piñero et al. 2020)	159,052 Disease, Gene and Variant entities	839,138 Gene-Disease,Variant-Disease relations	A database that integrates data from expert-curated repositories for genes and variants associated with human diseases.	https://www.disgenet.org/home/
IntAct (Orchard et al. 2014)	119,281 Chemical and Gene entities	1,130,596 relations	A database for molecular interaction data	https://www.ebi.ac.uk/intact/
STRING (Szklarczyk et al. 2019)	24,584,628 Protein entities	3,123,056,667 Protein-Protein Interaction relations	A database for Protein-Protein Interaction netword	https://string-db.org/
SIDER (Kuhn et al. 2016)	7,298 Drug and Side-effect entities	139,756 Drug-Side effect relations	A database contains medicines and their recorded adverse drug reactions	http://sideeffects.embl.de/
SIGNOR (Licata et al. 2020)	7,095 entities from 10 entity types	26,523 relations	A database for signaling information published in the scientific literature	https://signor.uniroma2.it/
TISSUE (Palasca et al. 2018)	26,260 entities in Tissue and Gene	6,788,697 relations	A database for Tissue-Gene Expression by literature curated manually	https://tissues.jensenlab.org/
DrugBank (Wishart et al. 2018)	15,128 Drug entities	28,014 Drug-Target, Drug-Enzyme, Drug-Carrier, Drug-Transporter relations	A database for the information on drugs and drug targets	https://go.drugbank.com/
KEGG (Kanehisa and Goto, 2000)	33,756,186 entities in Drug, Pathway, Gene, etc.	-	A database for genomes, biological pathways, diseases, drugs, and chemical substances.	https://www.kegg.jp/kegg/
PharmGKB (Whirl-Carrillo et al. 2012)	43,112 entities in Genes, Variant, Drug/Chemical and Phenotype	61,616 relations	A database for drugs and drug-related relationships.	https://www.pharmgkb.org/
Reactome (Jassal et al. 2020)	21.087 Pathway entities	-	A manually curated database for peer-reviewed pathway	https://reactome.org/
Semantic MEDLINE Database (Kilicoglu et al. 2012)	-	109,966,978 relations	A database contains Semantic predictions from the literature	https://skr3.nlm.nih.gov/index.html
Gene Ontology (Ashburner et al. 2000)	44,085 Gene entities	-	An ontology the functions of genes	http://geneontology.org/

24.3 Inference on Knowledge Graphs

In KG inference, one usually needs to address two important attributes of KGs: 1) the KG's local and global structure properties, and 2) heterogeneity of entities and relations(Wang et al, 2017d; Cai et al, 2018b; Zhang et al, 2018c; Goyal and Ferrara, 2018; Su et al, 2020c; Zhao et al, 2019d). In this context, a standard pipeline for KG inference typically contains two major steps: 1) learning embeddings (i.e., representation vectors) for entities (and relations) while preserving their structural properties and entity and relation attributes in the KG; and 2) performing downstream tasks such as entity classification and link prediction using the learned embeddings. Of note, one can perform these two steps separately, but also build an end-to-end model that can jointly learn the embeddings and perform downstream tasks. In this section, we review the existing techniques for inference on KGs, including the conventional inference techniques and the GNN-based models.

24.3.1 Conventional KG inference techniques

This subsection surveys the conventional KG inference techniques.

Semantic matching models typically exploit the similarity-based energy functions by matching latent semantics of entities and relations in the embedding spaces. A well-known semantic matching model, RESCAL (Nickel et al, 2011; Jenatton et al, 2012), was proposed based on the idea that entities are similar if connected to similar entities via similar relations(Nickel and Tresp, 2013). By associating each relation r_k with a matrix M_k , it defines the energy function by a bilinear model $f(e_i, r_k, e_j) = \mathbf{h}_i^\top M_k \mathbf{h}_j$, where $\mathbf{h}_i, \mathbf{h}_j \in \mathbb{R}^d$ are d-dimensional embedding vectors for entities e_i and e_j, respectively. RESCAL jointly learns embedding results for entities by e_i and e_j and for relation by M_k. Another model, DistMult (Yang et al, 2015a) simplifies RESCAL by restricting matrix M_k for relation r_k as a diagonal matrix. Though DistMult is more efficient than RESCAL, it can only deal with the undirected graphs. To address this, HolE (Nickel et al, 2016b) composes e_i and e_j by their circular correlation. Consequently, power of RESCAL and efficiency of DistMult are inherited by HolE. Other semantic matching models refer to the neural network architecture by considering embedding as the input layer and energy function as the output layer, such as the the semantic matching energy (SME) model (Bordes et al, 2014) and multi-layer perceptron (MLP) (Dong et al, 2014).

Translational distance models are based on the idea that, for each triplet (e_i, r_k, e_j) , the relation r_k can be considered as a translation from head entity e_i to tail entity e_j in the embedding space. Accordingly, they exploit distance-based energy functions to model the triplets in KG. In this context, TransE (Bordes et al, 2013) is the famous pioneer of the translational distance model family. It typically represents relation r_k as the translation vector $\mathbf{g}_k$, such that e_i and e_j are closely connected by r_k. Therefore, the energy function is defined as $f(e_i, r_k, e_j) = \|\mathbf{h}_i + \mathbf{g}_k + \mathbf{h}_j\|_2$. Since all parameters to learn are entity and relation embedding vectors lying in a same

low-dimensional space, TransE is obviously easy to train. A drawback of TransE is that it cannot do well with N-to-1, N-to-1 and N-to- N structures in KGs. To address this issue, TransH (Wang et al, 2014) extends TransE by introducing a hyperplane for each relation r_k and projecting e_i and e_j into the hyperplane before constructing the translation scheme. In this way, TransH improves model capacity while preserving efficiency. Similarly, TransR (Lin et al, 2015) extends TransE by introducing the relation-specific space. Further, for more fine-grained embedding, TransD (Ji et al, 2015) extends TransE by constructing two matrices M_k^1 and M_k^2 for each r_k to project e_i and e_j, respectively. Hence it captures both entity diversity and relation diversity. Further, TranSparse (Ji et al, 2016) simplifies TransR by using adaptive sparse matrices to model different types of relations, and TransF (Feng et al, 2016) relaxes the translation restriction as $\mathbf{h}_i + \mathbf{g}_k \approx \alpha \mathbf{h}_j$.

Meta-path-based approaches. A potential issue for both semantic matching models and translational distance models is that they mainly focus on one-hop information (i.e., modeling neighboring entities within a triplet) and hence may ignore the global structure properties of KGs. To address this, the meta-path based models aim at capturing local and global structure properties, as well as entity and relation types for KG inference. Typically, a meta-path is defined as a sequence of node types separated by edge types (Sun et al, 2011). For example, a meta-path of length l is $a_1 \overset{b_1}{\rightarrow} a_2 \overset{b_2}{\rightarrow} ... \overset{b_{l-1}}{\rightarrow} a_l$, where $\{a_1, a_2, ..., a_l\}$ and $\{b_1, b_2, ..., b_{l-1}\}$ are the sets of node type and relation type, respectively. Following this idea, Heterogeneous Information Network Embedding (HINE) (Huang and Mamoulis, 2017) defines meta-path-based proximity. It preserves heterogeneous structure by minimizing the difference between meta-path-based proximity and expected proximity in the embedding space. Moreover, metapath2vec (Dong et al, 2017) formalizes meta-path-based random walks and extends the word embedding model SkipGram to learn entity embeddings, by considering each walk path as a sentence and entities as words.

Convolutional neural network (CNN) models have also been used to address the KG inference task. For example, ConvE (Dettmers et al, 2018) uses CNN architecture for link prediction in KGs. For each triplet (e_i, r_k, e_j), ConvE first reshapes embedding vectors of e_i and r_k as two matrices and concatenate them. The resulting matrix is then fed to the convolutional layers to produce feature maps, which are then transformed into the entity embedding space to match the embedding of e_j. In addition, ConvKB (Nguyen et al, 2017) directly concatenate embedding vectors of e_i , r_k, and e_j, for each triplet (e_i, r_k, e_j), into a 3-column matrix. Then the matrix is fed to the convolutional layers to learn the entity and relation embeddings.

24.3.2 GNN-based KG inference techniques

This subsection discusses KG inference techniques based on the novel GNN architectures.

Graph convolution network (GCN)-based architectures. A pioneer effort using and extending GCN in KG inference is the Relational GCN (R-GCN) (Schlichtkrull et al, 2018). In contrast to the original application scenario, the structure property of a KG is usually heterogeneous as having diverse entity types and relation types. To address this, R-GCN introduces two subtle modifications on the regular GCN architecture (Berg et al, 2017). Specifically, for each entity, instead of simply aggregating information from all of its neighbors, R-GCN uses a relation-specific transformation mechanism, which first gathers information from neighboring entities based on relation types and relation directions separately and then accumulates them together. Specifically,

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{r_k \in \mathbb{R}} \sum_{j \in \mathscr{N}_i^k} \frac{1}{c_{i,k}} W_k^{(l)} \mathbf{h}_j^{(l)} + W_0^{(l)} \mathbf{h}_i^{(l)}\right) \tag{24.1}$$

Here $\mathbf{h}_i^{(l+1)}$ is the embedding vector of entity e_i at the $(l+1)$-th graph convolutional layer. $\mathscr{R}$ is the set of all relations and $\mathscr{N}_i^k$ is the neighbors of entity e_i under relation r_k. The problem-specific normalization coefficient $c_{i,k}$ can be either learned or pre-defined. Using softmax for each entity, R-GCN can be trained for entity classification. In link prediction, R-GCN is used as an encoder for learning embedding vectors of the entities while the factorization model, DistMult, is used as the decoder to predict missing links in the KG based on the learned entity embeddings. It resulted in a significantly improved performance compared to the baseline models like DistMult and TransE.

Cai et al (2019) proposed the TransGCN, which combines the GCN architecture with the translational distance models (e.g., TransE and RotatE) for link prediction in KGs. Compared to R-GCN, TransGCN aims to address the link prediction task without a task-specific decoder like R-GCN and learn both entity embeddings and relation embeddings simultaneously. For each triplet (e_i, r_k, e_j), TransGCN assumes that r_k is the transformation from the head e_i to the tail e_j in the embedding space. Then it extends the GCN layer to update e_i's embedding as

$$\mathbf{m}_i^{(l+1)} = \frac{1}{c_i} W_0^{(l)} \left(\sum_{(e_j, r_k, e_i) \in \mathscr{N}_i^{(in)}} \mathbf{h}_i^{(l)} \circ \mathbf{g}_k^{(l)} + \sum_{(e_i, r_k, e_j) \in \mathscr{N}_i^{(out)}} \mathbf{h}_j^{(l)} \star \mathbf{g}_k^{(l)}\right) \tag{24.2}$$

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{m}_i^{(l+1)} + \mathbf{h}_i^{(l)}\right) \tag{24.3}$$

where $\circ$ and $\star$ are transformation operators that can be defined based on specific translational mechanism used. $\mathscr{N}_i^{(in)}$ and $\mathscr{N}_i^{(out)}$ are incoming and outgoing triplet of e_i, respectively. The normalization constant $c_i$was defined by the total degree of entity e_i. Meanwhile, embedding of each relation r_k was updated by simply $\mathbf{g}_k^{(l+1)} = \sigma(W_1^{(l)} \mathbf{g}_k^{(l)})$. The authors engaged two translational mechanisms, TransE and RotatE, and defined $\circ$, $\star$, and scoring functions accordingly. Both result-

ing architectures, TransE-GCN and RotatE-GCN, showed higher performance than TransE, RotatE, and R-GCN in the experiments.

Structure-Aware Convolutional Network (SACN) (Shang et al, 2019) is another architecture for knowledge graph inference based on GCN. Similar to R-GCN, it engaged a weighted graph convolutional network (WGCN) as the encoder to capture the structure property of the KG. WGCN considers a KG with multiple relation types as a combination of multiple sub-graphs with single relation type. Then, the embedding vector of each entity e_i can be obtained by a weighted combination of information propagation based on each sub-graph,

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_k^{(l)} \mathbf{h}_j^{(l)} W^{(l)} + \mathbf{h}_i^{(l)} W^{(l)}\right) \tag{24.4}$$

where $\alpha_k^{(l)}$ is the weight of relation r_k at the l-th layer. The learned embedding from WGCN was then fed to a decoder, Conv-TransE, a CNN with TransE's translational mechanism, for link prediction.

Graph attention network (GAT)-based architectures. A potential drawback of the GCN architectures is that, for each entity, they treat the neighbors equally to gather information. However, different neighboring entities, relations or triplets may have different importance in indicating a specific entity, and the weights of neighboring entities under the same relation may be also distinct. To address this, GATs have been used to involved in the KG inference problems. One of the early efforts is the GATE-KG (i.e., graph attention-based embedding in KG) (Nathani et al, 2019). It introduces an extended and generalized attention mechanism as the encoder to produce the entity and relation embeddings while capturing the diverse relation type in KG. For each triplet (e_i, r_k, e_j), GATE-KG first produces a representation vector $\mathbf{c}_{ijk}^{(l)}$ of this triplet by

$$\mathbf{c}_{ijk}^{(l)} = W_1^{(l)}[\mathbf{h}_i^{(l)} || \mathbf{h}_j^{(l)} || \mathbf{g}_k^{(l)}] \tag{24.5}$$

Here $||$ is the concatenation operation. The attention coefficient α_{ijk} is obtained by

$$\beta_{ijk}^{(l)} = \text{LeakyReLU}\left(W_2^{(l)} \mathbf{c}_{ijk}^{(l)}\right) \tag{24.6}$$

$$\alpha_{ijk}^{(l)} = \frac{\exp(\beta_{ijk}^{(l)})}{\sum_{j'\in\mathcal{N}_i}\sum_{k'\in\mathcal{R}_{ij'}} \exp_(\beta_{ij'k'}^{(l)})} \tag{24.7}$$

where $\mathcal{R}_{ij}$ is the set of all relations between e_i and e_j . By aggregating information from neighbors according to different relations, entity e_i's embedding vector $\mathbf{h}_i^{(l+1)}$ at the $(l+1)$-th layer can be calculated as

$$\mathbf{h}_i^{(l+1)} = \sigma\left(\sum_{j\in\mathcal{N}_i}\sum_{k\in\mathcal{R}_{ij}} \alpha_{ijk}^{(l)} \mathbf{c}_{ijk}^{(l)}\right) \tag{24.8}$$

In addition, by using the auxiliary relation between n-hop neighbors and iteratively accumulates information of n-hop neighbors at the n-th graph attention layer, GATE-KG gives high weights to the 1-hop neighbors while lower weights to the n-hop neighbors. Hence it capture the multi-hop structure information of KG.

Relational Graph neural network with Hierarchical ATtention (RGHAT) (Zhang et al, 2020i) is another GAT-based model to address link prediction in KGs. Specifically, it engages a two-level attention mechanism. First, a relational-level attention defines the weight of each relation r_k indicating a specific entity e_i as

$$\mathbf{a}_{ik} = W_1 [\mathbf{h}_i || \mathbf{g}_k] \tag{24.9}$$

$$\alpha_{ik} = \frac{\exp(\sigma(\mathbf{z}_1 \cdot \mathbf{a}_{ik}))}{\sum_{r_x \in \mathcal{N}_i} \exp(\sigma(\mathbf{z}_1 \cdot \mathbf{a}_{ix}))} \tag{24.10}$$

where $\mathbf{z}_1$is a learnable parameter vector and σ is LeakyReLU. $\mathcal{N}_i$ is the neighboring relations of entity e_i. Second, it defines an entity-level attention as

$$\mathbf{b}_{ikj} = W_2 [\mathbf{a}_{ik} || \mathbf{h}_j] \tag{24.11}$$

$$\beta_{kj} = \frac{\exp(\sigma(\mathbf{z}_2 \cdot \mathbf{b}_{ikj}))}{\sum_{r_y \in \mathcal{N}_{i,k}} \exp(\sigma(\mathbf{z}_1 \cdot \mathbf{b}_{iyj}))} \tag{24.12}$$

where $\mathbf{z}_2$is a learnable parameter vector and $\mathcal{N}_{i,k}$ denotes the set of tail entities of entity e_i under relation r_k. The final attention coefficient for gathering information via triplet (e_i, r_k, e_j) is calculated as $\mu_{ikj} = \alpha_{ik} \cdot \beta_{kj}$. Similar to GATE-KG, the RGHAT engages ConvE as the decoder for link prediction.

Wang et al (2019j) proposed the Knowledge Graph Attention Network (KGAT) for recommendation based on KG, which contains three types of layers. First, a embedding layer learns embeddings for entities and relations using TransR. Second, the attentive embedding propagation layers extend GAT to capture the high-order structure properties (i.e., multi-hop neighbor information) of KG. Specifically, they defined the attention coefficient for each triplet (e_i, r_k, e_j), depending on distance between e_i and e_j in the r_k's space, i.e.,

$$\beta_{ijk} = (W_k \mathbf{h}_i)^\top \tanh (W_k \mathbf{h}_j + \mathbf{g}_k) \tag{24.13}$$

$$\alpha_{ijk} = \frac{\exp(b_{ijk})}{\sum_{j' \in \mathcal{N}_i} \sum_{k' \in \mathcal{R}_{ij'}} \exp_(\beta_{ij'k'})} \tag{24.14}$$

KGAT then stacks multiple attentive embedding propagation layers to capture information of multiple-hop neighbors of each entity, specifically, entity e_is embedding at the $(l+1)$-th layer, i.e., $\mathbf{h}_i^{(l+1)} = \sigma\left(\mathbf{h}_i^{(l)}, \mathbf{h}_{\mathcal{N}_i}^{(1)}\right)$, where $\mathbf{h}_{\mathcal{N}_i}^{(1)} = \sum_{(e_i, r_k, e_j) \in \mathcal{N}_i} \alpha_{ijk} \mathbf{h}_j^{(l)}$. Finally, a prediction layer concatenates embeddings at each graph attention layer for each entity to make prediction.

In addition, Heterogeneous graph Attention Network (HAN) (Wang et al, 2019m) uses GAT to address the node (i.e., entity) classification in the heterogeneous graphs (the KG can be considered as a specific type of heterogeneous graph). HAN couples graph attention mechanism with meta-paths to capture the heterogeneous structure properties. A hierarchical attention mechanism that contains a node-level attention and semantic-level attention was proposed. The node-level attention aims to learn the importance of the meta-path-based neighbors in indicating a node. Specifically, it first projects different types of entities into a same space by $\mathbf{h}_i = M_{\phi_i}\mathbf{h}'_i$, where ϕ_i is the type of entity e_i, and $\mathbf{h}_i$ and $\mathbf{h}'_i$ are the projected and original embeddings of e_i, respectively. It then calculates the attention weight α_{ij}^{Φ} of entity pair (e_i, e_j) under a specific meta-path Φ, as

$$\alpha_{ij}^{\Phi} = \frac{\exp(\mathbf{a}_{\Phi}^{\top} \cdot [\mathbf{h}_i || \mathbf{h}_j])}{\sum_{j' \in \mathscr{N}_i^{\Phi}} \exp(\mathbf{a}_{\Phi}^{\top} \cdot [\mathbf{h}_i || \mathbf{h}_{j'}])} \tag{24.15}$$

where $\mathscr{N}_i^{\Phi}$ is the neighbors of e_i under meta-path Φ and $\mathbf{a}_{\Phi}$ is the node-level attention vector. In addition, the semantic attention layer learns importance of each meta-path Φ in the task (i.e., classification) by

$$w_{\Phi} = \frac{1}{|\mathscr{V}|} \sum_{e_i \in \mathscr{V}} \mathbf{q}^{\top} \cdot \tanh\left(W \cdot \mathbf{z}_i^{\Phi} + \mathbf{b}\right) \tag{24.16}$$

where $\mathscr{V}$ is all entities, $\mathbf{q}$ is the learnabel semantic-level attention vector, and $\mathbf{b}$ is the bias. Then the semantic-level attention weight is calculated as $\beta_{\Phi} = \frac{\exp(w_{\Phi})}{\sum_{\Phi'} \exp(w_{\Phi'})}$. The final embeddings of all entities, $Z = \sum_{\Phi} \beta_{\Phi} Z_{\Phi}$, are used for classification.

24.4 KG-based hypothesis generation in computational drug development

Generally, the drug repurposing procedure includes three major steps: hypothesis generation, assessment, and validation (Pushpakom et al, 2019). Among them, the first and foremost step is hypothesis generation. Typially, the hypothesis generation for drug repurposing aims at identifying candidate drugs that has a high confidence to be associated with the therapeutic indication of interest. Today's largly available BKGs, encoding huge volume of biomedical knowledge, have become a valuable resouce for drug repurposing. In KG, the hypothesis generation procedure can be formulated as a link prediction problem, i.e., computational identification of potential drug-target or drug-disease associations with a high confidence level based on existing knowledge (KG's structure properties). This section introduces some preliminary efforts of hypothesis generation for drug repurposing, using computational approaches in the BKGs.

24.4.1 A machine learning framework for KG-based drug repurposing

One of the previous efforts using computational inference in BKG for drug repurposing is Zhu et al.'s study (Zhu et al, 2020b). The main contributions of this study is two-folds: 1) KG construction via data integration, and 2) building the KG-based machine learning pipeline for drug repurposing.

First, by integrating six drug knowledge bases, including PharmGKB (Whirl-Carrillo et al, 2012), TTD (Yang et al, 2016a), KEGG DRUG (Kanehisa et al, 2007), DrugBank (Wishart et al, 2018), SIDER (Kuhn et al, 2016), and DID (Sharp, 2017), they curated a drug-centric KG consisting of five entity types including drugs, diseases, genes, pathways, and side-effects and nine relation types including drug-disease TREATS, drug-drug INTERACTS, and drug-gene REGULATES, BINDS, and ASSOCIATES, drug-side effect CAUSES relations, gene-gene ASSOCIATES, gene-disease ASSOCIATES, and gene-pathway PARTICIPATES relations.

Second, based on the drug-centric KG, a machine learning pipeline was built for drug repurposing. Specifically, the target of the proposed model was to predict the existence of relation between a pair of drug and disease entities. In this way, the task fell into the supervised classification setting where the input samples were the drug-disease pairs. To this end, representation for each sample (drug-disease pair) was calculated in two ways: 1) meta-path-based representation and 2) KG embedding-based representation. For meta-path-based representation, 99 possible meta-paths between drugs and diseases with length 2-4 were enumerated, such as Drug $\overset{\text{TREATS}}{\rightarrow}$ Gene $\overset{\text{ASSOCIATES}}{\rightarrow}$ Disease and Drug $\overset{\text{TREATS}}{\rightarrow}$ Gene $\overset{\text{ASSOCIATES}}{\rightarrow}$ Gene $\overset{\text{ASSOCIATES}}{\rightarrow}$ Disease. Then a 99-dimensional representation vector was calculated for a drug-disease pair, of which each element indicates the connectivity measure between this two entities based on a specific meta-path. In this study, four different connectivity measures were used, under a specific meta-path Φ, including

- Path count, $PC_{\Phi}(e_{dr}, e_{di})$, the number of paths between drug e_{dr} and disease e_{di};
- Head normalized path count $HNPC_{\Phi} = \frac{PC_{\Phi}(e_{dr},e_{di})}{PC_{\Phi}(e_{dr},*)}$;
- Tail normalized path count $TNPC_{\Phi} = \frac{PC_{\Phi}(e_{dr},e_{di})}{PC_{\Phi}(*,e_{di})}$;
- Normalized path count $NPC_{\Phi} = \frac{PC_{\Phi}(e_{dr},e_{di})}{PC_{\Phi}(e_{dr},*)+PC_{\Phi}(e_{dr},*)}$;

For KG embedding-based representation, three translational distance models, including TransE (Bordes et al, 2013), TransH (Wang et al, 2014), and TransR (Lin et al, 2015), were used. Specifically, for each pair of drug e_{dr} and disease e_{di}, using each of the three models, their embedding vectors $\mathbf{h}_{dr}$ and $\mathbf{h}_{di}$ were first learned. Then representation of the drug-disease pair (e_{dr}, e_{di}) was calculated by $\mathbf{h}_{di} - \mathbf{h}_{dr}$.

After that, a machine learning pipeline was built of which the input are representations of the drug-disease pairs. A drug-disease pair was labeled as positive if there is a relation between them. However, the drug-disease pair without a relation between them isn't really negative, instead, it was marked as unknown/unlabeled.

To address this, a positive and unlabeled (PU) learning framework (Elkan and Noto, 2008) was used. Decision Tree, Random Forest, and support vector machine (SVM) were used as basic classifiers of this PU learning framework, respectively. In this study, drug-disease relations related to eight diseases were used as the testing set, while the remaining drug-disease relations (positive) and 143,830 pairs associating the eight diseases with other drugs (unlabeled) were used as the training set. Experimental results showed that the KG-driven pipeline can produce high prediction results on known diabetes mellitus treatments with only using treatment information of other diseases.

24.4.2 Application of KG-based drug repurposing in COVID-19

The sudden outbreak of the human coronavirus disease 2019 (COVID-19) has led to a pandemic that heavily strikes the healthcare system and tremendously impacts people' life around the world. To date, many drugs have been under investigation to treat COVID-19, costing tremendous investment, however, very limited COVID-19 antiviral medications are approved. In this context, there is the urgent need for a more efficient and effective way for drug development against the pandemic, and computational drug repurposing can be a promising approach to address this.

Zeng et al.'s work (Zeng et al, 2020b) is a pioneer effort that computationally repurposes antiviral medications in COVID-19 based on KG inference. First of all, a comprehensive biomedical KG was constructed by integrating the two biomedical relational data resources, Global Network of Biomedical Relationships (GNBR) (Percha and Altman, 2018) and DrugBank (Wishart et al, 2018), and experimentally discovered COVID-gene relationships (Zhou et al, 2020f), resulting a KG consisting of 145,179 entities of four types (drugs, disease, genes, and drug side information) and 15,018,067 relationships of 39 types. Secondly, a deep KG embedding model, RotatE, was performed to learn low-dimensional representations for the entities and relations. Using such learned embedding vectors, the top 100 drugs most close to the COVID-19 entity in the embedding space were prioritized as the candidate drugs. Using drugs in ongoing COVID-19 clinical trials (`https://covid19-trials.com/`) as a validation set, the results achieved a desirable performance with an area under the receiver operating characteristic curve (AUROC) of 0.85. Moreover, gene set enrichment analysis (GSEA), which involved transcriptome data from peripheral blood and Calu-3 cells, and proteome data from Caco-2 cells, was performed to validate the candidate drugs. Finally, 41 drugs were identified as potential repurposable candidates for COVID-19 therapy, especially 9 are under ongoing COVID-19 trials. Among the 41 candidates, three types of drugs were highlighted by the author: 1) the Anti-Inflammatory Agents such as dexamethasone, indomethacin, and melatonin; 2) the Selective Estrogen Receptor Modulators (SERMs) such as clomifene, bazedoxifene, and toremifene; and 3) the Antiparasitics including hydroxychloroquine and chloroquine phosphate.

Another work (Hsieh et al, 2020), has been focused on using GNN in KG to address the drug repurposing problem. By extracting and integrating drug-target interactions, pathways, gene/drug-phenotype interactions from CTD (Davis et al, 2019), a SARS-CoV-2 KG was built, which consists of 27 SARS-CoV-2 baits, 5,677 host genes, 3,635 drugs, and 1,285 phenotypes, as well as 330 virus-host protein-protein interactions, 13,423 gene-gene sharing pathway interactions, 16,972 drug-target interactions, 1,401 gene-phenotype associations, and 935 drug-phenotype associations. Nest, a variational graph autoencoder (Kipf and Welling, 2016), which engages R-GCN (Schlichtkrull et al, 2018) as encoder, was used to learn entity embeddings in the SARS-CoV-2 KG. Since the SARS-CoV-2 KG has a specific focus on COVID-19 related knowledge, some general yet meaningful biomedical knowledge may be missing. To address this, a transfer learning framework was introduced. Specifically, it first used entity embeddings of Zeng et al.'s work (Zeng et al, 2020b) that encode general biomedical knowledge to initialize entity embeddings in SARS-CoV-2 KG. Then the embeddings were fine-tuned in SARS-CoV-2 KG through the proposed GNN. Using a customized neural network ranking model, 300 drugs most relevant to the COVID-19 were selected as the candidate drugs. Similar to Zeng et al.'s work (Zeng et al, 2020b) , the authors engaged GSEA, retrospective in-vitro drug screening, and populiation-based treatment effect analysis in electronic health records (EHRs), to further validate the repurposable candidates. Through such a pipeline, 22 drugs were highlighted for potential COVID-19 treatment, including Azithromycin, Atorvastatin, Aspirin, Acetaminophen, and Albuterol.

In summary, these studies shed light on the importance of the KG-based computational approaches in drug repurposing to fight against the complex diseases like COVID-19. The reported good performance in terms of the high overlapping ratio between the repurposed candidate drug set and the drugs under ongoing COVID-19 trials, not only demonstrated the effectiveness of the KG-based techniques but also provided biological evidence of the ongoing clinical trials. Moreover, they proposed feasible ways using other publicly available data to validate or refine the hypothesis derived from KGs, which therefore enhances the usability of KG-based approaches.

24.5 Future directions

KGs have been playing a more and more important role in biomedicine. An increasing number of KG-based machine learning and deep learning approaches have been used in biomedical studies such as hypothesis generation in computational drug development. As one of the latest advances in artificial intelligence (AI), GNNs, which have led to tremendous progress in image and text data mining (Kipf and Welling, 2017b; Hamilton et al, 2017b; Veličković et al, 2018), have been introduced to address the KG inference problems. In this context, the use of GNN in biomedical KGs has a great potential in improving hypothesis generation in computational drug development. However, there remain significant gaps between the novel technique and the success of computational drug development. This section discusses the potential

opportunities and future research possibilities in this field toward improvements of hypothesis generation for computational drug development.

24.5.1 KG quality control

The procedures of constructing and curating a biomedical KG typically include manually gathering, annotating, and extracting knowledge from text (e.g., literature or experimental reports), automatically or manually normalizing terminology to integrate multiple data resources, and automatically text mining for knowledge extraction, etc. However, none of them are perfect. Therefore, the quality issue has been challenging the KG inference approaches. In KG-based hypothesis generation for drug repurposing, a poor quality of KG will lead to uninformative or wrong representations and hence result in incorrect hypothesis generated (drug-disease associations) and even failure of the entire drug repurposing project. Therefore, there is an urgent need for accurate and appropriate KG quality control. In general, there are two categories of quality issues in KGs: the incorrectness and incompleteness.

Incorrectness refers to incorrect triplets in the KG, i.e., a triplet exists in KG but the corresponding relationship between the two entities is inconsistent with real-world evidence. To address this, a common strategy is manual annotation with sampled small subsets. Such a procedure is time- and cost-consuming, if one wants to evaluate sufficient triplets to reach the statistic criteria. To address this, for example, Gao et al (2019a) proposed an iterative evaluation framework for KG accuracy evaluation. Specifically, inspired by the properties of the annotation cost function observed in practice, the authors developed a cluster sampling strategy with unequal probability theory. Their framework resulted in a 60% shrunk annotation cost and can be easily extended to address evolving KG. In addition, the use of well-designed biomedical vocabularies such as the Unified Medical Language System (UMLS) (Bodenreider, 2004) will improve entity term normalization and hence reduce the risk of errors caused by the ambiguous biomedical entities. Moreover, learning based on KG structure to refine the KG is also a potential way to solve this issue. Early efforts, such as (Zhao et al, 2020d), have been focused on this field.

Incompleteness mainly refers to the missing of biologically or clinically meaningful triplets in the KG. To address the incompleteness in biomedical KG, a common way is to integrate multiple data resources, biomedical data bases, and biomedical KGs to construct and curate a more comprehensive one. CKG (Santos et al, 2020), Hetionet (Himmelstein et al, 2017), DRKG (Ioannidis et al, 2020), KG (Zhu et al, 2020b), etc. are good examples of this strategy. However, there is no guarantee they are comprehensive enough to cover all biomedical knowledge. In addition, today's largely available biomedical literature and medical data (e.g., EHRs) are great treasure of biomedical knowledge. In this context, previous studies have been focused on deriving knowledge from biomedical literature (Zhao et al, 2020e; Xu et al, 2013; Zhang et al, 2018h; Sahu and Anand, 2018) and EHR data (Rotmensch et al, 2017; Chen et al, 2020e), and the derived knowledge could be a good

complement for the biomedical KGs. Moreover, the computational methods such as the KG embedding models (e.g., TransE and TransH) and the GNNs (e.g., R-GCN) have been used in KG completion (Arora, 2020), which predict missing relations within a KG according to its structure properties.

24.5.2 Scalable inference

An ultimate goal of biomedical KGs is always to comprehensively incorporate the biomedical knowledge. For example, by integration of 26 publicly available biomedical databases, CKG (Santos et al, 2020) has included over 16 million biomedical entities connected by over 220 million relationships; another KG, DRKG (Ioannidis et al, 2020), integrating six databases and data collected from recent COVID-19 publications, has included 10K entities and 5.8 million relationships. Meanwhile, today's advanced high-throughput techniques as well as computer software and hardware have led to an inrush of a continuously increasing number of relational data interlinking biomedical entities like drugs, genes, proteins, chemical compounds, diseases and medical concepts extracted from clinical data. This largely enables us to extract knew knowledge to enrich the biomedical KGs and hence these KGs keep expanding constantly.

In this context, the huge and even continuously increasing volume of KGs may challenge the computational models like GNNs. To this end, there is an urgent need for scalable techniques to address the high memory- and time-cost in KGs. For example, Deep Graph Library (DGL, `https://www.dgl.ai`) (Wang et al, 2019f) is an open-source, free Python package designed by Amazon for facilitating the implementation of GNN family models, running on the top of several deep learning framework including PyTorch (Paszke et al, 2019), TensorFlow (Abadi et al, 2016), and MXNet (Chen et al, 2015). As of Mach 1, 2021, it has released the version 0.6. By distilling GNN's message passing procedure as the generalized sparse tensor operations, DGL provides the implementations of optimization techniques like kernel fusion, multi-thread and multi-process acceleration, and automatic sparse format tuning to speed up training process and reduce memory load. In addition to GNNs, DGL also released DGL-KE (`https://github.com/awslabs/dgl-ke`) (Zheng et al, 2020c), an easy-to-use framework for implementation of KG representation models such as TransE, DistMult, RotatE, etc., which has been used in existing KG-based drug-repurposing studies such as (Zeng et al, 2020b).

24.5.3 Coupling KGs with other biomedical data

Apart from the KGs, there is an enormous volume of other biomedical data available such as clinical data and omics data, which are also promising resources for computational drug repurposing. The clinical data is an important resource for healthcare

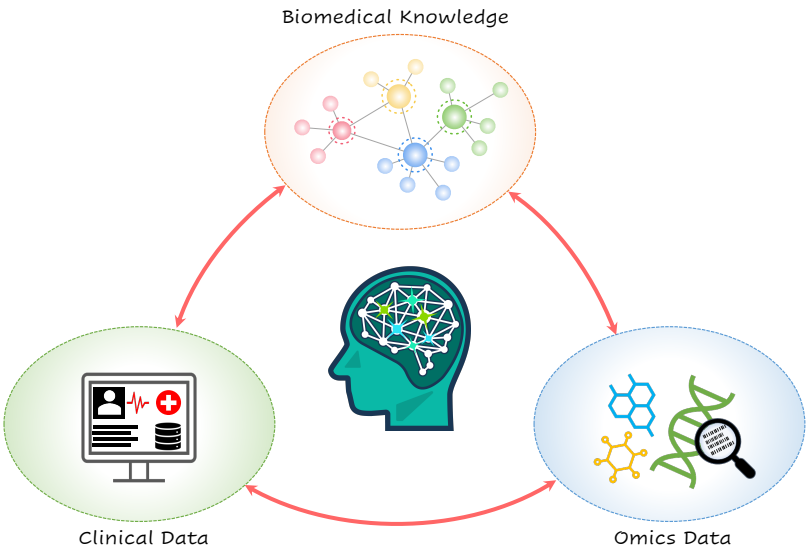

Figure 24.1: Coupling biomedical KGs with other biomedical data resources for improving computational drug development.

and medical research, mainly including EHR data, claim data, and clinical trial data, etc. The EHR data is routinely collected during the daily patient care, containing heterogeneous information of the patients, such as demographics, diagnoses, laboratory test results, medications, and clinical notes. Such rich information makes it possible for tracking patient's health condition changes, medication prescriptions, and clinical outcomes. In addition, a tremendous volume of EHR data has been collected and the volume is rapidly increasing, which largely strengthens the statistical power for EHR-based analysis. For this reason, beyond its common usage such as diagnostic and prognostic prediction (Xiao et al., 2018; Si et al., 2020; Su et al., 2020e,a), and phenotyping (Chiu and Hripcsak, 2017; Weng et al., 2020; Su et al., 2020d, 2021), EHR data has been used for computational drug repurposing (Hurle et al., 2013; Pushpakom et al., 2019). For example, Wu et al. (2019d) identified some non-cancer drugs as the repurposable candidates to treat cancer using EHR; Gurwitz (Gurwitz, 2020) analyzed EHR data to repurpose drugs for treating COVID-19.

Advanced by the high throughput sequencing techniques, an enormous volume of omics data, including genomics, proteomics, transcriptomics, epigenomics, and metabolomics, have been collected and publicly available for analysis. Integrating and analyzing the omics data enable us to derive new biomedical insights and better understand human health and diseases at the molecular level (Subramanian et al., 2020; Nicora et al., 2020; Su et al., 2020b). Due to the wealth of the omics data, it has also been involved in computational drug development (Pantziarka and Meheus, 2018; Nicora et al., 2020; Issa et al., 2020). For example, via mining multiple omics data, Zhang et al. (2016c) identified 18 proteins as the potential anti-Alzheimer's dis-

ease (AD) targets and prioritized 7 repurposable drugs inhibiting the targets. Mokou et al (2020) proposed a drug repurposing pipeline in bladder cancer based on patients' omics (proteomics and transcriptomics) signature data.

In this context, combining KGs, clinical data, and multi-omics data and jointly learning them is a promising route to advance computational drug development (Fig. 24.1). The benefits of combining of these data for inference can be two-way. First, computational models in clinical data and multi-omics data usually suffer from the data quality such as noise and limited cohort size especially for the population of a rare disease and model interpretability. The incorporation of KGs has been demonstrated to be able to address these issues effectively and accelerate the clinical data and omics data analysis. For example, Nelson et al (2019) linked EHR data with a biomedical KG and learned a barcode vector for each specific cohort (e.g., the obese cohort), which encodes both KG structure and EHR information and illustrates the importance of each biomedical entity (e.g., genes, symptoms, and medications) in indicating the cohort. Such cohort-specific barcode vectors further showed the effectiveness in link prediction (e.g., disease-gene associations prediction). Wang et al (2017c) bridged patient EHR data with the BKG and extended the KG embedding model for safe medicine recommendation, which comprehensively considered relevant knowledge such as drug-drug interactions. In addition, Santos et al (2020) developed an open platform that couples the CKG (i.e., Clinical Knowledge Graph) with the typical proteomics workflows. In this way, CKG facilitates analysis and interpretation of the protomics data. Second, the incorporation of clinical data and omics data can potentially improve KG inference. Current KG-based drug repurposing studies have involved the clinical data and omics data (Zeng et al, 2020b; Hsieh et al, 2020), which were typically used in an independent validation procedure to validate/refine the generated new hypotheses (i.e., novel disease-drug associations) . Moreover, previous studies have showcased that leveraging the clinical data (Rotmensch et al, 2017; Chen et al, 2020e; Pan et al, 2020c) and omics data (Ramos et al, 2019) can derive new knowledge. Therefore, we believe that incorporating clinical data and omics data in KG inference may largely reduce the impacts of KG quality issues especially the incompleteness. In total, when we design the next-generation GNN models for drug-repurposing, a considerable direction is the feasible and flexible architecture that can subtly harness KGs, clinical data, and multi-omics data to recursively improve each other.

Editor's Notes: Drug hypothesis generation aims to use biological and clinical knowledge to generate biomedical molecules. This knowledge is effectively stored in the form of knowledge graph (KG). The construction of KG is relevant to graph generation (Chapter 11) and some applications, such as text mining (Chapter 21). Based on KG, hypothesis generation process mainly contains graph representation learning (Chapter 2) and graph structure learning (Chapter 14). It can also be formulated as the link prediction (Chapter 10) problem and calculate the confidence level of candidate drugs. The future direction of drug developments focuses on scalability (Chapter 6) and interpretability (Chapter 7).

Table 24.3: Summary of existing BKGs.

Database	Number of Entities	Entity Types	Number of Relations	Relation Types	Focus	Available Formats	Source Type	URL
Clinical Knowledge Graph(Santos et al., 2020)	16 million	33 entity types, such as Drug, Gene, Disease, etc.	220 million	51 relation types, such as associate, has quantified protein, etc.	-	Neo4j	KG (Integration)	https://github.com/MannLabs/CKG
Drug Repurposing Knowledge Graph(Ioannidis et al., 2020)	97,238	13 entity types, such as Compound, Disease, etc.	5,874,261	107 relation types, such as interaction, etc.	-	TSV	KG (Integration)	https://github.com/gnn4dr/DRKG
Hetionet(Himmelstein et al., 2017)	47,031	11 entity types, such as Disease, Gene, Compound, etc.	2,250,197	24 relation types, such as treats, associates, etc.	-	Neo4j,TSV	KG (Integration)	https://het.io/
iDISK(Rizvi et al., 2019)	144,059	6 entity types, such as Semantic Dietary Supplement Ingredient, Dietary Supplement Product, Disease, etc.	708,164	6 relation types, such as has_adverse_reaction, is_effective_for, etc.	Dietary Supplements	Neo4j,RRF	KG (Integration)	https://conservancy.umn.edu/handle/11299/204783
PreMedKB(Yu et al., 2019b)	404,904	Drug, Variant, Gene, Disease	496,689	52 relation types, such as cause, associate, etc.	Variant	-	KG (Integration)	http://www.fudan-pgx.org/premedkb/index.html#/home
Zhu et al (2020b)	-	Drug, Side-effect, Disease, Gene, Pathway	-	9 relation types, such as Cause, Binds, Treats, etc.	Drug Repurposing	-	KG (Integration)	-
Zeng et al (2020b)	145,179	Drug, Gene, Disease, and Drug side	15,018,067	39 relation types, such as treatment, binding, etc.	Drug Repurposing	-	KG (Integration)	-

Database	Number of Entities	Entity Types	Number of Relations	Relation Types	Focus	Available Formats	Source Type	URL
COVID-19 Knowledge Graph(Domingo-Fernández et al, 2020)	3,954	10 entity types, such as proteins, genes, chemicals, etc.	9,484	Increases, Decreases, association, etc.	COVID-19	JSON	KG	https://github.com/covid19kg/covid19kg
COVID-KG(Wang et al, 2020e)	67,217	Diseases, Chemicals, Genes	85,126,762	Chemical-Gene, Chemical-Disease, Gene-Disease	-	CSV	KG	http://blender.cs.illinois.edu/covid19/
KGHC(Li et al, 2020d)	5,028	9 entity types, such as drug, protein, disease, etc.	13,296	Associate_with, Cause, etc.	Hepatocellular Carcinoma	Neo4j	KG	http://202.118.75.18:18895/browser/
Li et al (2020b)	22,508	9 entity types, such as disease, symptom, etc.	579,094	-	Disease-Symptom	-	KG	-
QMKG(Goodwin and Harabagiu, 2013)	634,000	-	1,390,000,000	-	-	-	KG	-
Rotmensch et al (2017)	647	Disease, Symptom	-	Disease-Symptom	The linkage between diseases and symptoms	-	KG	-
Sun et al (2020a)	1,616,549	62 entity types, such as Disease, Drug, etc	5,963,444	202 relation types	Clinical suspected claims detection	-	KG	https://web.archive.org/web/20191231152615if_/http://121.12.85.245:1347/kg_test/#/
Bgee(Bastian et al, 2021)	60,072	Anatomy, Gene	11,731,369	Expression_Present, Expression_Absent	Anatomy-Gene Expression	TSV	KB	https://bgee.org/
Comparative Toxicogenomics Database(Davis et al, 2019)	73,922	Disease, Gene, Chemical, Pathway	38,344,568	Chemical-Gene, Chemical-Disease, Chemical-Pathway, Gene-Disease, Gene-Pathway, Disease-Pathway	-	CSV,TSV	KB	http://ctdbase.org/

Database	Number of Entities	Entity Types	Number of Relations	Relation Types	Focus	Available Formats	Source Type	URL
Drug–Gene Interaction Database(Cotto et al, 2018)	160,054	Drug, Gene	96,924	-	Drug-Gene Interaction	TSV	KB	https://www.dgidb.org/
DISEASES(Pletscher-Frankild et al, 2015)	22,216	Disease, Gene	543,405	-	Disease-Gene Association	TSV	KB	https://diseases.jensenlab.org/
DisGeNET(Piñero et al, 2020)	159,052	Disease, Gene, Variant	839,138	Gene-Disease,Variant-Disease	Gene-Disease,Variant-Disease associations	TSV	KB	https://www.disgenet.org/home/
Global Network of Biomedical Relationships(Percha and Altman, 2018)	-	Chemical, Disease, Gene	2,236,307	36 relation types, such as causal mutations, treatment, etc.	-	TXT	KB	https://zenodo.org/record/1035500
IntAct(Orchard et al, 2014)	119,281	Chemical, Gene	1,130,596	-	Molecular Interaction	TXT	KB	https://www.ebi.ac.uk/intact/
STRING(Szklarczyk et al, 2019)	24,584,628	Protein	3,123,056,667	Protein-Protein Interaction	Protein-Protein Interaction	TXT	KB	https://string-db.org/
SIDER(Kuhn et al, 2016)	7,298	Drug, Side-effect	139,756	Drug-Side effect	Medicines and their recorded adverse drug reactions	TSV	KB	http://sideeffects.embl.de/
SIGNOR(Licata et al, 2020)	7,095	10 entity types, such as protein, chemical, etc.	26,523	-	Signaling information	TSV	KB	https://signor.uniroma2.it/
TISSUE(Palasca et al, 2018)	26,260	Tissue, Gene	6,788,697	Express	Tissue-Gene Expression	TSV	KB	https://tissues.jensenlab.org/
Catalogue of Somatic Mutations in Cancer(Tate et al, 2019)	12,339,359	Mutation	-	-	Somatic Mutations in Cancer	TSV	Database	https://cancer.sanger.ac.uk/cosmic

Database	Number of Entities	Entity Types	Number of Relations	Relation Types	Focus	Available Formats	Source Type	URL
ChEMBL(Mendez et al, 2019)	2,940,733	Molecule	-	-	Molecule	TXT	Database	https://www.ebi.ac.uk/chembl/
ChEBI(Hastings et al, 2016)	155,342	Molecule	-	-	Molecule	TXT	Database	https://www.ebi.ac.uk/chebi/init.do
DrugBank(Wishart et al, 2018)	15,128	Drug	28,014	Drug-Target, Drug-Enzyme, Drug-Carrier, Drug-Transporter	Drug	CSV	Database	https://go.drugbank.com/
Entrez Gene(Maglott et al, 2010)	30,896,060	Gene	-	-	Gene	TXT	Database	https://www.ncbi.nlm.nih.gov/gene/
HUGO Gene Nomenclature Committee(Braschi et al, 2017)	41,439	Gene	-	-	Gene	TXT	Database	https://www.genenames.org/
KEGG(Kanehisa and Goto, 2000)	33,756,186	Drug, Pathway, Gene, etc.	-	-	-	TXT	Database	https://www.kegg.jp/kegg/
PharmGKB(Whirl-Carrillo et al, 2012)	43,112	Genes, Variant, Drug/Chemical, Phenotype	61,616	-	-	TSV	Database	https://www.pharmgkb.org/
Reactome(Jassal et al, 2020)	21.087	Pathway	-	-	Pathway	TXT	Database	https://reactome.org/
Semantic MEDLINE Database(Kilicoglu et al, 2012)	-	-	109,966,978	Subject-Predicate-Object Triples	Semantic predictions from the literature	CSV	Database	https://skr3.nlm.nih.gov/index.html
UniPort(?)	243,658	Protein	-	-	Protein	XML,TXT	Database	https://www.uniprot.org/
Brenda Tissue Ontology(Gremse et al, 2010)	6,478	Tissue	-	-	Tissue	OWL	Ontology	http://www.BTO.brenda-enzymes.org
Disease Ontology(Schriml et al, 2019)	10,648	Disease	-	-	Disease	OWL	Ontology	https://disease-ontology.org/

Database	Number of Entities	Entity Types	Number of Relations	Relation Types	Focus	Available Formats	Source Type	URL
Gene Ontology(Ashburner et al. 2000)	44,085	Gene	-	-	Gene	OWL	Ontology	http://geneontology.org/
Uberon(Mungall et al. 2012)	14,944	Anatomy	-	-	Anatomy	OWL	Ontology	http://uberon.github.io/publications.html